

**SYSMAC
CQM1**

AUTOMATE PROGRAMMABLE

MANUEL DE PROGRAMMATION

OMRON

Table des matières

Chapitre 1

Configuration de l'API et caractéristiques afférentes . . . 1

1-1	Configuration de l'API	2
1-2	Fonctionnement fondamental du CQM1 et processus d'Entrée/Sortie	8
1-3	Sélection et utilisation des fonctions de sortie d'impulsion	12
1-4	Sélection et utilisation des fonction d'interruption	25
1-5	Fonction de communication	58

Chapitre 2

Caractéristiques des API CQM1 69

2-1	Instructions étendues	70
2-2	Instructions d'E/S étendues	72
2-3	Fonction Macro	83
2-4	Surveillance de changement d'état d'un bit	84
2-5	Sélections analogiques (CQM1-CPU42-E uniquement)	85

Chapitre 3

Zone de mémoire 87

3-1	Fonctions de la zone mémoire	88
3-2	Attribution des bits d'E/S	92
3-3	Stockage des DM et de l'UM dans les cartouches mémoire	93

Chapitre 4

Programmation en schéma à relais 99

4-1	Procédure fondamentale	100
4-2	Terminologie des instructions	100
4-3	Les bases du schéma à relais	101
4-4	Contrôle de l'état des bits	120
4-5	Bits de travail (relais internes)	122
4-6	Précautions de programmation	125
4-7	Execution du programme	126

Chapitre 5

Instructions 129

5-1	Notation	133
5-2	Format	133
5-3	Zones de données, données d'opérande et drapeaux	133
5-4	Instructions sur changement d'état	135
5-5	Codage des instructions de droite	136
5-6	Tableaux d'instructions	139
5-7	Instructions en schéma à relais	143
5-8	Instructions de contrôle de bit	144
5-9	NO OPERATION – NOP(00)	148
5-10	END – END(01)	148
5-11	INTERLOCK et INTERLOCK CLEAR – IL(02) et ILC(03)	149
5-12	JUMP et JUMP END – JMP(04) et JME(05)	151
5-13	Instructions d'erreur utilisateur : FAILURE ALARM AND RESET – FAL(06) et SEVERE FAILURE ALARM – FALS(07)	152
5-14	Instructions d'étape : STEP DEFINE et STEP START–STEP(08)/SNXT(09)	153
5-15	Instructions de temporisation/compteur	155

TABLE DES MATIERES

5-16	Registre à décalage	171
5-17	Instructions de déplacement de données	180
5-18	Instructions de comparaison	191
5-19	Instructions de conversion	202
5-20	Instructions de calcul BCD	229
5-21	Instructions de calcul binaire	240
5-22	Instructions mathématiques spéciales	251
5-23	Instructions logiques	260
5-24	Instructions d'incrément/décément	264
5-25	Instructions de sous-programme	265
5-26	Instructions spéciales	267
5-27	Instructions de communication	292
5-28	Instructions d'E/S étendues	295
Chapitre 6		
Commandes de liaison à l'ordinateur		299
6-1	Procédure de communication	300
6-2	Formats de commande et de réponse	302
6-3	Commandes de liaison à l'ordinateur	305
Chapitre 7		
Fonctionnement et temps de traitement du CQM1		329
7-1	Fonctionnement du CQM1	330
7-2	Temps de scrutation et temps de réponse des E/S	331
Chapitre 8		
En cas de problème		351
8-1	Introduction	352
8-2	Erreurs de fonctionnement de la console de programmation	352
8-3	Erreurs de programmation	353
8-4	Erreurs définies par l'utilisateur	354
8-5	Erreurs de fonctionnement	355
8-6	Stockage des erreurs	357
8-7	Erreurs de liaison à l'ordinateur	357
8-8	Diagrammes de résolution des problèmes	359
Annexes		
A	Références	367
B	Instructions de programmation	371
C	Fonctionnement des drapeaux d'erreur et arithmétiques	377
D	Zones mémoire	381
E	Configuration API	391
F	Durée de vie de la pile	397
G	Utilisation de la fonction horloge	399
H	Feuille d'attribution des E/S	401
I	Feuille de codage des programmes	403
J	Liste des numéros FAL	407
K	Codes ASCII étendus	409

Chapitre 1

Configuration de l'API et caractéristiques afférentes

Ce chapitre traite de la configuration de l'API et des caractéristiques afférentes, y compris les fonctions d'interruption et de communication. La configuration peut être utilisée pour contrôler les paramètres de l'API ; pour la modifier, reportez-vous au manuel d'utilisation du CQM1 concernant la console de programmation et au manuel LSS pour la procédure de ce logiciel.

Si vous ne connaissez pas très bien les API OMRON ou la programmation en schéma à relais, vous pouvez lire le chapitre 1-1, mais il faudra probablement compléter cette lecture par les chapitres 3, 4 et 5 en cours de route.

1-1	Configuration de l'API	2
1-2	Fonctionnement fondamental du CQM1 et processus d'Entrée/Sortie	8
1-3	Sélection et utilisation des fonctions de sortie d'impulsion	12
1-3-1	Types de sorties par impulsion	12
1-3-2	Sortie d'impulsion standard à partir d'un point de sortie	13
1-3-3	Sortie d'impulsion standard à partir des ports 1 et 2	15
1-3-4	Rapport cyclique des impulsions à partir des ports 1 et 2	24
1-3-5	Etat des ports 1 et 2	25
1-4	Sélection et utilisation des fonctions d'interruption	25
1-4-1	Types d'interruption	25
1-4-2	Interruption d'entrées	29
1-4-3	Comment masquer la totalité des interruptions	34
1-4-4	Interruptions par temporisation cyclique	34
1-4-5	Gamme/comptage non respectés pour le compteur rapide 0	37
1-4-6	Dépassements ou insuffisances du compteur rapide 0	43
1-4-7	Interruptions des compteurs rapide 1 et 2 (CQM1-CPU43-E)	45
1-4-8	Interruptions absolues de compteur rapide (CQM1-CPU44-E)	52
1-5	Fonction de communication	58
1-5-1	Configuration de l'API	59
1-5-2	Câblage du port	62
1-5-3	Communications à l'ordinateur	62
1-5-4	Communications RS-232C	64
1-5-5	Communication point par point	66

1-1 Configuration de l'API

La configuration de l'API comprend plusieurs paramètres qui contrôlent le fonctionnement de l'automate. Pour tirer le bénéfice optimal du CQM1 lors de l'utilisation des fonctions d'interruption et de communication, la configuration de l'API doit être effectuée en fonction des conditions de fonctionnement de l'API. Des valeurs par défaut sont proposées d'origine et le CQM1 peut être utilisé sans les modifier, mais il vaut mieux effectuer une vérification de ces valeurs avant de faire fonctionner l'automate.

Valeurs par défaut

Pour la configuration de l'automate, les valeurs proposées sont égales à 0000 pour tous les mots. Elles peuvent être remises à zéro à tout moment en plaçant SR 25210 à ON.

Rem.: lorsque la zone de mémoire de données (DM) est effacée d'un appareil de programmation, les sélections de configuration de l'API sont toutes remises à zéro également.

Modification de la configuration

Les changements de configuration sont effectifs et affectent le fonctionnement de l'API à différents moments selon la sélection concernée :

DM 6600 à DM 6614: lorsque l'alimentation du CQM1 est branchée
DM 6615 à DM 6644: lorsque le programme commence son exécution
DM 6645 à DM 6655: à tout moment où le CQM1 est sous tension.

Rem.: les modifications de configuration ne sont effectives qu'aux moments ci-dessus. Veillez à prendre les mesures qui s'imposent pour que les changements soient effectifs avant de poursuivre.

Bien que la configuration de l'API soit stockée dans les DM 6600 à DM 6655, les sélections peuvent être effectuées et modifiées seulement à partir d'un appareil de programmation (par ex. LSS ou console de programmation). Les DM 6600 à DM 6644 ne peuvent être sélectionnés ou modifiés qu'en mode PROGRAM. Les DM 6645 à DM 6655 peuvent être sélectionnés ou modifiés en mode PROGRAM ou MONITOR.

Rem.: la configuration de l'API peut être lue à partir du programme utilisateur mais on ne peut y effectuer d'écriture ; elle n'est possible qu'à partir d'un appareil de programmation.

Si la configuration de l'API n'est pas correcte, une erreur non fatale (code erreur 9B) est générée à la lecture du CQM1 et le bit adéquat entre AR 2400 et AR 2402 passe à ON (la sélection erronée est lue comme valeur par défaut).

Les sélections suivantes peuvent être effectuées en mode PROGRAM à partir du menu LSS. Toutes les autres sélections peuvent être effectuées en utilisant la sélection hexadécimale.

Mode lancement (DM 6600)
Etat du bit de maintien d'E/S et du bit de maintien d'état forcé (DM 6601)
Temps réel du cycle automate (DM 6618)
Temps de scrutation (DM 6619)
Paramétrage du port RS-232C (DM 6645 à DM 6649)

Contenu de la configuration

La configuration se divise en quatre phases : 1) Sélections relatives aux processus d'E/S et au fonctionnement fondamental du CQM1 2) Sélections relatives aux fonctions de sortie par impulsion 3) Sélections relatives aux interruptions 4) Sélections relatives aux communications. Ce chapitre se fonde sur ces catégories pour expliquer le fonctionnement des sélections.

La table suivante montre les diverses sélections par ordre dans la zone DM. Reportez-vous aux pages indiquées pour de plus amples détails.

Canaux	Bit(s)	Fonction
Lancement (DM 6600 à DM 6614)		
Les sélections suivantes sont effectives après transfert vers l'API et uniquement après un redémarrage de l'API.		
DM 6600	00 à 07	Mode de lancement (effectif lorsque les bits 08 à 15 sont à 02). 00: PROGRAM; 01: MONITOR 02: RUN
	08 à 15	Désignation du mode démarrage automate 00: console de programmation (mode de la console de programmation) 01: dernier mode de fonctionnement avant coupure de tension 02: sélection dans les bits 00 à 07
DM 6601	00 à 07	Réservé (sélectionné à 00).
	08 à 11	Etat des bits de maintien IOM (SR 25212) 0: RAZ ; 1: Maintien
	12 à 15	Etat des bits de maintien d'état forcé (SR 25211) 0: RAZ ; 1: Maintien
DM 6602 à DM 6610	00 à 15	Réservés
DM 6611	00 à 15	CQM1-CPU43-E: sél. de mode pour les ports 1 et 2 0000: mode du compteur rapide ; 0001: mode de sortie par impulsion CQM1-CPU44-E: sél. de compensation d'origine pour le port 1 (4 chiffres BCD)
DM 6612	00 à 15	CQM1-CPU44-E: sélection de compensation d'origine pour le port 2 (4 chiffres BCD)
Sélection du temps de scrutation et de la sortie d'impulsion (DM 6615 à DM 6619)		
Les sélections suivantes sont effectives après transfert vers l'API et après un redémarrage de l'API.		
DM 6615	00 à 07	Canaux pour la sortie impulsion. 00: IR 100; 01: IR101; 02: IR 102... 15: IR 115
	08 à 15	Réservé (sélectionné à 00).
DM 6616	00 à 07	Temps de service du port RS-232C (effectif lorsque les bits 08 à 15 sont sélectionnés à 01) 00 à 99 (BCD): pourcentage du temps de cycle utilisé pour le service du port RS-232C.
	08 à 15	Activation de la sélection de service du port RS-232C 00: ne pas sélectionner de temps de service 01: utiliser le temps des bits 00 à 07.
DM 6617	00 à 07	Temps de service du port périphérique (effectif lorsque les bits 08 à 15 sont à 01) 00 à 99 (BCD): pourcentage de temps de cycle utilisé pour le service des périphériques.
	08 à 15	Activation de la sélection du service de port périphérique. 00: ne pas sélectionner de temps de service 01: utiliser le temps des bits 00 à 07.
DM 6618	00 à 07	Temps de surveillance du cycle (effectif lorsque les bits 08 à 15 sont à 01, 02, ou 03) 00 à 99 (BCD): sélection (voir 08 à 15)
	08 à 15	Activation de la surveillance de cycle (sélection dans les bits 00 à 07 x unité; 99 s max.) 00: 120 ms (sélection dans les bits 00 à 07 désactivée) 01: unité de sélection : 10 ms 02: unité de sélection : 100 ms 03: unité de sélection : 1 s
DM 6619	00 à 15	Temps de structation 0000: variable (pas de minimum) 0001 à 9999 (BCD): temps min. en ms
Processus d'interruption (DM 6620 à DM 6639)		
Les sélections suivantes sont effectives après transfert ver l'API et après un redémarrage de l'API.		
DM 6620	00 à 03	Constante d'entrée pour IR 00000 à IR 00007 00: 8 ms; 01: 1 ms; 02: 2 ms; 03: 4 ms; 04: 8 ms; 05: 16 ms; 06: 32 ms; 07: 64 ms; 08: 128 ms
	04 à 07	Constante d'entrée pour IR 00008 à IR 00015 (sél. identique aux bits 00 à 03)
	08 à 15	Constante d'entrée pour IR 001 (sél. identique aux bits 00 à 03)
DM 6621	00 à 07	Constante d'entrée pour IR 002 (sél. identique aux bits 00 à 03 de DM 6620)
	08 à 15	Constante d'entrée pour IR 003 (sél. identique aux bits 00 à 03 de DM 6620)

Canaux	Bit(s)	Fonction
DM 6622	00 à 07	Constante d'entrée pour IR 004 (sél. identique aux bits 00 à 03 de DM 6620)
	08 à 15	Constante d'entrée pour IR 005 (sél. identique aux bits 00 à 03 de DM 6620)
DM 6623	00 à 07	Constante d'entrée pour IR 006 (sél. identique aux bits 00 à 03 de DM 6620)
	08 à 15	Constante d'entrée pour IR 007 (sél. identique aux bits 00 à 03 de DM 6620)
DM 6624	00 à 07	Constante d'entrée pour IR 008 (sél. identique aux bits 00 à 03 de DM 6620)
	08 à 15	Constante d'entrée pour IR 009 (sél. identique aux bits 00 à 03 de DM 6620)
DM 6625	00 à 07	Constante d'entrée pour IR 010 (sél. identique aux bits 00 à 03 de DM 6620)
	08 à 15	Constante d'entrée pour IR 011 (sél. identique aux bits 00 à 03 de DM 6620)
DM 6626 à DM 6627	00 à 15	Réservés
DM 6628	00 à 03	Activation d'interruption pour IR 00000 (0: entrée normale ; 1: entrée interrompue)
	04 à 07	Activation d'interruption pour IR 00001 (0: entrée normale ; 1: entrée interrompue)
	08 à 11	Activation d'interruption pour IR 00002 (0: entrée normale ; 1: entrée interrompue)
	12 à 15	Activation d'interruption pour IR 00003 (0: entrée normale ; 1: entrée interrompue)
DM 6629	00 à 07	Nombre de temporisations rapides pour le rafraîchissement d'interruption 00 à 15 (BCD; par ex., sél. 15 pour 00 à 14)
	08 à 15	Activation du rafraîchissement des tempo. rapides 00: 16 tempo. (sél. dans les bits 00 à 07 désactivée) 01: utiliser la sélection des bits 00 à 07
DM 6630	00 à 07	1er canal de rafraîchissement d'entrée pour l'interruption d'E/S 0: 00 à 11 (BCD)
	08 à 15	Nombre de canaux de rafraîchiss. d'entrée pour l'interruption d'E/S 0: 00 à 12 (BCD)
DM 6631	00 à 07	1er canal de rafraîchissement d'entrée pour l'interruption d'E/S 1: 00 à 11 (BCD)
	08 à 15	Nombre de canaux de rafraîchiss. d'entrée pour l'interruption d'E/S 1: 00 à 12 (BCD)
DM 6632	00 à 07	1er canal de rafraîchissement d'entrée pour l'interruption d'E/S 2: 00 à 11 (BCD)
	08 à 15	Nombre de canaux de rafraîchiss. d'entrée pour l'interruption d'E/S 2: 00 à 12 (BCD)
DM 6633	00 à 07	1er canal de rafraîchissement d'entrée pour l'interruption d'E/S 3: 00 à 11 (BCD)
	08 à 15	Nombre de canaux de rafraîchiss. d'entrée pour l'interruption d'E/S 3: 00 à 12 (BCD)
DM 6634	00 à 07	1er canal de rafraîchiss. d'entrée pour le compteur rapide 1: 00 à 11 (BCD)
	08 à 15	Nombre de canaux de rafraîchiss. d'entrée pour le compteur rapide 1: 00 à 12 (BCD)
DM 6635	00 à 07	1er canal de rafraîchiss. d'entrée pour le compteur rapide 1: 00 à 11 (BCD)
	08 à 15	Nombre de canaux de rafraîchiss. d'entrée pour le compteur rapide 1: 00 à 12 (BCD)
DM 6636	00 à 07	1er canal de rafraîchiss. d'entrée pour la tempo. cyclique 0: 00 à 07 (BCD)
	08 à 15	Nombre de canaux de rafraîchiss. d'entrée pour la tempo. cyclique 0: 00 à 08 (BCD)
DM 6637	00 à 07	1er canal de rafraîchiss. d'entrée pour la tempo. cyclique 1: 00 à 07 (BCD)
	08 à 15	Nombre de canaux de rafraîchiss. d'entrée pour la tempo. cyclique 1: 00 à 08 (BCD)
DM 6638	00 à 07	1er canal de rafraîchiss. d'entrée pour la tempo. cyclique 2 (également utilisé pour le compteur rapide 0): 00 à 07 (BCD)
	08 à 15	Nombre de canaux de rafraîchiss. d'entrée pour la tempo. cyclique 2 (également utilisé pour le compteur rapide 0): 00 à 08 (BCD)
DM 6639	00 à 07	Mode de rafraîchissement de sortie 00: Cyclique; 01: Direct
	08 à 15	Nombre de chiffres pour l'instruction DIGITAL SWITCH (DSW) 00: 4 chiffres; 01: 8 chiffres

Canaux	Bit(s)	Fonction
Sélections du compteur rapide (DM 6640 à DM 6644)		
Les sélections suivantes sont effectives après transfert vers l'API et après un redémarrage de l'API.		
DM 6640 à DM 6641	00 à 15	Réservés
DM 6642	00 à 03	Mode du compteur rapide 0 0: mode du compteur incrémental/décrémental (bidirectionnel) 5 kHz 4: mode du compteur incrémental (unidirectionnel) 2,5 kHz
	04 à 07	Mode de remise à zéro du compteur rapide 0: RAZ de la phase Z et RAZ par programme 1: RAZ par programme uniquement
	08 à 15	Activation du compteur rapide 0 00: ne pas utiliser le compteur rapide 01: utiliser le compteur rapide avec sélections dans les bits 00 à 07
DM 6643	00 à 03	CQM1-CPU43-E: sélection d'entrée à zéro du port 1 0: entrée en phase différentielle ; 1: entrée impulsion/direction ; 2 : entrée incrémentation/décrémentation CQM1-CPU44-E: sélection d'entrée du port 1 (résolution des bits) 0: entrée 8 bits; 1: entrée 10 bits; 2: entrée 12 bit s
	04 à 07	CQM1-CPU43-E: entrée de RAZ du port 1 0: RAZ de la phase Z et RAZ par programme 1: RAZ par programme uniquement CQM1-CPU44-E: non utilisé ; sélection à 0
	08 à 11	CQM1-CPU43-E: sélection du mode de comptage du port 1 0: mode linéaire ; 1: mode anneau CQM1-CPU44-E: sélection de mode du port 1 0: BCD mode; 1: mode 360°
	12 à 15	CQM1-CPU43-E: sélection du type d'impulsion du port 1 0: sortie d'impulsion standard (rapport cyclique 0,5); 1: sortie d'impulsion à rapport cyclique CQM1-CPU44-E: non utilisé ; sélection à 0.
DM 6644	00 à 15	Sélections du port 2 (identiques à celles du port 1 dans le DM 6643)

Canaux	Bit(s)	Fonction																																																																
Sélections du port RS-232C																																																																		
Les sélections suivantes sont effectives après transfert vers l'API et après un redémarrage de l'API.																																																																		
DM 6645	00 à 07	Sélections du port (standard) 00: Standard (1 bit de départ, donnée 7 bits, parité paire;2 bits d'arrêt, 9 600 bps) 01: Sélections dans le DM 6646																																																																
	08 à 11	Canaux de liaison pour la liaison 1:1 0: LR 00 à LR 63; 1: LR 00 à LR 31; 2: LR 00 à LR 15																																																																
	12 à 15	Mode de communication 0: liaison à l'ordinateur ; 1: RS-232C (pas de protocole); 2: 1:1 esclave de liaison de données 3: 1:1 maître de liaison de données 1:1																																																																
DM 6646	00 à 07	Vitesse 00: 1,2 Kb ; 01: 2,4 Kb ; 02: 4,8 Kb ; 03: 9,6 Kb ; 04: 19,2 Kb																																																																
	08 à 15	Format de trame <table><tr><th></th><th>Départ</th><th>Longueur</th><th>Arrêt</th><th>Parité</th></tr><tr><td>00:</td><td>1 bit</td><td>7 bits</td><td>1 bit</td><td>Paire</td></tr><tr><td>01:</td><td>1 bit</td><td>7 bits</td><td>1 bit</td><td>Impaire</td></tr><tr><td>02:</td><td>1 bit</td><td>7 bits</td><td>1 bit</td><td>Aucune</td></tr><tr><td>03:</td><td>1 bit</td><td>7 bits</td><td>2 bits</td><td>Paire</td></tr><tr><td>04:</td><td>1 bit</td><td>7 bits</td><td>2 bits</td><td>Impaire</td></tr><tr><td>05:</td><td>1 bit</td><td>7 bits</td><td>2 bits</td><td>Aucune</td></tr><tr><td>06:</td><td>1 bit</td><td>8 bits</td><td>1 bit</td><td>Paire</td></tr><tr><td>07:</td><td>1 bit</td><td>8 bits</td><td>1 bit</td><td>Impaire</td></tr><tr><td>08:</td><td>1 bit</td><td>8 bits</td><td>1 bit</td><td>Aucune</td></tr><tr><td>09:</td><td>1 bit</td><td>8 bits</td><td>2 bits</td><td>Paire</td></tr><tr><td>10:</td><td>1 bit</td><td>8 bits</td><td>2 bits</td><td>Impaire</td></tr><tr><td>11:</td><td>1 bit</td><td>8 bits</td><td>2 bits</td><td>Aucune</td></tr></table>		Départ	Longueur	Arrêt	Parité	00:	1 bit	7 bits	1 bit	Paire	01:	1 bit	7 bits	1 bit	Impaire	02:	1 bit	7 bits	1 bit	Aucune	03:	1 bit	7 bits	2 bits	Paire	04:	1 bit	7 bits	2 bits	Impaire	05:	1 bit	7 bits	2 bits	Aucune	06:	1 bit	8 bits	1 bit	Paire	07:	1 bit	8 bits	1 bit	Impaire	08:	1 bit	8 bits	1 bit	Aucune	09:	1 bit	8 bits	2 bits	Paire	10:	1 bit	8 bits	2 bits	Impaire	11:	1 bit	8 bits	2 bits
	Départ	Longueur	Arrêt	Parité																																																														
00:	1 bit	7 bits	1 bit	Paire																																																														
01:	1 bit	7 bits	1 bit	Impaire																																																														
02:	1 bit	7 bits	1 bit	Aucune																																																														
03:	1 bit	7 bits	2 bits	Paire																																																														
04:	1 bit	7 bits	2 bits	Impaire																																																														
05:	1 bit	7 bits	2 bits	Aucune																																																														
06:	1 bit	8 bits	1 bit	Paire																																																														
07:	1 bit	8 bits	1 bit	Impaire																																																														
08:	1 bit	8 bits	1 bit	Aucune																																																														
09:	1 bit	8 bits	2 bits	Paire																																																														
10:	1 bit	8 bits	2 bits	Impaire																																																														
11:	1 bit	8 bits	2 bits	Aucune																																																														
DM 6647	00 à 15	Temps de transmission (liaison à l'ordinateur) 0000 à 9999 (BCD): sélection par unité de 10 ms, ex., sélection de 0001 = 10 ms																																																																
DM 6648	00 à 07	Numéro de l'API (liaison à l'ordinateur) 00 à 31 (BCD)																																																																
	08 à 11	Activation du code de départ (RS-232C) 0: inhibé ; 1: sélectionné																																																																
	12 à 15	Activation du code de fin (RS-232C) 0: inhibé (nombre d'octets reçus) 1: sélectionné (code fin spécifié) 2: CR, LF																																																																
DM 6649	00 à 07	Code de départ (RS-232C) 00 à FF (binaire)																																																																
	08 à 15	12 à 15 du DM 6648 sélectionnés à 0: Nombre d'octets reçus 00: sélection par défaut (256 octets) 01 à FF: 1 à 255 octets 12 à 15 de DM 6648 sélectionnés à 1: Code fin (RS-232C) 00 à FF (binaire)																																																																

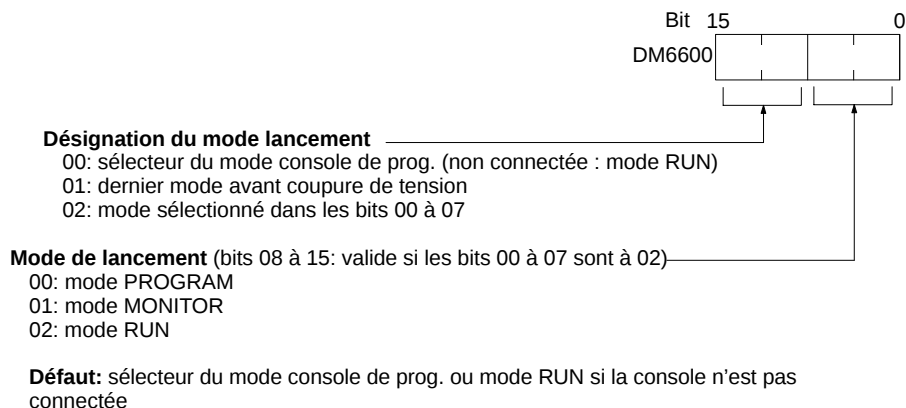
Canaux	Bit(s)	Fonction																																																																
Sélections du port périphérique																																																																		
Les sélections suivantes sont effectives après transfert vers l'API.																																																																		
DM 6650	00 à 07	Sélections de port 00: Standard (1 bit de départ, donnée à 7 bits, parité paire; 2 bits d'arrêt, 9 600 bps) 01: Sélections dans le DM 6651																																																																
	08 à 11	Réservés																																																																
	12 à 15	Mode de communication 0: liaison à l'ordinateur ; 1: RS-232C																																																																
DM 6651	00 à 07	Vitesse 00: 1,2Kb, 01: 2,4Kb, 02: 4,8Kb, 03: 9,6Kb, 04: 19,2Kb																																																																
	08 à 15	Format de trame <table><tr><th></th><th>Départ</th><th>Longueur</th><th>Arrêt</th><th>Parité</th></tr><tr><td>00:</td><td>1 bit</td><td>7 bits</td><td>1 bit</td><td>Paire</td></tr><tr><td>01:</td><td>1 bit</td><td>7 bits</td><td>1 bit</td><td>Impaire</td></tr><tr><td>02:</td><td>1 bit</td><td>7 bits</td><td>1 bit</td><td>Aucune</td></tr><tr><td>03:</td><td>1 bit</td><td>7 bits</td><td>2 bits</td><td>Paire</td></tr><tr><td>04:</td><td>1 bit</td><td>7 bits</td><td>2 bits</td><td>Impaire</td></tr><tr><td>05:</td><td>1 bit</td><td>7 bits</td><td>2 bits</td><td>Aucune</td></tr><tr><td>06:</td><td>1 bit</td><td>8 bits</td><td>1 bit</td><td>Paire</td></tr><tr><td>07:</td><td>1 bit</td><td>8 bits</td><td>1 bit</td><td>Impaire</td></tr><tr><td>08:</td><td>1 bit</td><td>8 bits</td><td>1 bit</td><td>Aucune</td></tr><tr><td>09:</td><td>1 bit</td><td>8 bits</td><td>2 bits</td><td>Paire</td></tr><tr><td>10:</td><td>1 bit</td><td>8 bits</td><td>2 bits</td><td>Impaire</td></tr><tr><td>11:</td><td>1 bit</td><td>8 bits</td><td>2 bits</td><td>Aucune</td></tr></table>		Départ	Longueur	Arrêt	Parité	00:	1 bit	7 bits	1 bit	Paire	01:	1 bit	7 bits	1 bit	Impaire	02:	1 bit	7 bits	1 bit	Aucune	03:	1 bit	7 bits	2 bits	Paire	04:	1 bit	7 bits	2 bits	Impaire	05:	1 bit	7 bits	2 bits	Aucune	06:	1 bit	8 bits	1 bit	Paire	07:	1 bit	8 bits	1 bit	Impaire	08:	1 bit	8 bits	1 bit	Aucune	09:	1 bit	8 bits	2 bits	Paire	10:	1 bit	8 bits	2 bits	Impaire	11:	1 bit	8 bits	2 bits
	Départ	Longueur	Arrêt	Parité																																																														
00:	1 bit	7 bits	1 bit	Paire																																																														
01:	1 bit	7 bits	1 bit	Impaire																																																														
02:	1 bit	7 bits	1 bit	Aucune																																																														
03:	1 bit	7 bits	2 bits	Paire																																																														
04:	1 bit	7 bits	2 bits	Impaire																																																														
05:	1 bit	7 bits	2 bits	Aucune																																																														
06:	1 bit	8 bits	1 bit	Paire																																																														
07:	1 bit	8 bits	1 bit	Impaire																																																														
08:	1 bit	8 bits	1 bit	Aucune																																																														
09:	1 bit	8 bits	2 bits	Paire																																																														
10:	1 bit	8 bits	2 bits	Impaire																																																														
11:	1 bit	8 bits	2 bits	Aucune																																																														
DM 6652	00 à 15	Temps de transmission (liaison à l'ordinateur) 0000 à 9999: en ms.																																																																
DM 6653	00 à 07	Numéro d'API (liaison à l'ordinateur) 00 à 31 (BCD)																																																																
	08 à 11	Activation du code de départ (RS-232C) 0: inhibé; 1: sélectionné																																																																
	12 à 15	Activation du code de fin (RS-232C) 0: inhibé (nombre d'octets reçus) 1: sélectionné (code fin spécifié) 2: CR, LF																																																																
DM 6654	00 à 07	Code de départ (RS-232C) 00 à FF (binaire)																																																																
	08 à 15	12 à 15 de DM 6653 sélectionnés à 0: Nombre d'octets reçus 00: sélection par défaut (256 octets) 01 à FF: 1 à 255 octets 12 à 15 de DM 6653 sélectionnés à 1: Code fin (RS-232C) 00 à FF (binaire)																																																																

Canaux	Bit(s)	Fonction
Sélections de l'enregistrement des erreurs (DM 6655)		
Les sélections suivantes sont effectives après transfert vers l'API.		
DM 6655	00 à 03	Style 0: décalage après 10 enregistrements d'erreurs 1: stockage des 10 enregistrements seulement (pas de décalage) 2 à F: pas d'enregistrement des erreurs
	04 à 07	Réservés
	08 à 11	Activation de la surveillance de temps de cycle 0: détection de cycles longs comme erreur non fatale 1: pas de détection des cycles longs
	12 à 15	Activation de l'erreur pile défaillante 0: détection de défaillance de pile comme erreur non fatale 1: pas de détection de la défaillance de pile

1-2 Fonctionnement fondamental CQM1 et processus d'E/S

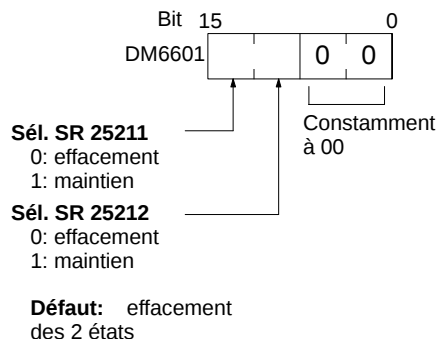
Ce chapitre détaille les sélections de configuration de l'API afférentes au fonctionnement fondamental de l'API et aux processus d'E/S.

Mode lancement (DM 6600) Le premier mode de fonctionnement du CQM1 à la mise sous tension peut être sélectionné de la façon suivante.



Etat du bit de maintien (DM 6601)

Effectuez les sélections ci-dessous afin de déterminer si les bits de maintien d'état forcé (SR 25211) et/ou les bits de maintien IOM (SR 25212) doivent retenir l'état effectif des bits à la coupure de tension ou si cet état doit être effacé.

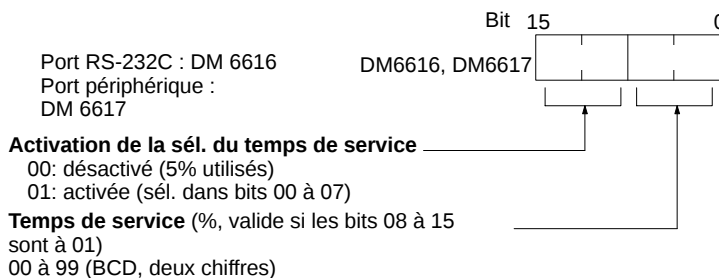


Le bit de maintien d'état forcé (SR 25211) détermine si l'état forcé de sélection/remise à zéro est maintenu lors du passage du mode PROGRAM en mode MONITOR.

Temps de service des ports RS-232C et périphériques (DM 6616 et DM 6617)

Le bit de maintien IOM (SR 25212) détermine si l'état des bits IR et LR est maintenu lorsque le CQM1 est arrêté et remis sous tension.

Les sélections suivantes s'utilisent pour déterminer le pourcentage du cycle de temps dévolu au service des ports RS-232C et périphériques.



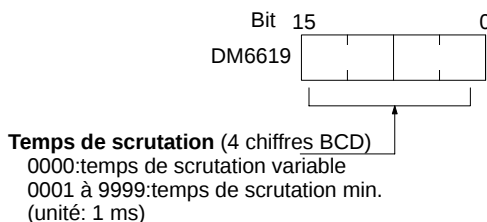
Défaut: 5% du temps de scrutation

Exemple: si le DM 6616 est à 0110, le port RS-232C est en service pendant une période équivalente à 10% du temps de cycle. Si le DM 6617 est à 0115, le port périphérique est en service pendant une période équivalente à 15 % du temps de cycle.

Le temps de service est de 0,34 ms minimum.

Le temps de service n'est utilisé en entier qu'une fois si une demande de traitement a été effectuée.

Temps de scrutation (DM 6619) Effectuez les sélections suivantes pour standardiser le temps de scrutation et éliminer les variations du temps de réponse des E/S par sélection d'un temps de cycle min.



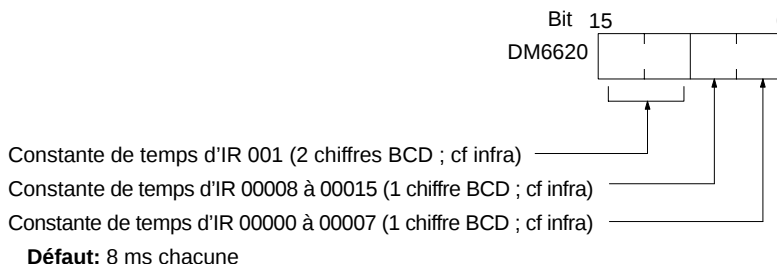
Défaut: temps de scrutation variable

Si le véritable temps de scrutation est inférieur au temps de scrutation minimum, l'exécution est suspendue jusqu'à expiration du temps minimum. Si le véritable temps de scrutation est supérieur au temps de scrutation minimum, le fonctionnement s'effectue en fonction du véritable temps de scrutation. AR 2405 passe à ON si le temps de scrutation est dépassé.

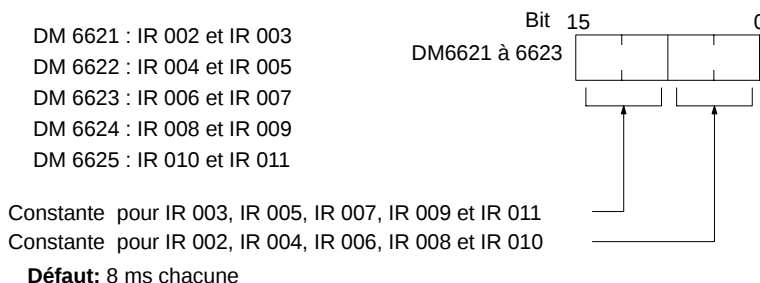
Constantes du temps d'entrée (filtrage) (DM 6620 à DM 6625)

Effectuez les sélections ci-dessous pour choisir le temps qui s'écoule entre le moment où les véritables entrées provenant de la carte d'entrée c.c. passent à ON ou à OFF et le moment où les bits d'entrée correspondants sont remis à jour (c'est-à-dire jusqu'au changement d'état ON/OFF). Effectuez ces sélections lorsque vous voulez régler le temps de stabilisation de l'entrée.

Constantes du filtrage d'entrée IR 000 et IR 001



Constantes de filtrage d'entrée pour IR 002 à IR 011

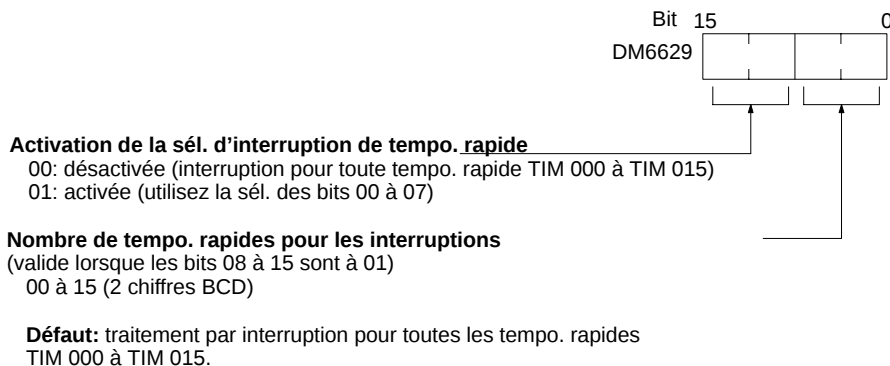


Voici les neuf sélections possibles pour la constante de temps d'entrée. Sélectionnez uniquement le chiffre d'extrême droite pour IR 000.

00: 8 ms	01: 1 ms	02: 2 ms	03: 4 ms	04: 8 ms
05: 16 ms	06: 32 ms	07: 64 ms	08: 128 ms	

Tempo. rapide (DM 6629)

Effectuez les sélections suivantes pour choisir le nombre de tempo. rapides créées avec TIMH(15) et utilisant un traitement par interruption.



La sélection indique le nombre de tempo. utilisant un processus d'interruption commençant par TIM 000. Par exemple, si "0108" est spécifié, huit tempo., TIM 000 à TIM 007 utilisent un processus d'interruption.

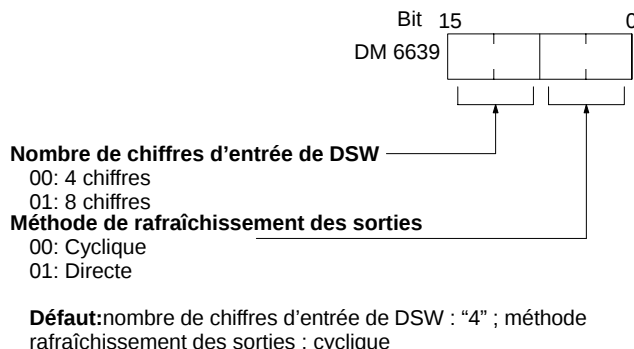
Rem.: les tempo. rapides ne peuvent être précises sans processus d'interruption, sauf le temps de cycle est de 10 ms maximum.

Le temps de réponse pour les autres interruptions est amélioré si le processus d'interruption est sélectionné à 00 alors que le traitement par tempo. rapide n'a pas été demandé. Cela est valable pour tout temps de cycle inférieur à 10 ms.

Rem.: si l'on utilise l'instruction SPED(—) et que les impulsions sont sorties à une fréquence de 500 Hz au moins, choisissez alors un nombre de tempo. rapides à interruption égale à 4 maximum. Pour de plus amples détails, reportez-vous à l'instruction SPED(—).

Chiffres d'entrée de DSW et méthode de rafraîchissement des sorties (DM 6639)

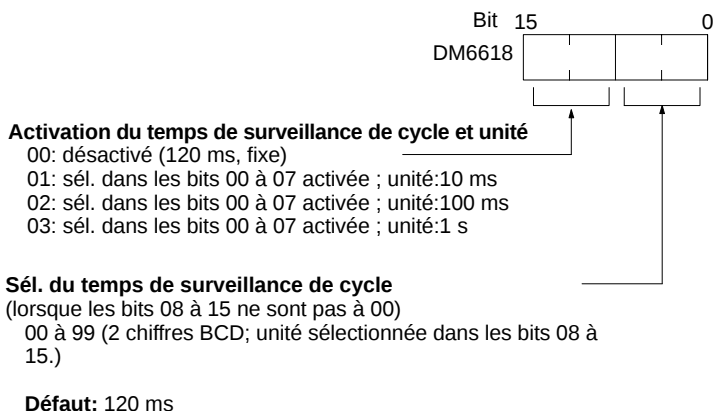
Effectuez les sélections ci-dessous pour choisir le nombre de chiffres d'entrée de l'instruction DSW et pour sélectionner la méthode de rafraîchissement des sorties.



Reportez-vous au chapitre 3 pour les détails concernant l'instruction DSW et au chapitre 6 pour la méthode de rafraîchissement des E/S.

Sélection de l'enregistrement des erreurs

Détection et stockage des erreurs :

Temps de surveillance de cycle (DM 6618)

Le temps de surveillance de cycle s'utilise pour vérifier les temps de scrutation très longs, comme ceux que l'on peut rencontrer en cas de programme en boucle infinie. Si le temps de scrutation dépasse la valeur de sélection de surveillance, une erreur fatale (FALS 9F) est générée.

Rem.: 1. L'unité de temps utilisée pour le temps de cycle en cours et maximum enregistrée dans les canaux AR 26 et AR 27 change selon l'unité sélectionnée pour le temps de surveillance de cycle.

Bits 08 à 15 sélectionnés à 00 ou 01:	0,1 ms
Bits 08 à 15 sélectionnés à 02:	1 ms
Bits 08 à 15 sélectionnés à 03:	10 ms

2. Même si le temps de cycle est supérieur ou égal à 1 s, le temps de cycle lu à partir des appareils de programmation ne dépasse pas 999,9 ms. L'unique temps de cycle en cours et maximum correct est enregistré dans les canaux AR 26 et AR 27.

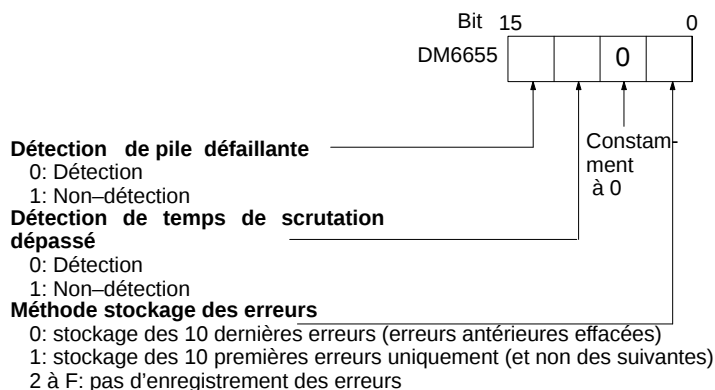
Exemple

Si 0230 est sélectionné dans le DM 6618, une erreur FALS 9F n'apparaît que si le temps de cycle dépasse 3 s. Si le temps réel est de 2,59 s, le contenu de AR 27 est égal à 2590 (ms), mais le temps de cycle lu à partir des appareils de programmation est de 999,9 ms.

Une erreur "cycle time over" (non fatale) est générée lorsque le temps de cycle dépasse 100 ms, sauf si la détection de temps de cycle long est inhibée à l'aide de la sélection du mot DM 6655.

Détection et stockage des erreurs (DM 6655)

Effectuez les sélections ci-dessus de façon à générer ou non une erreur non fatale si le temps de cycle dépasse 100 ms ou si la tension de la pile intégrée est défectueuse et de façon à sélectionner la méthode d'enregistrement des erreurs.



Défaut: erreurs de pile défectueuse et de temps de scrutation dépassé détectées et 10 dernières erreurs stockées.

Les erreurs de pile de temps de cycle dépassé sont des erreurs non fatales. Pour de plus amples détails concernant le stockage des erreurs, reportez-vous au chapitre 7.

1-3 Sélection et utilisation des fonctions de sortie d'impulsion

Reportez-vous au manuel d'utilisation du CQM1 pour de plus amples détails concernant les connexions du matériel aux ports et points de sortie du CQM1.

1-3-1 Types de sortie par impulsion

Tous les CQM1 peuvent produire des impulsions standard à partir d'un bit de sortie ; la CQM1-CPU43-E peut également produire des impulsions standard ou à rapport cyclique à partir des ports 1 et 2. Les impulsions standard ont un rapport cyclique de (t_{on}/T) de 50%. Le rapport cyclique des sorties par impulsion à rapport cyclique peut être sélectionné entre 1 et 99 % par incréments de 1 %.

Rem. : avec la CQM1-CPU43-E, les sorties par impulsion décrites ci-après peuvent être produites à partir de 3 ports à la fois. En outre, 2 ports peuvent être utilisés pour les entrées par compteur indépendantes de la sortie par impulsion.

Sortie par impulsion standard à partir d'un point de sortie

Les impulsions standard (rapport cyclique = 50%) peuvent être produites à partir d'un point de sortie à une fréquence de 20 Hz à 1 kHz. Le canal d'E/S est spécifié dans la configuration de l'API et le bit est spécifié dans l'instruction de sortie par l'impulsion elle-même (cf. chapitre 1-3-2).

Sortie par impulsion standard à partir des ports 1 et 2

Avec la CQM1-CPU43-E, les impulsions standard (rapport cyclique = 50%) peuvent être produites à partir des ports 1 et/ou 2 à une fréquence de 10 à 50 kHz (20 kHz max. pour un moteur pas à pas). La sortie par impulsion peut être effectuée en sens horaire ou anti-horaire et les modifications de fréquence peuvent être introduites en souplesse.

Sortie par impulsion à rapport cyclique à partir des ports 1 et 2

PLS2(—) et le mode 0 de ACC(—) ne peuvent être utilisés lorsque la configuration de l'API (DM 6611) est positionnée en mode compteur rapide.

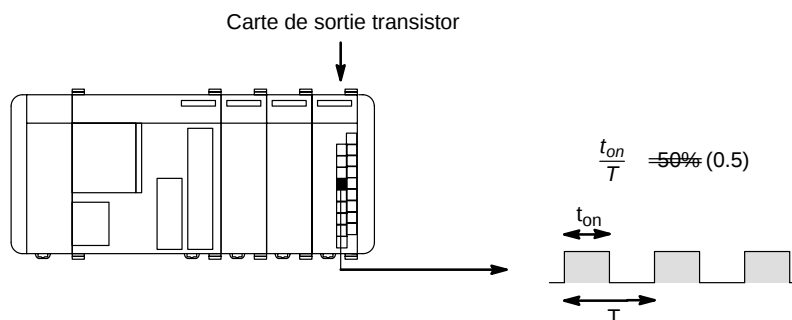
CTBL (--) ne peut être utilisée avec les ports 1 et 2 lorsque la configuration de l'API (DM 6611) est positionnée en mode de sortie par impulsion. Reportez-vous au chapitre 1-3-3.

Avec la CQM1-CPU43-E, les impulsions à rapport cyclique peuvent être produites (rapport cyclique = 0 à 99%) à partir des ports 1 et/ou 2 à des fréquences de 91,6 Hz, 1,5 kHz ou 5,9 kHz. Un seul sens de sortie est possible et la sortie par impulsion se poursuit jusqu'à ce qu'elle soit arrêtée par INI(--).

Reportez-vous au chapitre 1-3-4.

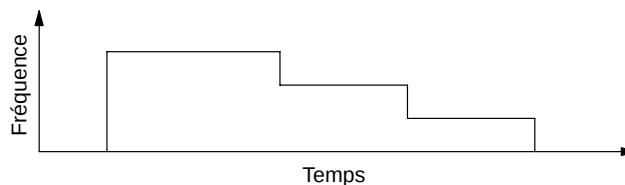
1-3-2 Sortie d'impulsion standard à partir d'un point de sortie

Les impulsions peuvent être produites à partir d'un bit de sortie spécifié à l'aide de SPED(—). Les impulsions peuvent être produites à partir d'un seul bit à la fois. Le schéma suivant illustre les impulsions produites à partir du point de sortie de la carte de sortie transistor. Le rapport cyclique de la sortie par impulsion est de 50% et la fréquence peut être sélectionnée entre 20 Hz et 1 kHz.



- Rem.: 1. La carte de sortie transistor est indispensable pour cette application.
 2. Les impulsions ne peuvent être produites lorsque la tempo. cyclique 0 fonctionne.
 3. Lorsqu'une sortie par impulsion supérieure à 500 Hz est produite, sélectionnez le nombre de tempo. rapides avec interruption à 4, par positionnement des DM 6629 à 0104.

Lors de la production d'impulsions à partir d'un point de sortie, la fréquence peut être modifiée par étapes en exécutant SPED(—) à nouveau à différentes fréquences, comme ci-après :



Il y a deux façons d'arrêter la sortie d'une impulsion :

- 1, 2, 3...**
1. Après exécution de SPED(—), la sortie par impulsion s'arrête si INI(—) est exécutée avec C=003 ou si SPED(—) est exécutée à nouveau avec une fréquence égale à 0.
 2. Le nombre total d'impulsions à sortir peut être sélectionné à l'aide de PULS(—) avant l'exécution de SPED(—). Dans ce cas, SPED(—) doit être exécutée en mode indépendant. La sortie d'impulsions s'arrête automatiquement lorsque le nombre d'impulsions sélectionné par PULS(—) a été atteint.

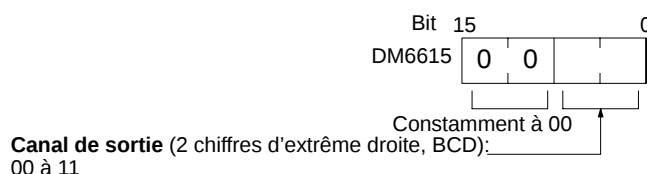
Rem.: pour de plus amples détails, reportez-vous aux chapitres concernant les instructions SPED(—) et PULS(—).

Sélections de la configuration API

Avant l'exécution de SPED(—) pour la sortie d'impulsions à partir d'une carte de sortie, placez l'API en mode PROGRAM et effectuez les sélections suivantes en configuration :

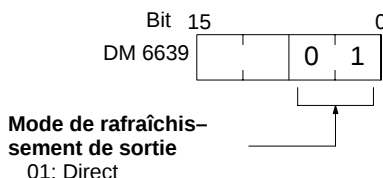
Dans le DM 6615, spécifiez le mot de sortie utilisé pour la sortie d'impulsions SPED(—) vers les cartes de sortie (le bit est spécifié dans la première opérande de SPED(—)).

Le contenu des DM 6615 (0000 à 0011) spécifie les canaux de sortie IR 100 à IR 111. Par exemple, si le DM 6615 est sélectionné à 0002, les impulsions seront sorties dans IR 102.



Défaut: impulsion sortie dans IR 100 (canal 100).

Avec les UC CQM1-CPU11/21-E, sélectionnez le rafraîchissement de sortie direct dans le DM 6639, comme ci-dessous (avec les UC CQM1-CPU4 ☐-E, le mode de rafraîchissement de sortie peut être direct ou cyclique.



Défaut: mode de rafraîchissement cyclique

Sortie d'impulsions continues Les impulsions commencent à être sorties sur le bit de sortie spécifié lorsque SPED(—) est exécutée. Positionnez le bit de sortie entre 00 et 15 (D=000 à 150) et la fréquence entre 20 Hz et 1000 Hz (F=0002 à 0100). Choisissez le mode continu (M=001).



La sortie d'impulsions peut être arrêtée en exécutant INI(—) avec C=003 ou en ré-exécutant SPED(—), la fréquence étant à 0. La fréquence peut être modifiée en ré-exécutant SPED(—) avec une fréquence différente.

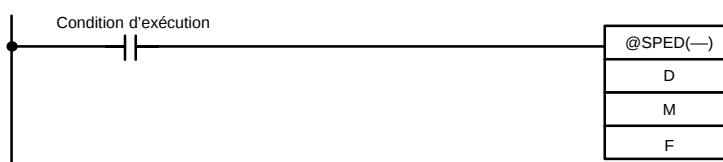
Sélection du nombre d'impulsions

Le nombre total d'impulsions peut être sélectionné à l'aide de PULS(—) avant l'exécution de SPED(—) en mode indépendant. La sortie des impulsions

s'arrête automatiquement lorsque le nombre d'impulsions sélectionné par PULS(—) a été atteint.



PULS(—) sélectionne le nombre d'impulsions (8 chiffres) P1+1, P1. Ces impulsions vont de 00000001 à 16777215. On accède au nombre d'impulsions sélectionné avec PULS(—) lorsque SPED(—) est exécutée en mode indépendant (le nombre d'impulsions ne peut pas être modifié pour les impulsions en cours de sortie).



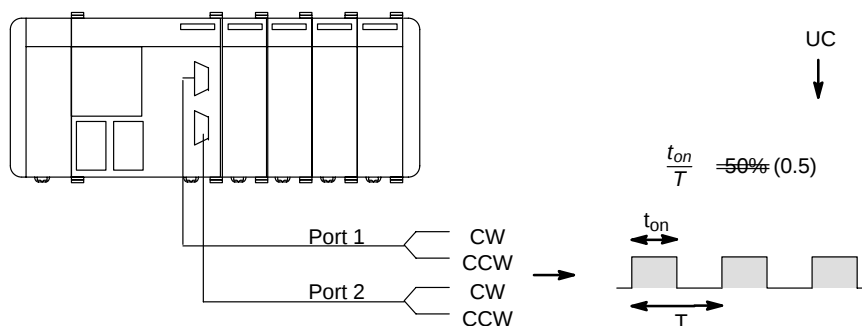
Lorsque SPED(—) est exécutée, les impulsions commencent à être sorties sur le bit de sortie spécifié (D=000 à 150: bit 00 à 15) à la fréquence spécifiée (F=0002 à 0100: 20 Hz à 1000 Hz). Choisissez le mode indépendant (M=001) pour sortir le nombre d'impulsions sélectionné à l'aide de PULS(—). La fréquence peut être modifiée en ré-exécutant SPED(—) avec une fréquence différente.

Changement de fréquence

La fréquence de la sortie d'impulsion peut être modifiée en ré-exécutant SPED(—) à une fréquence différente. Utilisez les mêmes sélections de bit de sortie (P) et de mode (M) utilisées pour le début de sortie des impulsions. La nouvelle fréquence doit être située entre 20 Hz et 1000 Hz (F=0002 à 0100).

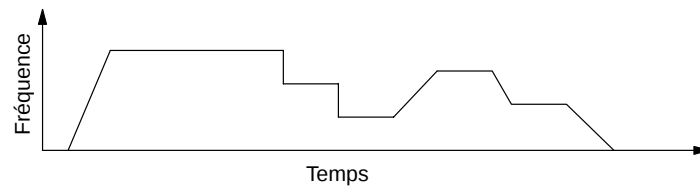
1-3-3 Sortie d'impulsion standard à partir des ports 1 et 2

Avec l'UC CQM1-CPU43-E, on peut sortir des impulsions standard à partir des ports 1 et 2 à l'aide de SPED(—), PLS2(—) ou ACC(—). La fréquence par impulsions doit être sélectionnée entre 10 Hz et 50 kHz (20 kHz max. pour un moteur pas à pas). La sortie d'impulsion peut se faire en sens horaire (CW) ou anti-horaire (CCW) et les modifications de fréquence peuvent être effectuées en douceur.



Rem. : seule l'UC CQM1-CPU43-E peut sortir des impulsions à partir des ports 1 et 2.

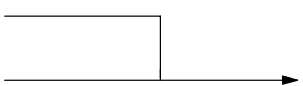
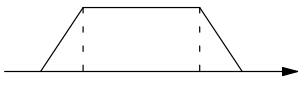
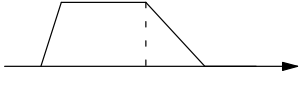
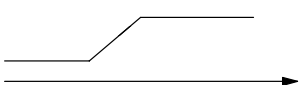
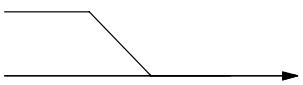
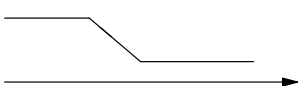
Lors de la sortie d'impulsions à partir d'un port, la fréquence peut être modifiée en douceur ou par étapes avec SPED(—), PLS2(—) et ACC(—) :



Il y a deux façons d'arrêter la sortie d'impulsions :

- 1, 2, 3...**
1. Après l'exécution de SPED(—), la sortie par impulsion s'arrête si INI(—) est exécutée avec C=003 ou si SPED(—) est ré-exécutée avec une fréquence sélectionnée à 0.
 2. Le nombre total d'impulsions sorties peut être sélectionné avec PULS(—) avant l'exécution de SPED(—). Dans ce cas, SPED(—) doit être exécutée en mode indépendant. La sortie d'impulsions s'arrête automatiquement lorsque le nombre d'impulsions sélectionné par PULS(—) a été atteint.

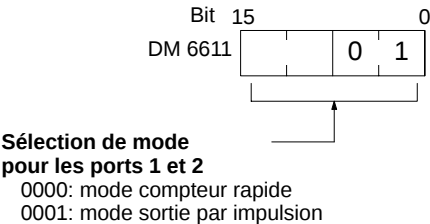
Changements de fréquence pouvant être effectués en combinant PULS(65), SPED(64), INI(61), PLS2(—) et ACC(—) :

Changements de fréquence	Instruction	Sélection d'opérande
	PULS(—)	CW/CCW (Nombre d'impulsions)
	SPED(—)	Port Mode Fréquence
	SPED(—) ou INI(—)	Port Fréquence= 0
		Mot de contrôle=0
	PLS2(—)	Port sens CW/CCW Vitesse d'accélération Fréquence spécifiée Nombre d'impulsions
	PULS(—) + ACC(—) (Mode 0)	Sens CW/CCW Nombre d'impulsions Point de décélération
		Port Taux d'accélération Fréquence spécifiée1 Vitesse de décélération Fréquence spécifiée 2
	PULS(—) ACC(—) (Mode 1)	CW/CCW
		Port Vitesse d'accélération Fréquence spécifiée
	PULS(—) + ACC(—) (Mode 2)	CW/CCW Nombre d'impulsions
		Port Vitesse de décélération Fréquence spécifiée
	PULS(—) + ACC(—) (Mode 3)	CW/CCW
		Port Vitesse de décélération Fréquence spécifiée

La CPU43 permet de gérer 2 axes en boucle ouverte ou fermée suivant le DM 6611. Les ports 1 et 2 peuvent compter des impulsions et générer des impulsions pour piloter un moteur pas à pas à l'aide des instructions PULS-SPED.

PULS2 et ACC en mode 0 ne peuvent être utilisées. En mode sortie "pulse" (DM 6611 = 0001), les ports 1 et 2 génèrent des rampes programmées sans possibilité de comptage puisque l'instruction CTBL n'est plus disponible.

Sélections de configuration Avant de produire des impulsions à partir des ports 1 et 2, placez l'API en mode PROGRAM et effectuez les sélections suivantes en configuration :
Dans le DM 6611, spécifiez la sélection de mode pour les ports 1 et 2 :



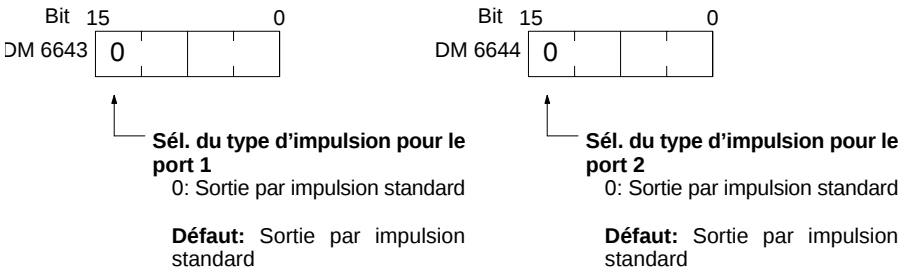
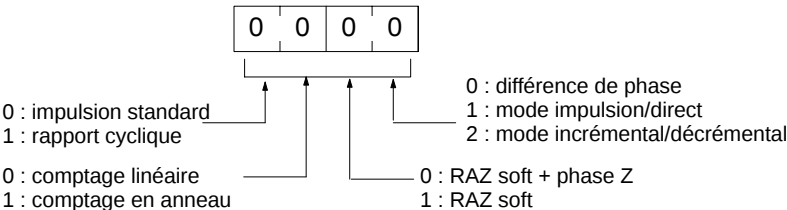
Défaut: mode compteur rapide

Certaines instructions peuvent être utilisées ou non selon la sélection dans le DM 6611.

Sélection DM 6611	Instructions affectées
Mode compteur rapide (0000)	PLS2(—) et le mode 0 de ACC(—) ne peuvent être utilisés.
Mode de sortie par impulsion (0001)	CTBL(—) ne peut être utilisée avec les ports 1 et 2.

La sélection du DM 6611 ne peut être lue que lorsque le CQM1 est en fonctionnement. Si cette sélection est modifiée, coupez et remettez l'API sous tension pour que la nouvelle sélection soit prise en compte.

Spécifiez les sorties par impulsion standard dans les DM 6643 (port 1) et/ou DM 6644 (port 2).



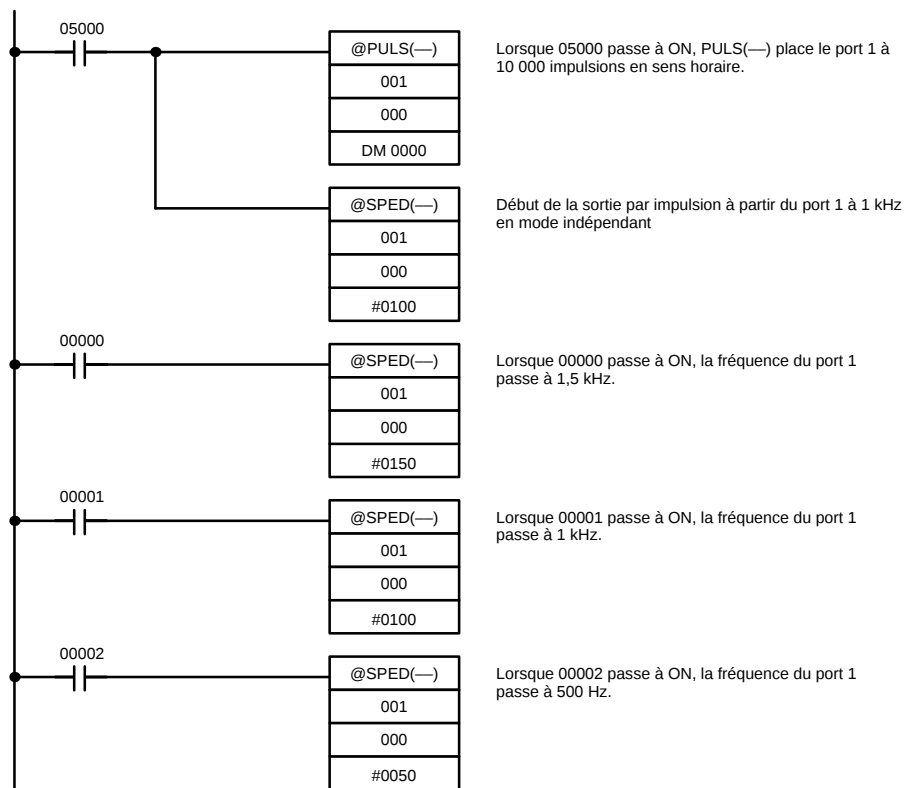
Le rapport cyclique des impulsions ne peut être sorti à partir d'un port qui aurait été sélectionné pour une sortie par impulsion standard dans les DM 6643 ou DM 6644.

Exemple 1:
Démarrage de sortie par impulsion avec PULS(—) et SPED(—)

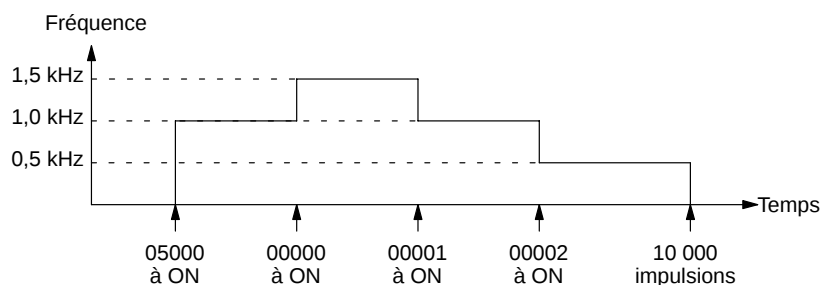
L'exemple suivant montre l'utilisation de PULS(—) et SPED(—) dans le contrôle d'impulsion à partir du port 1. Le nombre d'impulsions spécifié dans

PULS(—) (10 000) est sorti lors du changement de fréquence par l'exécution de SPED(—) avec une sélection de fréquence différente.

Avant l'exécution du programme, assurez-vous que le DM6611 soit placé à 0000 (sélection d'impulsion standard pour le port 1).



Le schéma suivant montre la fréquence des sorties par impulsion à partir du port 1, selon l'exécution du programme :



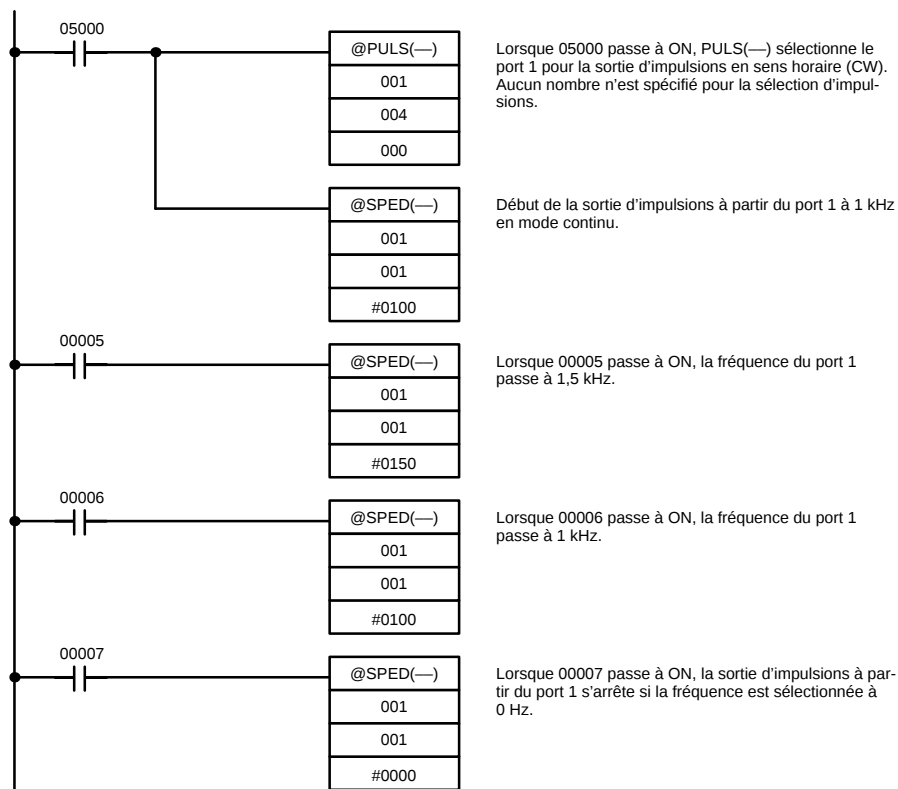
Rem. : veillez à ce que la fréquence d'impulsion soit comprise dans la gamme de fréquences compatible avec la vitesse maximum du moteur.

Attention : la synchronisation du contrôle de vitesse est très précise lorsque les modifications de fréquence sont effectuées en tant que traitements par interruption d'entrée.

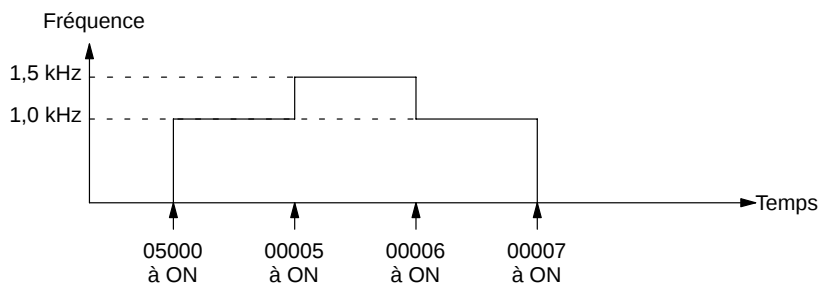
Exemple 2: arrêt de sortie d'impulsion par SPED(—)

L'exemple suivant montre l'utilisation de PULS(—) et SPED(—) pour le contrôle d'une sortie d'impulsion à partir du port 1. La fréquence est modifiée par l'exécution

tion de SPED(—) avec des sélections de fréquence différentes, puis arrêtée avec une sélection de fréquence égale à 0.



Le schéma suivant montre la fréquence des sorties d'impulsions à partir du port 1 pendant l'exécution du programme :

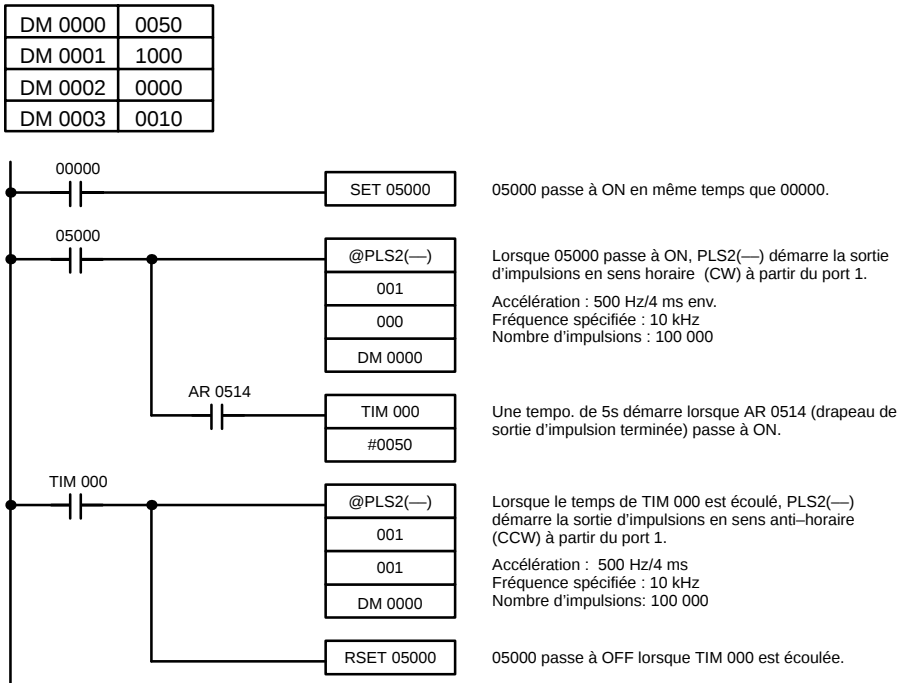


Attention : veillez à ce que la fréquence d'impulsion soit comprise dans la gamme de fréquences compatible avec la vitesse maximum du moteur.

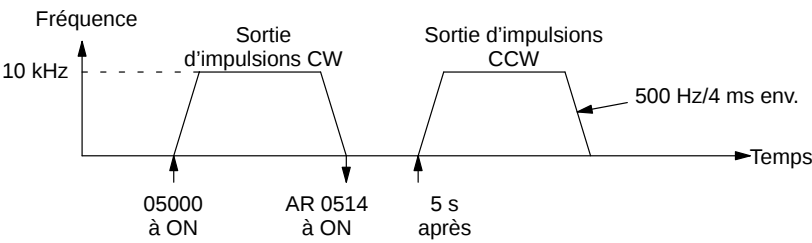
Exemple 3: PLS2(—)

L'exemple suivant montre PLS2(—) utilisées pour sortir 100 000 impulsions en sens horaire à partir du port 1. La fréquence s'accélère jusqu'à 10 kHz au rythme de 500 Hz/4 ms environ et la décélération s'effectue à la même vitesse.

Cinq secondes après la sortie des impulsions en sens horaire (CW), une nouvelle instruction PLS2(—) sort 100 000 impulsions en sens anti-horaire (CCW) avec les mêmes sélections.



Le schéma suivant montre la fréquence des sorties d'impulsions à partir du port 1 pendant l'exécution du programme :

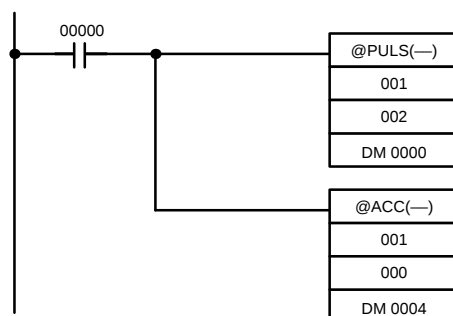


Exemple 4: ACC(—) mode 0 L'exemple suivant illustre le mode 0 de ACC(—), utilisé pour la sortie de 10 000 impulsions en sens horaire à partir du port 1. La fréquence s'accélère jusqu'à

10 kHz à 1 kHz/4 ms env. et décélère jusqu'à 1 kHz à 250 Hz/4 ms env. La décélération commence après la sortie de 9 100 impulsions.

DM 0000	0000
DM 0001	0001
DM 0002	9100
DM 0003	0000

DM 0004	0100
DM 0005	1000
DM 0006	0025
DM 0007	0050

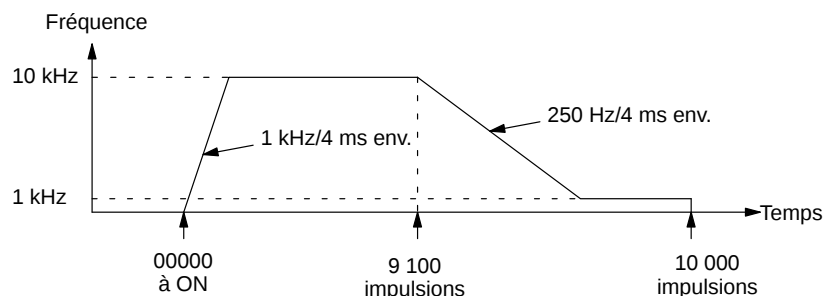


Lorsque 00000 passe à ON, PULS(---) place le port 1 en sortie d'impulsions en sens horaire. Le nombre total d'impulsions est sélectionné à 10 000 et le point de décélération est sélectionné à 9 100 impulsions.

Début de la sortie d'impulsions en sens horaire (CW) à partir du port 1.

Accélération : 1000 Hz/4 ms env.
Fréquence après accélération : 10 kHz
Décélération : 250 Hz/4 ms env.
Fréquence après décélération: 1 kHz

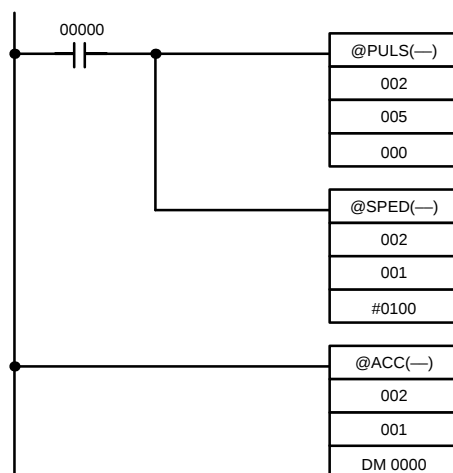
Le schéma suivant montre la fréquence des sorties d'impulsions à partir du port 1 pendant l'exécution du programme :



Exemple 5: ACC(---) mode 1

L'exemple suivant illustre le mode 1 de ACC(---), utilisé pour augmenter la fréquence d'une sortie d'impulsions à partir du port 1. La fréquence s'accélère à partir de 1 kHz jusqu'à 20 kHz à 500 Hz/4 ms env.

DM 0000	0050
DM 0001	2000

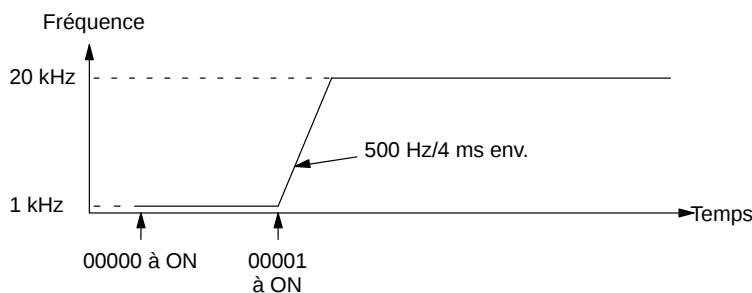


Lorsque 00000 passe à ON, PULS(---) sélectionne le port 2 pour la sortie d'impulsions en sens anti-horaire (CCW). Aucun nombre d'impulsions n'est sélectionné.

Début de la sortie d'impulsions de 1 kHz à partir du port 2 en mode continu.

Lorsque 00001 passe à ON, ACC(---) commence l'accélération de la sortie d'impulsions du port 2 à 500 Hz/4 ms env. jusqu'à atteindre la fréquence spécifiée de 20 kHz.

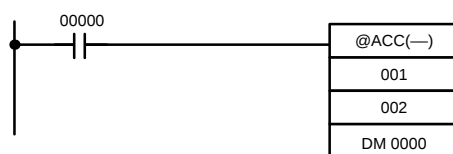
Le schéma suivant montre la fréquence des sorties d'impulsions à partir du port 2 pendant l'exécution du programme :



Exemple 6: ACC(---) mode 2

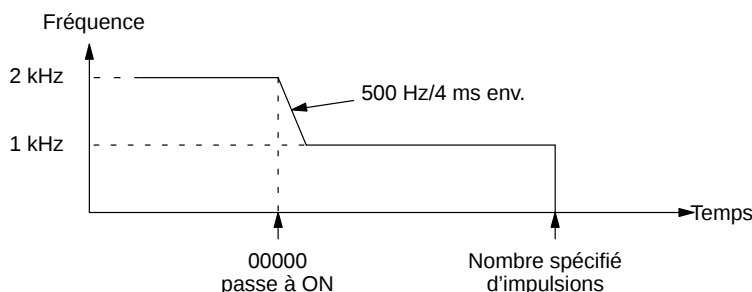
L'exemple suivant illustre le mode 2 de ACC(---), utilisé pour diminuer la fréquence de la sortie d'impulsions à partir du port 1. La sortie d'impulsions de 2 kHz est déjà en cours en mode indépendant et s'arrête automatiquement lorsque le nombre d'impulsions a été atteint.

DM 0000	0050
DM 0001	0001



Lorsque 00000 passe à ON, ACC(---) commence la décélération de la sortie d'impulsions à partir du port 1 à 500 Hz/4 ms env. jusqu'à la fréquence spécifiée de 10 Hz.

Le schéma suivant illustre la fréquence des sorties d'impulsions à partir du port 1 pendant l'exécution du programme :

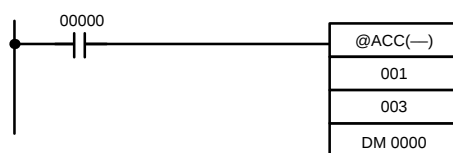


Attention : la sortie d'impulsions peut être arrêtée par l'exécution de ACC(---) en mode 2 avec une fréquence spécifiée de 0, mais la sortie d'impulsions ne peut être vraiment arrêtée à un nombre précis d'impulsions ; cette méthode doit donc être réservée aux cas d'urgence.

Exemple 7: ACC(---) mode 3

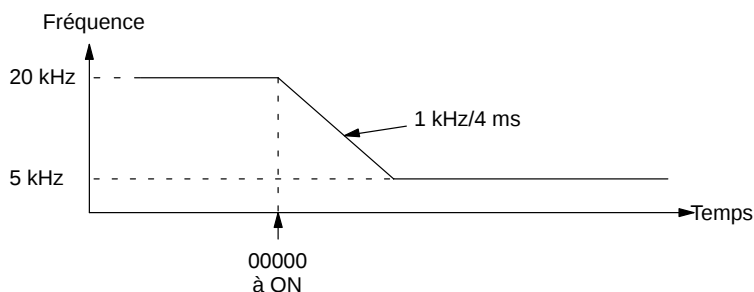
L'exemple suivant illustre le mode 3 de ACC(---), utilisé pour diminuer la fréquence d'une sortie d'impulsion à partir du port 1. La sortie d'impulsion de 20 kHz est déjà en cours en mode continu :

DM 0000	0100
DM 0001	0500



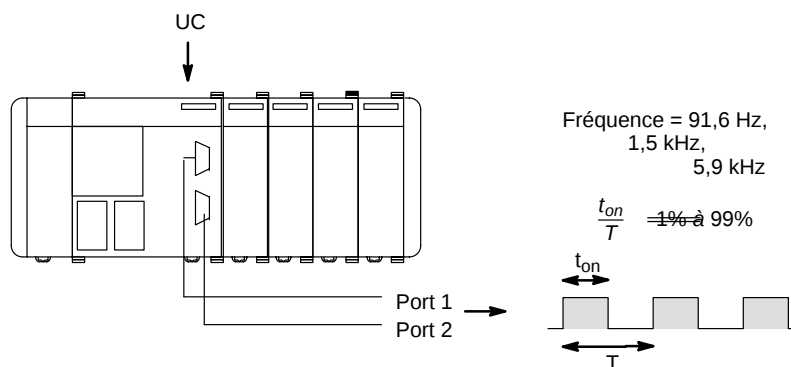
Lorsque 00000 passe à ON, ACC(---) commence la décélération de la sortie d'impulsions du port 1 à 1 kHz/4 ms env. jusqu'à la fréquence spécifiée de 5 kHz.

Le schéma suivant illustre la fréquence des sorties d'impulsions à partir du port 1 pendant l'exécution du programme :



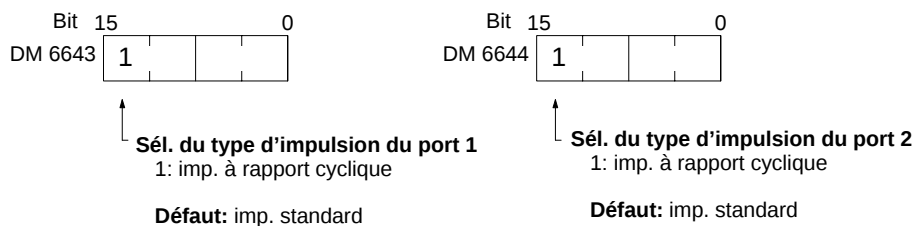
1-3-4 Rapport cyclique des impulsions à partir des ports 1 et 2

Avec la CQM1-CPU43-E, les rapports cycliques peuvent être sortis à partir des ports 1 et/ou 2 à l'aide de PWM(—). La fréquence d'impulsion peut être sélectionnée à 91,6 Hz, 1,5 kHz ou 5,9 kHz. Cette fonction peut être utilisée pour différentes sorties de contrôle comme les sorties d'intensité de lumière ou de contrôle de vitesse pour un variateur de fréquence.



Attention : seule l'UC CQM1-CPU43-E peut produire des impulsions à partir des ports 1 et 2.

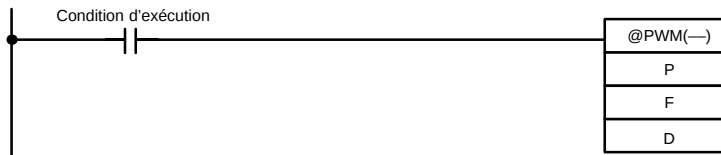
Configuration des paramètres Avant d'utiliser le rapport cyclique à partir des ports 1 ou 2, placez l'API en mode PROGRAM et effectuez les sélections suivantes dans la configuration de l'API : Spécifiez le rapport cyclique d'impulsion dans les DM 6643 (port 1) et/ou DM 6644 (port 2).



Les impulsions standard ne peuvent être produites à partir d'un port s'il a été sélectionné pour le rapport cyclique d'impulsion dans les DM 6643 ou DM 6644.

Début de sortie d'impulsions Les impulsions commencent à être sorties à partir du port spécifié lorsque PWM(—) est exécuté. Spécifiez les ports 1 ou 2 (P=001 à 002). Sélectionnez la

fréquence à 5,9 kHz, 1,5 kHz ou 91,6 Hz (F=000, 001 ou 002). Sélectionnez le rapport cyclique entre 1 et 99 % (D=0001 à 0099, BCD).



La sortie d'impulsion continue à la fréquence spécifiée et au rapport cyclique spécifié jusqu'à ce que PWM(—) soit ré-exécutée avec des sélections différentes ou que INI(—) soit exécutée pour arrêter la sortie d'impulsions à partir du port spécifié.

Arrêt de sortie d'impulsions

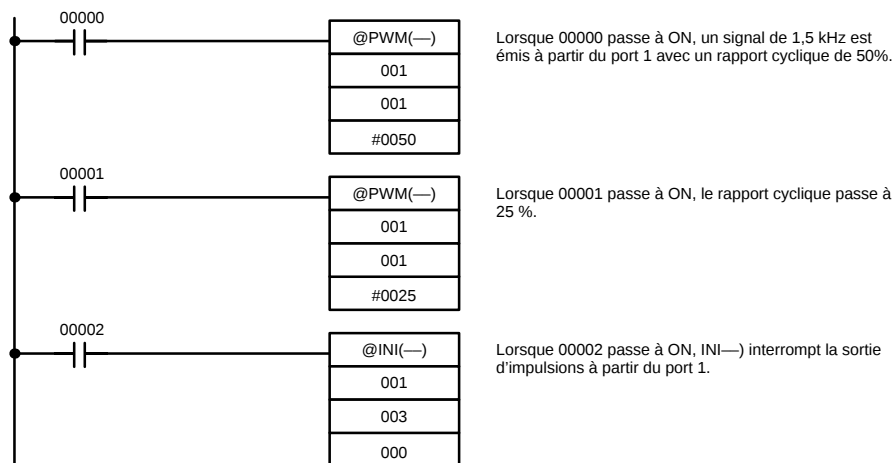
La sortie d'impulsions à partir d'un port peut être arrêtée en exécutant INI(—) avec C=003. Spécifiez le port 1 ou 2 (P=001 à 002).



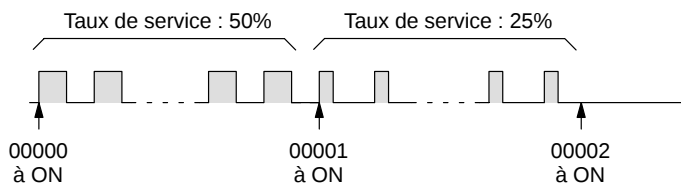
Exemple: avec PWM(—)

L'exemple suivant montre l'utilisation de PWM(—) pour le début d'une sortie d'impulsion à 1,5 kHz à partir du port 1 et pour la modification du rapport cyclique de 50 à 25 %. La sortie d'impulsion est alors arrêtée avec INI(—).

Avant l'exécution du programme, vérifiez que le DM 6643 soit à 1000 (rapport cyclique d'impulsion du port 1).



Le schéma suivant illustre le rapport cyclique de la sortie d'impulsion du port 1 pendant l'exécution du programme.



1-3-5 Etat des ports 1 et 2

L'état des sorties d'impulsion (impulsions standard ou à rapport cyclique) des ports 1 et 2 peut être déterminé par la lecture de l'état des drapeaux correspondants dans les zones SR et AR ou par l'exécution de PRV(—).

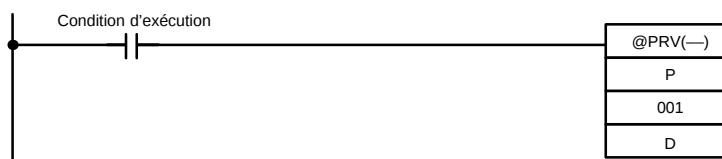
Lecture de l'état du drapeau L'état des sorties d'impulsions peut être déterminé par la lecture du contenu des canaux et drapeaux de la table suivante :

Canaux	Bit(s)	Fonction	Description
SR 236 et SR 237	00 à 15	Port 1 : PV (valeur en cours)	Indique la valeur en cours à 8 chiffres du nombre d'impulsions sorties du port 1 ; les quatre chiffres les plus importants sont dans SR 237.
SR 238 et SR 239	00 à 15	Port 2 : PV (valeur en cours)	Indique la valeur en cours à 8 chiffres du nombre d'impulsions sorties du port 2 ; les quatre chiffres les plus importants sont dans SR 239.
AR 04	08 à 15	Etat de la sortie d'impulsions	Indique l'état de la sortie d'impulsions 00: normal 01 ou 02: erreur système 03: erreur de configuration API 04: arrêt pendant la sortie d'impulsions
AR 05	12	Drapeau de décélération du port 1	Indique la décélération. (0: non spécifiée ; 1: spécifiée)
	13	Drapeau de nombre d'impulsions du port 1	Indique la sélection du nombre d'impulsions (0: non spécifié ; 1: spécifié)
	14	Drapeau de sortie d'impulsions terminée du port 1	Indique l'état de la sortie d'impulsions (0: non terminée ; 1: terminée)
	15	Drapeau d'état de la sortie d'impulsions du port 1	Indique la sortie d'impulsions (0: pas de sortie ; 1: sortie en cours)
AR 06	12	Drapeau de décélération du port 2	Indique la décélération (0: non spécifiée ; 1: spécifiée)
	13	Drapeau de nombre d'impulsions du port 2	Indique la sélection du nombre d'impulsions (0: non spécifié ; 1: spécifié)
	14	Drapeau de sortie d'impulsions terminée du port 2	Indique l'état de la sortie d'impulsions (0: non terminée ; 1: terminée)
	15	Drapeau d'état de la sortie d'impulsions du port 2	Indique la sortie d'impulsions (0: pas de sortie ; 1: sortie en cours)

Exécution de PRV(—)

L'état des sorties d'impulsions peut être déterminé par l'exécution de PRV(—). Spécifiez le port 1 ou 2 (P=001 à 002) et le canal de destination D. L'information concernant l'état du port est inscrite dans les bits 04 à 07 de D et les bits 00 à 03 et 08 à 15 sont placés à 0.

Lorsque l'on utilise PRV(—) pour lire l'état du port, c'est l'information la plus récente qui est lue et le temps de scrutation de l'API n'entre donc pas en compte.



Les bits 04 à 07 de D contiennent l'information concernant l'état du port.

Bit	Fonction	Description
04	Drapeau de décélération	Indique la décélération (0: pas de décélération ; 1: décélération)
05	Drapeau du nombre d'impulsions	Indique la spécification du nombre total d'impulsions (0: non spécifiée ; 1: spécifiée)
06	Drapeau de sortie d'impulsions terminée	Indique l'état de la sortie d'impulsions (0: non terminée ; 1: terminée)
07	Drapeau d'état de la sortie d'impulsions	Indique la sortie d'impulsions (0: pas de sortie ; 1: sortie en cours)

1-4 Sélection et utilisation des fonctions d'interruption

1-4-1 Types d'interruption

Il existe trois types de traitement d'interruption pour le CQM1 :

Interruption d'entrée :

L'interruption est exécutée lorsqu'une entrée provenant d'une source externe place à ON l'un des bits UC 00000 à 00003.

Interruption par temporisation cyclique :

L'interruption est exécutée par une temporisation cyclique avec une précision de 0,1 ms.

Interruption par compteur rapide :

L'interruption est exécutée selon la valeur en cours (PV) d'un compteur rapide intégré. Toutes les UC du CQM1 sont équipées d'un compteur rapide numéro 0 qui compte les entrées d'impulsion dans l'une des entrées UC 00004 à 00006. Les impulsions à deux phases jusqu'à 2,5 kHz peuvent être comptées.

Les UC CQM1-CPU43/44-E peuvent également compter les entrées à partir des ports 1 et 2.

CQM1-CPU43-E: les compteurs rapides 1 et 2 comptent les entrées d'impulsions rapides sur les ports 1 et 2. Les impulsions à deux phases jusqu'à 25 kHz ou 50 kHz unidirectionnelles peuvent être comptées.

CQM1-CPU44-E: les compteurs rapides 1 et 2 comptent les entrées de code des codeurs rotatifs absolus sur les ports 1 et 2 (1 kHz).

Processus d'interruption

Lorsqu'une interruption est générée, le sous-programme d'interruption spécifié est exécuté. Les interruptions sont traitées selon les priorités suivantes (l'interruption d'entrée 0 est traitée en priorité et l'interruption de compteur rapide numéro 0 est traitée en dernier) :

- 1, 2, 3...** 1. Interruption d'entrée 0 > Interruption d'entrée 1 > Interruption d'entrée 2 > Interruption d'entrée 3
2. Interruption de compteur rapide 1 > Interruption de compteur rapide 2
3. Interruption de tempo. cyclique 0 > Interruption de tempo. cyclique 1 > Interruption de tempo. cyclique 2 (l'interruption de tempo. cyclique 2 est égale à l'interruption de compteur rapide numéro 0.

Lorsqu'une interruption supérieure en priorité est reçue au cours d'un processus d'interruption, le processus en cours est arrêté et la nouvelle interruption est traitée en lieu et place. Après exécution de ce sous-programme, l'interruption précédemment arrêtée est reprise.

Lorsqu'une interruption inférieure ou égale en priorité est reçue pendant un processus d'interruption, elle est traitée dès que le sous-programme en cours est terminé.

Comme pour les sous-programmes ordinaires, les processus d'interruption sont définis à l'aide de SBN(92) et de RET(93) à la fin du programme principal. Lorsque des sous-programmes d'interruption sont exécutés, une gamme de bits d'entrée spécifiée peut être rafraîchie.

Lorsqu'un sous-programme d'interruption est défini, un message "no SBS error" ("pas d'erreur SBS") est généré pendant la vérification du programme mais l'exécution continue normalement. Si cette erreur se produit, vérifiez tous les sous-programmes normaux pour vous assurer que SBS(91) a bien été programmée auparavant.

Instructions de sortie d'impulsions et interruptions

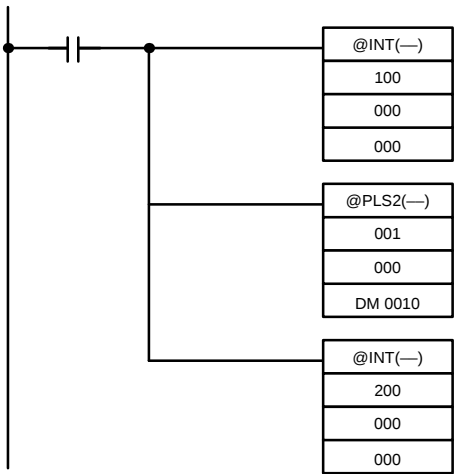
Avec les UC CQM1-CPU43/44-E, les instructions suivantes ne peuvent être exécutées dans un sous-programme d'interruption pendant l'exécution d'une

instruction contrôlant les E/S d'impulsions ou les compteurs rapides dans le programme principal (25503 passe à ON) :

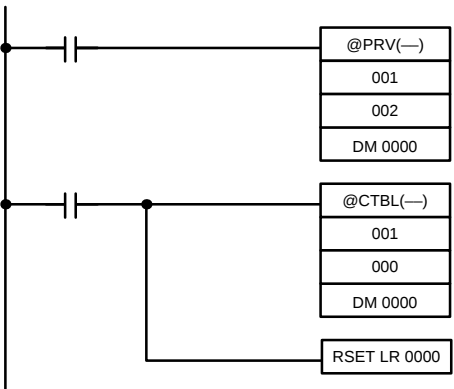
INI(—), PRV(—), CTBL(—), SPED(—), PULS(—), PWM(—), PLS2(—) et ACC(—)

Voici plusieurs façons de contourner ces limites :

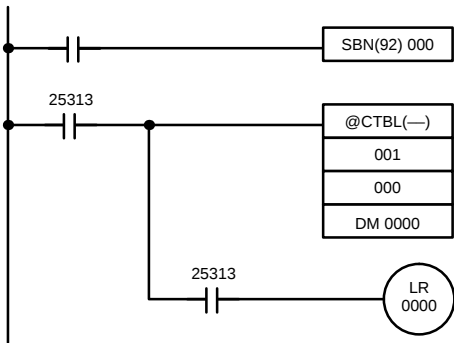
- 1
- Toutes les interruptions peuvent être masquées pendant l'exécution de l'instruction :



- 2
- Ré-exécutez l'instruction dans le programme principal.
Partie concernée dans le programme principal :



Partie concernée dans le sous-programme d'interruption :



1-4-2 Interruption d'entrée

Les entrées de l'UC IR 00000 à IR 00003 peuvent être utilisées pour les interruptions provenant de sources extérieures. Les interruptions d'entrée 0 à 3 correspondent respectivement à ces bits et sont toujours utilisées pour appeler les sous-programmes 000 à 003. Lorsque les interruptions d'entrée ne sont pas utilisées, les sous-programmes 000 à 003 peuvent être utilisés comme sous-programmes ordinaires.

Processus

Il existe deux mode d'interruption d'entrée : le mode d'interruption d'entrée, pour lequel l'interruption correspond à une entrée externe et le mode compteur pour lequel des signaux provenant d'une source extérieure sont comptés à grande vitesse, l'interruption étant effectuée à chaque fois qu'un certain nombre de signaux est atteint.

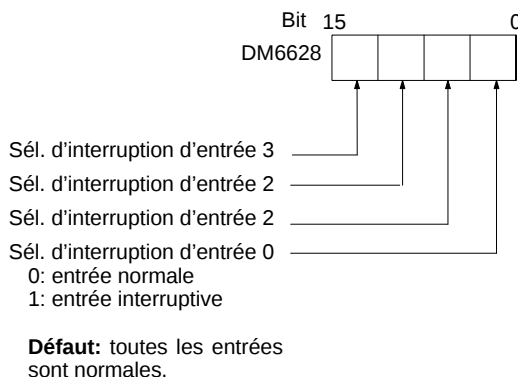
En mode d'interruption d'entrée, des signaux d'une longueur de 100 µs minimum peuvent être détectés. En mode compteur, on peut compter les signaux allant jusqu'à 1 kHz.

Paramètres de configuration

Avant d'exécuter le programme, effectuez les sélections suivantes dans la configuration de l'API en mode PROGRAM :

Sélections d'entrée d'interruption (DM 6628)

Sans ces sélections, les interruptions ne pourront être exécutées dans le programme :



Sélection du canal de rafraîchissement d'entrée (DM 6630 à DM 6633)

Effectuez ces sélections lorsqu'il est nécessaire de rafraîchir les entrées :



Défaut: pas de rafraîchissement d'entrée

Exemple: si le DM 6630 est à 0100, IR 000 est rafraîchi à la réception d'un signal pour l'interruption 0.

Rem. : si le rafraîchissement d'entrée n'est pas utilisé, l'état du signal d'entrée dans le sous-programme d'interruption n'est plus fiable, ni même l'état du bit d'entrée d'interruption qui a activé l'interruption. Par ex., IR 00000 ne serait pas à ON dans le sous-programme d'interruption pour l'interruption d'entrée 0 si cette

entrée n'était pas rafraîchie (dans ce cas, le drapeau constamment à ON pourrait être utilisé à la place de IR 00000).

Mode interruption d'entrée

Utilisez les instructions suivantes pour programmer les interruptions d'entrée en mode interruption d'entrée :

Interruptions masquées

Avec l'instruction INT(—), sélectionnez ou effacez les masques d'interruption d'entrée selon vos besoins.

(@)INT
000
000
D

Effectuez les sélections avec les bits D 0 à 3 correspondant aux interruptions d'entrée 0 à 3 :

- 0: masque effacé (interruption d'entrée permise)
- 1: masque sélectionné (interruption d'entrée non permise)

Au début, toutes les interruptions d'entrée sont masquées.

Effacement des interruptions masquées

Si le bit correspondant à une interruption d'entrée passe à ON alors qu'il est masqué, cette interruption est sauvegardée en mémoire et sera exécutée dès que le masque sera effacé. Pour que cette interruption ne soit pas exécutée au moment où le masque est effacé, l'interruption doit être effacée de la mémoire. Un seul signal d'interruption est sauvegardé en mémoire pour chaque numéro d'interruption.

Avec l'instruction INT(—), effacez l'interruption d'entrée de la mémoire.

(@)INT
001
000
D

Si les bits D 0 à 3 correspondant aux interruptions d'entrée 0 à 3 sont placés sur "1", alors les interruptions d'entrée sont effacées de la mémoire.

- 0: interruptions d'entrée retenues
- 1: interruptions d'entrée effacées

Lecture de l'état du masque

Avec l'instruction INT, lisez l'état du masque d'interruption d'entrée.

(@)INT
002
000
D

L'état du chiffre d'extrême droite de la donnée stockée dans le canal D (bits 0 à 3) révèle l'état du masque.

- 0: masque effacé (interruption permise)
- 1: masque sélectionné (interruption non permise)

Mode compteur

Voici la programmation d'interruption d'entrée en mode compteur :

Rem. : les canaux SR utilisés en mode compteur (SR 244 à SR 251) contiennent tous des données binaires (hexadécimales et non BCD).

- 1, 2, 3...**
1. Ecrivez les valeurs sélectionnées pour le fonctionnement par compteur dans les canaux SR correspondant aux interruptions 0 à 3. Ces valeurs sélectionnées sont écrites entre 0000 et FFFF (0 à 65535). La valeur 0000 inhibe l'opération de comptage jusqu'à la sélection d'une nouvelle valeur et l'étape 2 (ci-dessous) est répétée.

Rem. : ces bits SR sont effacés en début d'opération et doivent être ré-écrits à partir du programme.

Le signal d'entrée max. pouvant être compté est de 1 kHz.

Interruption	Canal
Entrée d'interruption 0	SR 244
Entrée d'interruption 1	SR 245
Entrée d'interruption 2	SR 246
Entrée d'interruption 3	SR 247

Si l'on n'utilise pas le mode compteur, ces bits SR peuvent être utilisés comme bits de travail.

2. Avec l'instruction INT(—), rafraîchissez les valeurs sélectionnées pour le mode compteur et activez les interruptions.

(@)INT
003
000
D

Si les bits D 0 à 3 correspondant aux interruptions d'entrée 0 à 3 sont placés à "0," les valeurs sélectionnées sont rafraîchies et les entrées sont permises.

0: valeurs du mode compteur rafraîchies et masque effacé

1: aucune action (placez les bits de toutes les interruptions qui n'ont pas été modifiées à "1".)

L'interruption d'entrée pour laquelle la valeur sélectionnée est rafraîchie est activée en mode compteur. Lorsque le compteur atteint la valeur sélectionnée, une interruption se produit, le compteur est remis à zéro, les interruptions et le comptage continuent jusqu'à l'arrêt du compteur.

Attention : 1. Si l'instruction INT(—) est utilisée pendant le comptage, la valeur en cours (PV) revient à la valeur sélectionnée (SV). On doit donc absolument utiliser la forme impulsionnelle de l'instruction pour obtenir une interruption.

2. La valeur sélectionnée est choisie pendant l'exécution de l'instruction INT(—). Si les interruptions sont déjà en cours, la valeur sélectionnée ne sera pas modifiée par le simple changement des contenus de SR 244 à SR 247 ; c'est-à-dire que si les contenus sont modifiés, la valeur sélectionnée doit être rafraîchie par la ré-exécution de l'instruction INT(—).

Les interruptions peuvent être masquées par le même processus que celui du mode d'interruption d'entrée, mais si les masques sont également effacés par ce processus, le mode compteur n'est pas maintenu ; le mode interruption d'entrée est alors utilisé en lieu et place. Les signaux d'interruption reçus pour les interruptions masquées peuvent également être effacés par le même processus que celui du mode d'interruption d'entrée.

Valeur sélectionnée (PV) en mode compteur

Lorsque les interruptions d'entrée sont utilisées en mode compteur, la valeur sélectionnée du compteur est stockée dans le canal SR correspondant aux interruptions d'entrée 0 à 3. Les valeurs sont comprises entre 0000 et FFFE (0 à 65 534) et sont égales à la valeur sélectionnée du compteur moins 1.

Interruption	Canal
Interruption d'entrée 0	SR 248
Interruption d'entrée 1	SR 249
Interruption d'entrée 2	SR 250
Interruption d'entrée 3	SR 251

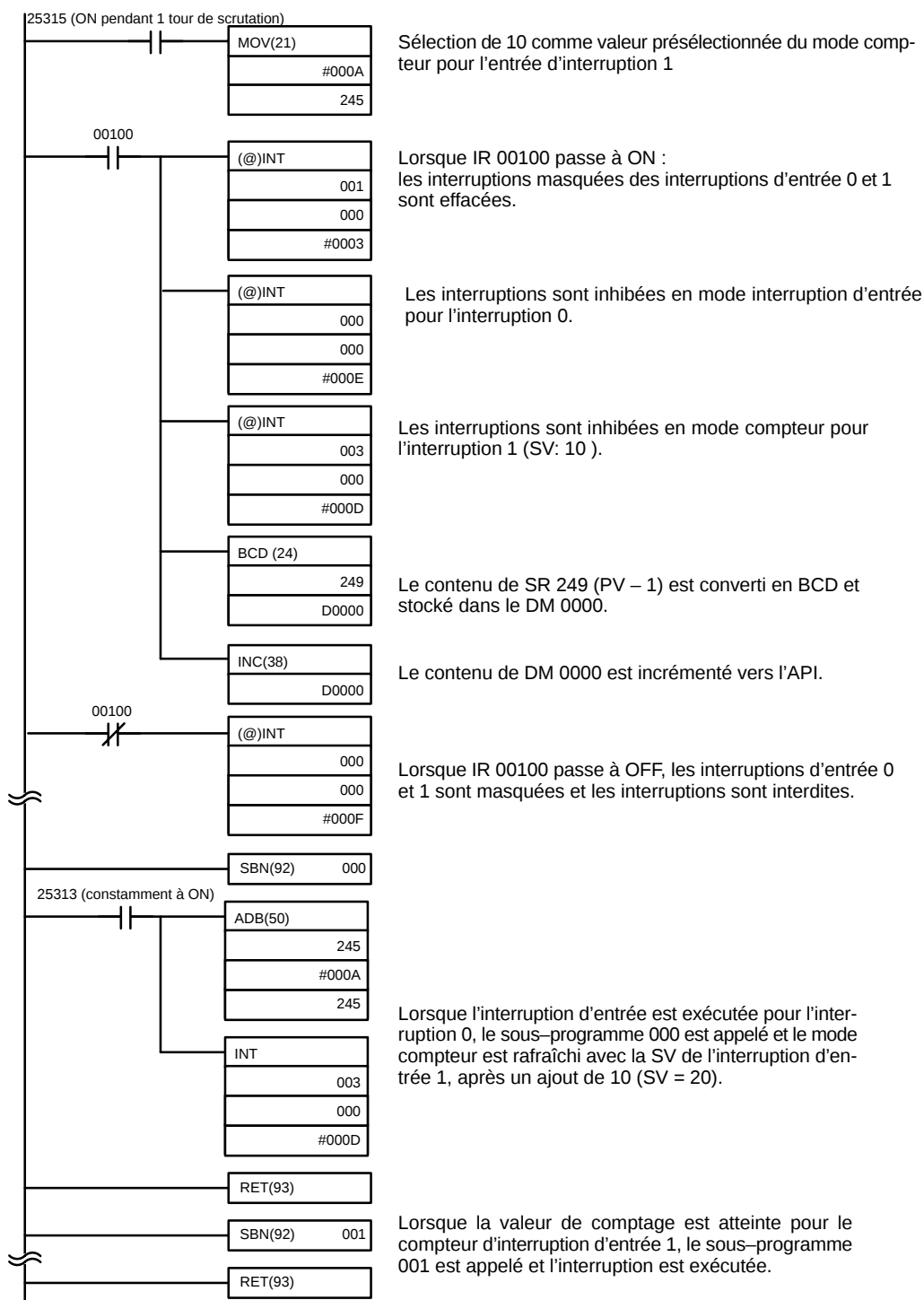
Exemple: la valeur en cours d'une interruption dont la valeur sélectionnée est égale à 000A est enregistrée sous la forme de 0009 tout de suite après l'exécution de INT(—).

Attention : même si les interruptions d'entrée ne sont pas effectuées en mode compteur, les canaux SR ne peuvent être utilisés comme bits de travail.

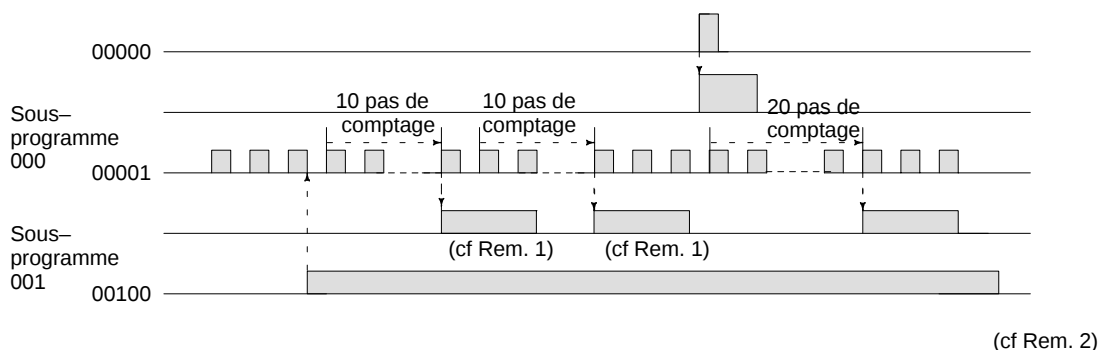
Exemple d'application

Dans cet exemple, l'interruption d'entrée 0 est utilisée en mode interruption d'entrée et l'interruption d'entrée 1 en mode compteur. Avant l'exécution du programme, vérifiez bien la configuration de l'API.

Configuration de l'API : DM 6628: 0011 (IR 00000 et IR 00001 utilisés pour les interruptions d'entrée). Les sélections par défaut sont utilisées pour tous les autres paramètres de la configuration (les entrées ne sont pas rafraîchies au moment de l'interruption).



Lorsque le programme est exécuté, le fonctionnement est le suivant :



- Remarque : 1. Le compteur continue de fonctionner pendant l'exécution du sous-programme d'interruption.
2. L'interruption d'entrée reste masquée.

1-4-3 Comment masquer la totalité des interruptions

Toutes les interruptions, y compris celles d'entrée d'interruption, de temporisation cyclique et de compteur rapide peuvent être masquées et démasquées en groupe, grâce à l'instruction INT(—). Sur les interruptions individuelles, le masque s'ajoute aux autres masques. L'effacement de la totalité des interruptions n'efface donc pas les masques individuels, mais restaure les conditions masquées qui existaient avant l'exécution d'INT(—) pour masquer les interruptions en groupe.

N'utilisez pas INT(—) pour masquer les interruptions, sauf en cas de nécessité de masquer temporairement toutes les interruptions. Pour cela, utilisez toujours les instructions INT(—) en paire ; la première instruction INT(—) sert à masquer les interruptions et la seconde à ôter le masque.

INT(—) ne peut être utilisée pour masquer toutes les interruptions et supprimer le masque depuis l'intérieur des sous-programmes d'interruption.

Comment masquer

Utilisez l'instruction INT pour désactiver toutes les interruptions :

(@)INT
100
000
000

Si une interruption est générée pendant que l'on masque les interruptions, le processus d'interruption n'est pas exécuté mais l'interruption est enregistrée pour l'entrée, la tempo. cyclique ou le compteur rapide. Les interruptions sont ensuite mises en service dès que les masques sont supprimés.

Comment supprimer les masques

Utilisez l'instruction INT(—) de la façon suivante :

(@)INT
200
000
000

1-4-4 Interruptions par temporisation cyclique

Des interruptions par temporisation rapide et à haute précision peuvent être effectuées par les temporisations cycliques. Le CQM1 est équipé de trois temporisations cycliques, numérotées de 0 à 2.

Rem. : 1. La temporisation cyclique 0 ne peut être utilisée lorsque des impulsions sont sorties vers des cartes de sortie par l'instruction SPED(—).

2. La temporisation cyclique 2 ne peut être utilisée en même temps que le compteur rapide.

Processus

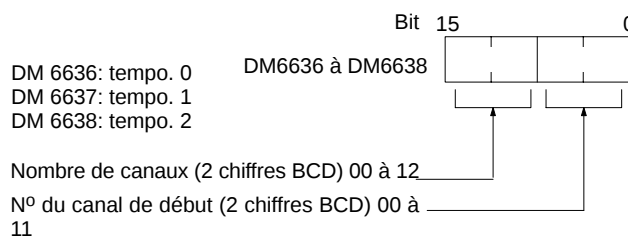
Il existe deux modes de fonctionnement de la temporisation cyclique : le mode à une impulsion, dans lequel une seule interruption a été exécutée lorsque le temps s'est écoulé et le mode d'interruption programmée, dans lequel l'interruption est répétée à intervalle fixe.

Configuration de l'API

Lors de l'utilisation d'interruptions par tempo. cyclique, effectuez les sélections suivantes pour la configuration de l'API en mode PROGRAM avant l'exécution du programme.

Sélection des canaux de rafraîchissement d'entrée (DM 6636 à DM 6638)

Effectuez les sélections suivantes lorsqu'il est nécessaire de rafraîchir les entrées :



Défaut: pas de rafraîchissement d'entrée

Sélections du compteur rapide (DM 6642)

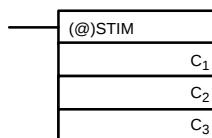
Lorsque vous utilisez la tempo. cyclique 2, vérifiez avant le début du fonctionnement que le compteur (configuration API : DM 6642) soit bien en sélection par défaut (0000: compteur rapide non utilisé).

Fonctionnement

Utilisez l'instruction suivante pour activer et contrôler la tempo. cyclique :

Démarrage en mode à une impulsion

Utilisez l'instruction STIM(—) pour démarrer la tempo. cyclique en mode à une impulsion.



C₁: n° de tempo. cyclique
tempo. cyclique 0 : 000
tempo. cyclique 1 : 001
tempo. cyclique 2 : 002

C₂: valeur de sélection de la tempo. (1ère adresse de canal)

C₃: n° de sous-programme (4 chiffres BCD): 0000 à 0255

C₂: valeur de sélection du compteur décremental (4 chiffres BCD): 0000 à 9999

C₂ + 1 : intervalle de décrementation (4 chiffres BCD; unité: 0,1 ms) : 0005 à 0320 (0,5 ms à 32 ms)

A chaque fois que le temps spécifié dans le canal C₂ + 1 est écoulé, le contenu de C₂ est décrementé de 1. Lorsque cette valeur est égale à 0, le sous-programme désigné est appelé une seule fois et la tempo. s'arrête.

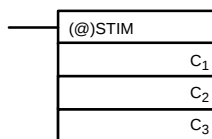
Le temps séparant le départ de l'instruction STIM et l'exécution du sous-programme est calculé comme suit :

(contenu du canal C₂) x (contenu du canal C₂ + 1) x 0,1 ms = (0,5 à 319 968 ms)

Si une constante a été sélectionnée pour C₂, la valeur sélectionnée du compteur décremental prend alors cette valeur et l'intervalle de décrementation est égal à 10 (1 ms). La valeur sélectionnée est exprimée en ms.

Départ en mode d'interruption cyclique

Utilisez l'instruction STIM(—) pour lancer la temporisation cyclique en mode d'interruption programmée.



C₁: n° de tempo. cyclique + 3
 tempo. cyclique 0 : 003
 tempo. cyclique 1 : 004
 tempo. cyclique 2 : 005

C₂: valeur de sélection de la tempo. (n° du 1er canal)

C₃: n° de sous-programme (4 chiffres BCD) : 0000 à 0255

C₂: valeur de sélection du compteur décrémental (4 chiffres BCD): 0001 à 9999

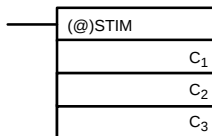
C₂ + 1 : cycle de décrémentation (4 chiffres BCD ; unité : 0,1 ms) :
 0005 à 0320 (0,5 ms à 32 ms)

La signification des sélections est identique à celle du mode à une impulsion, mais en mode programmé, la valeur en cours est remplacée à la valeur sélectionnée ; la décrémentation recommence après l'appel du sous-programme. En mode programmé, les interruptions continuent à être répétées à intervalle fixe jusqu'à la fin.

Rem. : les UC CQM1-CPU11-E/CPU21-E sont équipées des sous-programmes 0000 à 0127 seulement.

Lecture du temps écoulé de la temporisation

Utilisez l'instruction STIM(—) pour lire le temps écoulé.



C₁: n° de tempo. cyclique + 6
 tempo. cyclique 0 : 006
 tempo. cyclique 1 : 007
 tempo. cyclique 2 : 008

C₂: 1er canal du paramètre 1

C₃: paramètre 2

C₂: nombre de décréments appliqués au compteur décrémental (4 chiffres BCD)

C₂ + 1: intervalle de temps du compteur décrémental (4 chiffres BCD ;
 unité: 0,1 ms)

C₃: temps écoulé à partir de la décrémentation précédente (4 chiffres BCD ;
 unité : 0,1 ms)

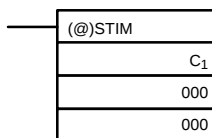
Le temps écoulé entre le lancement de la tempo. cyclique et l'exécution de cette instruction se calcule comme suit :

{(contenu du canal C2) x (contenu du canal C2 + 1) + (contenu du canal C3)} x 0,1 ms

En cas d'arrêt de la tempo. cyclique spécifiée, la valeur "0000" est stockée.

Arrêt de temporisation

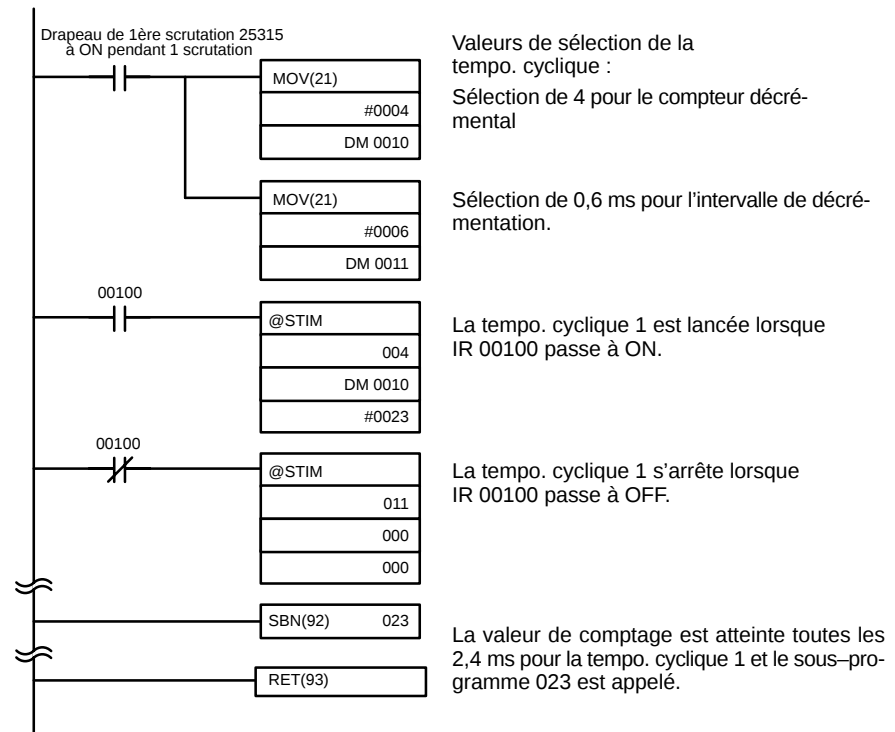
Utilisez l'instruction STIM(—) pour arrêter la tempo. cyclique spécifiée :



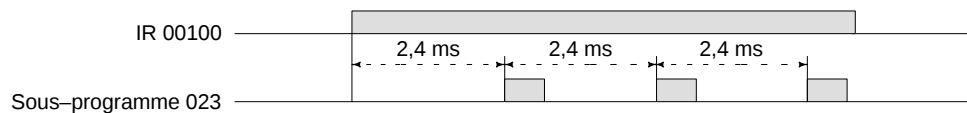
C₁: n° de tempo. cyclique + 10
 tempo. cyclique 0 : 010
 tempo. cyclique 1 : 011
 tempo. cyclique 2 : 012

Exemple d'application

Cette interruption est exécutée toutes les 2,4 ms (0,6 ms x 4) par la tempo. cyclique 1. La configuration API a été effectuée selon les sélections par défaut (les entrées ne sont pas rafraîchies pendant l'interruption).



Lorsque le programme est exécuté, le sous-programme 023 est exécuté toutes les 2,4 ms, tandis que IR 00100 est à ON.

**1-4-5 Interruptions du compteur rapide 0**

Les signaux d'impulsion provenant d'un codeur d'impulsions et envoyés vers les bits UC 00004 à 00006 peuvent être comptés à grande vitesse et l'interruption peut être exécutée en fonction du comptage.

Processus**Types de signaux d'entrée et modes compteur**

Deux types de signaux peuvent être produits à partir d'un codeur d'impulsions. Le mode utilisé pour le compteur rapide 0 dépend du type de signal.

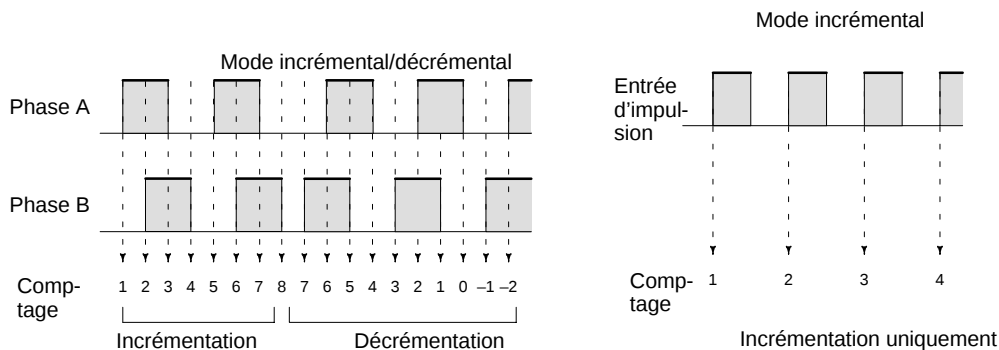
Mode incrémental/décrémental:

une différence de phase égale à 4 x le signal à deux phases (phases A et B) et une phase Z sont utilisées pour les entrées. Le comptage est incrémenté ou décrémental selon les différences des signaux à 2 phases.

Mode incrémental :

un signal d'impulsion à une phase et un signal de remise à

zéro du comptage sont utilisés pour les entrées. le comptage est incrémenté en fonction du signal à phase unique.



Rem. : il faut obligatoirement utiliser une des méthodes suivantes pour remettre à zéro le compteur lorsqu'on le relance. Il est automatiquement remis à zéro au départ ou à l'arrêt de l'exécution du programme.

Les transitions de signaux suivantes sont traitées comme des impulsions d'incrémentations : front montant de phase A vers front montant de phase B vers front descendant de phase A vers front descendant de phase B. Les transitions de signaux suivantes sont traitées comme des impulsions de décrémentations : front montant de phase B vers front montant de phase A vers front descendant de phase B vers front montant de phase A.

La gamme de comptage s'étend de - 32 767 à 32 767 pour le mode incrémental/décémental et de 0 à 65 535 pour le mode incrémental. Les signaux d'impulsion peuvent être comptés jusqu'à 2,5 kHz en mode incrémental/décémental et jusqu'à 5 kHz en mode incrémental.

Le mode incrémental/décémental utilise toujours une entrée égale à 4 x la différence de phase. Le nombre de comptages pour chaque tour de codeur est égal à 4 fois la résolution du compteur. Sélectionnez le codeur en fonction des gammes de comptage possibles.

Modes de remise à zéro

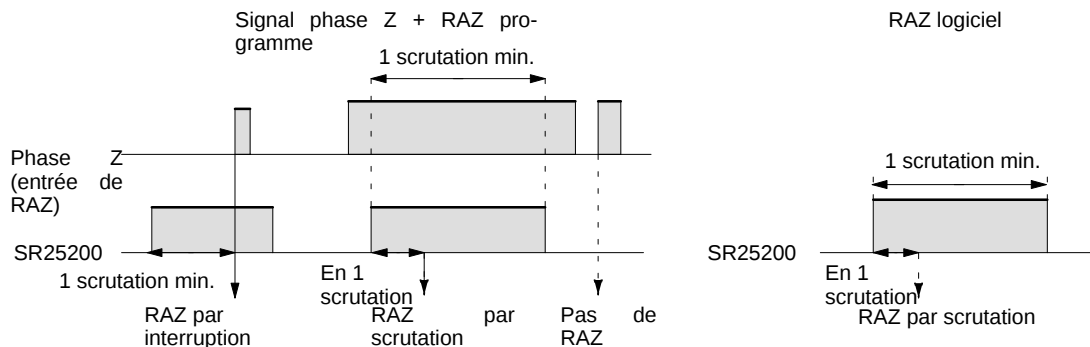
L'une des deux méthodes suivantes doit être utilisée pour la remise à zéro de la valeur en cours du compteur (c'est-à-dire que la sélection sera égale à 0) :

Signal de phase Z + RAZ à partir du programme :

la valeur en cours est remise à zéro lorsque le signal de phase Z (entrée de RAZ) passe à ON après passage à ON du bit de RAZ du compteur rapide 0 (bit SR 25200).

RAZ à partir du programme :

la valeur en cours est remise à zéro lorsque le bit de remise à zéro du compteur rapide 0 (SR 25200) passe à ON.



Rem. : le bit de RAZ du compteur rapide 0 (SR 25200) est rafraîchi une fois par scrutation ; pour en faire une lecture fiable, il doit être à ON pendant au moins une scrutation.

Le "Z" de la phase Z est l'abréviation de "zéro". Ce signal montre que le codeur a terminé une scrutation.

Comptage d'interruptions du compteur rapide

Pour les interruptions du compteur rapide 0, une table comparative est utilisée. La vérification du comptage peut être effectuée par l'une des deux méthodes suivantes. Dans la table, les conditions de comparaison (comparaison des valeurs en cours) et les combinaisons de sous-programmes d'interruption sont sauvegardées.

Valeur spécifiée : un maximum de 16 conditions de comparaison (valeurs spécifiées et sens de comptage) et de combinaisons de sous-programmes sont sauvegardés dans la table de comparaison. Lorsque la valeur en cours du compteur et le sens de comptage correspondent aux conditions de comparaison, le sous-programme d'interruption spécifié est exécuté.

Comparaison de gamme: Huit conditions de comparaison (limites haute et basse) et combinaisons de sous-programmes d'interruption sont sauvegardées dans la table. Lorsque la valeur en cours est supérieure ou égale à la limite basse et inférieure ou égale à la limite haute, le sous-programme d'interruption spécifié est alors exécuté.

Câblage

En fonction du mode de comptage, les signaux d'entrée provenant du codeur et envoyés vers le terminal d'entrée de la carte UC sont les suivants :

N° du terminal	Mode incrémental/décrémental	Mode incrémental
4	Phase A du codeur	Entrée de comptage des impulsions
5	Phase B du codeur	---
6	Phase Z du codeur	Entrée RAZ

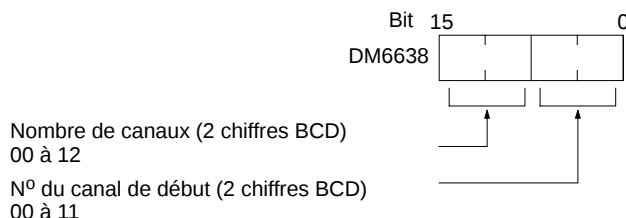
Si vous désirez utiliser uniquement la remise à zéro programme, l'entrée 6 peut être utilisée comme une entrée ordinaire. En mode incrémental, l'entrée 5 peut être utilisée comme une entrée ordinaire.

Configuration API

Lorsque vous utilisez les interruptions par compteur rapide 0, effectuez les sélections suivantes en mode PROGRAM avant l'exécution du programme.

Sélection du canal de rafraîchissement d'entrée (DM 6638)

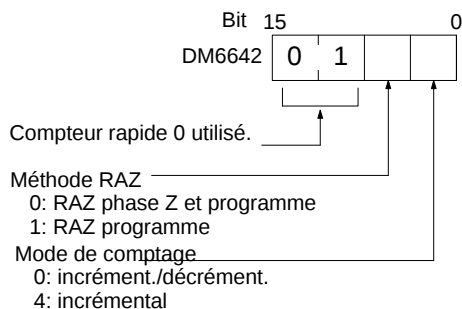
Effectuez les sélections suivantes pour rafraîchir les entrées. La sélection est identique à celle de la tempo. cyclique 2.



Défaut: pas de rafraîchissement d'entrée

Sélections du compteur rapide 0 (DM 6642)

Si ces sélections ne sont pas effectuées, le compteur rapide 0 ne pourra être utilisé dans le programme :



Défaut: compteur rapide 0 non utilisé

Les changements de sélection du DM6642 ne sont effectifs que lors de la mise sous tension ou au départ de l'exécution du programme.

Programmation

Programmation du compteur rapide 0 :

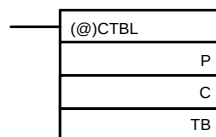
Le compteur rapide 0 commence à fonctionner lorsque les sélections de la configuration API ont été correctement effectuées mais les comparaisons avec la table de comparaison ne sont pas faites et les interruptions ne sont pas générées si l'instruction CTBL(—) n'est pas exécutée.

Le compteur rapide 0 est remis à 0 à la mise sous tension, au début du fonctionnement de l'appareil et à la fin de son fonctionnement.

La valeur en cours du compteur rapide 0 est maintenue dans SR 230 et SR 231.

Contrôle des interruptions du compteur rapide 0

- 1, 2, 3... 1. Utilisez l'instruction CTBL(—) pour sauvegarder la table de comparaison dans le CQM1 et pour commencer les opérations de comparaison.



C: (3 chiffres BCD)

- 000: table sélectionnée et comparaison lancée
- 001: table de gammes sél. et comparaison lancée
- 002: table sélectionnée
- 003: table de gammes sélectionnée

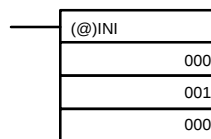
TB: canal de début de la table de comparaison

Si C est à 000, les comparaisons sont alors effectuées par la méthode de comparaison des valeurs spécifiées ; si C est à 001, elles sont effectuées par la méthode de comparaison des gammes. La table est sauvegardée et à la fin de la sauvegarde, les comparaisons commencent. Pendant l'exécution des comparaisons, les interruptions rapides sont effectuées selon la table. Pour tous détails sur le contenu des tables sauvegardées, reportez-vous aux explications concernant CTBL(—) dans le chapitre 5.

Rem. : le résultat des comparaisons est normalement stocké dans AR 1100 à AR 1107 pendant la comparaison par gamme.

Si C est à 002, les comparaisons sont effectuées par la méthode de comparaison des valeurs spécifiées ; si C est à 003, elles sont effectuées par la méthode de comparaison des gammes. Pour ces deux sélections, la table est sauvegardée mais la comparaison ne commence qu'avec l'exécution de l'instruction INI(—).

2. Pour arrêter les comparaisons, exécutez l'instruction INI(—) comme suit :



Pour reprendre les comparaisons, placez la seconde opérande à "000" (exécution de la comparaison) et exécutez l'instruction INI(—).

Une table sauvegardée est retenue dans l'API pendant son fonctionnement (pendant l'exécution du programme) jusqu'à la prochaine sauvegarde de table.

Lecture de la valeur en cours

Il y a deux façons de lire la valeur en cours. La première est la lecture de SR 230 et SR 231 et la seconde est l'instruction PRV(—).

Lecture de SR 230 et SR 231

La valeur en cours du compteur rapide 0 est stockée dans SR 230 et SR 231 comme suit. Le bit d'extrême gauche est égal à F pour les valeurs négatives.

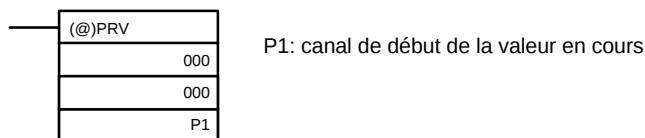
4 chiffres d'extrême gauche	4 chiffres d'extrême droite	Mode incrémental/décrémental	Mode incrémental
SR 231	SR 230	F0032767 à 00032767 (- 32767)	00000000 à 00065535

Rem. : ces canaux sont rafraîchis une fois par scrutation ; il se peut donc qu'il existe une différence entre ces valeurs et la valeur en cours.

Lorsque le compteur rapide 0 n'est pas utilisé, les bits de ce canal peuvent être utilisés comme bits de travail.

Utilisation de l'instruction PRV(--)

Lisez la valeur en cours du compteur rapide 0 à l'aide de l'instruction PRV(—).



La valeur sélectionnée du compteur rapide 0 est stockée de la façon suivante. Le bit d'extrême gauche est égal à F pour les valeurs négatives.

4 chiffres d'extême gauche	4 chiffres d'extême droite	Mode incrémental/décrémental	Mode incrémental
P1+1	P1	F0032767 à 00032767 (-32767)	00000000 à 00065535

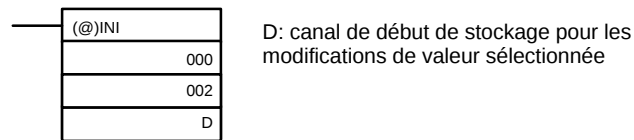
La valeur sélectionnée est lue lorsque l'instruction PRV(—) est en cours d'exécution.

Changement de la valeur sélectionnée

Il y a deux façons de modifier la valeur sélectionnée du compteur rapide 0. La première méthode consiste à remettre la valeur à zéro par les différents modes de RAZ ; la deuxième utilise l'instruction INI(—).

Voici la méthode utilisant l'instruction INI. Pour la méthode RAZ, reportez-vous au début de la description du compteur rapide 0.

Modifiez la valeur sélectionnée par l'instruction INI(—) comme suit :



4 chiffres d'extrême
gauche
D+1

4 chiffres d'extrême
droite
D

Mode incrémental/
décrémental
F0032767 à 00032767

Mode incrémental
00000000 à 00065535

Pour spécifier un nombre négatif, sélectionnez F comme chiffre d'extrême gauche.

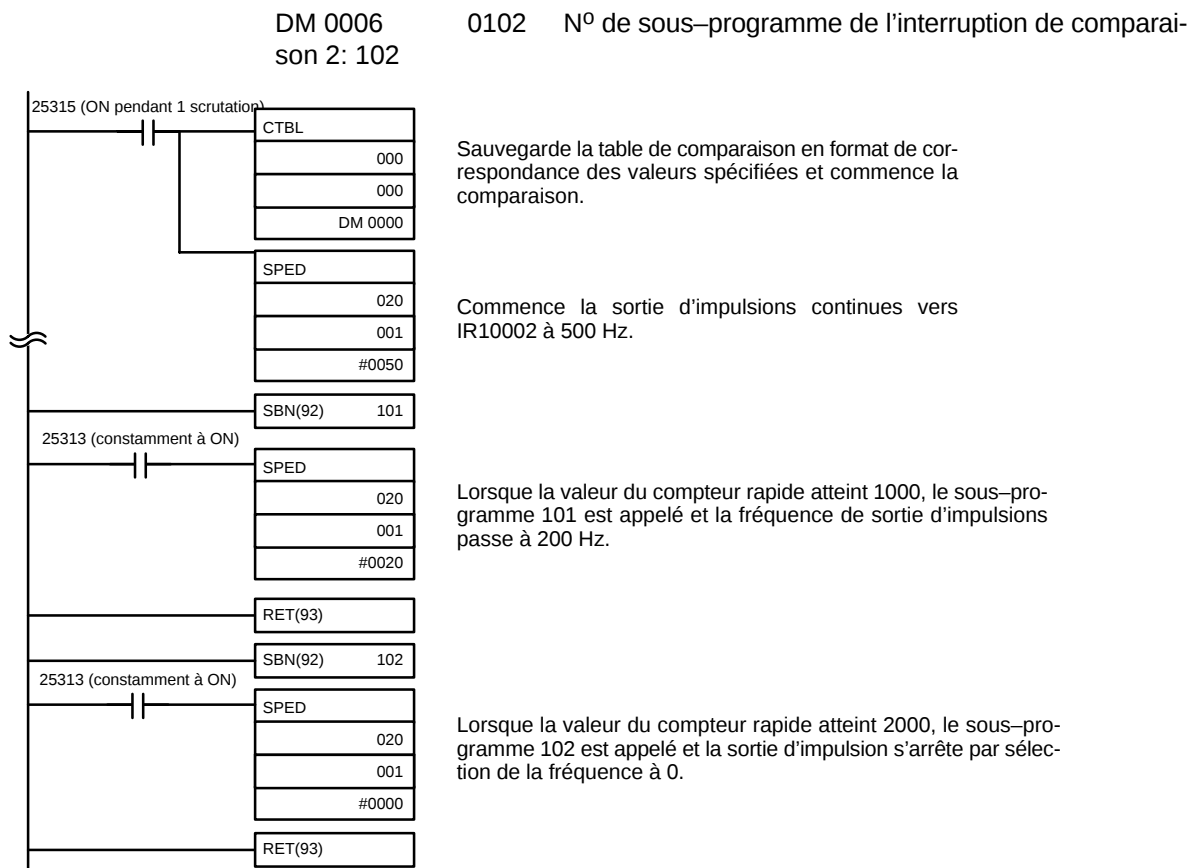
Exemple de fonctionnement

Voici un exemple de programme d'utilisation du compteur rapide 0 en mode incrémental, en effectuant des comparaisons par la méthode de rapprochement des valeurs spécifiées et en changeant la fréquence des sorties d'impulsion selon la valeur sélectionnée du compteur. Avant d'exécuter le programme, effectuez les sélections suivantes dans la configuration de l'API :

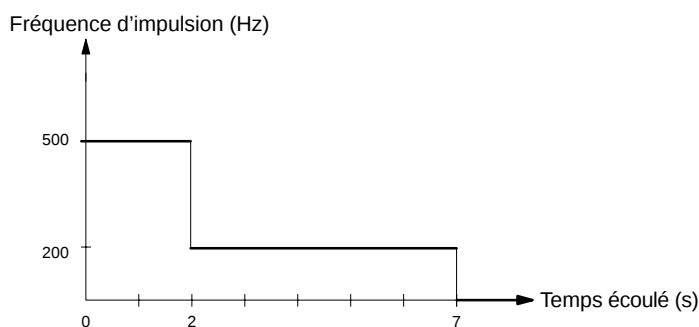
DM 6642: 0114 (compteur rapide 0 utilisé avec la RAZ programme et le mode incrémental). Pour toutes les autres sélections en configuration API, utilisez les sélections par défaut (les entrées ne sont pas rafraîchies au moment de l'interruption et les sorties d'impulsion sont exécutées IR 100).

Les données suivantes sont stockées :

DM 0000	0002	Nombre de conditions de comparaison : 2
DM 0001	1000	Valeur spécifiée 1 : 1000
DM 0002	0000	
DM 0003	0101	N° de sous-programme de l'interruption de comparaison 1 : 101
DM 0004	2000	Valeur spécifiée 1 : 2000
DM 0005	0000	



Fonctionnement pendant l'exécution du programme :



1-4-6 Gamme de comptage non respectée pour le compteur rapide 0

Si la gamme de comptage admissible du compteur rapide 0 n'est pas respectée, un état de "gamme de comptage non respectée" est produit et la valeur sélectionnée du compteur reste à 0FFF FFFF pour les dépassements et à FFFF FFFF pour les valeurs insuffisantes, et ce jusqu'à ce que l'état de gamme non respectée soit effacé par remise à zéro du compteur.

Voici les gammes admissibles :

Mode incrémental/décémental : F003 2767 à 0003 2767

Mode incrémental : 0000 0000 à 0006 5535

Rem. : 1. Les valeurs ci-dessus sont théoriques et supposent un temps de scrutation raisonnablement court. Les valeurs sont en fait celles du cycle précédent le non-respect de la gamme.

2. Les 6^e et 7^e chiffres de la valeur sélectionnée du compteur rapide 0 sont normalement égales à 00, mais elles peuvent être utilisées comme drapeaux

de "non-respect de la gamme" par détection des valeurs inférieures aux gammes de comptage admissibles.

Le compteur rapide 0 peut être remis à zéro comme indiqué au chapitre précédent ou de façon automatique par re-démarrage de l'exécution du programme. Le compteur rapide 0 ne peut fonctionner normalement avant effacement de l'état de non respect de la gamme. Pendant le maintien de l'état de non-respect de la gamme, le fonctionnement est le suivant :

- La table de comparaison ne fonctionne plus.
- La table de comparaison n'est pas effacée.
- Les sous-programmes d'interruption du compteur rapide ne sont pas exécutés.
- L'instruction CTBL(—) peut être utilisée pour enregistrer la table de comparaison seulement. Si l'on essaie de faire débiter le fonctionnement de la table de comparaison, celui-ci n'est pas exécuté et la table n'est pas enregistrée.
- L'instruction INI(—) ne peut être utilisée pour faire débiter ou arrêter le fonctionnement de la table de comparaison ni pour modifier la valeur en cours.
- La lecture des valeurs en cours par l'instruction PRV(—) ne donne que 0FFF FFFF ou FFFF FFFF comme résultat.

Sortie de l'état de non-respect

Utilisez la procédure suivante pour sortir de l'état de non-respect de la gamme :

Si les tables de comparaison sont enregistrées :

- 1, 2, 3...**
1. Remise à zéro du compteur
 2. Choix de la valeur sélectionnée par PRV(—) si nécessaire
 3. Sélection de la table de comparaison par CTBL(—) si nécessaire
 4. Début du fonctionnement du table de comparaison par INI(—)

Si les tables de comparaison ne sont pas enregistrées :

- 1, 2, 3...**
1. Remise à zéro du compteur
 2. Choix de la valeur sélectionnée par PRV(—) si nécessaire
 3. Sélection de la table de comparaison et début du fonctionnement par CTBL(—) et INI(—)

Rem. : les résultats de la comparaison de gammes de AR 11 sont maintenus même après sortie de l'état de non-respect de la gamme. Le sous-programme d'interruption d'une condition d'interruption rencontré immédiatement après la sortie de cet état de non-respect ne sera pas exécuté si la condition a déjà été rencontrée avant le passage à cet état. Si l'exécution d'un sous-programme d'interruption est nécessaire, effacez AR 11 avant de continuer.

Remise à zéro

Lorsque le compteur rapide 0 est remis à zéro, la PV est remise à zéro. Le comptage commence à partir de zéro. La table de comparaison, l'état d'exécution et les résultats de l'exécution sont maintenus.

Etat de lancement du compteur

Lors du lancement du compteur rapide 0, le mode compteur de la configuration API est lu, puis utilisé ; la valeur sélectionnée est remise à 0 ; l'état de non-respect de la gamme de comptage est effacé, l'enregistrement et l'état d'exécution de la table de comparaison sont effacés et les résultats de l'exécution de la gamme sont également effacés (ces résultats sont toujours effacés au début du fonctionnement ou lorsque la table de comparaison est enregistrée).

Etat d'arrêt du compteur

Lorsque le compteur rapide 0 s'arrête, la valeur sélectionnée est maintenue, l'enregistrement et l'état d'exécution de la table de comparaison sont effacés et les résultats d'exécution de la gamme sont maintenus.

1-4-7 Interruption des compteurs rapides 1 et 2 (CQM1-CPU43-E)

Les signaux d'impulsion provenant d'un codeur incrémental et envoyés vers les ports 1 et 2 du CQM1-CPU43-E peuvent être comptés à grande vitesse et les interruptions peuvent être exécutées en fonction du comptage.

Les 2 ports peuvent fonctionner séparément. Le compteur du port 1 est appelé compteur rapide 1 et le compteur du port 2 compteur rapide 2. Ce chapitre décrit l'utilisation des compteurs rapides 1 et 2. Reportez-vous au manuel d'utilisation du CQM1 pour de plus amples détails sur le matériel et le câblage.

Rem. : 1. Les compteurs rapides 1 et 2 peuvent être utilisés avec l'UC CQM1-CPU43-E uniquement.

2. Certaines instructions ne peuvent être utilisées lorsque la configuration API (DM 6611) est en mode compteur rapide.

Sélection du DM 6611	Instructions affectées
Mode compteur rapide (0000)	PLS2(—) et le mode 0 de ACC(—) ne peuvent être utilisés.
Mode de sortie par impulsion (0001)	CTBL(—) ne peut être utilisée avec les ports 1 et 2.

Processus

Signaux d'entrée et modes de comptage

Trois types de signaux peuvent être envoyés vers les ports 1 et 2. Les modes de comptage utilisés pour les compteurs rapides 1 et 2 sont sélectionnés dans les DM 6643 et DM 6644 respectivement.

1, 2, 3...

1. Mode de phase différentielle (vitesse de comptage = 25 kHz):

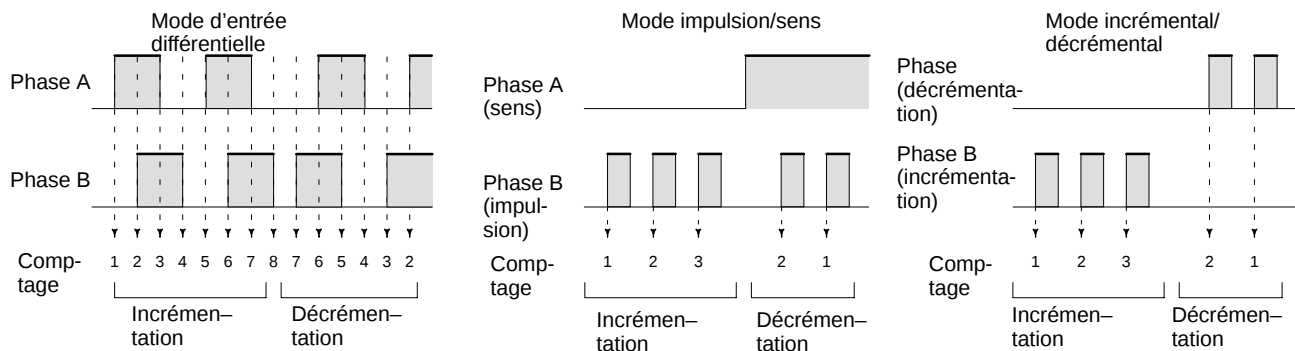
Une différence de phase de 4 x le signal à deux phases (phase A et phase B) et un signal de phase Z sont utilisés pour les entrées. Le comptage est incrémenté ou décrémenté selon les différences des signaux à 2 phases. Ce mode est identique à au mode incrémental/décrémental du compteur rapide 0.

2. Mode impulsion/sens (vitesse de comptage = 50 kHz) :

La phase A représente le signal de sens et la phase B l'impulsion de comptage. Le compteur incrémente lorsque le signal de phase A est à OFF et incrémente lorsqu'il est à ON.

3. Mode incrémental/décrémental (vitesse de comptage = 50 kHz) :

La phase A représente le signal de décrément et la phase B le signal d'incrément. Le compteur décrément lorsque une impulsion de phase A est détectée et incrément lorsque un signal de phase B est détecté.



Modes de comptage

Les modes de comptage (en anneau ou linéaire) pour les compteurs rapides 1 et 2 sont spécifiés dans les DM 6643 et DM 6644 respectivement.

1, 2, 3...

1. Mode en anneau :

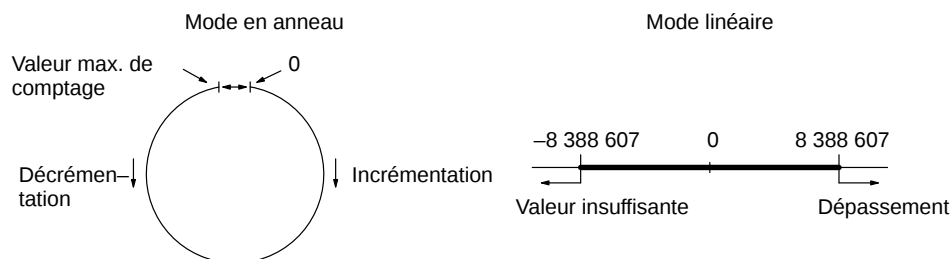
La valeur de comptage maximum +1 est sélectionnée dans CTBL(—). Le

compteur passe de la valeur de comptage maximum à 0 lors de l'incréméntation et de 0 à la valeur de comptage maximum lors de la décrémentation. Il n'y a pas de valeurs négatives.

Le nombre de points de l'anneau (valeur de comptage maximum +1) peut être sélectionné de 1 à 65 000.

2. Mode linéaire :

La gamme de comptage s'étend de - 8 388 607 à 8 388 607. Si le niveau de comptage admissible des compteurs rapides 1 et 2 est dépassé, un état de niveau non respecté apparaît et la valeur en cours du compteur est maintenue à 0838 8607 pour les dépassements et à F838 8607 en cas de valeur insuffisante. Le comptage et les comparaisons s'arrêtent et AR 0509 (port 1) ou AR 0609 (port 2) passent à ON.



- Rem. : 1. Il faut toujours remettre le compteur à zéro lorsque l'on désire le relancer et utiliser l'une de ces deux méthodes. Le compteur se remet automatiquement à zéro au début ou à l'arrêt de l'exécution du programme.
2. Les transitions de signaux suivantes sont traitées comme impulsions d'incréméntation : front montant d'entrée A vers front montant d'entrée B vers front descendant d'entrée A vers front descendant d'entrée B. Les transitions de signaux suivantes sont traitées comme impulsions de décrémentation : front montant d'entrée B vers front montant d'entrée A vers front descendant d'entrée B vers front descendant d'entrée A.

Méthodes de remise à zéro

Pour remettre à zéro la valeur en cours de comptage, on peut utiliser le signal de phase Z + la RAZ programme ou la RAZ programme. Ce sont des méthodes identiques à celles du compteur rapide 0.

- Rem. : 1. Les bits de remise à zéro des compteurs rapides 1 et 2 (SR 25201 et SR 25202) un fois par scrutation. Assurez-vous qu'un bit de remise à zéro est à ON pendant au moins un cycle complet afin d'en assurer une lecture fiable.
2. La table de comparaison, l'état d'exécution et les résultats de comparaison de gamme seront retenus grâce à une remise à zéro (les comparaisons se poursuivent après RAZ).

Comptage d'interruptions du compteur rapide

Les tables de comparaison des compteurs rapides 1 et 2 sont identiques à celles du compteur rapide 0.

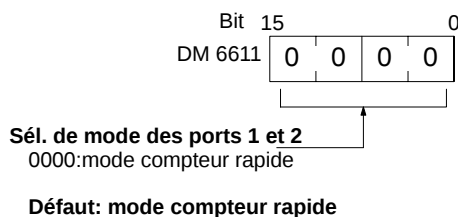
Configuration API

Lorsque vous utilisez les interruptions des compteurs rapides 1 et/ou 2, effectuez les sélections en mode PROGRAM avant l'exécution du programme.

Sélection de mode des ports 1 et 2 (DM 6611)

Spécifiez le mode compteur rapide pour les ports 1 et 2. Si ce mode n'est pas spécifié, CTBL(—) ne peut être utilisée pour les comparaisons de comptage.

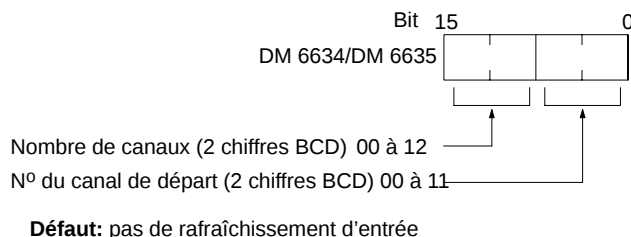
La lecture de cette sélection s'effectue à la mise sous tension de l'API. En cas de modification, l'API doit être débranché puis remis sous tension avant l'exécution du programme.



Rem. : si le DM 6611 est en mode par impulsion, une instruction de comparaison supplémentaire comme BCMP(—) peut être utilisée pour comparer la valeur en cours des compteurs rapides 1 et 2.

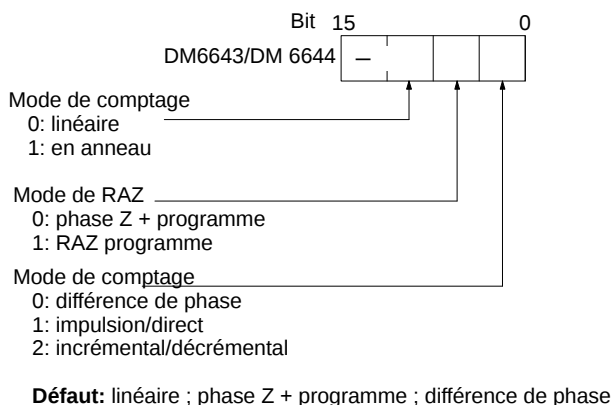
Sélection des canaux de rafraîchissement d'entrée (DM 6634 et DM 6635)

Le DM 6634 contient les sélections du canal de rafraîchissement d'entrée du compteur rapide 1 et le DM 6635 contient celles du compteur rapide 2. Effectuez ces sélections lorsqu'il est nécessaire de rafraîchir les entrées.



Sélections des compteurs rapides 1 et 2 (DM 6643 et DM 6644)

Le DM 6643 contient les sélections du compteur rapide 1 et le DM 6644 celles du compteur rapide 2. Ces sélections déterminent les paramètres de fonctionnement de ces compteurs rapides.



Programmation

Programmation des compteurs rapides 1 et 2 :

Les compteurs rapides 1 et 2 commencent leur comptage lorsque les sélections de configuration API ont été correctement effectuées ; si l'instruction CTBL(—)

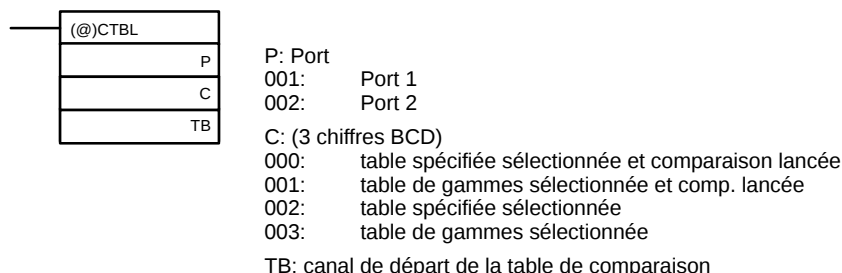
n'est pas exécutée, les comparaisons à l'aide des tables ne sont pas exécutées et les interruptions ne sont pas générées.

Les compteurs rapides 1 et 2 sont remis à zéro à la mise sous tension, à chaque début de fonctionnement de l'appareil et lorsqu'il s'arrête.

La valeur en cours du compteur rapide 1 est maintenue dans SR 232 et SR 233 et la valeur en cours du compteur rapide 2 est maintenue dans SR 234 et SR 235.

Contrôle des interruptions des compteurs rapides 1 et 2

- 1, 2, 3... 1. Utilisez l'instruction CTBL(—) pour sauvegarder la table de comparaison dans l'automate et commencez l'exécution des comparaisons :

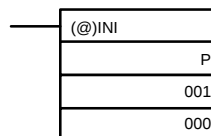


Si C est à 000, les comparaisons sont effectuées par la méthode de comparaison des valeurs spécifiées, si C est à 001, elles sont effectuées par la méthode de comparaison des gammes. La table de comparaison est sauvegardée et ensuite, les comparaisons commencent. Pendant leur exécution, les interruptions rapides sont effectuées en fonction de la table de comparaison. Pour de plus amples détails, reportez-vous à l'instruction CTBL(—) au chapitre 5.

Rem. : les résultats des comparaisons sont normalement stockés dans les AR 1100 à 1107 pendant la comparaison par gamme.

Si C est égal à 002, les comparaisons sont alors effectuées par la méthode de comparaison des valeurs spécifiées. Si C est égal à 3, elles sont effectuées par la méthode de comparaison de gammes. Pour ces deux sélections, la table de comparaison est sauvegardée mais la comparaison ne commence qu'avec l'exécution de l'instruction INI(—).

2. Pour arrêter la comparaison, exécutez l'instruction INI(—) de la façon suivante : spécifiez le port 1 ou 2 dans P (P=001 ou 002).



Pour relancer les comparaisons, sélectionnez "000" pour la deuxième opérande (exécution de comparaison) et exécutez l'instruction IN(—).

Une table sauvegardée peut être maintenue dans l'automate pendant l'exécution du programme jusqu'à la prochaine sauvegarde de table.

Lecture de la valeur en cours

Il existe deux façons de la lire : à partir de SR 232 et SR 233 (port 1) ou SR 234 et SR 235 (port 2), ou bien par l'utilisation de PRV(—).

Lecture de SR 232 et SR 233 ou SR 234 et SR 235

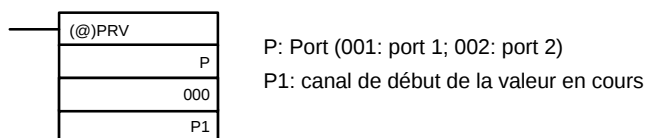
La valeur en cours du compteur rapide 1 est stockée dans SR 232 et SR 233 et la valeur en cours du compteur rapide 2 dans SR 234 et SR 235. En mode linéaire, le chiffre d'extrême gauche est égal à F pour les valeurs négatives.

4 chiffres d'extrême-gauche	4 chiffres d'extrême-droite	Mode linéaire	Mode en anneau
Port 1: SR 233	SR 232	F8388607 à 08388607 (-8388607 à 8388607)	00000000 à 0006499
Port 2: SR 235	SR 234		

Rem. : ces mots ne sont rafraîchis qu'une fois par cycle et il peut donc y avoir une différence entre eux et la valeur en cours réelle.

Utilisation de l'instruction PRV(--)

Lisez la valeur en cours du compteur rapide 0 en utilisant l'instruction PRV(—). Spécifiez le compteur rapide 1 ou 2 dans P (P=001 ou 002).



La valeur en cours du compteur rapide spécifié est stockée comme suit. En mode linéaire, le bit d'extrême gauche est égal à F pour les valeurs négatives.

4 chiffres d'extrême-gauche	4 chiffres d'extrême-droite	Mode linéaire	Mode en anneau
D+1	D	F8388607 à 08388607 (-8388607 à 8388607)	00000000 à 0006499

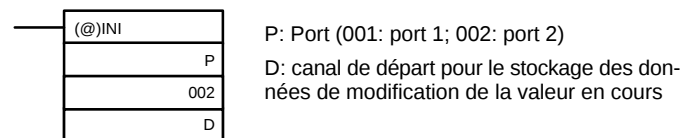
La valeur en cours est lue lorsque l'instruction PRV(—) est vraiment exécutée.

Modification de la valeur en cours

Il y a deux façons de modifier la valeur en cours des compteurs rapides 1 et 2 : on peut la remettre à zéro en utilisant les diverses méthodes de RAZ ou utiliser l'instruction INI(—).

Voici la méthode utilisant l'instruction INI(—) ; les méthodes de RAZ sont expliquées au début de la description des compteurs rapides 1 et 2.

Modification de la valeur en cours par l'instruction INI(—) :



4 chiffres d'extrême-gauche	4 chiffres d'extrême-droite	Mode linéaire	Mode en anneau
D+1	D	F8388607 à 08388607 (-8388607 à 8388607)	00000000 à 0006499

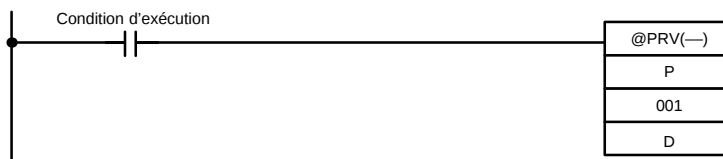
Pour spécifier un nombre négatif en mode linéaire, sélectionnez F comme chiffre d'extrême gauche.

Etat du compteur rapide

L'état des compteurs rapides 1 et 2 peut être déterminé en lisant l'état des drapeaux correspondant dans la zone AR ou en exécutant PRV(—). Voici les drapeaux de zone AR et leurs différentes fonctions :

Canal	Bit(s)	Fonction
AR 04	08 à 15	Indique l'état du compteur rapide 00: normal 01 ou 02: erreur du matériel 03: erreur de configuration API
AR 05	00 à 07	Drapeau de résultat de la comparaison du compteur rapide 1 pour les gammes 1 à 8 (0: hors de la gamme ; 1: dans la gamme)
	08	Drapeau de comparaison du compteur rapide 1 (0: arrêt ; 1: comparaison)
	09	Drapeau de non-respect de la gamme du compteur rapide 1 (0: normal ; 1: gamme non respectée)
AR 06	00 à 07	Drapeau de résultat de la comparaison du compteur rapide 2 pour les gammes 1 à 8 (0: hors de la gamme ; 1: dans la gamme)
	08	Drapeau de comparaison du compteur rapide 2 (0: arrêt ; 1: comparaison)
	09	Drapeau de non-respect de la gamme du compteur rapide 2 (0: normal ; 1: gamme non respectée)

L'état des compteurs rapides 1 et 2 peut également être déterminé par l'exécution de PRV(—). Spécifiez les compteurs rapides 1 ou 2 (P=001 à 002) et le canal de destination D. L'information concernant l'état est écrite dans les bits 00 et 01 de D. Les bits 02 à 15 sont à 0.



Les bits 00 et 01 de D contiennent l'état du compteur rapide spécifié.

Bit	Fonction
00	Drapeau de comparaison (0: arrêt ; 1: comparaison)
01	Drapeau de non-respect de la gamme (0: normal ; 1: gamme non respectée)

Exemple de fonctionnement

Cet exemple montre un programme émettant des impulsions standard à partir du port 1 et comptant les impulsions à partir du compteur rapide 1. Le compteur rapide fonctionne en mode incrémental/décrémental ; les impulsions en sens horaire incrémentent le compteur (entrée de phase B) et les impulsions en sens anti-horaire le décrémentent (entrée de phase A). Avant l'exécution du programme, sélectionnez la configuration de l'API comme suit et relancez l'API :

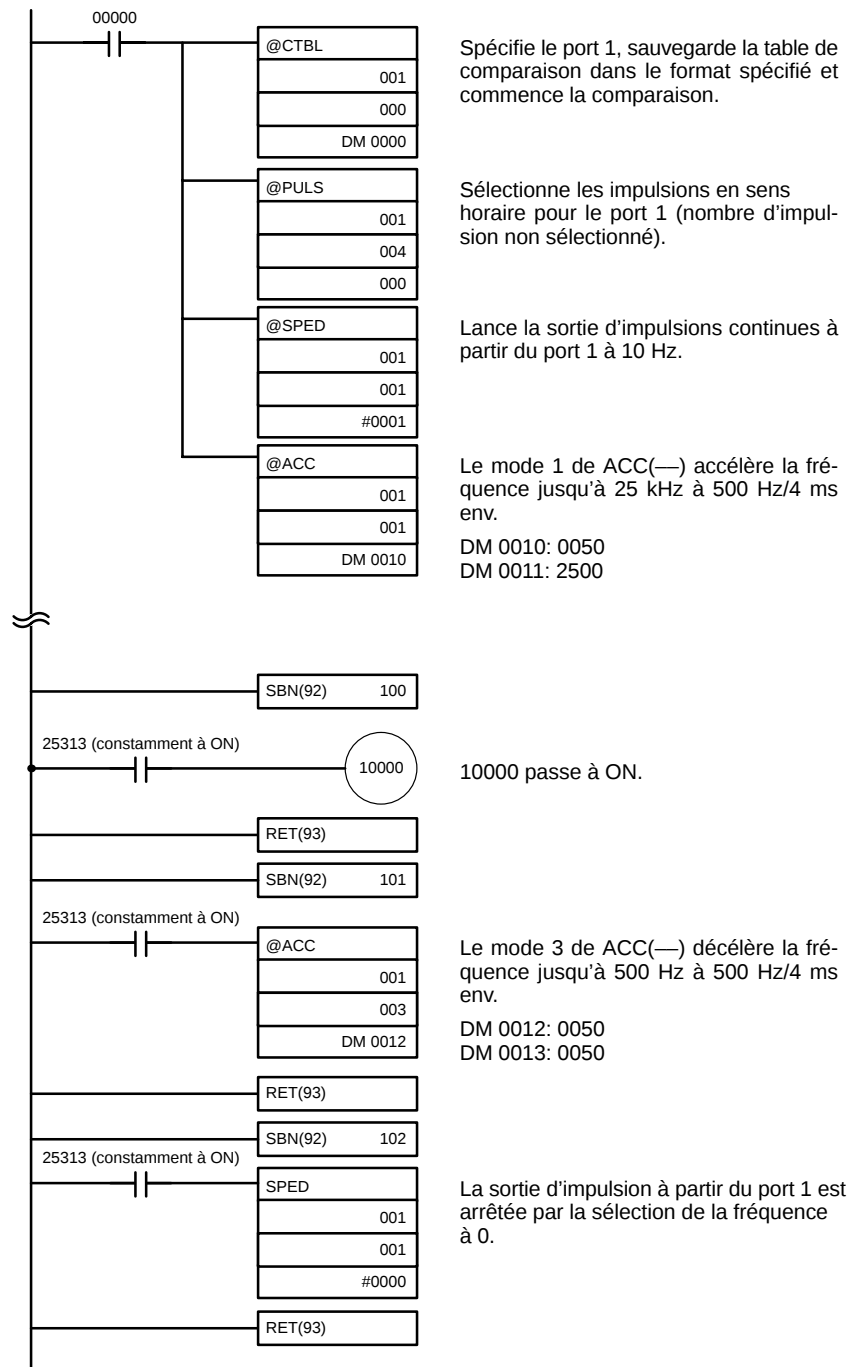
DM 6611: 0000 (mode compteur rapide)

DM 6643: 0002 (port 1 : sortie d'impulsions standard, mode de comptage linéaire, signal de phase Z + programme et mode incrémental/décrémental).

Les autres sélections de configuration utilisent les sélections par défaut (les entrées ne sont pas rafraîchies au moment de l'interruption).

En outre, les données suivantes sont stockées pour la table de comparaison :

DM 0000	0003	nombre de conditions de comparaison : 3
DM 0001	2500	valeur spécifiée 1 : 2500
DM 0002	0000	
DM 0003	0100	n° de sous-programme d'interruption de la comparaison 1 : 100
DM 0004	7500	valeur spécifiée 2: 7500
DM 0005	0000	
DM 0006	0101	n° de sous-programme d'interruption de la comparaison 2 : 101
DM 0007	0000	valeur spécifiée 2 : 10000
DM 0008	0001	
DM 0009	0102	n° de sous-programme d'interruption de la comparaison 3 : 102



1-4-8 Interruptions de compteur rapide absolu (CQM1-CPU44-E)

Les signaux à code binaire provenant d'un codeur absolu peuvent être envoyés vers les ports 1 et 2 de la CQM1-CPU44-E et comptés à 1 kHz. Le traitement des interruptions peut être exécuté selon le comptage.

Les 2 ports peuvent fonctionner séparément. Le compteur du port 1 est le compteur rapide absolu 1 et le compteur du port 2 est le compteur rapide absolu 2. Ce chapitre traite du fonctionnement des compteurs rapides absolus 1 et 2. Reportez-vous au manuel d'utilisation du CQM1 pour les questions de matériel telles que le câblage et l'équipement.

Processus

Signaux de comptage et modes de comptage

On peut utiliser 2 modes de comptage avec les compteurs rapides absolus 1 et 2. Le mode de comptage et les sélections de résolution sont spécifiés dans la configuration API (DM 6643 et DM 6644).

- 1, 2, 3...
- 1. Mode BCD :
Le code binaire du codeur absolu est d'abord converti en donnée binaire normale puis en BCD.
 - 2. Mode 360° :
Le code binaire du codeur absolu est converti en valeur angulaire (0° à 359°) en fonction de la sélection de résolution (les sélections de CTBL(—) se font par unités de 5°).

Valeurs en cours du compteur en mode BCD et 360° pour chaque sélection de résolution :

Résolution	Valeurs en cours possibles	
	Mode BCD	Mode 360°
8 bits	0 à 255	Sortie de la valeur en cours : 0° à 359° (unités de 1°)
10 bits	0 à 1023	
12 bits	0 à 4095	Sélections de la table de comparaison : 0° à 355° (unités de 5°)

Comptage d'interruptions du compteur rapide absolu

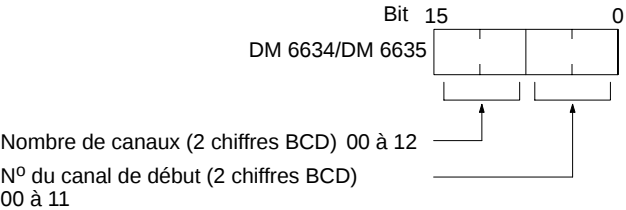
La valeur en cours du compteur peut être comparée à 16 valeurs spécifiées maximum ou à 8 gammes. Pour de plus amples détails sur les tables de comparaison utilisées avec les compteurs rapides absolus 1 et 2 reportez-vous au chapitre concernant l'enregistrement du table de comparaison de CTBL(—).

Configuration API

Lors de l'utilisation des compteurs rapides absolus 1 et/ou 2, effectuez les sélections en mode PROGRAM comme indiqué ci-dessous avant d'exécuter le programme.

Sélections du canal de rafraîchissement des entrées (DM 6634 et DM 6635)

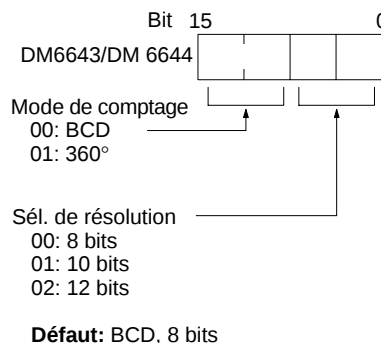
DM 6634 contient les sélections du canal de rafraîchissement des entrées du compteur rapide absolu 1 et DM 6635 celles du compteur rapide absolu 2. Effectuez ces sélections lorsqu'il est nécessaire de rafraîchir les entrées.



Défaut: pas de rafraîchissement d'entrée

Sélections du compteur rapide absolu (DM 6643 et DM 6644)

DM 6643 contient les sélections du compteur rapide absolu 1 et DM 6644 celles du compteur rapide absolu 2. Ces canaux déterminent les modes de comptage et les sélections de résolution.

**Compensation d'origine**

On peut compenser les décalages entre l'origine d'un codeur absolu et l'origine réelle. ce réglage peut être effectué séparément pour les ports 1 et 2.

Suivez la procédure suivante pour sélectionner la compensation d'origine ; une fois qu'elle a été sélectionnée, les données provenant du codeur seront ajustées avant d'être sorties comme valeurs en cours.

- 1, 2, 3... 1. Sélectionnez la position d'origine du codeur absolu.
2. Assurez-vous que la broche 1 de l'interrupteur DIP de l'UC est à OFF(activation des périphériques pour la ré-écriture des DM 6614 à 6655) et passez l'API en mode PROGRAM.
3. Sélectionnez la résolution dans DM 6643 ou DM 6644.
4. Assurez-vous qu'une erreur fatale ou un FALS 9C ne sont pas survenus.
5. Lisez la valeur en cours du compteur rapide dans IR 232 et IR 233 (port 1) ou IR 234 et IR 235 (port 2) pour déterminer la valeur en cours avant compensation d'origine.
6. Placez le drapeau de compensation d'origine du port 1 (SR 25201) ou le drapeau de compensation d'origine du port 2 (SR 25202) à ON à partir d'un périphérique.
La valeur de compensation sera écrite sur le DM 6611 (port 1) ou sur le DM 6612 (port 2) et le drapeau de compensation d'origine sera automatiquement placé à OFF. La valeur de compensation sera enregistrée en BCD entre 0000 et 4095, en fonction du mode compteur (BCD ou 360°).
7. Lisez la valeur en cours du compteur rapide pour déterminer la valeur en cours après compensation d'origine ; elle doit alors être égale à 0000.

La valeur de compensation est valide jusqu'à ce qu'elle soit modifiée par la procédure précédente.

Programmation

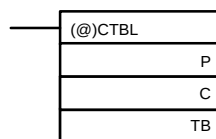
Voici comment programmer les compteurs rapides absolus 1 et 2 :

Les compteurs rapides absolus 1 et 2 commencent à compter lorsque leurs sélections de paramètres API ont été effectuées, mais si l'instruction CTBL(—) n'est pas exécutée, les tables de comparaison ne sont pas effectuées et les interruptions ne sont pas générées.

La valeur en cours du compteur rapide absolu 1 est maintenue dans IR 232 et IR 233 et la valeur en cours du compteur rapide absolu 2 est maintenue dans IR 234 et IR 235.

Contrôle des interruptions des compteurs rapides absolus

- 1, 2, 3... 1. Utilisez l'instruction CTBL(—) pour sauvegarder la table de comparaison dans l'API et commencer les comparaisons.



P: spécification du port (001 : port 1 ; 002 : port 2)

C: (3 chiffres BCD)

000: table spécifiée et comparaison lancée

001: table de gammes sélectionnée et comp. lancée

002: table spécifiée

003: table de gammes spécifiée

TB : premier canal de la table de comparaison

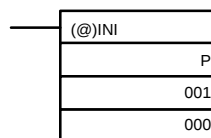
P sert à spécifier le port. Placez P à 001 pour spécifier le compteur rapide absolu 1 ou à 002 pour le compteur rapide absolu 2.

Si C est à 000, les comparaisons sont effectuées par la méthode de comparaison des valeurs spécifiées ; si C est à 001, elles sont effectuées par la méthode de comparaison des gammes. Les tables de comparaison sont sauvegardées et lorsque la sauvegarde est terminée, les comparaisons commencent. Lorsqu'elles ont été exécutées, les interruptions rapides sont effectuées en fonction de la table de comparaison. Pour de plus amples détails sur la structure des tables de comparaison, reportez-vous à l'instruction CTBL(—).

Rem. : les résultats des comparaisons sont normalement stockés dans AR 0500 à 0507 (port 1) et dans AR 0600 à 0607 (port 2) pendant l'exécution de la comparaison des gammes.

Si C est à 002, les comparaisons sont effectuées par la méthode de comparaison des valeurs spécifiées ; si C est à 003, elles sont effectuées par la méthode de comparaison des gammes. Pour ces deux sélections, la table de comparaison est sauvegardée mais les comparaisons ne commencent qu'avec l'exécution de l'instruction INI(—).

2. Pour arrêter les comparaisons, exécutez l'instruction INI(—) comme suit. Spécifiez le port 1 ou 2 dans P (P=001 ou 002).



Pour reprendre les comparaisons, placez la deuxième opérande à "000" (exécution de comparaison) et exécutez l'instruction INI(—).

Après sauvegarde d'une table, celle-ci est retenue dans l'API pendant l'exécution du programme et jusqu'à la prochaine sauvegarde.

Lecture de la valeur en cours

Il y a deux façons de lire la valeur en cours : la première consiste à la lire dans IR 232 et IR 233 (port 1) ou dans IR 234 et IR 235 (port 2) et la deuxième à utiliser l'instruction PRV(—).

Lecture de IR 232 et IR 233 ou de IR 234 et IR 235

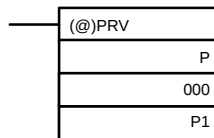
La valeur en cours du compteur rapide absolu 1 est stockée dans IR 232 et IR 233 et la valeur en cours du compteur rapide absolu 2 est stockée dans IR 234 et IR 235 :

4 chiffres d'extrême gauche	4 chiffres d'extrême droite	Mode BCD	Mode 360
Port 1: IR 233	IR 232	0000 0000 à 0000 4095	0000 0000 à 0000 0359
Port 2: IR 235	IR 234		

Rem. : ces canaux sont rafraîchis une fois par scrutation ; il peut donc exister une différence par rapport à la valeur en cours réelle.

Utilisation de l'instruction PRV(—)

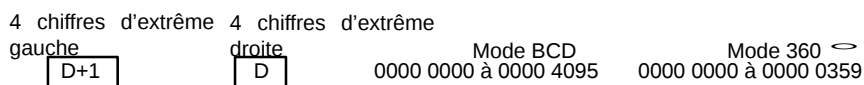
Lisez la valeur en cours du compteur rapide absolu par l'instruction PRV(—).
Spécifiez le compteur rapide absolu 1 ou 2 dans P (P=001 ou 002).



P: Port (001: port 1 ; 002: port 2)

P1: canal de départ de la valeur en cours

Stockage de la valeur en cours du compteur rapide absolu :



La valeur en cours est lue au moment de l'exécution de l'instruction PRV(—).

Lecture de l'état du compteur rapide absolu

L'état des compteurs rapides absolus 1 et 2 peut être déterminé par la lecture des drapeaux correspondant dans la zone AR ou par l'exécution de PRV(—).

Table des drapeaux de zone AR et de leurs fonctions :

Canal	Bit(s)	Fonction
AR 04	08 à 15	Indique l'état du compteur rapide absolu 00: normal 01 ou 02: erreur matériel 03: erreur de configuration API
AR 05	00 à 07	Drapeaux de résultat de comparaison du compteur 1 pour les gammes 1 à 8 (0: gamme non respectée ; 1: gamme respectée)
	08	Drapeau de comparaison du compteur 1 (0: arrêt ; 1: comparaison)
AR 06	00 à 07	Drapeaux de résultat de comparaison du compteur 1 pour les gammes 1 à 8 (0: gamme non respectée ; 1: gamme respectée)
	08	Drapeau de comparaison du compteur 2 (0: arrêt ; 1: comparaison)

L'état du drapeau de comparaison des compteurs rapides absolus 1 et 2 peut également être déterminé par l'exécution de PRV(—). Spécifiez le compteur rapide absolu 1 ou 2 (P=001 à 002) et le canal de destination D. L'état du drapeau (0 : arrêt ; 1: comparaison) est écrit dans le bit 00 de D. Les bits 01 à 15 sont placés à 0.



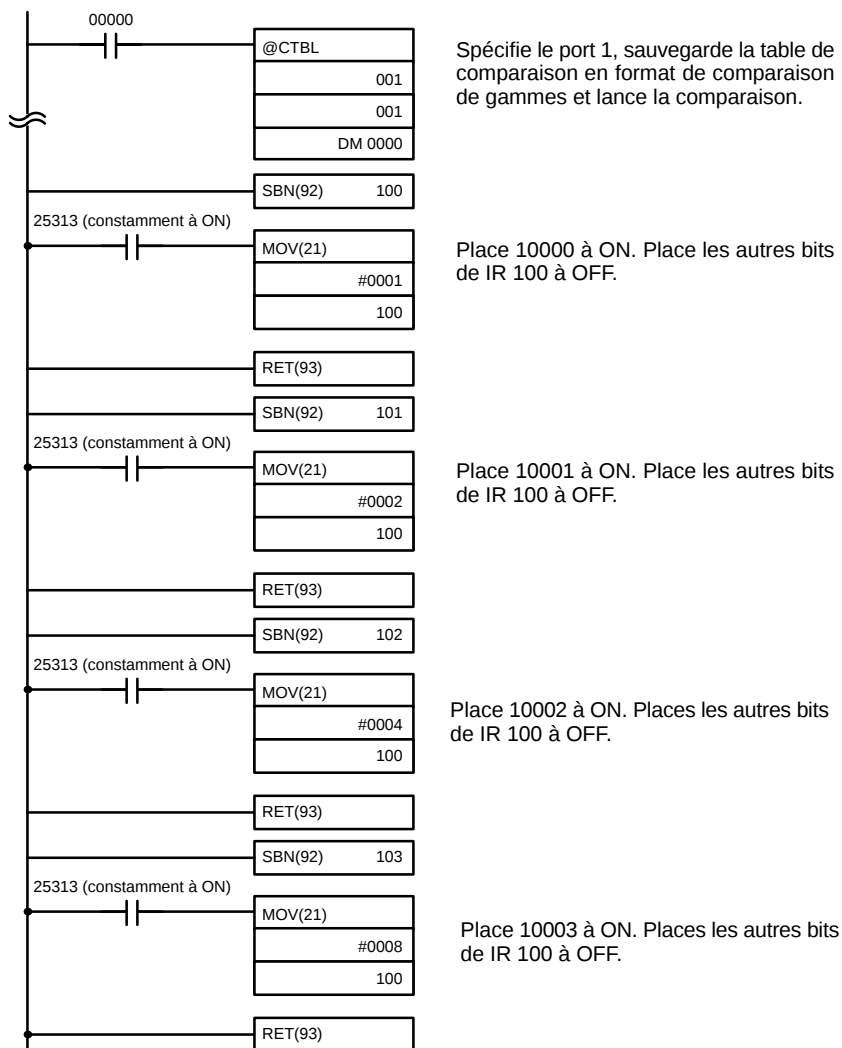
Exemple de fonctionnement

Cet exemple montre un programme recevant un signal d'entrée provenant d'un codeur rotatif absolu du port 1 et qui utilise cette entrée pour contrôler les sorties IR 10000 à 10003. Le compteur rapide absolu 1 est placé en résolution à 8 bits et en mode 360° ; les comparaisons se font par gammes. Avant l'exécution du programme, sélectionnez les DM 6643 à 0100 (port 1 : 360°, résolution : 8 bits).

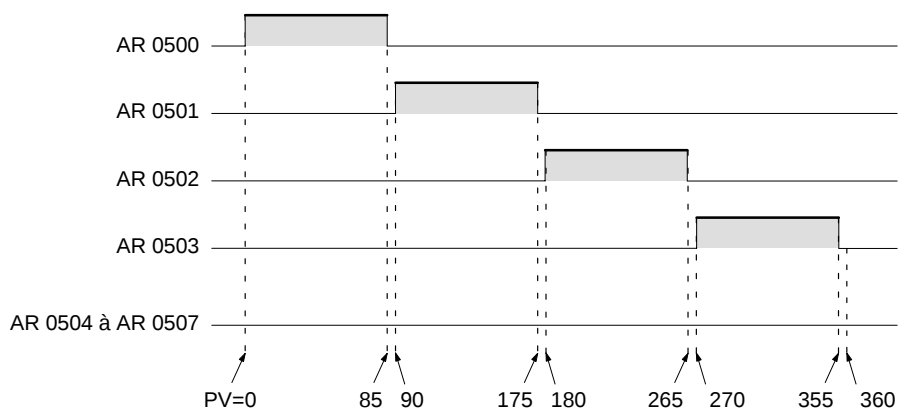
Les autres sélections de la configuration API sont celles de défaut (les entrées ne sont pas rafraîchies au moment de l'interruption).

Les données suivantes sont en outre stockées dans la table de comparaison :

DM 0000	0000	Limite basse #1 (0°)	→ Sélection 1re gamme (0° à 85°)
DM 0001	0085	Limite haute #1 (85°)	
DM 0002	0100	N° de sous-progr. 100	
DM 0003	0090	Limite basse #2 (90°)	→ Sélection 2e gamme (90° à 175°)
DM 0004	0175	Limite haute #2 (175°)	
DM 0005	0101	N° de sous-progr. 101	
DM 0006	0180	Limite basse #3 (180°)	→ Sélection 3e gamme (180° à 265°)
DM 0007	0265	Limite haute #3 (265°)	
DM 0008	0102	N° de ss-prog. 102	
DM 0009	0270	Limite basse #4 (270°)	→ Sélection 4e gamme (270° à 355°)
DM 0010	0355	Limite haute #4 (355°)	
DM 0011	0103	N° de ss-prog. 103	
DM 0012	0000	Limite basse #1 (0°)	→ Sélection 5e gamme (non utilisée)
DM 0013	0000	Limite haute #1 (0°)	
DM 0014	FFFF	Pas de n° de ss-prog.	
DM 0015	0000	Limite basse #1 (0°)	→ Sélection 6e gamme (non utilisée)
DM 0016	0000	Limite haute #1 (0°)	
DM 0017	FFFF	Pas de n° de ss-prog.	
DM 0018	0000	Limite basse #1 (0°)	→ Sélection 7e gamme (non utilisée)
DM 0019	0000	Limite haute #1 (0°)	
DM 0020	FFFF	Pas de n° de ss-prog.	
DM 0021	0000	Limite basse #1 (0°)	→ Sélection 8e gamme (non utilisée)
DM 0022	0000	Limite haute #1 (0°)	
DM 0023	FFFF	Pas de n° de ss-prog.	



Le schéma suivant illustre le lien existant entre la valeur en cours du compteur rapide absolu 1 et les drapeaux de résultat de la comparaison par gammes AR 0500 à 0507 pendant l'exécution du programme ci-dessus :



1-5 Fonction de communication

Différents types de communications possibles avec les ports du CQM1 :

Communication avec un ordinateur (SYSMAC-WAY)
 RS-232C avec ordinateur ou autre (E/S bidirectionnelles ASCII)
 Communication point par point avec un autre CQM1 (inter-automates)

Rem. : ces types de communication ne peuvent être effectués avec une UC CQM1-CPU11-E, qui n'est équipée que d'un port périphérique.

Voici les configurations API et les méthodes d'utilisation de ces types de communication.

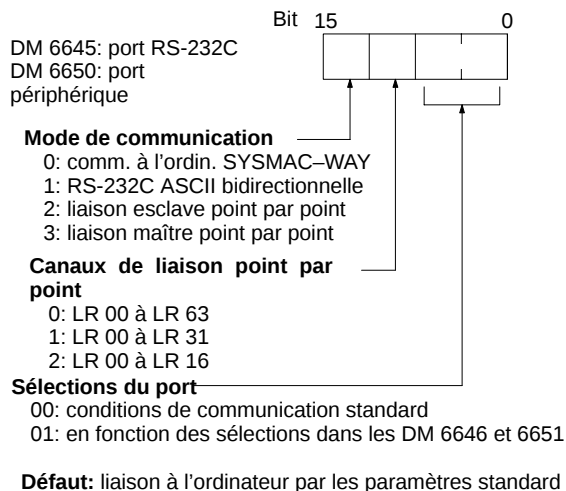
1-5-1 Configuration API

Les paramètres de configuration API des DM 6645 à 6654 sont utilisés pour la sélection des ports de communication. Les paramètres du port RS-232C dans les DM 6645 à 6649 peuvent être sélectionnés à partir du menu par le LSS.

Rem. : si la broche 5 de l'interrupteur DIP de l'API est à ON, les paramètres de communication de la configuration API sont ignorés et ce sont les paramètres suivants qui sont utilisés :

Mode:	communication à l'ordinateur
N° de noeud:	00
Bits de départ:	1 bit
Longueur de donnée:	7 bits
Bits d'arrêt :	1 bit
Parité:	paire
Vitesse:	2 400 bps
Délai de transmission :	aucun

Les sélections des DM 6645 et 6650 déterminent les paramètres de communication principaux de la façon suivante :



Liaisons point par point

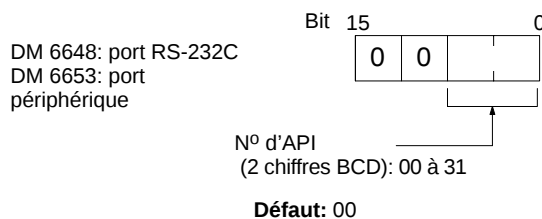
Liaison inter-automates/CQM1 (2 CQM1 maximum en liaison) : pour utiliser une liaison 1:1, les seules sélections nécessaires sont le mode de communication et les canaux de liaison. Choisissez 1:1 maître comme mode de communication d'un des API et 1:1 esclave pour l'autre ; sélectionnez ensuite les canaux de liaison dans l'API désigné comme maître. Les bits 08 à 11 ne sont valides que pour le maître de la liaison point par point.

Rem. : les communications en liaison point par point ne sont possibles que pour le port RS-232C. La sélection n'est pas possible pour le port périphérique.

N° d'API en liaison à l'ordinateur

Un numéro d'API doit être sélectionné pour les communications en liaison à l'ordinateur pour différencier les API e, réseau RS-422 lors de leur utilisation dans

les diverses communications. Ces sélections ne sont nécessaires que pour la liaison à l'ordinateur ; pour utiliser ce type de communication, la liaison doit être spécifiée dans le mode de communication et les paramètres de communication doivent être sélectionnés (cf chapitre suivant).

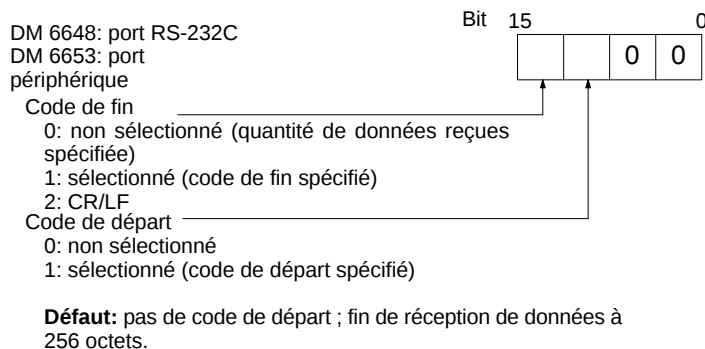


Choisissez 00 comme numéro d'API, sauf si plusieurs API se trouvent connectés en réseau.

Codes de fin, de départ et de réception de données RS-232C (ASCII)

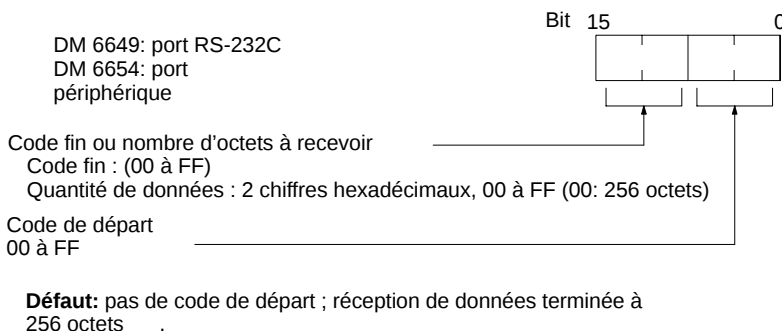
Les codes de fin et de départ ou de quantité de données à recevoir peuvent être sélectionnés comme sur le schéma suivant, si nécessaire. Ces sélections ne sont obligatoires que pour la communication RS-232C. Pour utiliser la communication RS-232C, celle-ci doit être spécifiée comme mode de communication ; la sélection des paramètres de communication est obligatoire (cf chapitre suivant).

Activation des codes de départ et de fin



Spécifiez la nécessité de sélectionner ou non un code de départ au début de la donnée et un code de fin à la fin de la donnée. A la place d'un code de fin, il est possible de spécifier un nombre d'octets à recevoir avant la fin de l'opération de réception. Le code de fin et le nombre d'octets à recevoir sont placés dans les DM 6649 ou 6654.

Sélection du code de départ, de fin et de la quantité de données à recevoir



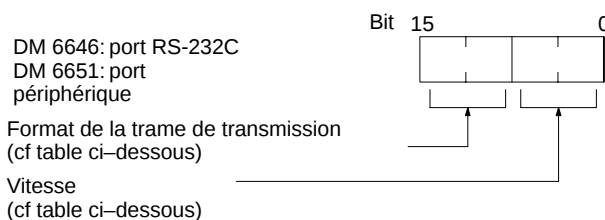
Liaison à l'ordinateur et paramètres de communication RS-232C

Sélectionnez la communication à l'ordinateur ou RS-232C, puis les paramètres de communication de façon à faire correspondre les conditions de communication aux sélections de l'appareil avec lequel la communication s'effectue.

Communications standard

Si les sélections suivantes sont satisfaisantes pour les conditions de communication, placez alors les deux chiffres d'extrême droite à 00. Les sélections des DM 6646 et 6651 sont ignorées pour cette sélection.

Bits de départ : 1 bit
 Longueur de données : 7 bits
 Bits d'arrêt : 2 bits
 Parité : paire
 Vitesse : 9 600 bps

Sélection des conditions de communication

Défaut: conditions de communication standard

Format de la trame de transmission

Sélection	Bits de départ	Longueur de donnée	Bits d'arrêt	Parité
00	1	7	1	Paire
01	1	7	1	Impaire
02	1	7	1	Aucune
03	1	7	2	Paire
04	1	7	2	Impaire
05	1	7	2	Aucune
06	1	8	1	Paire
07	1	8	1	Impaire
08	1	8	1	Aucune
09	1	8	2	Paire
10	1	8	2	Impaire
11	1	8	2	Aucune

Vitesse

Sélection	Vitesse
00	1 200 bps
01	2 400 bps
02	4 800 bps
03	9 600 bps
04	19 200 bps

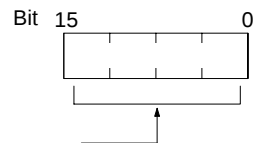
Retard à la transmission

Selon les appareils connectés au port RS-232, il peut être nécessaire de prévoir un temps de transmission.

DM 6647: port RS-232C
DM 6652: port
périphérique

Retard à la transmission
(4 chiffres BCD ; unité : 10 ms)

Défaut : pas de retard

**1-5-2 Câblage du port**

Reportez-vous au manuel d'utilisation du CQM1.

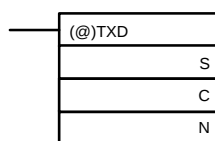
1-5-3 Communication à l'ordinateur

Ces types de communication ont été développés par OMRON pour la connexion d'un ordinateur et d'un ou plusieurs API par câble RS-232C (avec interface RS-422) et pour le contrôle des communications API à partir de l'ordinateur. En temps normal, l'ordinateur envoie une commande à l'API et celui-ci renvoie automatiquement une réponse. Les communications sont ainsi effectuées sans que les API soient activement impliqués, mais les API ont également la capacité de commencer la transmission de données lorsqu'un engagement direct est nécessaire.

En général, il y a deux façons de mettre en oeuvre la communication à l'ordinateur. L'une repose sur les commandes en mode C et l'autre sur les commandes FINS (mode CV). Le CQM1 permet d'effectuer uniquement des commandes en mode C. Pour de plus amples détails, reportez-vous au chapitre 6.

Procédure de communication Ce chapitre explique l'utilisation des communications à l'ordinateur pour l'exécution des transmissions de données à partir du CQM1. L'utilisation de cette méthode permet la transmission automatique de données à partir du CQM1 lorsque la donnée est modifiée et simplifie donc la communication en éliminant la nécessité d'une surveillance constante de l'ordinateur.

- 1, 2, 3...** 1. Vérifiez que AR 0805 (drapeau de transmission RS-232C prête) est à ON.
2. Utilisez l'instruction TXD(—) pour la transmission des données.



S: n° du canal de départ de transmission des données

C: données de contrôle

Bits 00 à 03

0: octets d'extrême gauche d'abord

1: octets d'extrême droite d'abord

Bits 12 à 15

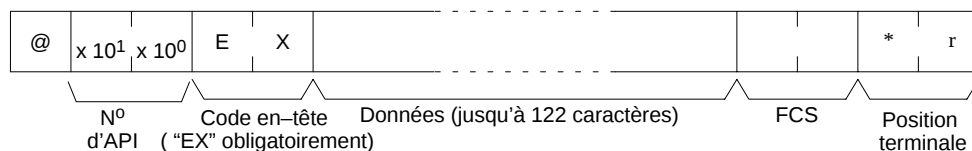
0: port RS-232C

1: port périphérique

N: nombre d'octets à envoyer (4 chiffres BCD)
0000 à 0256

De l'exécution de l'instruction à la fin de la transmission des données, AR 0805 (ou AR 0813 pour le port périphérique) reste à OFF. Il repasse à ON au moment où la transmission des données est terminée. L'instruction TXD(—) ne fournit pas de réponse ; pour recevoir confirmation de la réception des données par l'ordinateur, le programme de celui-ci doit être écrit de façon à confirmer l'écriture de données venant du CQM1.

La trame des données est la suivante pour les données transmises en mode RS-232C par l'instruction TXD(—).



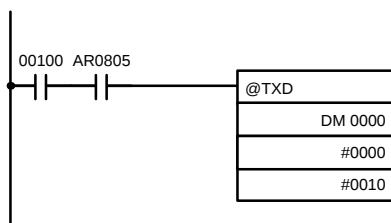
Pour remettre à zéro le port RS-232C (c'est-à-dire revenir à l'état initial), placez SR 25209 à ON. Pour remettre à zéro le port périphérique, placez SR 25208 à ON. Ces bits se placeront automatiquement à OFF après la remise à zéro.

Si l'instruction TXD(—) est exécutée alors que le CQM1 est en train de répondre à une commande provenant de l'ordinateur, la réponse est transmise avant l'exécution de l'instruction TXD(—). Dans tous les autres cas, la transmission par l'instruction TXD(—) est prioritaire.

Exemple d'application

Cet exemple montre un programme d'utilisation du port RS-232C en mode RS-232C, soit 5 DM pour la transmission de 10 octets de données (DM 0000 à 0004) vers l'ordinateur. On part du principe que les valeurs par défaut sont sélectionnées pour toutes les configurations API (c'est-à-dire que le port RS-232C est utilisé en mode RS-232C, que le numéro d'API est 00 et que les conditions de communication sont standard).

Du DM 0000 au DM 0004, "1234" est stocké dans chaque mot. A partir de l'ordinateur, exécutez un programme de réception des données du CQM1 avec les conditions de communication standard.



Si AR 0805 (drapeau de transmission prête) est à ON lorsque IR 00100 passe à ON, les dix octets de données (DM 0000 à 0004) sont transmis.

Le type de programme suivant doit être programmé dans l'ordinateur afin de recevoir les données. Ce programme permet à l'ordinateur de lire et d'afficher les données reçues en provenance de l'API tandis qu'une commande de lecture est exécutée afin de pouvoir lire les données venant de l'API.

```

10 'CQM1 SAMPLE PROGRAM FOR EXCEPTION
20 CLOSE 1
30 CLS
40 OPEN "COM:E73" AS #1
50 *KEYIN
60 INPUT "DATA -----", S$
70 IF S$="" THEN GOTO 190
80 PRINT "SEND DATA = "; S$
90 ST$=S$
100 INPUT "SEND OK? Y or N?=", B$
110 IF B$="Y" THEN GOTO 130 ELSE GOTO *KEYIN
120 S$=ST$
130 PRINT #1, S$                                ' Envoi de la commande vers l'API
140 INPUT #1, R$                                ' Réception de la réponse de l'API
150 PRINT "RECV DATA = "; R$
160 IF MID$(R$, 4, 2)="EX" THEN GOTO 210 ' Identification de la commande
                                         venant de l'API
170 IF RIGHT$(R$, 1)<>"*" THEN S$="" : GOTO 130
180 GOTO *KEYIN
190 CLOSE 1
200 END
210 PRINT "EXCEPTION!! DATA"
220 GOTO 140

```

La donnée reçue par l'ordinateur est la suivante (FCS est égal à "59.") :
 "@00EX1234123412341234123459*CR"

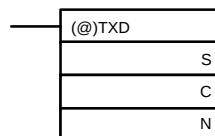
1-5-4 Communications RS-232C

Par la communication RS-232C, les données peuvent être imprimées sur une imprimante ou lues grâce à un lecteur de code barre. La fonction ON/OFF ("handshaking") n'est pas disponible en communication RS-232C.

Procédure de communication

Transmission

- 1, 2, 3... 1. Vérifiez que AR 0805 (drapeau de transmission du port RS-232C prête) est à ON.
2. Utilisez l'instruction TXD(—) pour la transmission des données :



S: n° du canal de départ des données à transmettre

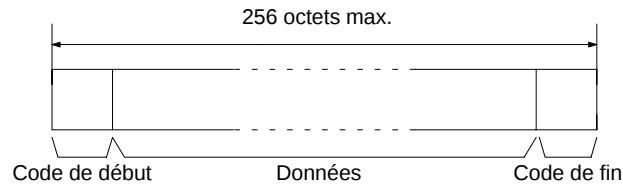
C: données de contrôle

N: nombre d'octets à transmettre (4 chiffres BCD), 0000 à 0256

De l'exécution de cette instruction jusqu'à la fin de la transmission des données, AR 0805 (ou AR0813 pour le port périphérique) reste à OFF (il repasse à ON à la fin de la transmission des données).

Les codes de début et de fin ne sont pas compris lorsque le nombre d'octets à transmettre est spécifié. La transmission la plus importante qui peut être émise avec ou sans code de début est de 256 octets. N est égal à 254 ou à 256 au maximum selon les désignations des codes de début et de fin. Si le nombre d'oc-

tets à envoyer est sélectionné à 0000, seuls les codes de début et de fin seront envoyés.

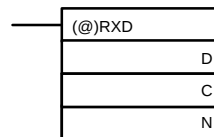


Pour remettre à zéro le port RS-232C (c'est-à-dire pour revenir à l'état initial), placez SR 25209 à ON. Pour remettre à zéro le port périphérique, placez SR 25208 à ON. Ces bits repassent à OFF automatiquement après la remise à zéro.

Réception

1, 2, 3...

- Vérifiez que AR 0806 (drapeau de réception RS-232C terminée) ou que AR 0814 (drapeau de réception de périphérique terminée) est à ON.
- Utilisez l'instruction RXD(—) pour recevoir les données.



D: n° du canal de début pour le stockage des données de réception

C: données de contrôle

Bits 00 à 03

0: octets d'extrême gauche d'abord

1: octets d'extrême droite d'abord

Bits 12 à 15

0: port RS-232C

1: port périphérique

N: Nombre d'octets stockés (4 chiffres BCD), 0000 à 0256

- Les résultats de la lecture de données reçues sont stockés dans la zone AR. Vérifiez que les opérations ont été correctement terminées. Le contenu de ces bits est remis à zéro à chaque exécution de l'instruction RXD(—).

Port RS-232C	Port périphérique	Erreur
AR 0800 à AR 0803	AR 0808 à AR 0811	Code d'erreur du port RS-232C (1 chiffre BCD) 0 : fin normale ; 1: erreur de parité ; 2 : erreur de trame ; 3 : erreur de dépassement
AR 0804	AR0812	Erreur de communication
AR 0807	AR0815	Drapeau de dépassement de réception (à la fin de la réception, les données suivantes ont été reçues avant d'être lues par l'instruction RXD.)
AR 09	AR10	Nombre d'octets reçus

Pour remettre à zéro le port RS-232C (c'est-à-dire pour revenir à l'état initial), placez SR 25209 à ON. Pour remettre à zéro le port périphérique, placez SR 25208 à ON. Ces bits passent automatiquement à OFF après la remise à zéro.

Les codes de départ et de fin ne sont pas compris dans AR 09 ni dans AR 10 (nombre d'octets reçus).

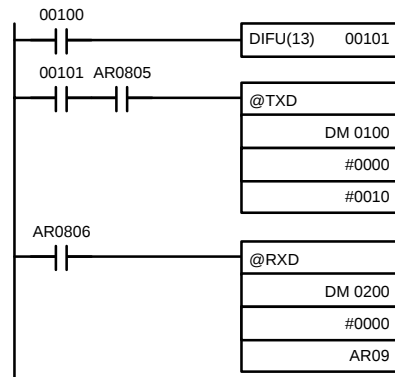
Exemple d'application

Cet exemple montre un programme d'utilisation du port RS-232C en mode RS-232C pour la transmission de 10 octets de données (DM 0100 à 0104) vers l'ordinateur et le stockage de données en provenance de l'ordinateur dans la zone DM commençant par DM 0200. Avant l'exécution du programme, la configuration suivante doit être réalisée :

DM 6645: 1000 (port RS-232C en mode RS-232C ; conditions de communication standard)

DM 6648: 2000 (pas de code de départ ; code de fin CR/LF)

On part du principe que les valeurs par défaut sont sélectionnées pour toutes les autres configurations API. Du DM 0100 au DM 0104, "3132" est stocké dans chaque mot. A partir de l'ordinateur, exécutez un programme pour recevoir les données du CQM1 avec les conditions de communication standard.



Si AR 0805 (drapeau de transmission prête) est à ON lorsque IR 00100 passe à ON, les dix octets de données (DM 0100 à DM 0104) sont transmis en commençant par les octets d'extrême gauche.

Lorsque AR 0806 (drapeau de réception terminée) est à ON, le nombre d'octets de données spécifié dans AR 09 est lu à partir de la mémoire tampon de réception du CQM1 et stocké en mémoire à partir du DM 0200 en commençant par les octets d'extrême gauche.

Les données sont les suivantes :
"31323132313231323132CR LF"

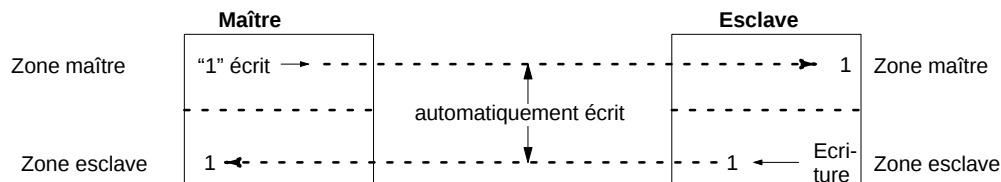
1-5-5 Communication en liaison point par point (inter-automates, 2 CQM1 maximum)

Si deux CQM1 sont reliés en liaison point par point par la connexion de leurs ports RS-232C, ils peuvent partager la même zone LR. L'un sert de maître et l'autre d'esclave.

Rem. : le port périphérique ne peut être utilisé pour les liaisons 1:1.

Liaison point par point

La liaison point par point permet à deux CQM1 de partager les données de leurs zones LR. Comme le montre le schéma suivant, lorsqu'une donnée est inscrite dans un canal de la zone LR d'une des cartes de liaison, elle est automatiquement ré-écrite à l'identique dans le canal correspondant de l'autre carte. Chaque API a des canaux spécifiés dans lesquels il peut écrire et d'autres canaux spécifiés dans lesquels l'autre API peut écrire. On peut lire mais non ré-écrire les canaux qui ont déjà été écrits par l'autre API.



Les canaux utilisés par chaque ordinateur sont ceux de la table suivante, en fonction des sélections du maître, de l'esclave et des canaux de liaison.

Sélections du DM 6645	LR 00 à LR 63	LR 00 à LR 31	LR 00 à LR 15
Canaux maîtres	LR00 à LR31	LR00 à LR15	LR00 à LR07
Canaux esclaves	LR32 à LR63	LR16 à LR31	LR08 à LR15

Procédure de communication Si les sélections des maître et esclave sont effectuées correctement, la liaison point par point commence alors automatiquement en alimentant simplement les deux CQM1. Le fonctionnement est indépendant des modes de chaque API.

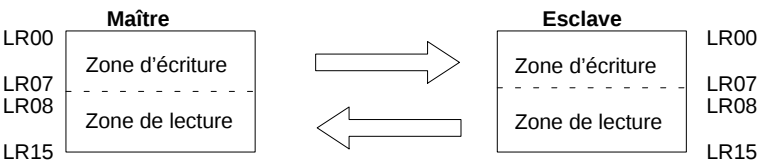
Exemple d'application

Cet exemple montre un programme de vérification des conditions par l'exécution d'une liaison point par point grâce aux ports RS-232C. Avant l'exécution du programme, effectuez les sélections suivantes dans la configuration API :

Maître : DM 6645: 3200 (maître de liaison point par point ;
zone utilisée : LR 00 à LR 15)

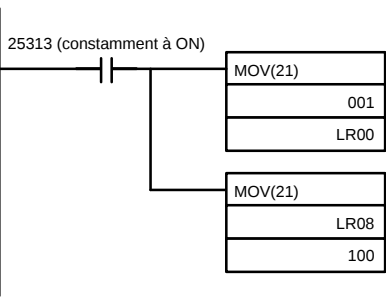
Esclave : DM 6645: 2000 (esclave de liaison point par point)

On part du principe que les valeurs par défaut sont sélectionnées pour toutes les autres configurations API. Canaux utilisés pour la liaison point par point :

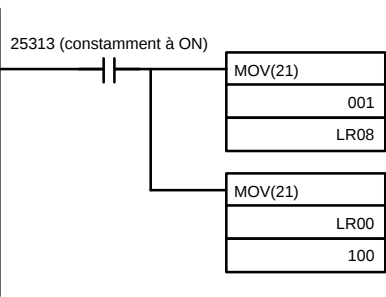


Lorsque le programme est utilisé dans le maître et dans l'esclave, l'état de IR 001 dans chaque carte est reflété dans le canal IR 100 de l'autre carte. De la même façon, l'état du canal IR 001 de l'autre carte est reflété dans le canal IR 100 de chaque carte. IR 001 est le canal d'entrée et IR 100 le canal de sortie.

Maître



Esclave



Chapitre 2

Caractéristiques des API CQM1

Ce chapitre présente les nouvelles caractéristiques du CQM1, y compris les nouvelles instructions disponibles grâce aux instructions étendues, une nouvelle caractéristique de surveillance de changement d'état d'un bit et la fonction de sélection analogique du CQM1-CPU42-E.

Si vous ne connaissez pas bien les API OMRON ni la programmation en schéma à relais, il vous faudra lire le chapitre 3 avant celui-ci. Les détails concernant les instructions de programmation du CQM1 se trouvent dans le chapitre 5.

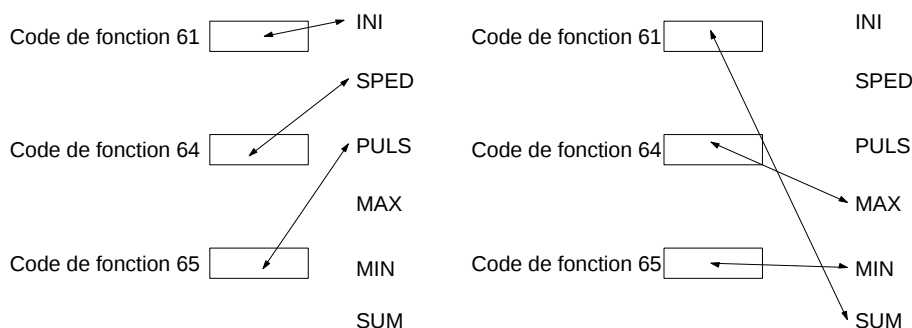
2-1	Instructions étendues	70
2-2	Instructions d'E/S étendues	72
2-2-1	TEN-KEY INPUT – TKY(—)	73
2-2-2	HEXADECIMAL KEY INPUT – HKY(—)	75
2-2-3	DIGITAL SWITCH INPUT – DSW(—)	77
2-2-4	7-SEGMENT DISPLAY OUTPUT – 7SEG(—)	80
2-2-5	Bits d'E/S alternés	82
2-3	Fonction macro	83
2-4	Surveillance de changement d'état d'un bit	84
2-5	Sélections analogiques (CQM1-CPU42-E uniquement)	85

2-1 Instructions étendues

Les instructions étendues du CQM1 répondent à certains besoins de programmation. Des codes de fonction peuvent être assignés à 18 instructions de fonction maximum pour permettre leur utilisation dans les programmes. Cela permet à l'utilisateur de choisir les instructions utiles à chaque programme du CQM1 et d'utiliser de façon plus efficace les codes de fonction pour l'entrée des fonctions.

Les abréviations des instructions sont suivies de "(—)", indiquant par là que l'utilisateur doit ajouter le code de fonction dans le tableau des instructions avant l'utilisation en programmation.

Exemple



A l'origine, les codes de fonction sont ceux cités ci-dessus (dans cet exemple, les instructions sont toutes liées à des sorties d'impulsion).

Si l'on n'utilise pas de sorties d'impulsion et si l'on a besoin de valeurs max., de valeurs min. et de sommes, la sélection d'instructions peut être utilisée comme ci-dessus pour ré-attribuer les instructions dans le tableau.

Les 18 codes de fonction suivants peuvent être utilisés pour les instructions d'extension 17, 18, 19, 47, 48, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 87, 88 et 89.

Les instructions d'extension peuvent être utilisées comme ci-dessous avec les codes de fonction par défaut attribués à l'origine. Les instructions suivies d'une astérisque "*" ne sont valables qu'avec les UC CQM1-CPU4 [-E.]

Instruction	Code de fonction
ASFT	17
TKY	18
MCMP	19
RXD	47
TXD	48
CMPL	60
INI	61
PRV	62
CTBL	63
SPED	64
PULS	65
SCL	66
BCNT	67
BCMP	68
STIM	69
DSW	87
7SEG	88
Instruction	Code de fonction
INT	89
HKY	---
FPD	---
SRCH	---
MAX	---
MIN	---
APR	---
LINE	---
COLM	---
SEC	---
HMS	---
SUM	---
FCS	---
HEX	---
AVG	---
PWM*	---
PID*	---
Instruction	Code de fonction
ADBL*	---
SBBL*	---
MBS*	---

Instruction	Code de fonction
DBS*	---
MBSL*	---
DBSL*	---
CPS*	---
CPSL*	---
NEG*	---
NEGL*	---
ZCP*	---
ZCPL*	---
XFRB*	---
PLS2*	---
ACC*	---
SCL2*	---
SCL3*	---

Avant de pouvoir être utilisés en programmation, les codes de fonction non attribués devront être attribués dans le tableau d'instructions que l'appareil de programmation et l'API utilisent. Les attributions des instructions étendues modifient la signification des instructions et opérandes ; veillez à bien effectuer les sélections du tableau avant de programmer et de transférer le tableau vers le CQM1 et avant l'exécution du programme.

Les tableaux d'instruction peuvent également être stockés sur cartouche mémoire lorsqu'ils sont utilisés. Soyez vigilants lors de l'utilisation d'une cartouche ayant déjà été utilisée avec un autre CQM1 et assurez-vous que le bon tableau soit présent.

Attention : si la broche numéro 4 de l'interrupteur DIP du CQM1 est à OFF, seules les instructions d'extension du tableau par défaut peuvent être utilisées ; le tableau créé par l'utilisateur sera ignoré. Le tableau d'instructions par défaut sera également sélectionné lorsque l'alimentation sera branchée et effacera les éventuelles sélections antérieures.

Assurez-vous que la broche 4 de l'interrupteur DIP de l'UC est à ON lors de la lecture de programme à partir d'une cartouche mémoire ayant un tableau d'instructions étendues. Si cette broche 4 est à OFF, le tableau d'instructions par défaut sera utilisé pour les instructions d'étendues dans les programmes lus à partir d'une cartouche mémoire (dans ce cas, le programme lu à partir de la cartouche et le programme réellement contenu dans la cartouche diffèrent lors de l'opération de comparaison).

2-2 Instructions d'E/S étendues

Les instructions d'E/S avancées permettent de contrôler, avec une seule instruction, les opérations auparavant complexes et qui impliquent l'usage d'appareils d'E/S externes (interrupteurs digitaux, affichages à 7 segments, etc.). Ce chapitre présente les instructions d'E/S avancées, également traitées à la fin du chapitre 5.

Il existe quatre instructions d'E/S avancées, comme le montre le tableau suivant. Toutes sont des instructions d'extension et on doit leur attribuer des codes de fonction avant de les utiliser.

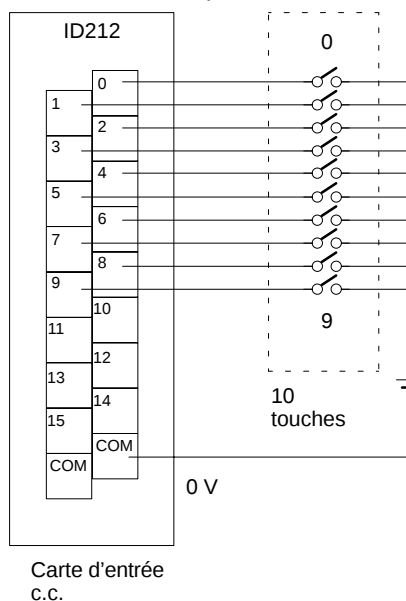
Dénomination	Abréviation	Fonction
TEN-KEY INPUT	TKY(—)	Entrée BCD à partir d'un clavier à 10 touches
HEXADECIMAL KEY INPUT	HKY(—)	Entrée hexadécimale à partir d'un clavier à 16 touches
DIGITAL SWITCH INPUT	DSW(—)	Entrée de la valeur sélectionnée à partir d'un interrupteur digital
7-SEGMENT DISPLAY OUTPUT	7SEG(—)	Sortie BCD vers un affichage à 7 segments

2-2-1 TEN-KEY INPUT (entrée 10 touches) – TKY(—)

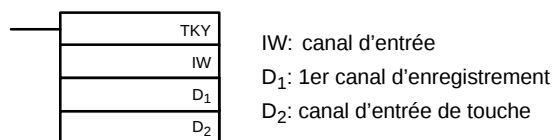
Cette instruction permet d'entrer 8 chiffres BCD à partir d'un clavier à 10 touches et utilise 10 points d'entrée.

Matériel

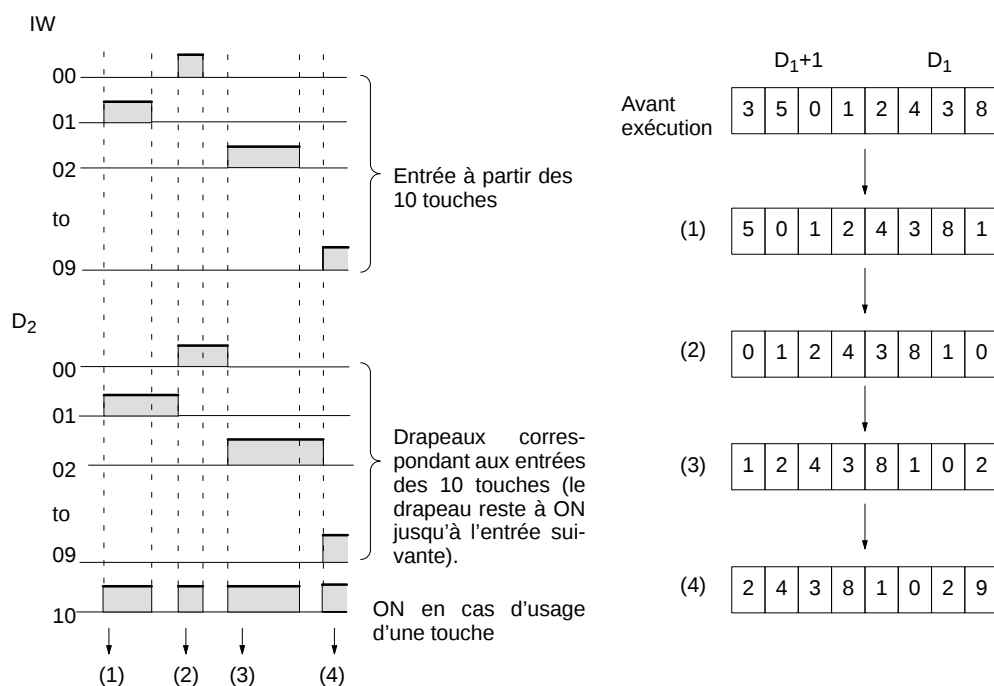
Préparez un clavier à 10 touches et connectez-le de façon à ce que les interrupteurs des touches numériques 0 à 9 soient entrés sur les points 0 à 9 comme sur le schéma suivant. On peut utiliser les bornes d'entrée de l'UC ou les entrées d'une carte d'entrée c.c. à 16 points d'entrée minimum.



Utilisation de l'instruction



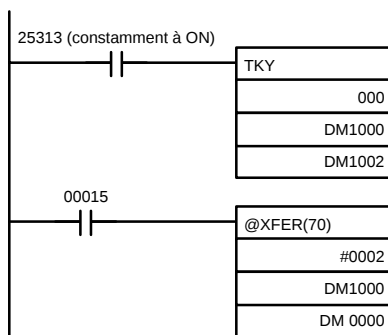
Si le canal d'entrée pour la connexion du clavier à 10 touches est spécifié pour IW, le fonctionnement se poursuit comme ci-dessous lorsque le programme est exécuté (on part du principe que plusieurs nombres ont déjà été entrés).



- Rem. : 1. Pendant l'utilisation d'une touche, les autres entrées de touches ne sont pas acceptées.
2. Si l'on entre plus de 8 chiffres, les chiffres sont effacés en commençant par l'extrême gauche.
3. Les bits d'entrée non utilisés ici peuvent l'être comme bits d'entrée ordinaires.

Exemple d'application

Dans cet exemple, un programme d'entrée de nombres à partir du clavier à 10 touches est illustré. Le clavier est ici connecté à IR 000.



L'information à 10 touches entrée sur IR 000 avec TKY(—) est convertie en BCD et stockée dans les DM 1000 et 1001. L'information clé est stockée dans le DM 1002.

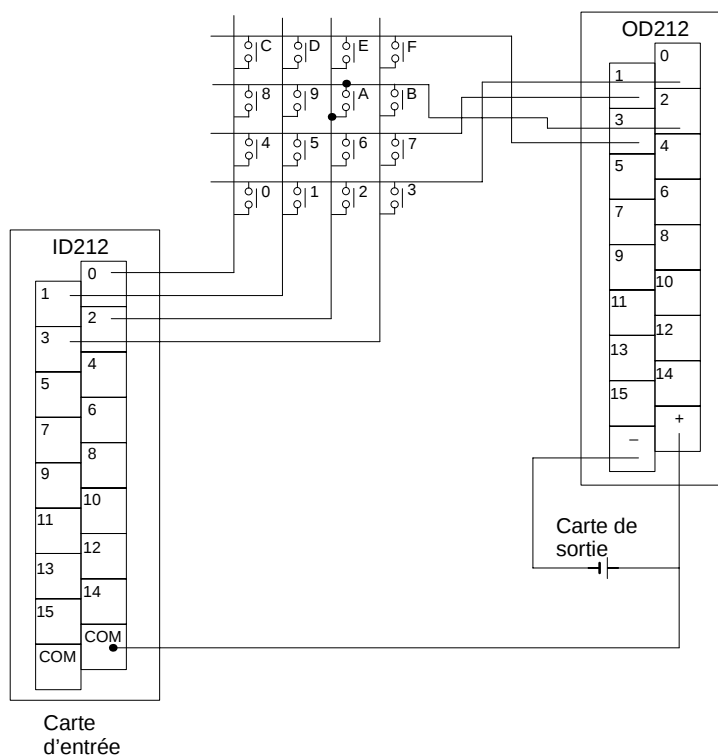
IR 00015 est utilisé comme touche "ENTER" et lorsque IR 00015 passe à ON, la donnée stockée dans les DM 1000 et 1001 est transférée vers les DM 0000 et 0001.

2-2-2 HEXADECIMAL KEY INPUT (entrée de touches hexadécimales) – HKY(--)

Cette instruction permet d'entrer 8 chiffres hexadécimaux à partir d'un clavier hexadécimal. On utilise 5 bits de sortie et 4 bits d'entrée.

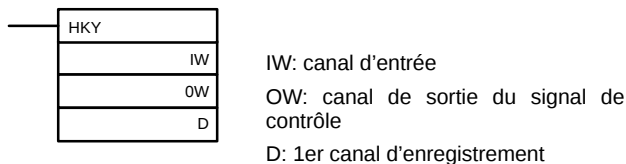
Matériel

Préparez le clavier hexadécimal et connectez les touches numériques 0 à F comme ci-dessous, pour les points d'entrée 0 à 3 et les points de sortie 0 à 3. Le point de sortie 4 passe à ON à l'occasion de toute utilisation de touche mais il n'est pas nécessaire de le connecter.



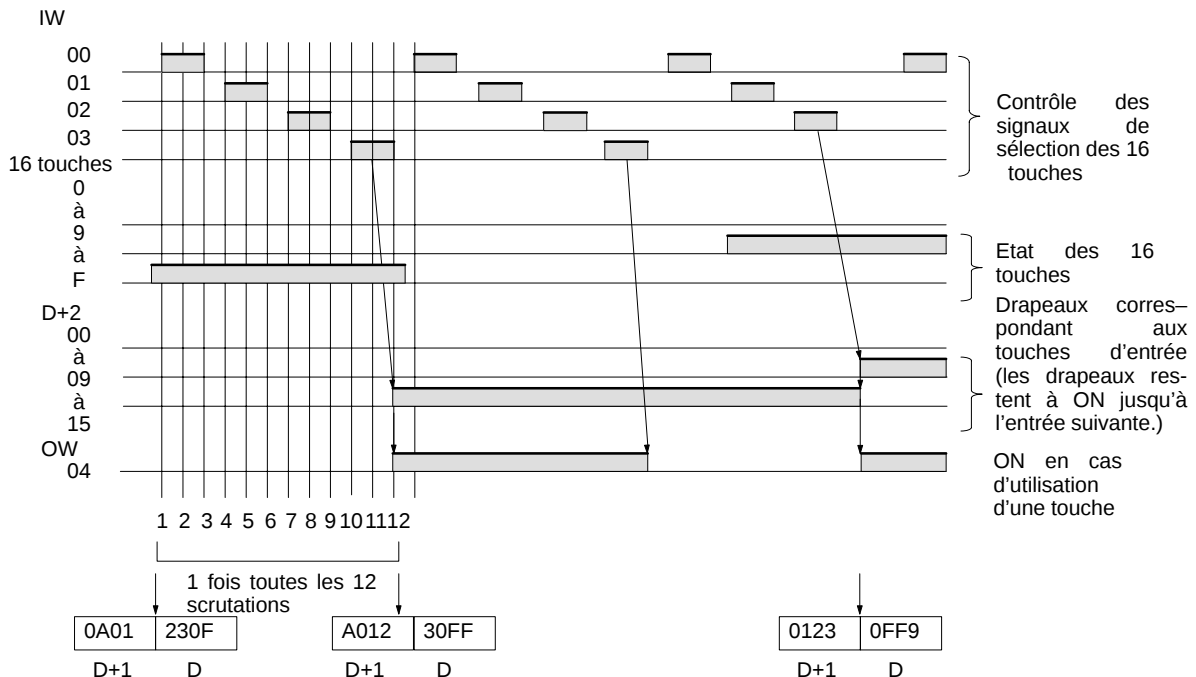
Les entrées peuvent être connectées aux bornes d'entrée de l'UC ou à une carte d'entrée c.c. ayant 8 points d'entrée minimum ; les sorties peuvent être connectées à partir d'une carte de sortie transistor ayant 8 points minimum.

Utilisation de l'instruction



Si le canal d'entrée de connexion du clavier hexadécimal est spécifié pour IW et que le canal de sortie est spécifié pour OW, le fonctionnement se poursuit

comme ci-dessous lors de l'exécution du programme (on considère que plusieurs nombres ont déjà été entrés).



SR 25408 passe à ON pendant l'exécution de HKY(—).

Rem. : 1. N'utilisez pas HKY(—) plus d'une fois dans le même programme.

2. Lors de l'utilisation de HKY(—), sélectionnez le filtre d'entrée du canal d'entrée correspondant de façon à ce qu'il soit inférieur au temps de scrutation (les filtres d'entrée peuvent être changés à partir du DM 6620).

3. Pendant l'utilisation d'une touche, les autres entrées de touches ne sont pas acceptées.

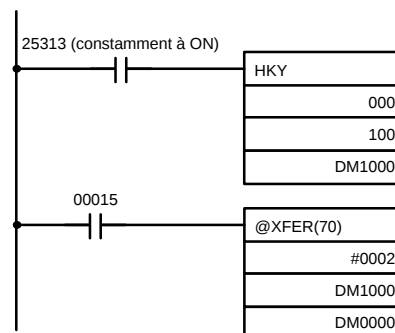
4. Si l'on entre plus de 8 chiffres, les chiffres sont effacés en commençant par l'extrême gauche.

5. Les bits d'entrée et de sortie non utilisés ici peuvent être comme bits d'entrée et de sortie ordinaires.

Avec cette instruction, une entrée de touche peut être lue en 3 à 12 scrutations. Il faut plus d'un cycle car les touches ON ne peuvent être déterminées que lorsque les sorties passent à ON.

Exemple d'application

Cet exemple illustre un programme d'entrée de nombres à partir d'un clavier hexadécimal. On suppose que le clavier hexadécimal est connecté à IR 000 (entrée) et à IR 100 (sortie).



L'information hexadécimale clé entrée sur IR 000 par HKY(—) est convertie en code hexadécimal et stockée dans les mots DM1000 et DM1001.

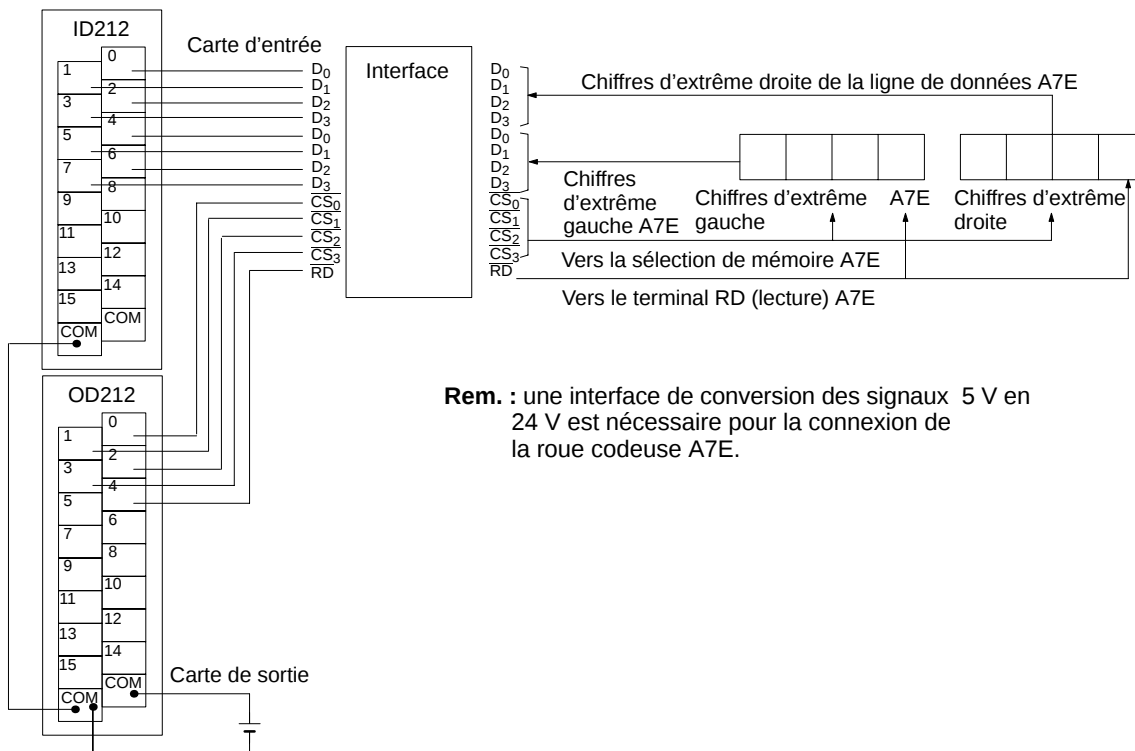
IR 00015 est utilisé comme touche "ENTER" et lorsque IR 00015 passe à ON, les nombres stockés dans les DM 1000 et DM 1001 sont transférés vers les DM 0000 et 0001.

2-2-3 DIGITAL SWITCH INPUT (entrée roue codeuse)– DSW(—)

A l'aide de cette instruction, les valeurs sélectionnées à 4 ou 8 chiffres BCD sont lues à partir d'une roue codeuse. DSW(—) utilise 5 bits de sortie et 4 ou 8 bits d'entrée (pour 4 ou 8 chiffres).

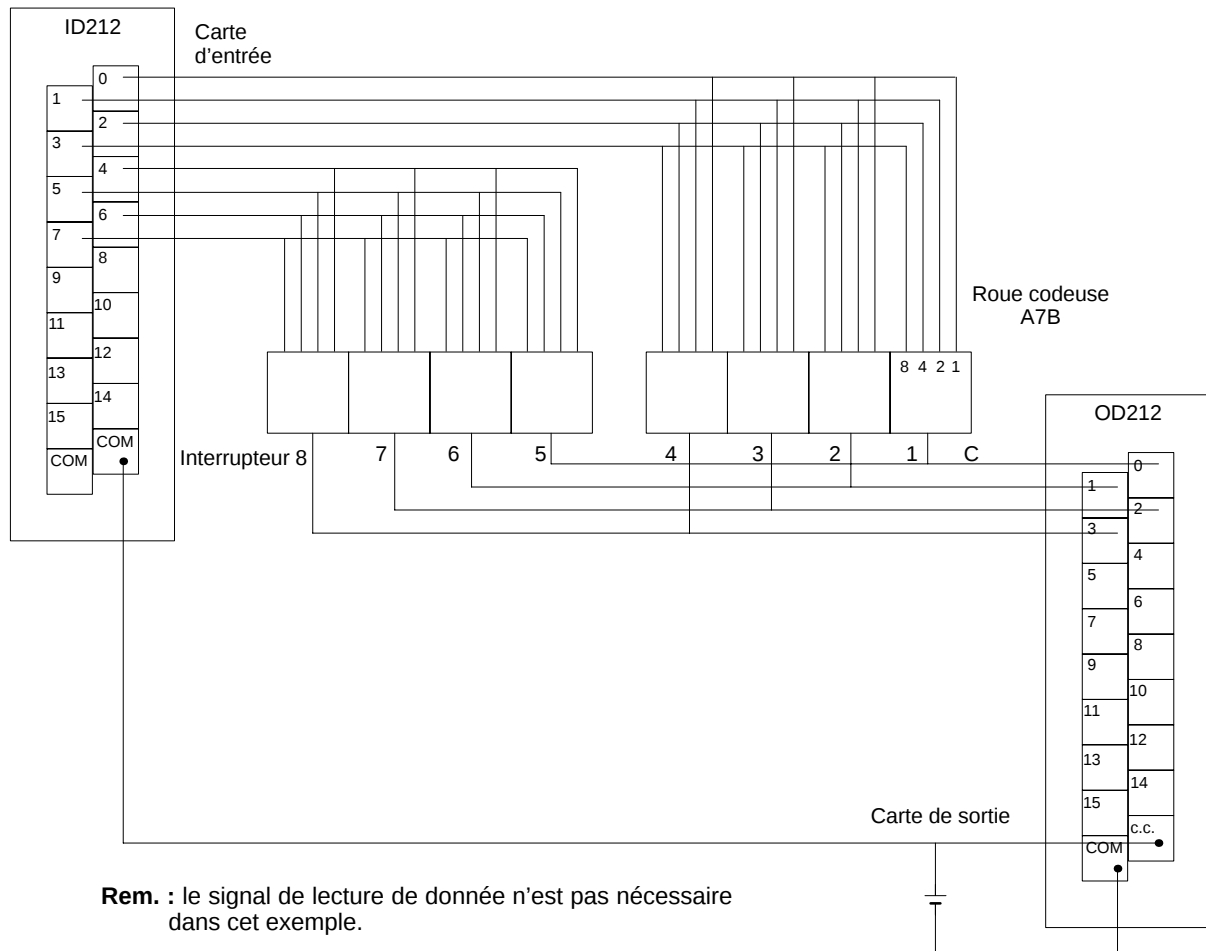
Matériel

Connectez la roue codeuse et les cartes d'entrée et de sortie comme sur le schéma ci-dessous. Une entrée à 8 chiffres est illustrée ici ; si vous utilisez une entrée à 4 chiffres, connectez D0 à D3 entre la roue codeuse et les points d'entrée 0 à 3. Dans les deux cas, le point d'entrée 5 passe à ON lorsqu'un tour de données a été lu mais il n'est pas nécessaire de connecter le point de sortie 5, sauf si l'application l'exige.



Rem. : une interface de conversion des signaux 5 V en 24 V est nécessaire pour la connexion de la roue codeuse A7E.

L'exemple suivant illustre les connexions d'une roue codeuse A7B.

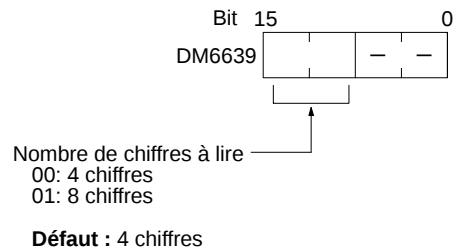


Les entrées peuvent être connectées aux bornes d'entrée de l'UC ou à une carte d'entrée c.c. ayant 8 points d'entrée max. ; les sorties peuvent être connectées à partir d'une carte de sortie transistor ayant 8 points de sortie max.

Préparatifs

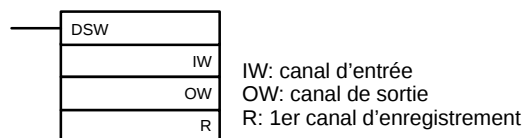
Avec DSW(—), effectuez les sélections suivantes dans la configuration API en mode PROGRAM avant l'exécution du programme.

Sélections de la roue codeuse (configuration API)

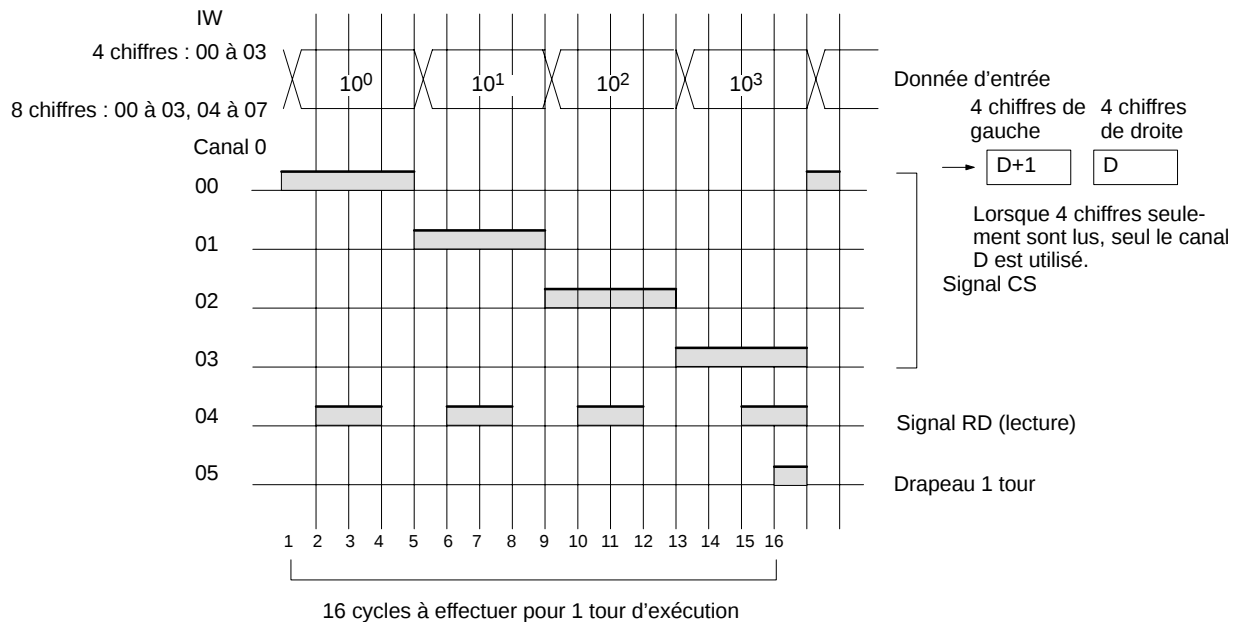


N'effectuez aucun changement dans les bits 0 à 7 ; ils ne sont pas liés à l'instruction DSW(—).

Utilisation de l'instruction



Si le canal d'entrée de la roue codeuse est spécifié pour IW et que le canal de sortie est spécifié pour OW, le fonctionnement est le suivant lors de l'exécution du programme :



SR 25410 passe à ON pendant l'exécution de DSW(—).

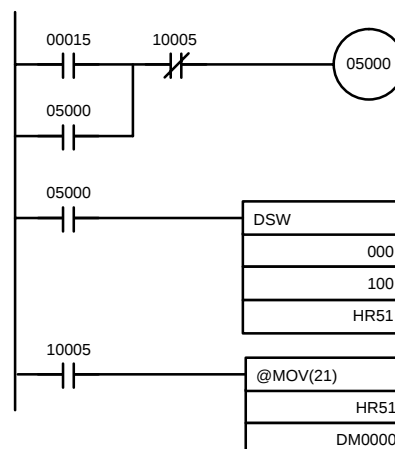
Rem. : 1. N'utilisez pas DSW(—) plus d'une fois dans le même programme.

2. Lors de l'utilisation de DSW(—), sélectionnez le filtre d'entrée du canal d'entrée correspondant pour une valeur inférieure au temps de scrutation (les filtres d'entrée peuvent être modifiées à partir de DM 6620). Les caractéristiques de la roue codeuse doivent également être considérées lors de la conception du système et du programme.
3. Les bits d'entrée et de sortie non utilisés ici peuvent l'être comme bits d'entrée et de sortie ordinaires.

Avec cette instruction, les valeurs sélectionnées à 4 ou 8 chiffres peuvent être lues en 16 cycles.

Exemple d'application

Cet exemple montre un programme de lecture de 4 chiffres BCD à partir de la roue codeuse. On suppose que la roue codeuse est connectée à IR 000 (entrée) et à IR 100 (sortie) et que les sélections par défaut sont valables pour toute la configuration API (4 chiffres à lire).



Lorsque IR 00015 passe à ON, IR 05000 se maintient lui-même à ON jusqu'à ce que le drapeau "1 tour" (IR 10005) passe à ON à la fin du premier tour de lecture accompli par DSW(—).

Les données sélectionnées à partir de la roue codeuse digitale par DSW(—) sont stockées dans HR 51.

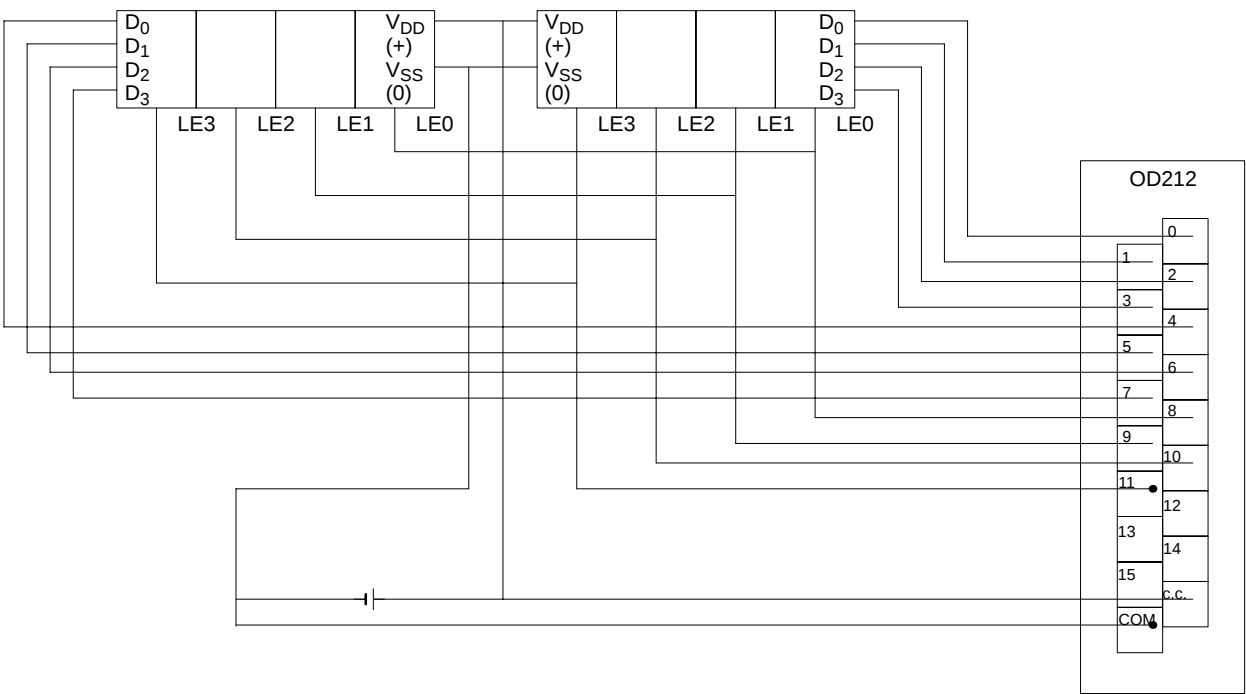
Lorsque le drapeau "1 tour" passe à ON (10005) à la fin de la lecture, le nombre stocké dans HR 51 est transféré vers DM 0000.

2-2-4 7-SEGMENT DISPLAY OUTPUT (affichage 7 segments) – 7SEG(—)

Cette instruction permet de sortir des données de canal sur un afficheur à 7 segments. On utilise 8 bits de sortie (pour 4 chiffres) ou 12 bits de sortie (pour 8 chiffres).

Matériel

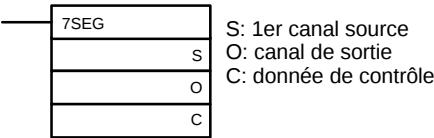
L'afficheur à 7 segments est connecté à une carte de sortie comme sur le schéma ci-dessous. Pour l'affichage de 4 chiffres, les données de sortie (D0 à D3) sont connectées aux points 0 à 3 et les sorties verrouillées (CS0 à CS3) sont connectées aux points de sortie 4 à 7. Le point de sortie 12 (affichage de 8 chiffres) ou le point de sortie 8 (affichage de 4 chiffres) passe à ON lorsqu'un tour de données est affiché, mais il n'est pas nécessaire de les connecter, sauf si l'application l'exige.



Les sorties peuvent être connectées à partir d'une carte de sortie à transistor ayant 8 points d'entrée min. (4 chiffres) ou 16 points d'entrée min. (8 chiffres).

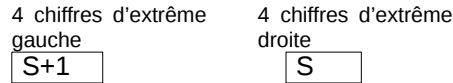
- Rem. : 1. Les sorties des cartes de sortie fonctionnent normalement en logique négative (seul le type de sortie PNP fonctionne en logique positive).
2. L'afficheur à 7 segments peut exiger une logique positive ou négative, selon le modèle.

Utilisation de l'instruction



Si le premier canal maintenant les données à afficher est spécifié en S, que le canal de sortie est spécifié en O et que la valeur de sélection venant du tableau ci-dessous est spécifiée en C, le fonctionnement est le suivant pendant l'exécution du programme :

Format de stockage des données



Si 4 chiffres sont affichés, seul le canal S est utilisé.

Valeurs sélectionnées pour la logique et le nombre de chiffres (C)

Nombre de chiffres affichés	Logique de l'entrée de données sur l'afficheur et de la carte de sortie	Logique de l'entrée verrouillée de l'afficheur et de la carte de sortie	Donnée de sélection C
4 chiffres (4 chiffres, 1 bloc)	Identique	Identique	000
		Différente	001
	Différente	Identique	002
		Différente	003
8 chiffres (4 chiffres, 2 blocs)	Identique	Identique	004
		Différente	005
	Différente	Identique	006
		Différente	007

Rem. : ne pas attribuer à C de valeur qui ne soit comprise entre 000 et 007.

Fonction	Bit(s) en O		Etat de sortie (la logique des données et du verrouillage dépend de C)
	(4 chiffres /1 bloc)	(4 chiffres /2 blocs)	
Sortie de données	00 à 03	00 à 03 04 à 07	Rem. : 0 à 3 : sortie de données pour le canal S 4 à 7 : sortie de données pour le canal S+1
Sortie de verrouillage 0	04	08	
Sortie de verrouillage 1	05	09	
Sortie de verrouillage 2	06	10	
Sortie de verrouillage 3	07	11	
Drapeau "1 tour"	08	12	

SR 25409 passe à ON pendant l'exécution de 7SEG(—).

Rem. : 1. N'utilisez pas 7SEG(—) plus d'une fois dans le même programme.

2. Le temps de scrutation et les caractéristiques de l'afficheur à 7 segments doivent être considérés lors de la conception du système.

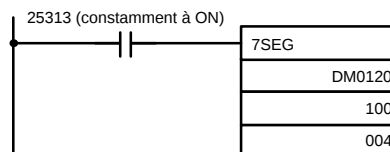
3. Les bits de sortie non utilisés ici peuvent l'être comme bits de sortie ordinaires.

Avec cette instruction, 4 ou 8 chiffres peuvent être affichés en 12 cycles.

Le fonctionnement commence à partir de la première exécution, quel que soit l'état précédent.

Exemple d'application

Cet exemple illustre un programme d'affichage de nombres BCD à 8 chiffres du CQM1 sur l'afficheur à 7 segments. On suppose que l'afficheur à 7 segments est connecté au canal de sortie IR 100, que la carte de sortie utilise une logique négative et que la logique de l'afficheur est également négative pour les signaux de données et de verrouillage.



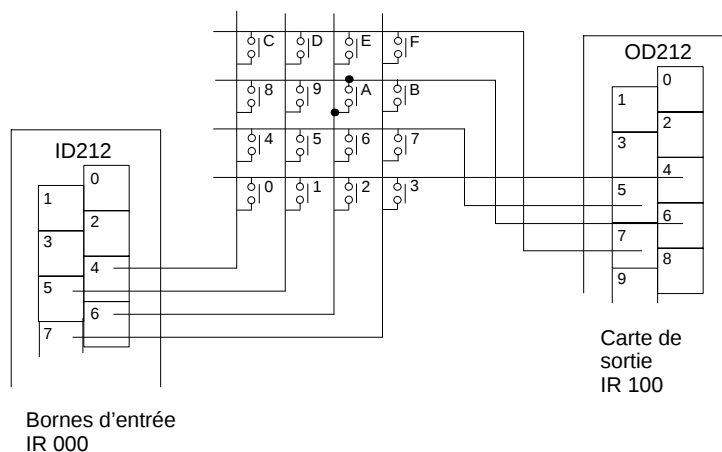
Les données BCD à 8 chiffres du DM 0120 (4 chiffres d'extrême droite) et du DM 0121 (4 chiffres d'extrême gauche) sont toujours affichées par 7SEG(—). Lorsque le contenu des DM 0120 et 0121 change, l'affichage est également modifié.

2-2-5 Bits d'E/S alternés

Bien que les instructions d'E/S avancées utilisent généralement des bits d'E/S à partir du bit 00 des canaux spécifiés, elles peuvent être programmées par le biais de canaux intermédiaires pour utiliser d'autres bits d'E/S. L'exemple suivant illustre le moyen d'y parvenir par HKY(—).

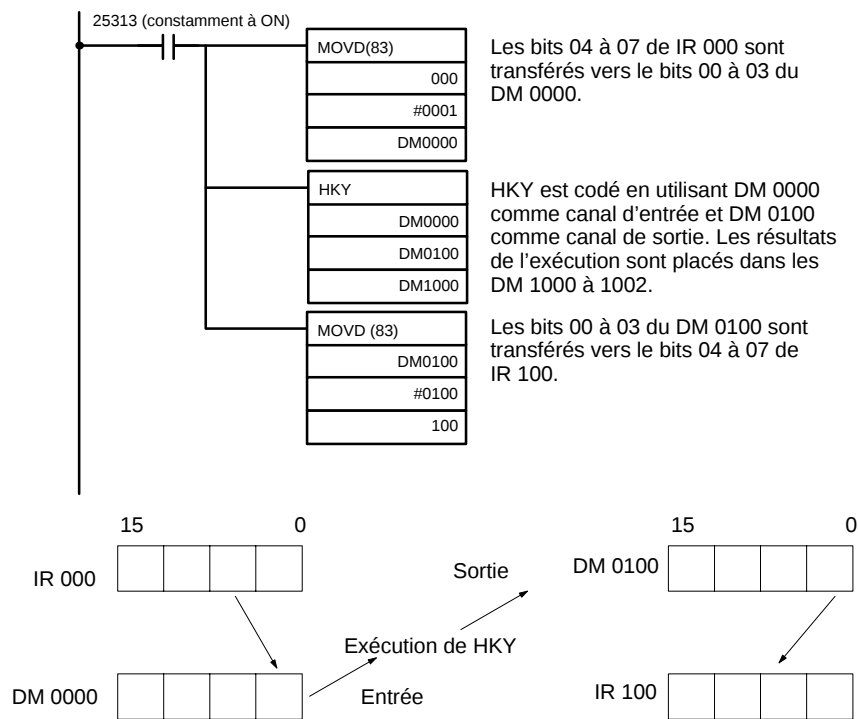
Exemple

Les exemples de programmation et de câblage suivants montrent comment utiliser les bits d'entrée IR 00004 à IR 00007 et les bits de sortie IR 10004 à IR 10007 pour entrer les valeurs à l'aide d'un clavier hexadécimal.

Schéma de câblage (partiel)

Rem. : les lignes de puissances ne figurent pas.

Programme

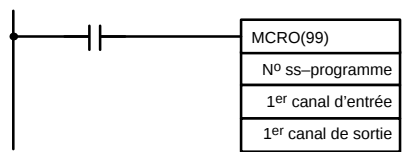


2-3 Fonction macro

La fonction macro permet d'utiliser un seul sous-programme (schéma de programmation) en changeant simplement les numéro de canal d'E/S. Un certain nombre de parties de programme similaires peuvent être gérées avec un seul sous-programme, cela permettant de réduire considérablement le nombre d'étapes et de clarifier le programme.

Utilisation des macros

Pour utiliser une macro, appelez un sous-programme par l'instruction MACRO MCRO(99) comme ci-dessous à la place de SBS(91) (SUBROUTINE ENTRY).



Lors de l'exécution de MCRO(99), la procédure est la suivante :

- 1, 2, 3...
1. Le contenu des 4 canaux consécutifs commençant par le premier canal d'entrée est transféré vers les IR 096 à 099. Le contenu des 4 canaux consécutifs commençant par le premier canal de sortie est transféré vers les IR 196 à 199.
 2. Le sous-programme spécifié est exécuté jusqu'à l'exécution de RET(93) (retour de sous-programme).
 3. Le contenu des IR 196 à 199 est transféré vers les 4 canaux consécutifs commençant par le premier canal de sortie.
 4. L'instruction MCRO(99) est alors terminée.

Lorsque MCRO(99) est exécutée, le même schéma d'instruction peut être utilisé selon les besoins en modifiant simplement le premier canal d'entrée ou de sortie.

Les restrictions suivantes s'appliquent lors de l'utilisation d'une fonction macro.

- Les seuls canaux à pouvoir être utilisés pour chaque exécution de macro sont les 4 canaux consécutifs commençant par le premier numéro de canal d'entrée (pour l'entrée) et les 4 canaux consécutifs commençant par le premier canal de sortie (pour la sortie).
- Les entrées et sorties spécifiées doivent bien correspondre aux canaux utilisés dans le sous-programme.
- Même lorsque le mode de sortie direct est utilisé pour les sorties, les résultats du sous-programme seront réellement reflétés dans les canaux de sortie spécifiés uniquement à la fin du sous-programme (étape 3 ci-dessus).

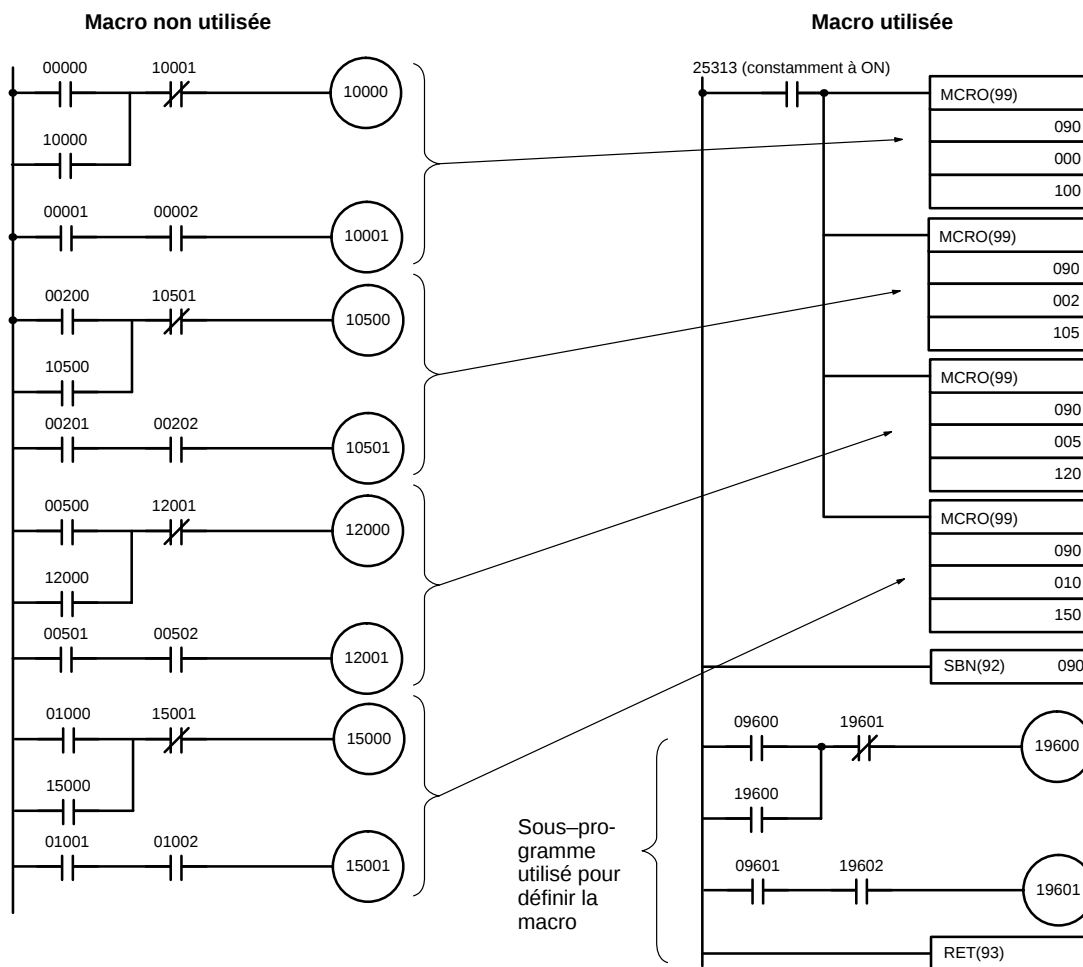
Rem. : si MCRO(99) n'est pas utilisée, les bits IR 096 à 099 et IR 196 à 199 peuvent être utilisés comme bits de travail.

Les premiers canaux d'entrée et de sortie peuvent être spécifiés non pas à l'aide des bits d'E/S, mais également avec d'autres bits (HR, bits de travail, etc.) ou à l'aide de canaux DM.

Les sous-programmes appelés par MCRO(99) sont définis par SBN(92) et RET(93), comme de simples sous-programmes.

Exemple d'application

Lorsqu'une macro est utilisée, le programme peut être simplifié comme suit :



2-4 Surveillance de changement d'état d'un bit

Le CQM1 permet la surveillance sur changement d'état de la console de programmation ou du LSS. On peut détecter la transition OFF à ON ou ON à OFF

dans un bit spécifié. Lorsque la transition spécifiée a lieu, elle est affichée et un signal sonore se met en route.

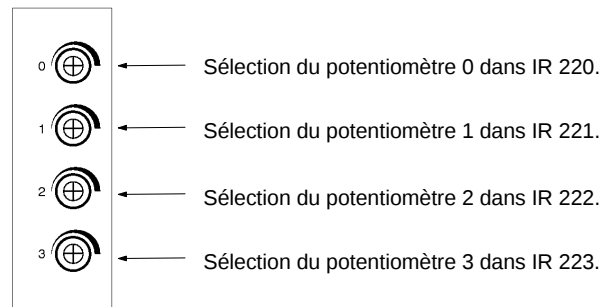
Reportez-vous au manuel d'utilisation du CQM1 pour de plus amples détails concernant la procédure de surveillance d'état d'un bit par console de programmation et au manuel d'utilisation du LSS pour la procédure LSS.

2-5 Sélections analogiques (CQM1-CPU42-E uniquement)

Avec l'UC CQM1-CPU42-E, la fonction de sélection analogique transfère automatiquement les sélections des potentiomètres de l'UC vers les IR 220 à 223. Cette fonction est très utile lorsque les valeurs sélectionnées doivent être précisément réglées en fonctionnement. Ces valeurs peuvent être modifiées en tournant simplement les potentiomètres de l'UC.

Rem. : avec des UC autres que la CQM1-CPU42-E, les IR 220 à 223 ne sont d'aucun usage spécial. Ils peuvent être utilisés comme des canaux de travail dans le programme.

Le schéma suivant illustre les potentiomètres et indique les canaux IR correspondants contenant les potentiomètres.



Les sélections sont stockées en BCD et les seuils de fonctionnement sont compris entre 0000 et 0200. Utilisez un tournevis cruciforme pour les réglages (tournez en sens horaire pour augmenter).

Attention : la CQM1-CPU42-E rafraîchit continuellement les IR 220 à 223 pour les sélections des potentiomètres aussi longtemps que l'appareil est sous tension. Ne ré-écrivez pas le contenu de ces canaux à partir du programme ou d'un périphérique.

Chapitre 3

Zones de mémoire

Ce chapitre décrit la structure des zones de mémoire du CQM1 et explique leur utilisation. Il décrit également le fonctionnement de la cartouche mémoire utilisée pour transférer les données entre le CQM1 et cette cartouche.

3-1	Fonctions de la zone mémoire	88
3-1-1	Structure de la zone mémoire	88
3-1-2	Fonctions des bits	89
3-2	Attribution des bits d'E/S	92
3-3	Stockage des DM et de l'UM dans les cartouches mémoire	93
3-3-1	Cartouches mémoire et contenu	93
3-3-2	Capacité des cartouches mémoire et taille de la zone UM	94
3-3-3	Ecriture de la cartouche mémoire	95
3-3-4	Lecture sur la cartouche mémoire	96
3-3-5	Comparaison du contenu de la cartouche mémoire	96

3-1 Fonctions de la zone mémoire

3-1-1 Structure de la zone mémoire

Zones mémoire pouvant être utilisées dans le CQM1 :

Zone de données		Taille	Canaux	Bits	Fonction
Zone IR ¹	Zone d'entrée	128 ou 192 bits	IR 000 à IR 015	IR 00000 à IR 01515	CQM1-CPU11/21-E: 8 canaux max. (7 + 1) (128 bits) peuvent être utilisés pour les bits d'E/S.
	Zone de sortie		IR 100 à IR 115	IR 10000 à IR 11515	CQM1-CPU4 11 -E: 12 canaux max. (11 + 1) (192 bits) peuvent être utilisés pour les bits d'E/S.
	Zones de travail (bits internes)	2 720 bits min. ²	IR 016 à IR 095	IR 01600 à IR 09515	Les bits de travail n'ont pas de fonction spécifique et peuvent être utilisés librement dans le programme.
			IR 116 à IR 195	IR 11600 à IR 19515	
			IR 216 à IR 219	IR 21600 à IR 21915	
			IR 224 à IR 229	IR 22400 à IR 22915	
Zone d'opé- rande- MA- CRO ¹	Zone d'entrée	64 bits	IR 096 à IR 099	IR 09600 à IR 09915	Utilisés lors de l'usage de l'instruction macro, MCRO(99). Si l'on n'utilise pas l'instruction macro, ces bits peuvent être utilisés comme bits de travail.
	Zone de sortie	64 bits	IR 196 à IR 199	IR 19600 à IR 19915	
Zone de valeur sélectionnée analogique ¹		64 bits	IR 220 à IR 223	IR 22000 à IR 22315	CQM1-CPU42-E: Utilisés pour stocker les valeurs sélectionnées analogiques (inutilisables comme bits de travail). Utilisables comme bits de travail avec d'autres UC.
Valeur en cours du compteur rapide 0 ¹		32 bits	IR 230 à IR 231	IR 23000 à IR 23115	Utilisés pour stocker les valeurs en cours du compteur rapide 0.
Valeurs en cours des sorties d'impulsion des ports 1 et 2 ¹		64 bits	IR 236 à IR 239	IR 23600 à IR 23915	CQM1-CPU43-E: Utilisé pour stocker les valeurs en cours des sorties d'impulsion des ports 1 et 2 (inutilisables comme bits de travail). CQM1-CPU44-E: Utilisé par le système (inutilisables comme bits de travail) Utilisables comme bits de travail avec d'autres UC.
Valeurs présélectionnées des compteurs rapides 1 et 2 ¹		64 bits	IR 232 à IR 235	IR 23200 à IR 23515	CQM1-CPU43/44-E: Utilisée pour stocker les valeurs en cours des compteurs rapides 1 et 2 pour les ports 1 et 2 (inutilisables comme bits de travail). Utilisables comme bits de travail avec d'autres UC.
Zones d'extension ¹		320 bits	IR 200 à IR 215 IR 240 à IR 243	IR 20000 à IR 21515 IR 24000 à IR 24315	Ces bits sont en attente d'utilisation pour des fonctions d'extension futures.
Zone SR		184 bits	SR 244 à SR 255	SR 24400 à SR 25507	Ces bits servent à des fonctions spécifiques comme les drapeaux et bits de contrôle. Utilisables comme bits de travail
Zone TR		8 bits	---	TR 0 à TR 7	Ces bits sont utilisés pour stocker temporairement l'état ON/OFF des branches du programme.

Zone de données		Taille	Canaux	Bits	Fonction
Zone HR		1 600 bits	HR 00 à HR 99	HR 0000 à HR 9915	Ces bits stockent les données et retiennent leur état ON/OFF lorsque la tension est coupée.
Zone AR		448 bits	AR 00 à AR 27	AR 0000 à AR 2715	Ces bits servent à des fonctions spécifiques comme les drapeaux et bits de contrôle.
Zone LR ¹		1 024 bits	LR 00 à LR 63	LR 0000 à LR 6315	Utilisés pour la liaison de données 1:1 par le port RS-232.
Zone de tempo./compteur ³		512 bits	TC 000 à TC 511 (n ^{os} des tempo./compteurs)		Les mêmes numéros servent aux tempo. et aux compteurs ; TC 000 à 002 servent pour les tempo. cycliques.
Zone DM	Lecture/écriture	1 024 canaux	DM 0000 à DM 1023	---	On peut accéder aux données de la zone DM par unités de mot seulement. Les valeurs des mots sont retenues lorsque l'alimentation est coupée.
		5 120 canaux	DM 1024 à DM 6143	---	Uniquement avec les UC CQM1-CPU4 
	Lecture seule ⁵	425 canaux	DM 6144 à DM 6568	---	Ré-écriture à partir du programme impossible
	Zone d'historique des erreurs ⁵	31 canaux	DM 6569 à DM 6599	---	Utilisés pour stocker l'heure de l'occurrence et de code des erreurs.
	Configuration API ⁵	56 canaux	DM 6600 à DM 6655	---	Utilisés pour stocker les divers paramètres de contrôle de l'API.
Zone de programme utilisateur (zone UM)		3 200 ou 7 200 canaux	---		Utilisés pour stocker le programme ; retenus en cas de coupure de l'alimentation. CQM1-CPU11/21-E: 3 200 canaux CQM1-CPU4  E: 7 200 canaux

Rem. : 1. Les bits IR et LR qui ne sont pas utilisés pour les fonctions qui leur sont attribuées peuvent être utilisés comme bits de travail.

2. Au moins 2 720 bits peuvent être utilisés comme bits de travail. Le nombre total de bits pouvant être utilisés dépend de la configuration API.

3. Lors de l'accès à une valeur en cours, les numéros de TC sont utilisés comme données de canaux ; lors de l'accès à des drapeaux de fin, ils sont utilisés comme données de bits.

4. Bien que les CQM1-CPU11-E et CQM1-CPU21-E ne possèdent pas de DM 1024 à 6143, il n'y a pas d'erreur dans le cas où on adresse ces mots. Toute tentative d'écriture ou de lecture donne cependant uniquement des zéros.

5. Les données des DM 6144 et 6655 ne peuvent être ré-écrites à partir du programme.

3-1-2 Fonctions des bits

Fonctions des diverses zones :

Zone IR

Les bits de la zone IR sont attribués à des bornes sur les cartes d'entrée et de sortie. Ils reflètent l'état ON ou OFF des signaux d'entrée et de sortie. Les bits d'entrée commencent à IR 00000 et ceux de sortie à IR 10000. Avec le CQM1, seuls les bits IR 00000 à 01515 peuvent être utilisés comme bits d'entrée et seuls les bits IR 10000 à 11515 peuvent être utilisés comme bits de sortie.

Pour de plus amples renseignements sur l'attribution des bits d'entrée et de sortie, reportez-vous ci-après.

Rem. : les bits d'entrée ne peuvent être utilisés dans des instructions de sortie. N'utilisez pas le même bit de sortie dans plusieurs instructions OUT et/ou OUT NOT car le programme ne pourrait pas fonctionner correctement.

Zone de bit de travail

Avec les UC CQM1-CPU11/21/41-E, n'importe quel bit situé entre IR 001 et 243 et non utilisé pour une fonction spécifique peut être utilisé comme bit de travail. Les quelques exceptions des UC CQM1-CPU42/43/44-E sont les suivantes :

UC	Bits ne pouvant être utilisés comme bits de travail
CQM1-CPU42-E	IR 22000 à IR 22315
CQM1-CPU43/44-E	IR 23200 à IR 23915

Les bits de travail peuvent être utilisés librement dans le programme, mais ils ne peuvent l'être que dans le programme et non pour les E/S externes directement. Les bits de travail sont remis à zéro (OFF) lorsque l'alimentation du CQM1 est coupée ou lorsque l'appareil entre en fonctionnement ou s'arrête.

Les bits des gammes suivantes ont des fonctions spécifiques mais peuvent être utilisés comme bits de travail si leurs fonctions spécifiques ne sont pas employées :

Gamme	Fonction
IR 001 à IR 015	Attribués à des cartes d'entrée, ces bits servent de bits d'entrée.
IR 096 à IR 099	Avec l'instruction macro, ces bits servent de bits d'entrée d'opérande.
IR 100 à IR 115	Attribués à des cartes de sortie, ces bits servent de bits de sortie.
IR 196 à IR 199	Avec l'instruction macro, ces bits servent de bits de sortie d'opérande.
IR 220 à IR 223	Avec l'UC CQM1-CPU42-E, ces bits servent à stocker les valeurs sélectionnées analogiques ; ils peuvent être utilisés comme bits de travail avec d'autres UC.
IR 230 à IR 231	Avec le compteur rapide 0, on utilise ces bits pour stocker la valeur en cours.
IR 232 à IR 235	Avec l'UC CQM1-CPU43/44-E, ces bits servent à stocker les valeurs en cours des compteurs rapides 1 et 2 ; ils peuvent être utilisés comme bits de travail avec d'autres UC.
IR 236 à IR 239	Avec l'UC CQM1-CPU43-E, ces bits servent à stocker les valeurs en cours des sorties d'impulsion des ports 1 et 2. Avec l'UC CQM1-CPU44-E, ils sont utilisés par le système ; ils peuvent être utilisés comme bits de travail avec d'autres UC.

Les IR 200 à 215 et les IR 240 à 243 seront utilisés plus tard mais peuvent être utilisés pour le moment comme bits de travail.

Les bits LR 00 à 63 sont utilisés comme bits de liaison mais peuvent également être utilisés comme bits de travail lorsqu'ils ne sont pas reliés à un autre CQM1.

Zone SR

Ces bits servent principalement de drapeaux pour les opérations du CQM1. Pour de plus amples détails sur les fonctions des bits, reportez-vous à l'annexe D de ce manuel.

Les bits SR 244 à 247 peuvent également être utilisés comme bits de travail, lorsque les interruptions d'entrée ne sont pas utilisées en mode compteur.

Zone TR

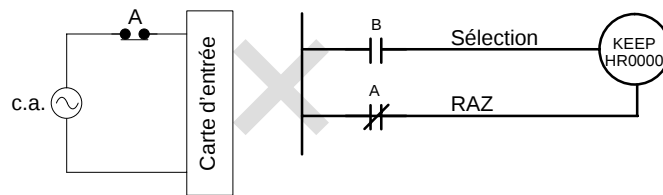
Lorsqu'un schéma à relais complexe ne peut être programmé en mnémotechnique tel quel, on utilise ces bits pour stocker temporairement les conditions d'exécution ON/OFF d'une équation logique du programme. Ils ne sont utilisés que pour les codes mnémotechniques. Lors d'une programmation directe en schéma à relais avec le LSS, les bits TR sont automatiquement traités.

On ne peut utiliser plus d'une fois les bits TR dans le même bloc d'instruction mais on peut les utiliser dans différents blocs. L'état ON/OFF des bits TR (8 TR max. 0 à 7) ne peut être surveillé à partir d'un appareil périphérique.

Zone HR

Ces bits retiennent leur état ON/OFF même après coupure de l'alimentation du CQM1 ou en début et fin fonctionnement. Ils s'utilisent de la même façon que les bits de travail.

Attention : n'utilisez jamais de bit d'entrée dans une condition NF pour la remise à zéro (RAZ) de KEEP(11) lorsque l'appareil d'entrée utilise une alimentation c.a. (cf schéma ci-dessous). Le retard à la fermeture de l'alimentation c.c. de l'API reliée à l'alimentation c.a. de l'appareil d'entrée peut provoquer une désignation du bit de KEEP(11) à la remise à zéro.



Zone AR

Ces bits servent principalement de drapeaux relatifs au fonctionnement du CQM1. Pour de plus amples détails sur les différentes fonctions de bits, reportez-vous aux chapitres correspondants dans ce manuel.

Zone LR

Lorsque deux CQM1 sont reliés point par point, ces bits sont utilisés pour le partage des données. Pour de plus amples détails, reportez-vous au chapitre des LR.

Les bits LR peuvent être utilisés comme bits de travail lorsqu'ils ne sont pas utilisés comme bits de liaison.

Zone de tempo./compteurs

Cette zone sert à la gestion des tempo./compteurs créés à l'aide de TIM, TIMH(15), CNT et CNTR(12). Les mêmes numéros sont utilisés pour les tempo./compteurs et chaque numéro peut être utilisé une seule fois dans le programme utilisateur. N'utilisez pas le même numéro de TC plusieurs fois, même pour des instructions différentes.

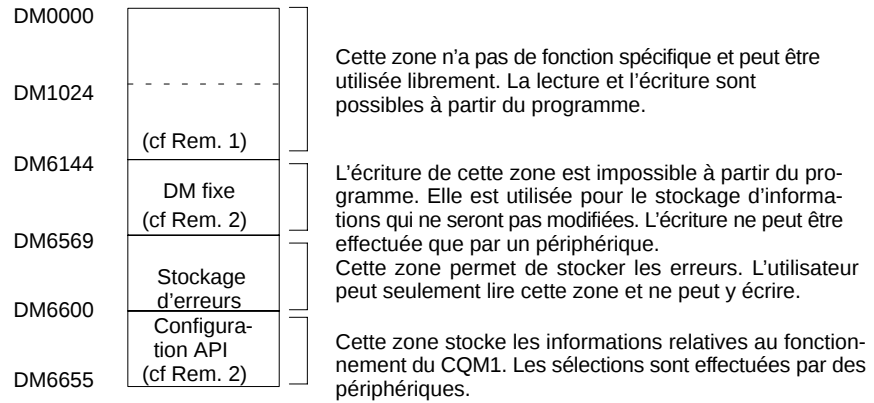
Les numéros de TC servent à la création de tempo./compteurs, à l'accès des drapeaux de fin et des valeurs en cours. Si on désigne un numéro de TC comme donnée de canal, on a alors accès à la valeur en cours ; si on l'utilise comme donnée de bit, on a accès au drapeau de fin de la tempo. ou du compteur.

La drapeau de fin passe à ON lorsque la valeur en cours de la tempo./ ou du compteur utilisée passe à 0.

- Rem. 1. Les numéros de TC 000 à 015 et les procédures d'interruption doivent être utilisées pour TIMH(15) à chaque fois que le temps de cycle dépasse 10 ms. L'utilisation d'autres numéros de TC ou le défaut d'utilisation des procédures d'interruption entraîne un fonctionnement imprécis des tempo. rapides. La procédure d'interruption peut être sélectionnée dans le DM 6629 de la configuration API.
2. Lorsque la condition d'entrée passe à OFF pour TIM ou TIMH(15), la valeur en cours est remise à zéro et retourne à la valeur présélectionnée. La valeur en cours est également remise à zéro au début de l'exécution du programme ou lorsque la condition de branchement secondaire passe à OFF dans une partie de programme secondaire (IL-ILC). La valeur en cours de CNT ou CNTR(12) n'est pas remise à zéro comme celle d'une instruction tempo. mais lorsque l'entrée de remise à zéro passe à ON.

Zone DM

On accède aux données par unités de mots. Comme sur le schéma ci-dessous, la zone DM contient une zone qui peut être librement utilisée et des zones ayant des fonctions spécifiques.



Rem. : 1. Les CQM1-CPU11-E et CQM1-CPU21-E ne possèdent pas de DM 1024 à 6143.

2. Le passage à ON de la borne 1 de l'interrupteur DIP de l'UC protège contre l'écriture, même à partir de périphériques.

On peut sauvegarder la configuration API, le programme utilisateur et le tableau d'instructions sur une cassette mémoire et les charger à partir de cette cassette tous ensemble. Reportez-vous au chapitre 3-3 pour de plus amples détails.

Attention : bien que les CQM1-CPU11-E et CQM1-CPU21-E n'aient pas de DM 1024 à 6143 il n'y aura pas d'erreur en cas d'attribution d'adresse. Toute tentative d'écriture sur ces mots n'aura aucun effet et la lecture donnera une suite de zéros.

Zone UM

La zone UM permet de stocker le programme utilisateur. On peut lire et écrire le contenu de la zone UM uniquement comme donnée de programme et non comme canal. La tableau suivant montre la taille de la zone UM dans les UC du CQM1.

UC	Taille de la zone UM
CQM1-CPU11/21-E	3,2 KW
CQM1-CPU4 [-E]	7,2 KW

3-2 Attribution des bits d'E/S

Les canaux d'E/S sont attribués par ordre à partir de la gauche, en commençant par IR 001 pour la carte d'entrée et IR100 pour la carte de sortie. Les points d'entrée de l'UC sont attribués à IR 000 (ce canal est fixe). Même en cas de montage aléatoire des cartes d'entrée et de sortie, les canaux d'entrée et de sortie affectés à 2 zones IR distinctes ne peuvent se mélanger.

On attribue un canal même pour les cartes d'E/S à 8 points. Voici l'utilisation des bits pour les cartes d'E/S à 8 points :

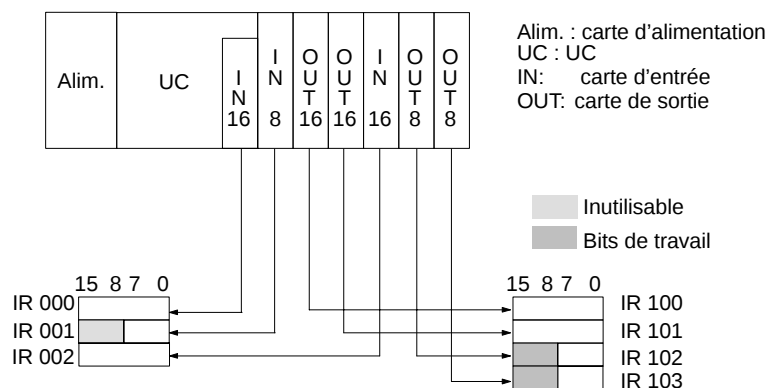
Carte	Bits 0 à 7	Bits 8 à 15
Carte d'entrée	Bits d'entrée	Constamment à OFF (0)
Carte de sortie	Bits de sortie	Bits de travail

Le nombre de bits d'E/S pouvant être attribués dépend de l'UC du CQM1 :

UC	Nombre de bits d'E/S
CQM1-CPU11/21-E	128 bits max. (8 canaux) peuvent être utilisés pour les bits d'E/S (7 + 1)
CQM1-CPU4 -E	192 bits max. (12 canaux) peuvent être utilisés pour les bits d'E/S (11 + 1)

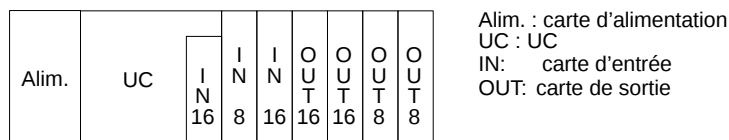
On peut attribuer jusqu'à 128 bits d'E/S (8 canaux) pour le CQM1. Une carte d'E/S à 16 points se voit attribuer un demi canal et doit représenter 16 points pour le calcul total et une carte 32 points prend 2 canaux de 16 bits.

Exemple d'attribution de canaux



Tous les bits se trouvant dans des canaux situés au-delà du dernier canal d'entrée et du dernier canal de sortie attribués peuvent être utilisés comme bits de travail.

De façon à clarifier l'attribution des canaux et à éliminer les problèmes d'interférence, il est conseillé de monter toutes les cartes d'entrée directement à la suite de l'UC. Pour l'exemple ci-dessus, voici la disposition conseillée :



Le nombre de canaux d'entrée attribués est stocké en BCD dans les bits AR 2200 à 2207 ; le nombre de canaux de sortie attribués est stocké en BCD dans les bits AR 2208 à 2215. Le CQM1 n'utilise pas de tableau d'E/S.

Attention : à chaque mise sous tension, ce canal est réactualisé.

Un programme de la table des E/S précédentes est donc nécessaire si l'on veut connaître l'ancienne configuration du CQM1 en nombre de cartes d'entrées et de sorties.

3-3 Stockage des DM et de l'UM dans les cartouches mémoire

En cas d'utilisation d'une cartouche mémoire optionnelle, la configuration API, le programme utilisateur, les DM fixes et les instructions du tableau peuvent être placées en ROM. Cela empêche les modifications intempestives. En outre, lors de modifications de process de contrôle, les sélections et programmes peuvent facilement être changés en remplaçant la cartouche mémoire.

Ce chapitre décrit la lecture, l'écriture et la comparaison des informations provenant de la cartouche mémoire et destinées à celle-ci.

3-3-1 Cartouches mémoire et contenu

Cartouches mémoire

Les six types de cartouches mémoires sont les suivants :

Référence	Remarques
CQM1-ME04K	EEPROM (sans horloge) 4 Koctets
CQM1-ME04R	EEPROM (avec horloge) 4 Koctets
CQM1-MP08K	EPROM (sans horloge) 8 Koctets
CQM1-MP08R	EPROM (avec horloge) 8 Koctets
CQM1-ME08K	EEPROM (sans horloge) 8 Koctets
CQM1-ME08R	EEPROM (avec horloge) 8 Koctets

Les EEPROM suivantes (vendues séparément) sont nécessaires pour les cartouches mémoire de type EPROM :

Référence	Version ROM	Capacité	Vitesse d'accès
ROM-ID-B	27128 ou équivalent	8 Koctets	150 ns
ROM-JD-B	27256 ou équivalent	16 Koctets	150 ns
ROM-KD-B	27512 ou équivalent	32 Koctets	150 ns

Sur les UC CQM1, on peut écrire 8 Koctets max. d'informations sur une cartouche mémoire. N'importe laquelle des EPROM citées ci-dessus possède donc la capacité nécessaire et son choix sera uniquement basé sur la facilité d'achat.

Pour de plus amples instructions sur l'utilisation des cartouches mémoires, reportez-vous au manuel d'utilisation du CQM1.

Contenu

Informations pouvant être écrites sur une cartouche mémoire :

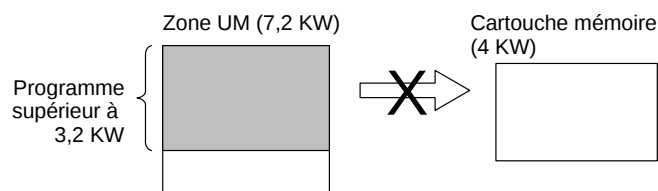
Information	Contenu
DM fixes	On ne peut écrire sur les DM fixes à partir du programme. La gamme s'étend du DM 6144 au DM 6568. Ces mots sont disponibles pour l'utilisateur.
Configuration API	La configuration API permet de sélectionner les paramètres de fonctionnement du CQM1 ; elle est stockée dans les DM 6600 à 6655.
Tableau d'instructions	Le tableau des instructions attribue des instructions d'extension aux codes de fonction de façon à permettre leur utilisation en programmation.
Mémoire de programme utilisateur (UM)	La zone UM maintient le programme utilisateur.

Les informations ci-dessus ne peuvent être lues, écrites ou comparées individuellement et doivent être traitées en groupe.

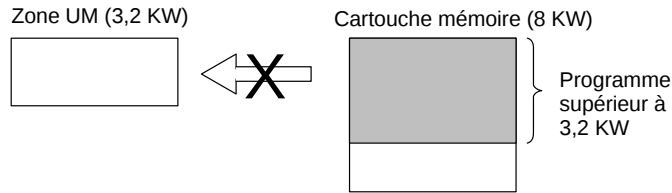
3-3-2 Capacité de la cartouche mémoire et taille de la zone UM (FAL90)

Une erreur non fatale se produit si l'on essaie de transférer un programme de taille trop importante. Il y a deux cas dans lesquels cela peut se produire :

- 1, 2, 3... 1. Lorsqu'une cartouche EEPROM de 4 KW est installée sur une UC CQM1-CPU4 ~~avec~~ avec une zone UM de 7,2 KW, les programmes de 3,2 KW peuvent être écrits sur la cartouche. Une erreur non fatale se produit si l'on essaie d'écrire un programme plus important que 3,2 KW sur la cartouche.



2. Lorsqu'une cartouche de 8 KW au moins est installée sur une UC CQM1-CPU11/21-E ayant une zone UM de 3,2 KW, on peut lire des programmes de 3,2 KW min. sur la cartouche. Une erreur non fatale se produit si l'on essaie de lire un programme supérieur à 3,2 KW à partir de la cartouche mémoire.



Rem. : les deux transferts ci-dessus seraient normalement portés à leur terme si les programmes ne dépassaient pas 3,2 KW.

Voici les tailles approximatives des programmes en zone UM et des cartouches mémoire, qui peuvent être déterminées par le contenu de AR 15 :

Emplacement du programme	Bits	Contenu	Signification
Cartouche mémoire	AR 1500 à AR 1507	00	Aucune cartouche mémoire installée ou aucune sauvegarde de programme dans la cartouche.
		04	Le programme est inférieur à 3,2 KW et ne peut être lu à partir de n'importe quelle UC CQM1.
		08	Le programme est inférieur à 7,2 KW et ne peut être lu qu'à partir des UC CQM1-CPU4 [-E.]
Zone UM	AR 1508 à AR 1515	04	Le programme est inférieur à 3,2 KW et ne peut être écrit sur n'importe quelle cartouche mémoire.
		08	Le programme est inférieur à 7,2 KW et ne peut être écrit que sur une cassette de 8 KW min.

Pour les UC CQM1-CPU11/21-E, le contenu des AR 1508 à 1515 est normalement égal à 04, ainsi que celui des AR 1500 à 1507, lorsqu'une cartouche mémoire de 4 KW a été installée.

La taille du programme indiqué dans AR 15 ne comprend pas les instructions NOP(00) se trouvant après END(01), mais toutes les autres instructions. Veillez à effacer toutes les instructions qui ne sont pas nécessaires après END(01) afin d'obtenir une évaluation précise de la taille du programme.

3-3-3 Ecriture sur la cartouche mémoire

L'écriture sur une cartouche de type EPROM s'effectue à l'aide du LSS et d'un programmeur d'EPROM. Pour de plus amples détails concernant le LSS, reportez-vous à son manuel d'utilisation.

Voici la procédure à suivre pour l'écriture :

- 1, 2, 3...**
 1. Vérifiez que l'interrupteur de protection contre l'écriture soit bien OFF sur la cartouche (c'est-à-dire désactivé). S'il est en position ON, (écriture impossible), coupez l'alimentation du CQM1 et démontez la cartouche avant de changer la position de l'interrupteur.
 2. Vérifiez que le CQM1 est en mode PROGRAM. S'il est en mode RUN ou MONITOR, utilisez le LSS pour changer de mode.
 3. Placez le bit AR 1400 à ON avec le LSS.

L'information sera écrite du CQM1 vers la cartouche mémoire. A la fin de l'opération, le bit AR 1400 passe automatiquement à OFF.

Attention : les données ne peuvent être inscrites sur la cartouche mémoire si une erreur mémoire s'est produite.

Rem. : si une erreur se produit lors de la transmission des données, une erreur non fatale (FAL 9D) est générée et le bit AR approprié (AR 1412 à 1415) passe à ON OFF. Dans ce cas, reportez-vous au chapitre 8 et effectuez les corrections nécessaires.

3-3-4 Lecture de la cartouche mémoire

Il existe deux méthodes de lecture des données de la cartouche mémoire sur le CQM1 : en utilisant un périphérique (par ex. le LSS) ou par lecture automatique du contenu au démarrage du CQM1.

Si le programme de la cartouche mémoire possède des instructions d'extension ayant des codes de fonction différents des sélections par défaut, assurez-vous que l'interrupteur DIP 4 de l'UC est à ON (indication des codes de fonction attribués par l'utilisateur).

Rem. : lorsque la donnée est lue de la cartouche mémoire sur le CQM1, l'interrupteur DIP 1 du CQM1 doit être à OFF (c'est-à-dire que l'écriture des DM doit être possible). Coupez l'alimentation du CQM1 avant de placer cette borne à OFF.

La lecture de la cartouche peut être exécutée quelle que soit le type de cartouche.

Si une erreur se produit pendant la transmission des données, une erreur non fatale (FAL 9D) est générée et le bit AR approprié (AR 1412 à AR 1415) passe à ON ou OFF (dans ce cas, reportez-vous au chapitre 8 et effectuez les corrections nécessaires).

Fonctionnement des périphériques

Procédure de lecture de la cartouche à partir d'un périphérique :

- 1, 2, 3...** 1. Vérifiez que le CQM1 est en mode PROGRAM. S'il est en mode RUN ou MONITOR, utilisez le périphérique ou LSS pour changer de mode.
2. Utilisez le périphérique ou LSS pour placer le bit AR 1401 à ON.

L'information est lue à partir de la cartouche sur le CQM1. Lorsque l'opération est terminée, AR 1401 passe automatiquement à OFF.

Lecture automatique

Si l'interrupteur DIP 2 du CQM1 est à ON (lancement automatique), les données sont automatiquement lues à partir de la cartouche mémoire à la mise sous tension du CQM1. Ce fonctionnement est impossible si une erreur survient pendant le transfert de données entre la cartouche mémoire et la mémoire du CQM1.

Attention : veillez à ce que l'alimentation soit bien coupée avant de changer les sélections des interrupteurs DIP sur le CQM1.

3-3-5 Comparaison du contenu de la cartouche mémoire

Le contenu de la cartouche mémoire peut être comparé à celui de la mémoire du CQM1 pour vérifier leur égalité ; cette opération peut être effectuée quelle que soit la cartouche. Utilisez la procédure suivante :

- 1, 2, 3...** 1. Vérifiez que le CQM1 est en mode PROGRAM. S'il est en mode RUN ou MONITOR, utilisez un périphérique pour passer en mode PROGRAM.
2. Placez AR 1402 à ON à partir du périphérique. Le contenu de la cartouche mémoire peut être comparé à celui de la mémoire du CQM1. Le bit AR 1402 passe automatiquement à OFF à la fin de la comparaison.

3. Vérifiez l'état du bit AR 1403 pour obtenir le résultat de la comparaison. Il est à ON si les contenus sont différents ou si la comparaison est impossible (si le CQM1 n'est pas en mode PROGRAM). Si le bit AR 1403 est à OFF, la comparaison a été effectuée et les contenus sont identiques.

Le bit AR 1403 ne peut être contrôlé à partir du programme ou d'un périphérique. Il ne peut être contrôlé que par le résultat de la comparaison.

Si l'on tente une comparaison avec le CQM1 dans un mode autre que le mode PROGRAM, une erreur non fatale se produit (FAL 9D) et le bit AR 1412 passe à ON. Bien que le bit AR 1403 passe également à ON, aucune comparaison n'a été effectuée. Le bit AR 1403 passe également à ON si l'on essaie d'effectuer une comparaison sans installer de cartouche mémoire sur le CQM1.

Chapitre 4

Programmation en schéma à relais

Ce chapitre décrit les étapes et concepts principaux de l'écriture en schéma à relais. Il présente les instructions qui sont utilisées pour construire la structure fondamentale du schéma à relais et pour contrôler son exécution. L'ensemble des instructions utilisées dans la programmation est décrite dans le chapitre 5.

4-1	Procédure fondamentale	100
4-2	Terminologie des instructions	100
4-3	Les bases du schéma à relais	101
4-3-1	Terminologie fondamentale	101
4-3-2	Codes mnémoniques	102
4-3-3	Instructions	104
4-3-4	OUTPUT et OUTPUT NOT	106
4-3-5	L'instruction END	106
4-3-6	Les instructions en bloc logique	107
4-3-7	Codage des instructions multiples de droite	114
4-3-8	Lignes secondaires	115
4-3-9	Sauts	119
4-4	Contrôle de l'état des bits	120
4-4-1	SET et RESET	121
4-4-2	DIFFERENTIATE UP et DIFFERENTIATE DOWN	121
4-4-3	KEEP	121
4-4-4	Bits de maintien automatique ("Seal")	122
4-5	Bits de travail (relais internes)	122
4-6	Précautions de programmation	125
4-7	Exécution du programme	126

4-1 Procédure fondamentale

Dans l'écriture d'un programme, il y a plusieurs étapes différentes :

- 1, 2, 3...** 1. Dressez la liste de tous les appareils d'E/S et des points d'E/S qui leur ont été attribués ; préparez un tableau des bits d'E/S attribués à chaque appareil d'E/S.
2. Si vous utilisez des bits LR pour relier deux API, préparez des feuilles détaillant l'utilisation de chaque bit.
3. Déterminez quels sont les canaux disponibles pour les bits de travail et préparez un tableau d'attribution de ces canaux selon leur utilisation.
4. Préparez également des tableaux comportant les numéros de TC et de sauts de façon à pouvoir les attribuer au fur et à mesure. Attention, la fonction d'un numéro de TC ne peut être définie qu'à l'intérieur du programme et les numéros de saut 01 à 99 ne peuvent être utilisés qu'une seule fois chacun (les numéros de TC sont décrits dans le chapitre 5 et les numéros de saut dans ce chapitre).
5. Dessinez le schéma à relais.
6. Entrez le programme dans l'UC. L'utilisation d'une console de programmation implique la conversion du programme en mnémonique.
7. Corrigez les éventuelles erreurs de syntaxe du programme.
8. Exécutez le programme pour corriger les éventuelles erreurs d'exécution.
9. Après installation du système de contrôle complet et lorsque celui-ci est prêt à l'usage, exécutez le programme et effectuez les réglages qui s'imposent.

Les lois fondamentales de la programmation en schéma à relais et de la codification en mnémonique sont décrites dans le chapitre 4–3. La préparation et l'entrée du programme par la console de programmation sont décrites dans le manuel d'utilisation du CQM1 ; pour l'entrée et la programmation par le LSS, reportez-vous au manuel d'utilisation du LSS. La suite du chapitre 4 traite de la programmation avancée, des précautions de programmation et de l'exécution du programme. Toutes les instructions spéciales sont traitées dans le chapitre 5. La mise au point est décrite dans le manuel d'utilisation CQM1 et dans le manuel LSS. Le chapitre 8 fournit également quelques informations concernant la mise au point.

4-2 Terminologie des instructions

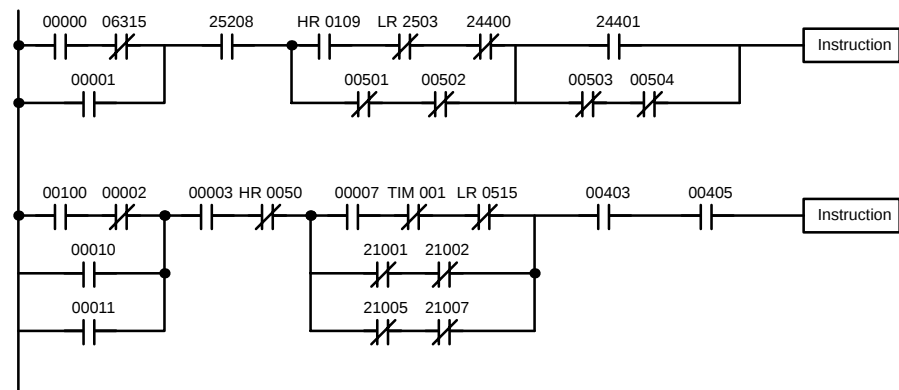
Il existe deux types d'instructions en schéma à relais : les instructions qui correspondent à des conditions, utilisées sous la forme d'instructions uniquement lors de la conversion en mnémonique et les instructions utilisées sur la partie de droite du schéma, exécutées en fonction des conditions se trouvant sur les lignes d'instruction.

La plupart des instructions possèdent au moins une opérande associée. Les opérandes fournissent des informations concernant l'exécution d'une instruction. Elles sont parfois entrées en valeurs numériques réelles mais correspondent généralement à des adresses de canaux de zones ou à des bits contenant des données à utiliser. Par exemple, une instruction MOVE ayant IR 000 comme opérande de source désignée déplace le contenu de IR 000. L'emplacement de destination est désigné comme opérande également. Un bit dont l'adresse est désignée comme opérande est appelé bit d'opérande. un canal dont l'adresse est désignée comme opérande est appelé canal d'opérande. Si la valeur réelle est entrée comme constante, elle est précédée de "#" pour indiquer que ce n'est pas une adresse.

D'autres termes relatifs à la description des instructions sont explicités dans le chapitre 5.

4-3 Les bases du schéma à relais

Un schéma à relais comporte une ligne et des embranchements. La ligne de gauche est la ligne de bus et les lignes d'embranchement sont des lignes d'instruction ou lignes secondaires. Le long des lignes d'instruction, on trouve des conditions, menant à d'autres instructions situées sur la droite. Les combinaisons logiques de ces conditions déterminent l'exécution des instructions de droite. Exemple de schéma à relais :



Comme on le voit, les lignes d'instructions peuvent avoir des embranchements secondaires et se rejoindre. Les lignes verticales sont des conditions. Les conditions dépourvues de lignes diagonales sont "normalement ouvertes" et correspondent à des instructions LOAD, AND et OR. Les conditions comportant une diagonale sont "normalement fermées" et correspondent à des instructions LOAD NOT, AND NOT ou OR NOT. Les numéros se trouvant au-dessus des conditions indiquent le bit d'opérande de l'instruction ; l'état de ce bit détermine la condition d'exécution des instructions suivantes. Le fonctionnement de chaque instruction selon les conditions d'exécution est décrit ci-après.

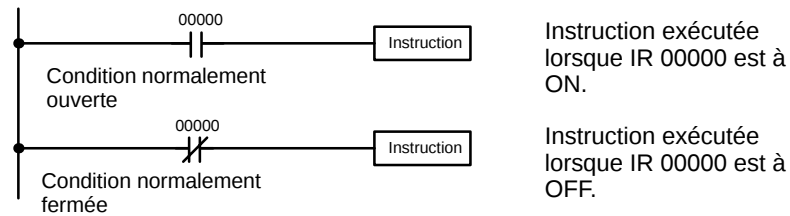
Rem. En cas d'affichage des schémas à relais par le LSS, une deuxième ligne de bus apparaît à droite, connectée à toutes les instructions de droite. Cela n'apporte pas de modification fonctionnelle au schéma à relais ; aucune condition ne doit être placée entre les instructions de droite et la ligne de bus de droite ; elles doivent toutes être connectées directement à la ligne de bus de droite. Reportez-vous au manuel LSS pour de plus amples détails.

4-3-1 Terminologie fondamentale

Conditions normalement ouvertes et normalement fermées

Une condition de schéma à relais est en position ON ou OFF, selon l'état du bit d'opérande qui lui a été attribué. Une condition normalement ouverte est à ON si le bit d'opérande est à ON ; elle est à OFF si le bit d'opérande est à OFF. Une condition normalement fermée est à ON si le bit d'opérande est à OFF ; elle est à OFF si le bit d'opérande est à ON. Généralement, on utilise une condition normalement ouverte lorsque l'on veut qu'un événement se

produise lorsque le bit est à ON et une condition normalement fermée pour qu'un événement se produise lorsque le bit est à OFF.



Conditions d'exécution

En programmation en schéma à relais, la combinaison logique des conditions ON et OFF avant une instruction détermine une condition composée selon laquelle une instruction est exécutée. Cette condition ON ou OFF est appelée condition d'exécution pour l'instruction. Toutes les instructions autres que LOAD ont des conditions d'exécution.

Bits d'opérande

Les opérands désignées pour toutes les instructions en schéma à relais peuvent correspondre à n'importe quel bit des zones IR, SR, HR, AR, LR ou TC. Les conditions en schéma à relais peuvent donc être déterminées par des bits d'E/S, des drapeaux, des bits de travail, des T/C, etc. Les instructions LOAD et OUTPUT peuvent également utiliser les bits des zones TR, mais uniquement lors d'applications spéciales. Reportez-vous au chapitre 4-3-8 pour de plus amples détails.

Blocs logiques

Les correspondances entre instructions et conditions sont déterminées par le lien existant entre les conditions des lignes d'instruction qui les relient. Tout groupe d'instructions créant un résultat logique s'appelle bloc logique. Bien que les schémas à relais puissent être écrits sans véritable analyse individuelle des blocs logiques, la compréhension des blocs est nécessaire pour une programmation efficace et devient essentielle lorsque les programmes doivent être entrés en code mnémotechnique.

Bloc d'instruction

Un bloc d'instructions comprend toutes les instructions reliées entre elles dans le schéma à relais. Un bloc d'instructions comprend donc toutes les instructions entre lesquelles on peut dessiner une ligne horizontale sans rencontrer de ligne verticale dans le schéma à relais et ce jusqu'à l'emplacement suivant où l'on peut à nouveau dessiner le même genre de ligne.

4-3-2 Codes mnémoniques

Le schéma à relais ne peut pas être directement entré dans un API par la console de programmation car le LSS est nécessaire. Pour l'entrer par la console de programmation, il est nécessaire de convertir le schéma à relais en mnémotechnique. Le code mnémotechnique fournit exactement les mêmes informations que le schéma à relais mais sous une forme qui peut être directement entrée dans l'API. On peut programmer directement en code mnémotechnique mais cela n'est pas conseillé aux débutants, ni dans les cas de programmes complexes. Quel que soit l'appareil de programmation, le programme est stocké en mémoire sous la forme mnémotechnique, d'où l'importance de ce code.

Nous proposons une introduction au code mnémotechnique, important pour la compréhension des programmes et pour l'utilisation de la console de programmation. Vous n'aurez pas obligatoirement à utiliser ce code pour la programmation par LSS, mais seulement si vous le souhaitez.

Structure de la mémoire programme

Le programme est entré dans la mémoire du programme sous la forme d'adresses. Les adresses de la mémoire du programme sont légèrement différentes de celles des autres zones de mémoire parce qu'elles ne retiennent pas obligatoirement la même quantité de données. Chaque adresse contient une instruction et toutes les données d'opérande et opérands (cf ci-après) nécessaires à cette instruction. Certaines instructions ne nécessitant pas d'opérands et d'autres jusqu'à 3 opérands, les adresses de mémoire programme peuvent posséder 1 à 4 canaux.

Les adresses de la mémoire programme commencent à 00000 et vont jusqu'à épuisement de la capacité. Le premier canal de chaque adresse définit l'instruction. Toutes les données d'opérands utilisées par l'instruction sont également contenues dans le premier canal. Si une instruction ne demande qu'une seule opérande de bit (sans donnée d'opérande), l'opérande se programme sur la même ligne que l'instruction. Le reste des canaux nécessaires à l'instruction contiennent les opérands spécifiant les données à utiliser. Lors de la conversion en mnémonique, toutes les instructions autres que celles en schéma à relais sont inscrites sous la forme identique d'un canal par ligne, selon leur apparition dans le schéma. Voici un exemple de code mnémonique ; les instructions utilisées sont décrites plus tard :

Adresse	Instruction	Opérande
00000	LD	HR 0001
00001	AND	00001
00002	OR	00002
00003	LD NOT	00100
00004	AND	00101
00005	AND LD	00102
00006	MOV(21)	
		000
		DM 0000
00007	CMP(20)	
		DM 0000
		HR 00
00008	LD	25505
00009	OUT	10000
00010	MOV(21)	
		DM 0000
		DM 0500
00011	DIFU(13)	00502
00012	AND	00005
00013	OUT	10003

Les colonnes d'adresse et d'instruction sont remplies pour le canal d'instruction seulement. Pour les autres lignes, les deux colonnes de gauche sont laissées vides. Si l'instruction ne nécessite pas de donnée d'opérande ni d'opérande de bit, la colonne d'opérande est vierge pour la première ligne. Vous pouvez mettre une croix dans les cases vides des canaux d'instructions ne nécessitant pas de données, de façon à vérifier d'un coup d'oeil qu'aucune adresse n'a été oubliée.

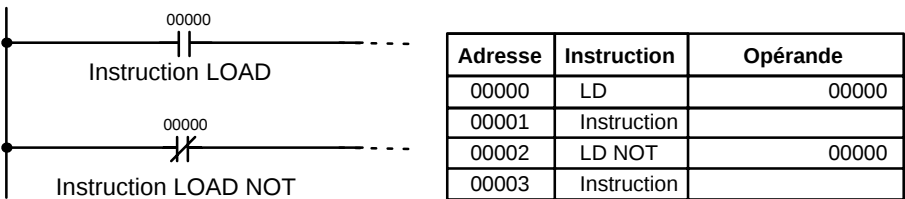
Lors de la programmation, les adresses sont automatiquement affichées et ne doivent pas nécessairement être entrées, sauf si l'on désire déplacer l'instruction. Pendant la conversion en code mnémonique, il vaut mieux commencer à l'adresse 00000 de la mémoire programme, mais ce n'est pas impératif.

4-3-3 Instructions

Les instructions des schémas à relais correspondent aux conditions ; qu'elles soient indépendantes ou combinées avec le bloc logique, elles forment les conditions sur lesquelles toutes les autres instructions sont fondées.

LOAD et LOAD NOT

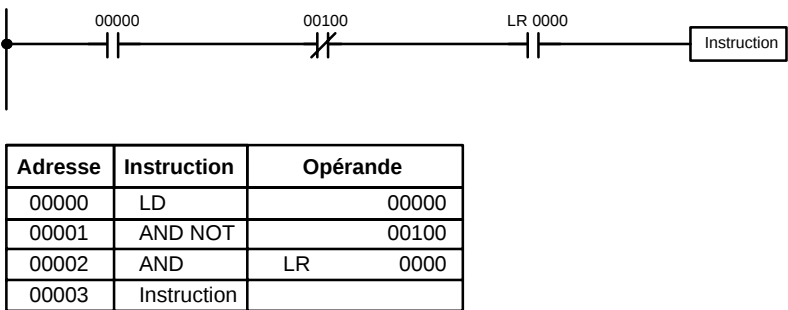
La première condition d'un bloc logique dans un schéma à relais correspond à une instruction LOAD ou LOAD NOT. Chacune de ces instructions demande une ligne de code mnémonique. Dans les schéma suivants, on appelle "instruction" une instruction servant d'exemple et qui pourrait être n'importe laquelle des instructions figurant à droite du schéma.



Lorsque cette condition est seule sur une ligne, la condition d'exécution de l'instruction de droite est à ON lorsque la condition est à ON. Pour l'instruction LOAD (condition normalement ouverte), la condition d'exécution doit être à ON lorsque IR 00000 est à ON ; pour l'instruction LOAD NOT (normalement fermée), elle doit être à ON lorsque 00000 est à OFF.

AND et AND NOT

Quand plusieurs conditions sont en série sur la même ligne, la première correspond à une instruction LOAD ou LOAD NOT et les autres aux instructions AND ou AND NOT. L'exemple suivant montre trois conditions correspondant de gauche à droite à des instructions LOAD, AND NOT et AND. Chacune de ces instructions demande une ligne de code mnémonique.



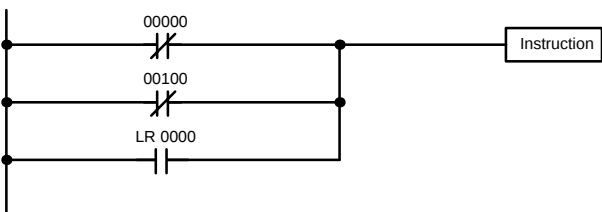
L'instruction ne peut avoir une condition ON que lorsque les trois conditions sont à ON, c'est-à-dire quand IR 00000 est à ON, IR 00100 à OFF et LR 0000 à ON.

Les instructions AND en série peuvent être considérées individuellement, chacune prenant en compte le AND logique de la condition d'exécution précédente (c'est-à-dire l'ensemble des conditions jusqu'à ce point) et l'état du bit d'opérande de l'instruction AND. Si ces deux états sont à ON, une condition d'exécution ON est produite pour l'instruction suivante. Si l'un des deux états est à OFF, le résultat est OFF. La condition d'exécution de la première instruction AND d'une série correspond à la première condition de la ligne d'instruction.

Chaque instruction AND NOT d'une série prend en compte le AND logique situé entre sa condition et l'inverse de son bit d'opérande.

OR et OR NOT

Lorsque plusieurs conditions sont présentes sur des lignes d'instruction séparées mais parallèles et se rejoignent, la première correspond à une instruction LOAD ou LOAD NOT ; le reste des conditions correspond à une instruction OR ou OR NOT. L'exemple suivant montre trois conditions correspondant de haut en bas à des instructions LOAD NOT, OR NOT et OR. Chacune de ces instructions demande une ligne mnémonique.



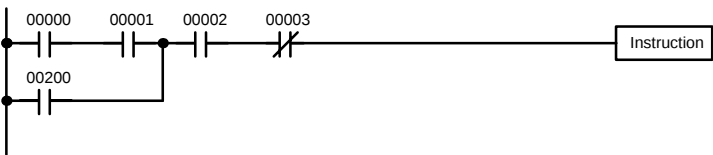
Adresse	Instruction	Opérande
00000	LD NOT	00000
00001	OR NOT	00100
00002	OR	LR 0000
00003	Instruction	

La condition de l'instruction est à ON lorsque l'une des trois conditions est à ON, c'est-à-dire lorsque IR 00000 est à OFF, IR 00100 à OFF, ou LR 0000 à ON.

Les instructions OR et OR NOT peuvent être considérées individuellement, chacune assumant le OR logique entre sa condition d'exécution et l'état du bit d'opérande de l'instruction OR. Si l'un de ces états est à ON, une condition d'exécution ON est produite pour l'instruction suivante.

Combinaison d'instructions AND et OR

Lorsque des instructions AND et OR sont combinées dans des schémas plus complexes, elles peuvent parfois être considérées individuellement, chaque instruction effectuant une opération logique sur la condition et sur l'état du bit d'opérande. Voici un exemple que vous pouvez étudier afin de bien saisir que le code mnémonique suit la même logique que le schéma à relais.



Adresse	Instruction	Opérande
00000	LD	00000
00001	AND	00001
00002	OR	00200
00003	AND	00002
00004	AND NOT	00003
00005	Instruction	

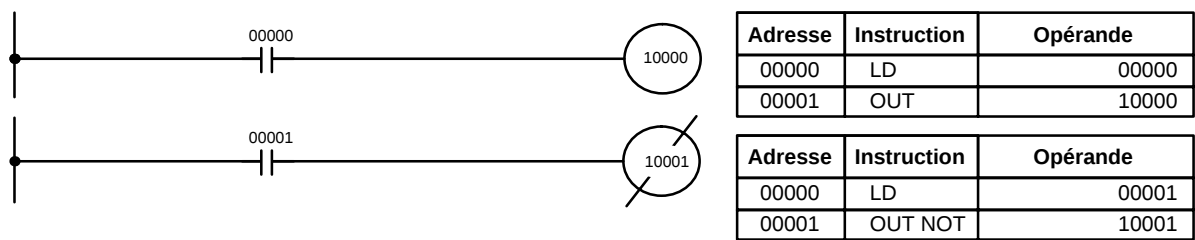
On prend ici un AND situé entre l'état de IR 00000 et celui de IR 00001 pour déterminer la condition d'un OR logique avec l'état de IR 00200. Le résultat de l'opération permet de déterminer la condition d'exécution d'un AND logique avec l'état de IR 00002, déterminant à son tour la condition d'un AND avec l'inverse (AND NOT) de l'état de IR 00003.

Dans les schémas plus compliqués, il est toutefois nécessaire de considérer les blocs logiques avant de pouvoir déterminer la condition d'exécution de

l'instruction finale ; on utilise AND LOAD et OR LOAD. Avant d'aller plus loin, voici les instructions nécessaires pour obtenir un simple programme d'entrée/sortie.

4-3-4 **OUTPUT et OUTPUT NOT**

La façon la plus simple de sortir les résultats d'une combinaison de conditions est de le faire directement avec OUTPUT et OUTPUT NOT. Ces instructions s'utilisent pour contrôler l'état du bit d'opérande désigné en fonction de la condition d'exécution. Avec l'instruction OUTPUT, le bit d'opérande passe respectivement à ON (ou OFF) aussi longtemps que la condition est à ON (ou OFF). Les voici en code mnémonique, chaque instructions nécessitant une ligne :

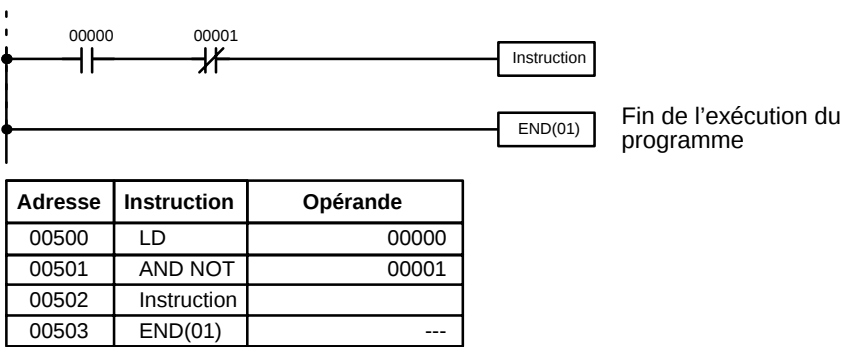


Dans l'exemple ci-dessus, IR 10000 est à ON aussi longtemps que IR 00000 est à ON et IR 10001 est à OFF aussi longtemps que IR 00001 est à ON. Ici, IR 00000 et IR 00001 sont les bits d'entrée et IR 10000 et IR 10001 les bits de sortie attribués aux cartes contrôlées par l'API, c'est-à-dire que les signaux entrant par les points d'entrées IR 00000 et IR 00001 contrôlent les points de sortie IR 10000 et IR 10001.

Le temps pendant lequel un bit reste à ON ou OFF peut être contrôlé par les instructions OUTPUT ou OUTPUT NOT à l'aide des instructions de temporisation (cf chapitre TIM).

4-3-5 **L'instruction END**

La dernière instruction d'un programme simple est END. Lorsque l'UC scrute le programme, elle exécute toutes les instructions jusqu'à la première instruction END et retourne au début du programme pour recommencer l'exécution. Bien qu'une instruction END puisse être placée à n'importe quel endroit du programme, ce qui se fait parfois lors de la mise au point, aucune instruction située au-delà de END ne sera exécutée. Le nombre qui suit l'instruction END en code mnémonique est son code de fonction, utilisé lors de l'entrée de la plupart des instructions dans l'API (description ci-après). L'instruction END n'a besoin d'aucune opérande et aucune condition ne peut être placée sur la même ligne.



En l'absence d'instruction END dans le programme, celui-ci n'est pas exécuté.

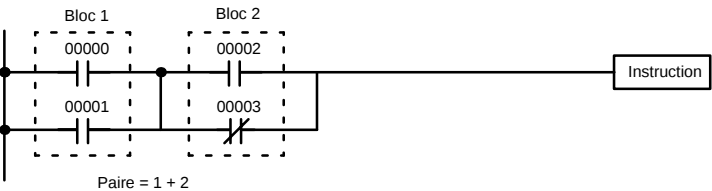
Vous avez maintenant toutes les instructions nécessaires à l'écriture d'un simple programme d'entrée/sortie. Avant d'en finir avec les bases du schéma à relais et de voir l'entrée du programme dans l'API, regardons les instructions en bloc logique (AND LOAD et OR LOAD), qui sont parfois nécessaires, même dans des schémas simples.

4-3-6 Les instructions en bloc logique

Les instructions en bloc logique ne correspondent à aucune condition spécifique sur le schéma à relais mais décrivent des relations existant entre blocs logiques. AND LOAD établit un AND logique entre les conditions d'exécution produites par deux blocs logiques ; OR LOAD établit un OR logique entre les conditions produites par deux blocs logiques.

AND LOAD

Bien que simple en apparence, le schéma ci-dessous nécessite une instruction AND LOAD :



Adresse	Instruction	Opérande
00000	LD	00000
00001	OR	00001
00002	LD	00002
00003	OR NOT	00003
00004	AND LD	---

Les deux blocs logiques apparaissent en pointillés. Cet exemple montre qu'une condition d'exécution ON est produite lorsque l'une des conditions situées dans le bloc de gauche est à ON (IR 00000 ou IR 00001 est à ON) et lorsque l'une des conditions du bloc de droite est à ON (IR 00002 est à ON ou IR 00003 est à OFF).

Le schéma à relais ci-dessus ne peut cependant pas être converti en code mnémonique en utilisant seulement AND et OR. Si l'on essaie d'effectuer un AND entre IR 00002 et les résultats d'un OR entre IR 00000 et IR 00001, le OR NOT entre IR 00002 et IR 00003 est perdu et le OR NOT ne se fait plus qu'entre IR 00003 et le résultat d'un AND entre IR 00002 et le premier OR. Il faut trouver une façon d'effectuer le OR (NOT) indépendamment et de combiner les résultats.

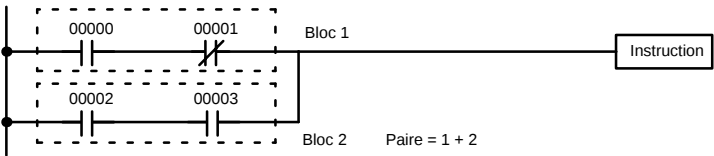
Pour y parvenir, on peut utiliser LOAD ou LOAD NOT au milieu d'une ligne d'instruction. Lorsque l'on exécute LOAD ou LOAD NOT ainsi, la condition d'exécution est sauvegardée dans des mémoires tampons spéciales et le processus logique recommence. Pour combiner les résultats de la condition d'exécution en cours et d'une autre condition précédente et "non utilisée", on utilise un AND LOAD ou un OR LOAD. "LOAD" renvoie au chargement de la dernière condition d'exécution non utilisée. Une condition d'exécution non utilisée est produite par l'utilisation de LOAD ou LOAD NOT pour les instruction d'une ligne, autres que la première instruction.

L'analyse du schéma ci-dessus en termes d'instructions mnémoniques révèle que la condition de IR 00000 est une instruction LOAD et que la

condition située au-dessous est un OR entre l'état de IR 00000 et celui de IR 00001. La condition de IR 00002 est une nouvelle instruction LOAD et la condition située au-dessous et un OR NOT (un OR entre l'état de IR 00002 et l'état inverse de IR 00003). Pour parvenir à la condition d'exécution de l'instruction de droite, il faut prendre le AND logique des conditions d'exécution résultant de celles de ces deux blocs, avec AND LOAD. On trouvera ci-dessous les codes mnémoniques correspondant au schéma. L'instruction AND LOAD ne nécessite aucune opérande particulière parce qu'elle fonctionne sur des conditions préalablement déterminées. Les pointillés indiquent que les opérandes n'ont pas besoin de désignation ni d'entrée. AND LOAD fonctionne par paire de blocs logiques.

OR LOAD

Le schéma suivant nécessite une instruction OR LOAD entre le bloc logique du haut et celui du bas. Une condition d'exécution ON est produite pour l'instruction de droite lorsque IR 00000 est à ON et IR 00001 à OFF ou lorsque IR 00002 et IR 00003 sont à ON. Le fonctionnement et le code mnémonique de OR LOAD sont exactement identiques à ceux de AND LOAD, à ceci près que la condition d'exécution en cours est reliée par OR avec la dernière condition non utilisée.



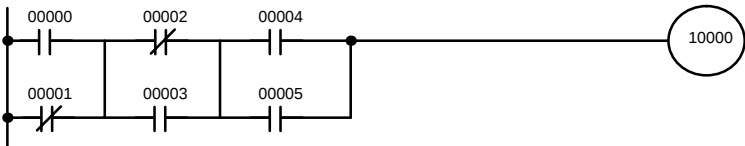
Adresse	Instruction	Opérande
00000	LD	00000
00001	AND NOT	00001
00002	LD	00002
00003	AND	00003
00004	OR LD	---

Certains schémas nécessitent des instructions AND LOAD et des instructions OR LOAD. OR LOAD fonctionne par paire de blocs logiques.

Instructions en blocs logiques en série

Pour le codage des schémas en instructions à blocs logiques en série, il faut diviser le schéma en blocs logiques. Chaque bloc est codé en utilisant une instruction LOAD pour coder la première condition ; AND LOAD ou OR LOAD s'utilisent pour effectuer des combinaisons logiques de blocs. Avec AND LOAD et OR LOAD, il y a deux façons d'y parvenir : on peut coder l'instruction en bloc logique après les deux premiers blocs et ensuite chaque bloc supplémentaire ; on peut également coder tous les blocs à combiner, en commençant chaque bloc par LOAD ou LOAD NOT et coder ensuite les instructions en bloc logique qui les combinent. Dans ce dernier cas, les instructions du dernier couple de blocs doivent être combinées en premier, puis chaque bloc précédent, remontant ainsi jusqu'au premier bloc. Ces méthodes produisent strictement les mêmes résultats mais la deuxième (codage des instructions en bloc) ne peut être utilisée que jusqu'à 8 instructions en blocs combinées (c'est-à-dire jusqu'à 7 instructions en bloc logiques).

Le schéma suivant nécessite la conversion de AND LOAD en mnémonique car trois paires de conditions parallèles se trouvent en série. Voici les deux méthodes de codage du programme :



Adresse	Instruction	Opérande	Adresse	Instruction	Opérande
00000	LD	00000	00000	LD	00000
00001	OR NOT	00001	00001	OR NOT	00001
00002	LD NOT	00002	00002	LD NOT	00002
00003	OR	00003	00003	OR	00003
00004	AND LD	—	00004	LD	00004
00005	LD	00004	00005	OR	00005
00006	OR	00005	00006	AND LD	—
00007	AND LD	—	00007	AND LD	—
00008	OUT	10000	00008	OUT	10000

La méthode de droite ne permet de combiner que 8 blocs. La première méthode n'admet aucune limite quant au nombre de blocs à combiner.

Le schéma suivant nécessite la conversion d'instructions OR LOAD en mnémonique car trois paires de conditions en série se trouvent en parallèle.



La première de chaque paire de conditions est convertie en LOAD avec l'opérande de bit attribuée, puis reliée par un AND avec l'autre condition. Les deux premiers blocs peuvent être codés en premier, suivis de OR LOAD, du dernier bloc et d'une nouvelle instruction OR LOAD. Les trois blocs peuvent également être codés en premier, suivis de deux OR LOAD. Voici les codes mnémoniques des deux méthodes :

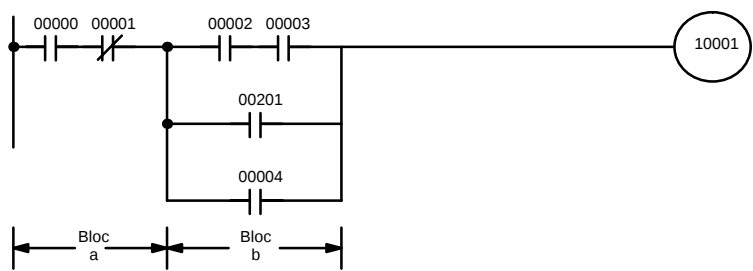
Adresse	Instruction	Opérande	Adresse	Instruction	Opérande
00000	LD	00000	00000	LD	00000
00001	AND NOT	00001	00001	AND NOT	00001
00002	LD NOT	00002	00002	LD NOT	00002
00003	AND NOT	00003	00003	AND NOT	00003
00004	OR LD	—	00004	LD	00004
00005	LD	00004	00005	AND	00005
00006	AND	00005	00006	OR LD	—
00007	OR LD	—	00007	OR LD	—
00008	OUT	10001	00008	OUT	10001

La méthode de droite ne permet de combiner que 8 blocs. La première méthode n'admet aucune limite quant au nombre de blocs à combiner.

Combinaison de AND LOAD et OR LOAD

Les deux méthodes de codage décrites ci-dessus peuvent également être utilisées lors de l'utilisation de AND LOAD et OR LOAD, aussi longtemps que le nombre de blocs combinés ne dépasse pas huit.

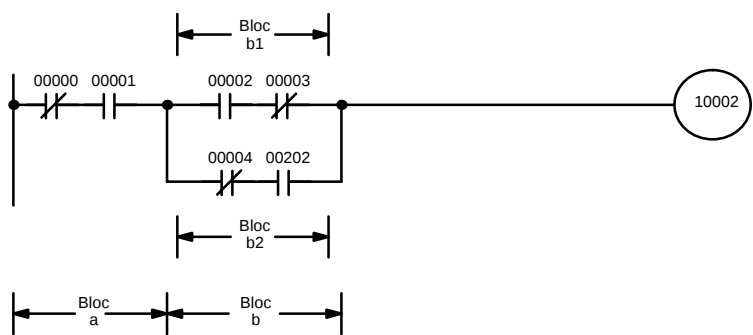
Le schéma suivant ne contient que deux blocs logiques. Il n'est pas nécessaire de séparer à nouveau les composants du bloc b car il peut être codé directement en utilisant seulement AND et OR.



Adresse	Instruction	Opérande
00000	LD	00000
00001	AND NOT	00001
00002	LD	00002
00003	AND	00003
00004	OR	00201
00005	OR	00004
00006	AND LD	—
00007	OUT	10001

Bien que le schéma suivant soit similaire au premier, le bloc b ci-dessous ne peut être codé sans qu'il soit séparé en deux blocs combinés par OR LOAD. Dans cet exemple, les trois blocs ont été codés en premier et OR LOAD a ensuite été utilisée pour combiner les deux derniers blocs, puis AND LOAD pour combiner la condition d'exécution produite par OR LOAD avec la condition d'exécution du bloc a.

Lors du codage d'instructions en blocs logiques à la fin des blocs combinés, l'opération doit être effectuée à l'envers : l'instruction en bloc logique des deux derniers blocs est codée en premier, puis celle destinée à combiner la condition d'exécution résultant de la première instruction en bloc logique, puis la condition du troisième bloc à partir de la fin, en remontant jusqu'au premier bloc combiné.



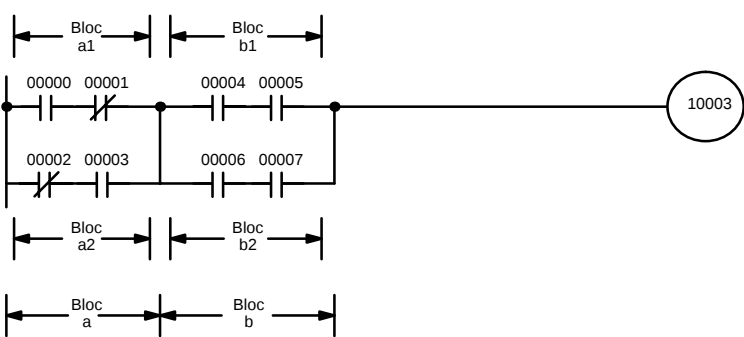
Adresse	Instruction	Opérande
00000	LD NOT	00000
00001	AND	00001
00002	LD	00002
00003	AND NOT	00003
00004	LD NOT	00004
00005	AND	00202
00006	OR LD	—
00007	AND LD	—
00008	OUT	10002

Schémas complexes

Lorsque l'on détermine les instructions logiques nécessaires au codage d'un schéma, il faut parfois diviser ce dernier en grands blocs puis diviser à leur tour ces blocs pour arriver à des blocs logiques pouvant être codés sans former d'instructions en blocs logiques. Ces blocs sont ensuite codés en combinant les petit blocs d'abord puis les blocs plus importants. AND LOAD ou OR LOAD est utilisée pour la combinaison des blocs : elles combinent toujours les deux dernières conditions d'exécution, qu'elles résultent d'une seule condition, de blocs logiques ou d'instructions à blocs logiques antérieures.

Dans les schémas complexes, les blocs sont codés en partant de la partie supérieure gauche puis en descendant. Quand on a le choix, on code donc OR LOAD avant AND LOAD.

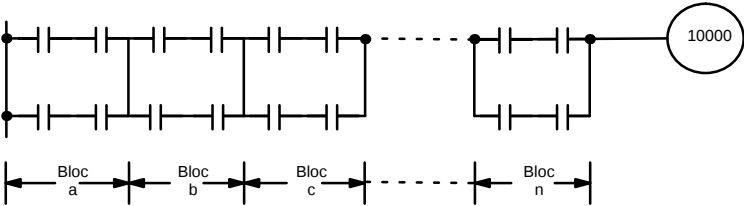
Le schéma suivant doit être séparé en deux blocs et chaque bloc en deux autres blocs avant le codage. Comme on le voit, les blocs a et b nécessitent une instruction AND LOAD. Avant AND LOAD, OR LOAD doit cependant être utilisée pour combiner les blocs du haut et du bas des deux côtés c'est-à-dire pour combiner a1 et a2 ainsi que b1 et b2.



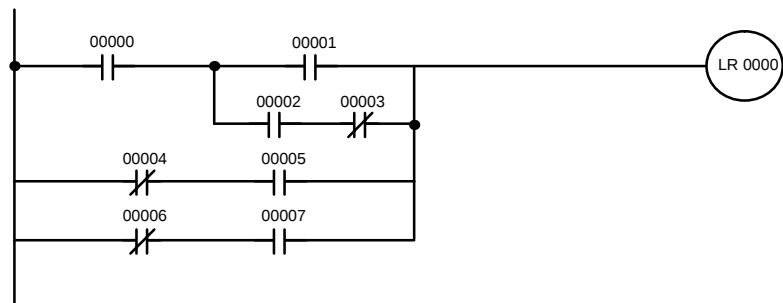
Adresse	Instruction	Opérande	
00000	LD	00000	
00001	AND NOT	00001	
00002	LD NOT	00002	
00003	AND	00003	
00004	OR LD	—	Blocs a1 et a2
00005	LD	00004	
00006	AND	00005	
00007	LD	00006	
00008	AND	00007	
00009	OR LD	—	Blocs b1 et b2
00010	AND LD	—	
00011	OUT	10003	Blocs a et b

Les schémas du type suivant peuvent facilement être codés si chaque bloc est codé selon l'ordre suivant : de haut en bas d'abord puis de gauche à droite. Dans le schéma suivant, les blocs a et b doivent être combinés par AND LOAD comme ci-dessus, puis le bloc c doit être codé par la suite ; on utilise ensuite une deuxième fois AND LOAD pour le combiner avec la condition d'exécution du premier AND LOAD. Le bloc d est codé ensuite, et on utilise alors un troisième AND LOAD pour combiner la condition d'exécution

du bloc d avec celle du deuxième AND LOAD, et ainsi de suite jusqu'au bloc n.

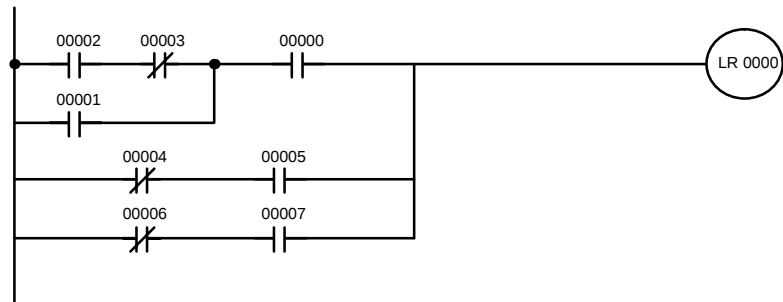


Le schéma suivant nécessite un OR LOAD puis un AND LOAD pour coder le haut des trois blocs, puis deux autres OR LOAD pour terminer le codage mnémonique.



Adresse	Instruction	Opérande
00000	LD	00000
00001	LD	00001
00002	LD	00002
00003	AND NOT	00003
00004	OR LD	--
00005	AND LD	--
00006	LD NOT	00004
00007	AND	00005
00008	OR LD	--
00009	LD NOT	00006
00010	AND	00007
00011	OR LD	--
00012	OUT	LR 0000

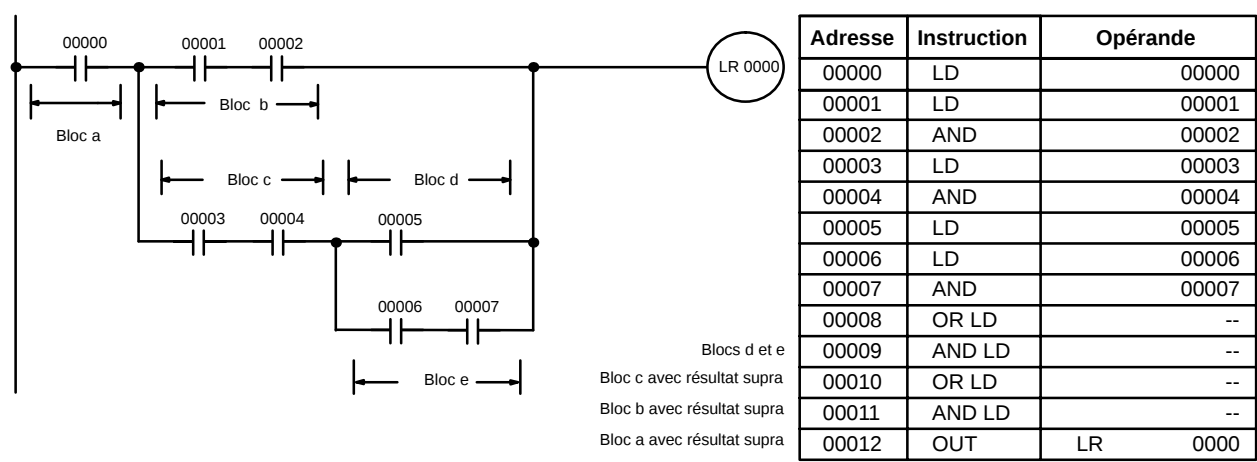
Bien que le programme soit exécuté selon ce qui est écrit, ce schéma pourrait être dessiné comme suit pour éviter le premier OR LOAD et le AND LOAD, ce qui simplifie le programme et permet de gagner de l'espace en mémoire.



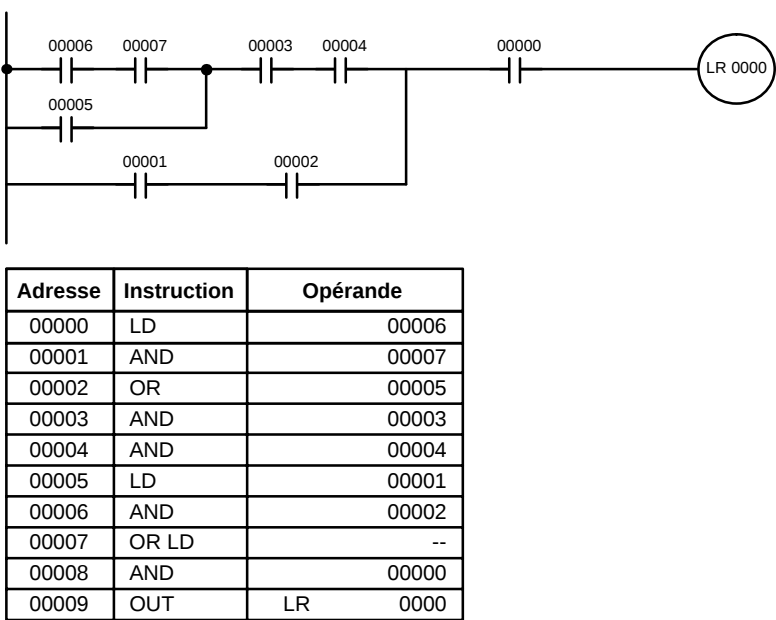
Adresse	Instruction	Opérande
00000	LD	00002
00001	AND NOT	00003
00002	OR	00001
00003	AND	00000
00004	LD NOT	00004
00005	AND	00005
00006	OR LD	--
00007	LD NOT	00006
00008	AND	00007
00009	OR LD	--
00010	OUT	LR 0000

Le schéma suivant nécessite cinq blocs, codés ici par ordre avant d'utiliser OR LOAD et AND LOAD pour les combiner en partant des deux derniers blocs et en remontant en sens inverse. Le OR LOAD à l'adresse 00008 com-

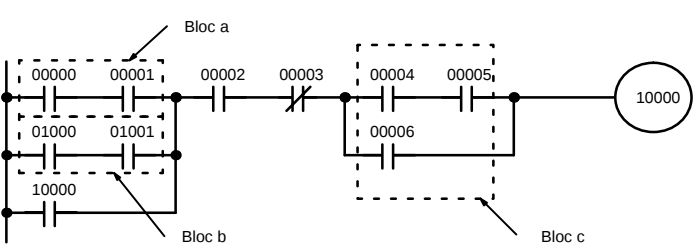
bine les blocs d et e ; le AND LOAD suivant combine la condition d'exécution résultante avec celle du bloc c, etc.



Ce schéma pourrait à nouveau être redessiné comme suit pour simplifier le programme et le codage et pour gagner de l'espace en mémoire :

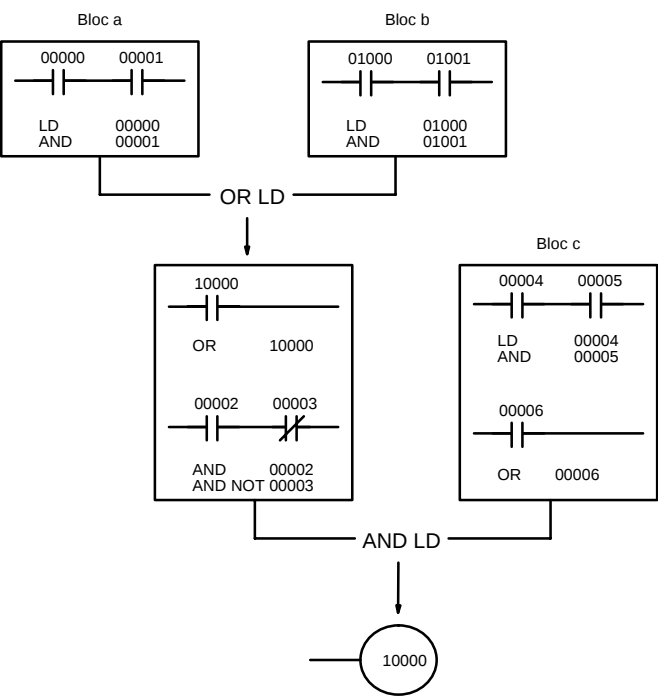


L'exemple suivant peut paraître très compliqué au départ mais il peut en fait être codé en utilisant simplement deux instructions en bloc logique. Le schéma se présente comme suit :



La première instruction en bloc logique s'utilise pour combiner les conditions d'exécution résultant des blocs a et b et la deuxième combine la condition d'exécution du bloc c avec celle résultant de la condition normale fer-

mée IR 00003. Le reste du schéma peut être codé avec OR, AND et AND NOT. Voici le schéma logique et le codage qui en découle :

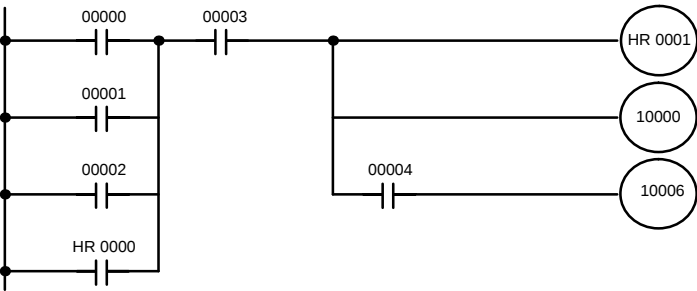


Adresse	Instruction	Opérande
00000	LD	00000
00001	AND	00001
00002	LD	01000
00003	AND	01001
00004	OR LD	--
00005	OR	10000
00006	AND	00002
00007	AND NOT	00003
00008	LD	00004
00009	AND	00005
00010	OR	00006
00011	AND LD	--
00012	OUT	10000

4-3-7 Codage des instructions multiples de droite

Dans le cas où plusieurs instructions de droite sont exécutées avec la même condition d'exécution, elles sont codées successivement à la suite de la dernière condition de la ligne. Dans l'exemple suivant, la dernière ligne d'instruc-

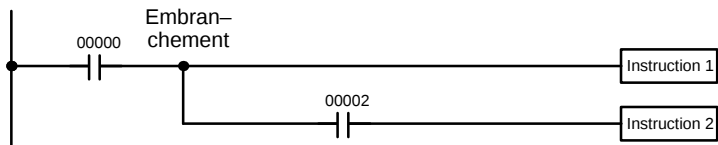
tion contient une instruction supplémentaire correspondant à un AND avec IR 00004.



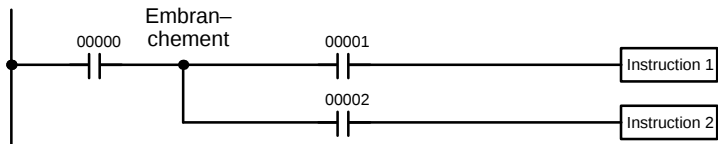
Adresse	Instruction	Opérande
00000	LD	00000
00001	OR	00001
00002	OR	00002
00003	OR	HR 0000
00004	AND	00003
00005	OUT	HR 0001
00006	OUT	10000
00007	AND	00004
00008	OUT	10006

4-3-8 Lignes secondaires

Lorsqu'une ligne d'instruction se sépare en une ou plusieurs autres lignes secondaires, il faut parfois utiliser des branchements ou des bits TR pour maintenir la condition d'exécution qui existait au niveau de l'embranchement. Cela vient du fait que les instructions sont exécutées de gauche à droite avant de retourner au point d'embranchement pour exécuter les instructions de la ligne secondaire. Si l'on trouve une condition située après un embranchement sur une des lignes d'instruction, la condition d'exécution peut changer pendant ce temps et rendre sa propre exécution impossible. Les schémas suivants illustrent ce problème ; l'instruction 1 est exécutée avant de revenir au point d'embranchement et de passer à la ligne secondaire menant à l'instruction 2.



Adresse	Instruction	Opérande
00000	LD	00000
00001	Instruction 1	
00002	AND	00002
00003	Instruction 2	



Adresse	Instruction	Opérande
00000	LD	00000
00001	AND	00001
00002	Instruction 1	
00003	AND	00002
00004	Instruction 2	

Si (comme sur le schéma A), la condition d'exécution existant au point d'embranchement ne peut être modifiée avant de revenir à la ligne secondaire (les instructions situées à l'extrême droite ne changent pas la condition d'exécution), la ligne secondaire est exécutée correctement et aucune mesure de programmation spéciale n'est nécessaire.

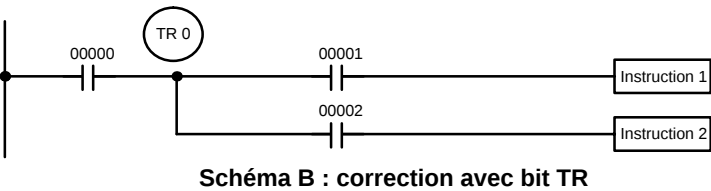
Si (comme sur le schéma B), une condition d'exécution existe entre le point d'embranchement et la dernière instruction de la ligne du haut, la condition d'exécution au point d'embranchement et celle qui suit l'instruction de la ligne du haut seront parfois différentes, ce qui rend impossible l'exécution correcte de la ligne secondaire.

Il y a deux façons de programmer les programmes d'embranchement préservant la condition d'exécution : en utilisant les bits TR ou les branchements (IL(02)/IL(03)).

Bits TR

La zone TR fournit 8 bits, TR 0 à 7, qui peuvent être utilisés pour préserver temporairement les conditions d'exécution. Si l'on place un bit TR sur un point d'embranchement, la condition d'exécution en cours est stockée dans le bit TR désigné. En revenant au point d'embranchement, le bit TR restaure l'état d'exécution qui a été sauvegardé lorsque le point d'embranchement a été atteint pour la première fois pendant l'exécution du programme.

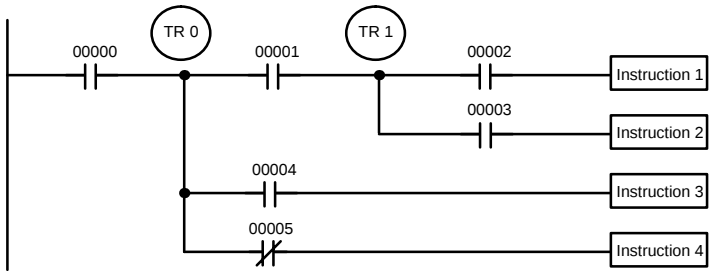
Le schéma B ci-dessus peut être écrit comme ci-dessous pour que l'exécution s'effectue correctement. En code mnémonique, la condition d'exécution est stockée au point d'embranchement par les bits TR comme opérande de l'instruction OUTPUT. Cette condition d'exécution est alors restaurée après exécution de l'instruction de droite par le même bit TR comme opérande d'une instruction LOAD.



Adresse	Instruction	Opérande
00000	LD	00000
00001	OUT	TR 0
00002	AND	00001
00003	Instruction 1	
00004	LD	TR 0
00005	AND	00002
00006	Instruction 2	

Les véritables instructions du schéma ci-dessus sont les suivantes : l'état de IR 00000 est chargé (instruction LOAD) pour établir la condition d'exécution initiale ; celle-ci est ensuite sortie par OUTPUT sur TR 0 pour stocker la condition d'exécution au point d'embranchement. La condition est ensuite reliée par un AND à l'état de IR 00001 et l'instruction 1 est exécutée en conséquence. La condition d'exécution stockée au point d'embranchement est alors rechargée (LOAD avec TR 0 comme opérande), reliée par un AND à l'état de IR 00002 et l'instruction 2 est exécutée.

Application avec deux bits TR :



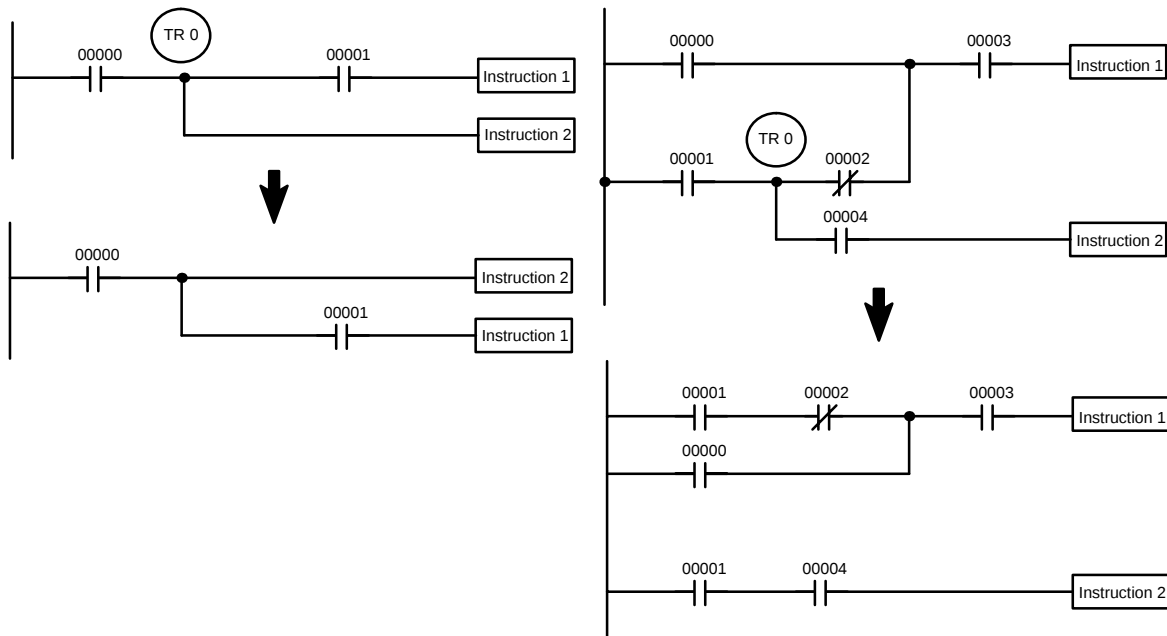
Adresse	Instruction	Opérande
00000	LD	00000
00001	OUT	TR 0
00002	AND	00001
00003	OUT	TR 1
00004	AND	00002
00005	Instruction 1	
00006	LD	TR 1
00007	AND	00003
00008	Instruction 2	
00009	LD	TR 0
00010	AND	00004
00011	Instruction 3	
00012	LD	TR 0
00013	AND NOT	00005
00014	Instruction 4	

Dans cet exemple, TR 0 et TR 1 sont utilisés pour stocker les conditions d'exécution au point d'embranchement. Après l'exécution de l'instruction 1, la condition stockée dans TR 1 est chargée pour effectuer un AND avec l'état de IR 00003. La condition d'exécution stockée dans TR 0 est chargée deux fois, la première pour effectuer un AND avec l'état de IR 00004 et la deuxième avec l'état inversé de IR 00005.

Les bits TR peuvent être utilisés autant de fois qu'on le désire, pourvu que le même bit ne soit pas utilisé deux fois dans le même bloc d'instructions. Ici, on commence un nouveau bloc d'instructions à chaque fois que l'exécution revient à la barre de bus. S'il faut plus de 8 points d'embranchement pour sauvegarder la condition d'exécution dans un seul bloc d'instructions, il est nécessaire de recourir aux embranchements (description ci-dessous).

Lorsque l'on dessine un schéma à relais, il faut veiller à ne pas utiliser les bits TR si ce n'est par vraiment nécessaire. Le nombre d'instructions nécessaires pour un programme peut souvent être réduit ; pour faciliter la compréhension d'un programme, on peut redessiner les schémas à relais et se passer ainsi de l'utilisation de bits TR. Dans les schémas suivants, les versions du bas ne nécessitent pas autant d'instructions que les autres et aucun bit TR. Dans le premier exemple, on ré-organise les différentes parties du bloc d'instructions : dans celui du bas, on sépare la deuxième instruction OUTPUT et on utilise une instruction LOAD supplémentaire pour créer sa propre condition d'exécution.

Rem. : Bien que la simplification des programmes soit toujours importante, il ne faut pas négliger l'ordre d'exécution des instructions. Par exemple, une instruction MOVE peut être nécessaire avant l'exécution d'une instruction BINARY ADD pour placer les bonnes données dans le canal d'opérande voulu. Veillez à respecter l'ordre d'exécution avant de simplifier les programmes.



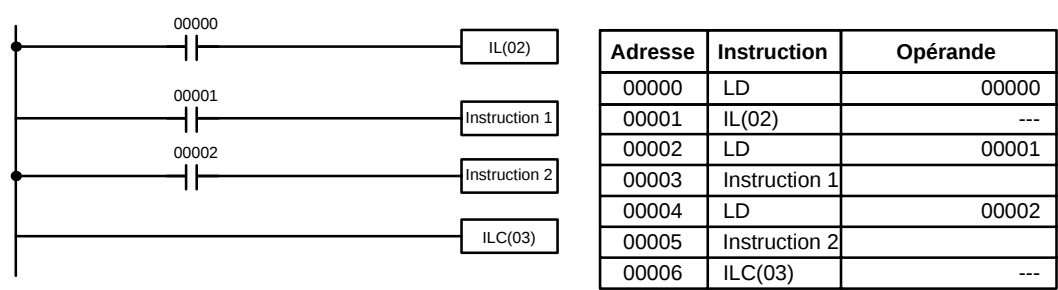
Rem. : Lorsque l'on programme en code mnémotechnique, les bits TR doivent absolument être entrés par l'utilisateur. Ils ne sont pas nécessaires lorsque l'on effectue les entrées directement en schéma à relais parce que le traitement se fait automatiquement. Les limites évoquées ci-dessus sur le nombre de points d'embranchements nécessitant des bits TR et les méthodes de réductions du nombre d'instructions s'appliquent également dans ce cas.

Embranchements

Le problème du stockage des conditions d'exécution aux points d'embranchement peut également être traité par les instructions INTERLOCK (IL(02)) et INTERLOCK CLEAR (ILC(03)) pour éliminer complètement le point d'embranchement tout en permettant à une condition d'exécution spécifique de contrôler un groupe d'instructions. Les instructions INTERLOCK et INTERLOCK CLEAR sont toujours utilisées ensemble.

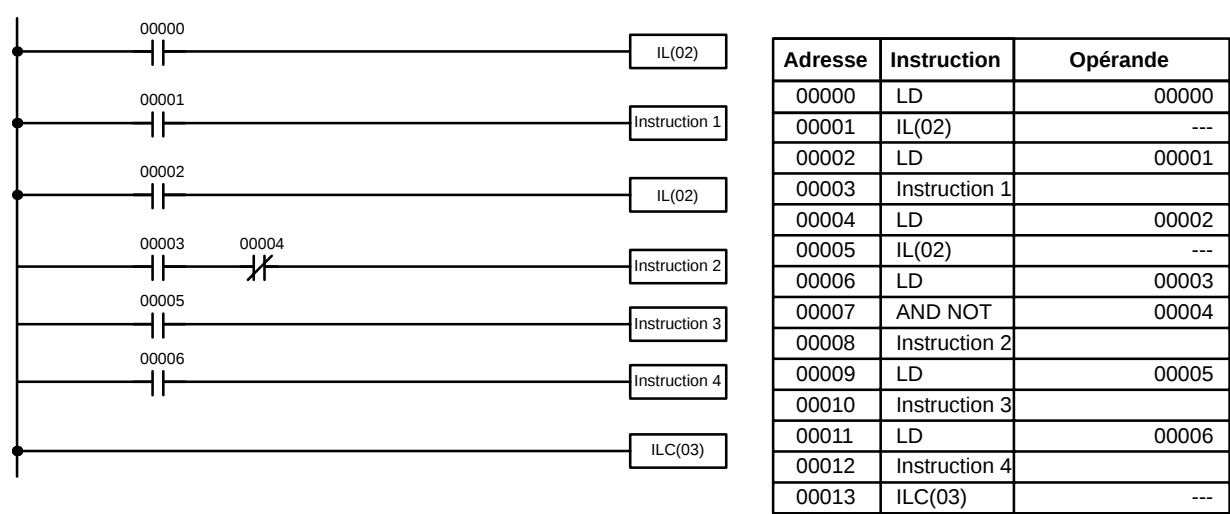
Lorsqu'une instruction INTERLOCK est placée avant une partie de programme en schéma à relais, la condition d'exécution de l'instruction INTERLOCK contrôle toutes les instructions jusqu'à la première INTERLOCK CLEAR. Si la condition d'exécution d'INTERLOCK est à OFF, toutes les instructions de droite jusqu'à la première INTERLOCK CLEAR sont exécutées avec les conditions à OFF pour la remise à zéro de toute cette partie du programme. L'effet de cette manoeuvre sur certaines instructions est décrit dans le chapitre concernant les instructions *INTERLOCK* et *INTERLOCK CLEAR – IL(02)* et *ILC(03)*.

Le schéma B peut également être corrigé par le moyen d'un embranchement. Ici, les conditions qui mènent au point d'embranchement sont placées sur une ligne d'instructions pour INTERLOCK et toutes les lignes partant du point d'embranchement sont figurées comme des lignes d'instructions séparées ; une ligne d'instruction supplémentaire est ajoutée pour l'instruction INTERLOCK CLEAR. Aucune condition n'est admise sur la ligne d'instructions pour INTERLOCK CLEAR. INTERLOCK et INTERLOCK CLEAR n'ont pas besoin d'opérande.



Si IR 00000 est à ON dans la deuxième version du schéma B ci-dessus, l'état de IR 00001 et de IR 00002 détermine les conditions d'exécution des instructions 1 et 2, respectivement. Comme IR 00000 est à ON, le résultat est le même que celui que produirait un AND entre les états de ces bits. Si IR 00000 est à OFF, l'instruction INTERLOCK produit une condition d'exécution OFF pour les instructions 1 et 2 et l'exécution continue avec la ligne d'instruction qui suit INTERLOCK CLEAR.

Comme on peut le voir sur le schéma suivant, il est possible d'utiliser plusieurs instructions INTERLOCK dans le même bloc d'instructions ; chacune reste effective jusqu'à la prochaine INTERLOCK CLEAR.



Si IR 00000 est à OFF dans le schéma ci-dessus (condition d'exécution de la première INTERLOCK à OFF), les instructions 1 à 4 doivent être exécutées avec les conditions d'exécution à OFF ; l'exécution passe ensuite à l'INTERLOCK CLEAR suivante. Si IR 00000 est à ON, l'état de IR 00001 doit être chargé comme condition d'exécution pour l'instruction 1, puis l'état de IR 00002 doit à son tour être chargé pour former la condition d'exécution de la deuxième instruction INTERLOCK. Si IR 00002 est à OFF, les instructions 2 à 4 doivent être exécutées avec des conditions d'exécution à OFF. Si IR 00002 est à ON, IR 00003, IR 00005 et IR 00006 déterminent la première condition d'exécution des nouvelles lignes d'instruction.

Remarque : ne jamais oublier de terminer une IL ou des conditions IL par une ILC ; dans le cas contraire, tous les programmes seraient conditionnés au dernier IL rencontré par le pointeur.

4-3-9 Sauts

On peut effectuer un saut d'une partie de programme par le moyen d'une condition d'exécution désignée. Bien que ce saut soit similaire à ce qu'il se passe lorsque la condition d'exécution d'INTERLOCK est à OFF, les sauts permettent de maintenir l'état de toutes les instructions grâce aux opérandes. On peut donc utiliser les sauts pour contrôler des appareils qui demandent une sortie maintenue (pneumatiques et hydrauliques par ex.), alors que les branchements peuvent être utilisés pour contrôler des appareils ne demandant pas de sortie maintenue (instruments électroniques par ex.).

On crée des sauts grâce aux instructions JUMP (JMP(04)) et JUMP END (JME(05)). Si la condition d'exécution d'une instruction JUMP est à ON, le programme est exécuté normalement, comme si le saut n'existait pas. Si la condition de JUMP est à OFF, l'exécution du programme passe immédiatement à JUMP END sans changer aucun état entre JUMP et JUMP END.

On attribue un numéro de saut à toutes les instructions JUMP et JUMP END entre 00 et 99. Il existe deux types de sauts : le numéro de saut utilisé détermine le type de saut.

On peut définir un saut à l'aide des numéros de saut 01 à 99 une seule fois. Chaque numéro peut être utilisé une fois par instruction JUMP et une fois par instruction JUMP END. Lorsqu'une instruction JUMP portant l'un de ces numéros est exécutée, l'exécution passe immédiatement à l'instruction JUMP END portant le même numéro, comme si les instructions situées entre JUMP et JUMP END n'existaient pas. Le schéma B à partir du bit TR et de l'exemple de branchement peut être redessiné comme suit, avec un saut. Bien que 01 soit utilisé comme numéro de saut, tout nombre situé entre 01 et 99 peut être utilisé dans la mesure où il n'est pas déjà utilisé dans une autre partie du programme. JUMP et JUMP END ne nécessitent pas d'autre opérande et JUMP END n'a jamais aucune condition sur sa ligne.

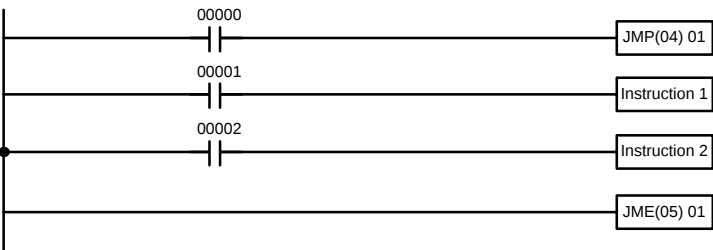


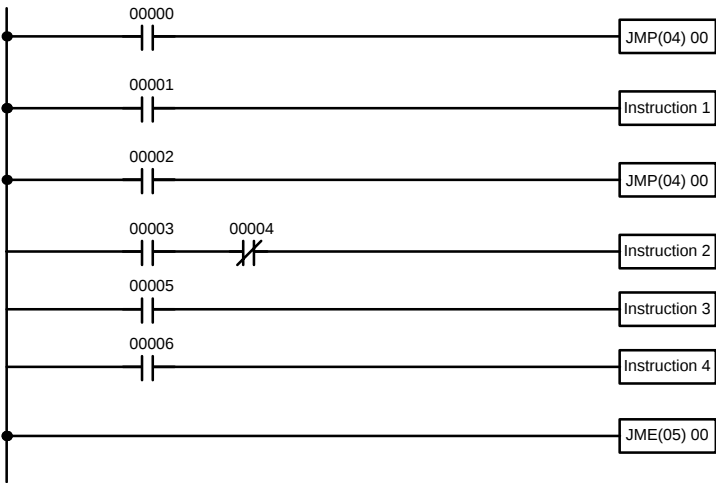
Schéma B : correction avec un saut

Adresse	Instruction	Opérande
00000	LD	00000
00001	JMP(04)	01
00002	LD	00001
00003	Instruction 1	
00004	LD	00002
00005	Instruction 2	
00006	JME(05)	01

Cette version du schéma B possède le temps d'exécution le plus court de toutes les versions lorsque IR 00000 est à OFF.

L'autre type de saut est créé à l'aide du numéro de saut 00. On peut créer autant de sauts qu'on le souhaite avec le numéro 00 et les instructions JUMP utilisant ce numéro peuvent être utilisées à la suite les unes des autres sans JUMP END 00 entre elles. Il est même possible pour toutes les instructions JUMP 00 de déplacer l'exécution du programme vers le même JUMP END 00 ; c'est-à-dire qu'une seule JUMP END 00 est nécessaire pour toutes les instructions JUMP 00 du programme. Lorsque 00 est utilisé comme numéro de saut pour une instruction JUMP, l'exécution du programme passe à l'instruction suivant immédiatement JUMP END avec un numéro de saut de 00. Bien que dans tous les sauts aucun état ne soit modifié et aucune instruction ne soit exécutée entre JUMP 00 et JUMP END 00, le programme recherche l'instruction JUMP END 00 suivante et produit un temps d'exécution légèrement plus long.

L'exécution des programmes contenant des instructions JUMP 00 multiples et une seule instruction JUMP END 00 est similaire à celle des parties créées par les branchements. Le schéma suivant est identique à celui qui est utilisé dans l'exemple de branchement ci-dessus mais il est redessiné avec des sauts. L'exécution de ce schéma est différente de celle du précédent : par exemple, les branchements précédents remettent à zéro certaines parties du branchement mais les sauts n'affectent pas l'état des bits entre JUMP et JUMP END).



Adresse	Instruction	Opérande
00000	LD	00000
00001	JMP(04)	00
00002	LD	00001
00003	Instruction 1	
00004	LD	00002
00005	JMP(04)	00
00006	LD	00003
00007	AND NOT	00004
00008	Instruction 2	
00009	LD	00005
00010	Instruction 3	
00011	LD	00006
00012	Instruction 4	
00013	JME(05)	00

4-4 Contrôle de l'état des bits

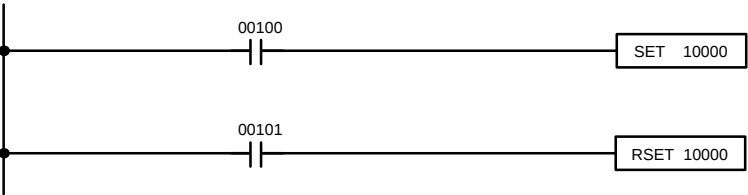
On peut utiliser 7 instructions de base pour contrôler individuellement l'état des bits : OUTPUT, OUTPUT NOT, SET, RESET, DIFFERENTIATE UP, DIFFERENTIATE DOWN et KEEP. Toutes ces instructions apparaissent à la fin d'une ligne d'instructions et ont une adresse de bit comme opérande. On trouvera le détail de ces instructions dans le chapitre 5 (sauf pour OUTPUT et OUTPUT NOT que nous avons déjà vues), mais elles sont évoquées ici car elles sont d'une importance capitale dans la plupart des programmes. Ces instructions s'utilisent pour placer les bits de sortie de la zone IR à ON ou OFF (c'est-à-dire pour envoyer ou arrêter des signaux de sortie vers les appareils externes) et pour contrôler l'état d'autres bits dans la zone IR ou dans d'autres zones.

4-4-1 SET et RESET

Les instructions SET et RESET sont très similaires aux instructions OUTPUT et OUTPUT NOT, à ceci près qu'elles changent l'état de leurs bits d'opérande pour les conditions d'exécution à ON. Aucune de ces deux instructions n'affecte l'état de son bit d'opérande lorsque la condition d'exécution est à OFF.

SET place à ON son bit d'opérande lorsque la condition d'exécution passe à ON, mais à la différence de l'instruction OUTPUT, SET ne place pas à OFF le bit d'opérande lorsque la condition d'exécution passe à OFF. RESET place à OFF le bit d'opérande lorsque la condition d'exécution est à OFF, mais contrairement à OUTPUT NOT, RESET ne passe pas à ON le bit d'opérande lorsque la condition d'exécution passe à OFF.

Dans l'exemple suivant, IR 10000 passe à ON lorsque IR 00100 passe à ON et reste à ON jusqu'à ce que IR 00101 passe à ON, quel que soit l'état de IR 00100. Lorsque IR 00101 passe à ON, RESET place IR 10000 à OFF.



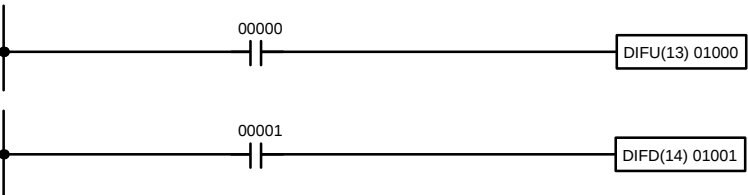
Adresse	Instruction	Opérande
00000	LD	00100
00001	SET	10000
00002	LD	00101
00003	RSET	10000

4-4-2 DIFFERENTIATE UP et DIFFERENTIATE DOWN

Les instructions DIFFERENTIATE UP (front montant) et DIFFERENTIATE DOWN (front descendant) s'utilisent pour placer le bit d'opérande à ON pour un tour de scrutation à la fois.

DIFFERENTIATE UP place à ON le bit d'opérande pour un tour de scrutation après le passage de OFF à ON de sa condition d'exécution ;

DIFFERENTIATE DOWN place à ON le bit d'opérande pour un tour de scrutation après le passage de ON à OFF de sa condition d'exécution. Ces deux instructions ne nécessitent qu'une ligne d'instruction en code mnémotique.



Adresse	Instruction	Opérande
00000	LD	00000
00001	DIFU(13)	01000

Adresse	Instruction	Opérande
00000	LD	00001
00001	DIFD(14)	01001

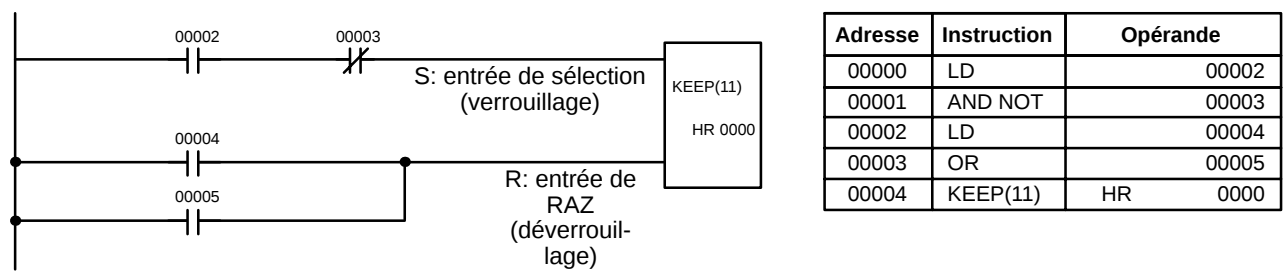
Ici, IR 01000 est placé à ON pendant un tour de scrutation après le passage de IR 00000 à ON. A l'exécution suivante de DIFU(13) 01000, IR 01000 passe à OFF, quel que soit l'état de IR 00000. Avec l'instruction DIFFERENTIATE DOWN, IR 01001 passe à ON pendant un tour de scrutation après le passage de IR 00001 à OFF (IR 01001 reste à OFF jusque-là) et à OFF à la prochaine exécution de DIFD(14) 01001.

4-4-3 KEEP

KEEP s'utilise pour maintenir l'état du bit d'opérande résultant de deux conditions d'exécution. Pour ce faire, KEEP est connectée à deux lignes

d’instruction. Lorsque la condition d’exécution de la fin de la première ligne est à ON (verrouillage), le bit d’opérande de KEEP est placé à ON (déverrouillage). Lorsque la condition d’exécution de la fin de la deuxième ligne est à ON, le bit d’opérande de KEEP est placé à OFF(verrouillage). Le bit d’opérande de KEEP maintient sont état ON ou OFF même s’il est situé dans un branchement. Le bit de déverrouillage est prioritaire sur le bit de verrouillage.

Dans l'exemple suivant, HR 0000 passe à ON lorsque IR 00002 est à ON et IR 00003 à OFF. HR 0000 reste ensuite à ON jusqu'à ce que IR 00004 ou IR 00005 passe à ON. Avec KEEP comme avec toutes les instructions demandant plus d'une ligne, on code d'abord les lignes avant de coder l'instruction qu'elles contrôlent.

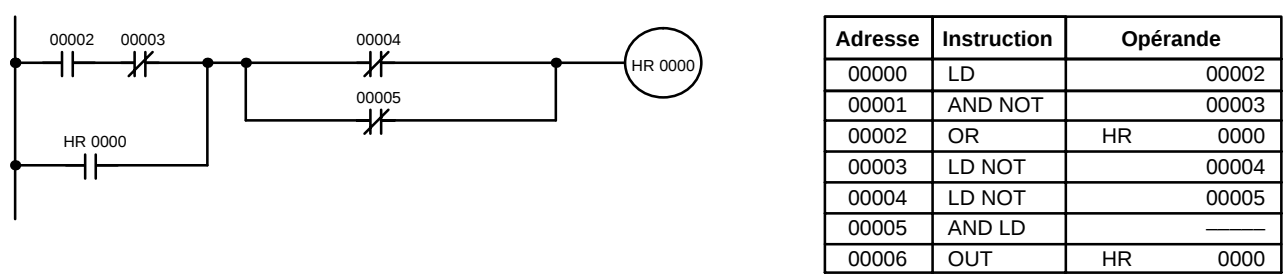


4-4-4 Bits de maintien automatique ("Seal")

Bien que l'on puisse utiliser l’instruction KEEP pour créer des bits de maintien automatique, il est parfois nécessaire d’en créer de façon à ce qu’ils puissent être placés à OFF lorsqu’il est situé dans un branchement.

Pour créer un bit de maintien automatique, on utilise le bit d’opérande d’une instruction OUTPUT comme condition de cet OUTPUT dans une configuration OR de façon à ce que le bit d’opérande d’OUTPUT reste à ON ou OFF jusqu’à ce que des changements se produisent dans d’autres bits. On utilise au moins une autre condition juste avant OUTPUT pour la remise à zéro. Sans cela, il est impossible de contrôler le bit d’opérande de l’instruction OUTPUT.

Le schéma ci-dessus avec KEEP peut être ré-écrit de la façon suivante. La seule différence entre ces schémas tient dans leur fonctionnement à l’intérieur des branchements lorsque la condition d’exécution d’INTERLOCK est à ON. Ici, comme dans l’autre schéma utilisant KEEP, on utilise deux bits de remise à zéro (HR 0000 peut être placé à OFF en plaçant IR 00004 ou IR 00005 à ON).



4-5 Bits de travail (relais internes)

En programmation, la combinaison des conditions pour produire directement une condition d’exécution est souvent très délicate. On vient cependant aisément à bout des difficultés en utilisant certains bits pour déclencher indirecte-

ment d'autres instructions. On utilise pour cela des bits de travail et même parfois des canaux entiers : ce sont des "canaux de travail".

Les canaux de travail ne se transfèrent pas à partir de l'API ou vers l'API. Ce sont des bits sélectionnés par le programmeur pour faciliter la programmation décrite ci-dessus. Les bits d'E/S et autres bits dédiés ne peuvent être utilisés comme bits de travail. Tous les bits de la zone IR qui ne sont pas des bits d'E/S et certains bits non utilisés dans la zone AR peuvent être utilisés comme bits de travail. Veillez à garder une trace des moments et endroits précis où vous utilisez ces bits de travail : cela aide à la planification, à l'écriture des programmes et à la mise au point.

Applications des bits de travail

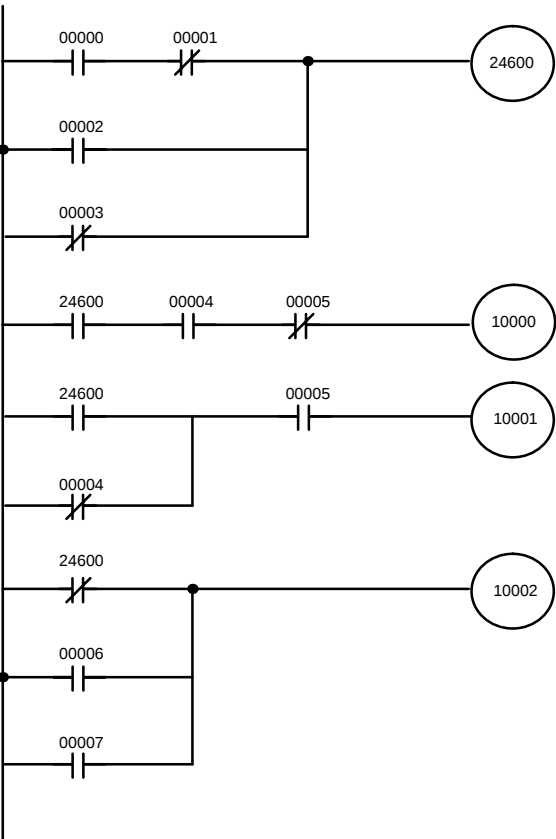
Les exemples suivants montrent les façons les plus communes d'utiliser les bits de travail et illustre les possibilités presque infinies qu'ils offrent dans leur utilisation. Lorsque des difficultés apparaissent dans la programmation d'une action de contrôle, il faut considérer l'utilisation possible des bits de travail et la simplification qu'ils apportent.

Les bits de travail s'utilisent souvent avec les instructions OUTPUT, OUTPUT NOT, DIFFERENTIATE UP, DIFFERENTIATE DOWN et KEEP. On utilise d'abord le bit de travail comme opérande d'une des instructions de façon à pouvoir ensuite l'employer comme condition déterminant l'exécution des autres instructions. On peut également utiliser les bits de travail avec d'autres instructions comme SHIFT REGISTER par exemple (SFT(10)). Il existe un exemple d'utilisation des bits et canaux de travail avec l'instruction SHIFT REGISTER dans le chapitre 5.

Bien qu'on ne les appelle pas toujours bits de travail, de nombreux bits utilisés dans le chapitre 5 emploient des bits de travail. La compréhension de l'utilisation de ces bits est essentielle si l'on veut programmer correctement.

Réduction de conditions complexes

On peut utiliser des bits de travail pour simplifier la programmation lorsqu'une certaine combinaison de conditions est employée de façon répétitive avec d'autres conditions. Dans l'exemple suivant, IR 00000, IR 00001, IR 00002 et IR 00003 sont combinés dans un bloc logique stockant la condition d'exécution qui en résulte, c'est-à-dire l'état de IR 24600. IR 24600 est alors combiné avec diverses autres conditions pour déterminer les conditions de sortie de IR 10000, IR 10001 et IR 10002, c'est-à-dire pour placer les sorties attribuées à ces bits à ON ou OFF.

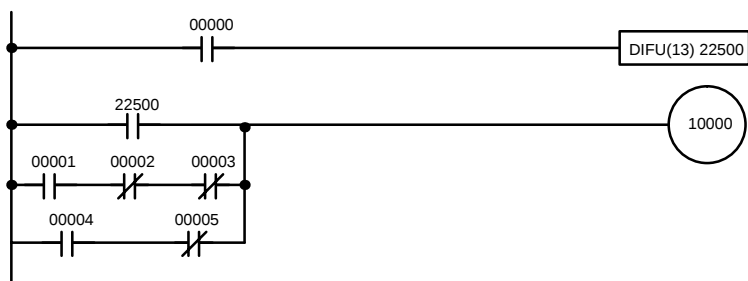


Adresse	Instruction	Opérande
00000	LD	00000
00001	AND NOT	00001
00002	OR	00002
00003	OR NOT	00003
00004	OUT	24600
00005	LD	24600
00006	AND	00004
00007	AND NOT	00005
00008	OUT	10000
00009	LD	24600
00010	OR NOT	00004
00011	AND	00005
00012	OUT	10001
00013	LD NOT	24600
00014	OR	00006
00015	OR	00007
00016	OUT	10002

Conditions sur changement d'état

Les bits de travail peuvent également être utilisés en changement d'état si besoin est, pour certaines conditions (mais pas pour toutes) nécessaires à l'exécution d'une instruction. Dans cet exemple, IR 10000 doit constamment rester à ON lorsque IR 001001 est à ON et IR 00002 et IR 00003 sont à OFF aussi longtemps que IR 00004 est à ON et que IR 00005 est à OFF. Il doit être placé à ON pendant un tour de scrutation seulement à chaque fois que IR 00000 passe à ON (sauf si l'une des conditions précédentes le maintient à ON continuellement).

Cette action se programme facilement en utilisant IR 22500 comme bit de travail et opérande de DIFFERENTIATE UP (DIFU(13)). Lorsque IR 00000 passe à ON, IR 22500 est placé à ON pour un tour de scrutation puis à OFF pour le tour suivant par DIFU(13). Si l'on part du principe que les autres conditions contrôlant IR 10000 ne le maintiennent pas à ON, le bit de travail IR 22500 place IR 10000 à ON pendant un tour de scrutation seulement.



Adresse	Instruction	Opérande
00000	LD	00000
00001	DIFU(13)	22500
00002	LD	22500
00003	LD	00001
00004	AND NOT	00002
00005	AND NOT	00003
00006	OR LD	---
00007	LD	00004
00008	AND NOT	00005
00009	OR LD	---
00010	OUT	10000

4-6 Précautions de programmation

le nombre de conditions pouvant être utilisées en série ou en parallèle est illimité tant que l'on ne dépasse pas la capacité mémoire de l'API. Il faut donc utiliser autant de conditions qu'il est nécessaire pour obtenir un schéma clair. Bien que des schémas très complexes puissent être dessinés avec des lignes d'instruction, il ne doit y avoir aucune condition sur les lignes verticales joignant les lignes d'instruction. Le schéma A ci-dessous, par exemple, n'est pas faisable, et doit être redessiné comme indiqué sur le schéma B. Le code mnémonique est indiqué seulement pour le schéma B car le codage du schéma A est impossible.

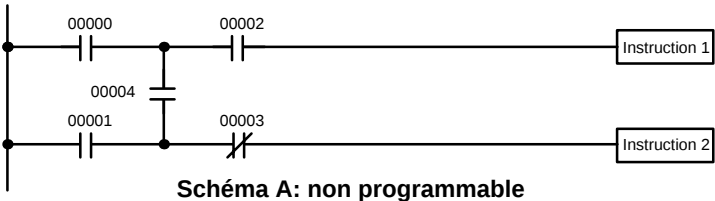


Schéma A: non programmable

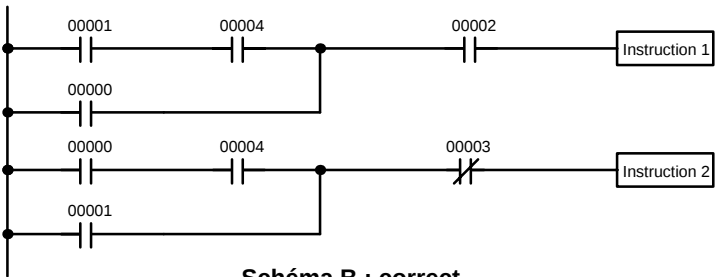


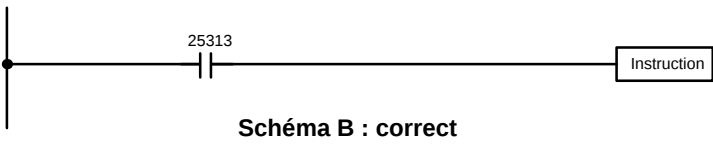
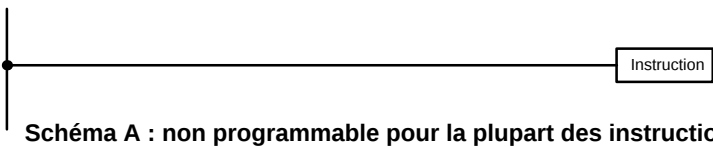
Schéma B : correct

Adresse	Instruction	Opérande
00000	LD	00001
00001	AND	00004
00002	OR	00000
00003	AND	00002
00004	Instruction 1	
00005	LD	00000
00006	AND	00004
00007	OR	00001
00008	AND NOT	00003
00009	Instruction 2	

On peut attribuer un bit autant de fois qu'on le désire ; il faut donc les utiliser à loisir pour simplifier le programme. Les programmes compliqués sont souvent le résultat d'un trop grand souci de réduction du nombre de fois où l'on utilise un bit.

Sauf pour les instructions qui n'admettent pas de conditions (INTERLOCK CLEAR et JUMP END, par ex., cf ci-dessous), toute ligne d'instruction doit posséder au moins une condition pour déterminer la condition d'exécution de l'instruction de droite. Le schéma A ci-dessus a donc dû être redessiné comme le schéma B le montre. Si une instruction doit être exécutée en continu (par ex. si une sortie doit être maintenue constamment à ON pendant

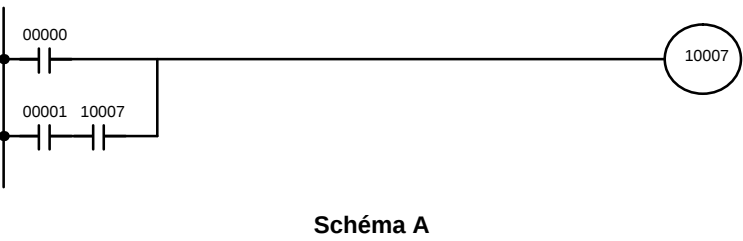
l'exécution du programme), le drapeau "Constamment à ON" (SR 25313) peut être utilisé dans la zone SR.



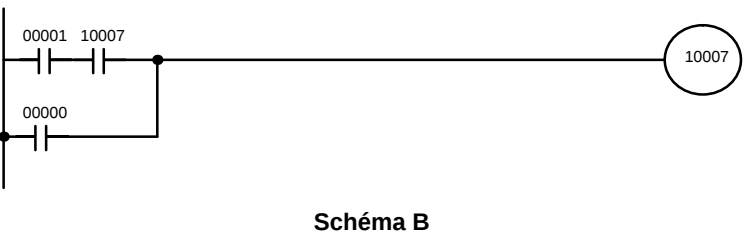
Adresse	Instruction	Opérande
00000	LD	25313
00001	Instruction	

Il y a peu d'exceptions à cette règle, y compris les instructions INTERLOCK CLEAR, JUMP END et les instructions d'étape. Elles sont utilisées comme deuxième instruction d'une paire d'instructions et sont contrôlées par la condition d'exécution de la première instruction. Les conditions ne doivent pas être placées sur la ligne d'instruction menant à ces instructions. Reportez-vous au chapitre 5 pour de plus amples détails.

Lorsque vous dessinez des schémas à relais, gardez à l'esprit le nombre d'instructions nécessaires à son entrée. Dans le schéma A ci-dessous, il faut une instruction OR LOAD pour combiner les lignes d'instruction du haut et du bas ; on peut éviter cela en redessinant le schéma B ci-dessous et en éliminant les instructions AND LOAD et OR LOAD. Reportez-vous au chapitre 5 pour de plus amples détails.



Adresse	Instruction	Opérande
00000	LD	00000
00001	LD	00001
00002	AND	10007
00003	OR LD	---
00004	OUT	10007



Adresse	Instruction	Opérande
00000	LD	00001
00001	AND	10007
00002	OR	00000
00003	OUT	10007

4-7 Exécution du programme

Au début de l'exécution d'un programme, l'UC scrute le programme de haut en bas, vérifie toutes les conditions et exécute toutes les instructions qui en résultent, en descendant la barre de bus. Il est important de bien ordonner les instructions de façon à ce que, par exemple, les données voulues soient transférées vers un canal avant que celui-ci ne soit utilisé comme opérande d'instruction. Souvenez-vous qu'une ligne d'instructions (ou équation logique) va jusqu'au niveau de l'instruction terminale de droite (OUT) avant d'exécuter une ligne dont le point d'embranchement serait situé sur la première ligne (LD) et qui mènerait à d'autres instructions à droite (AND ou OR).

L'exécution du programme n'est que l'une des tâches exécutées par l'UC dans le temps de scrutation. Reportez-vous au chapitre 7 pour de plus amples détails.

Chapitre 5

Instructions

Le CQM1 possède une vaste gamme d'instructions permettant une programmation aisée même pour les programmes complexes. Les instructions sont présentées individuellement, ainsi que leurs symboles en schéma à relais, les zones de données et les drapeaux utilisés. Les nombreuses instructions du CQM1 s'organisent par groupes : ils comprennent les instructions en schéma à relais, les instructions à codes fixes et les instructions de sélection. Certaines instructions, comme celles de temporisation et compteur s'utilisent pour contrôler l'exécution d'autres instructions, (par ex. un drapeau de fin TIM peut être utilisé pour placer à ON un bit lorsque le temps de temporisation s'est écoulé. Bien que ces instructions soient souvent utilisées pour contrôler les bits de sortie par l'instruction de sortie, ils peuvent être utilisés également pour contrôler l'exécution d'autres instructions. Les instructions de sortie utilisées dans les exemples de ce manuel peuvent généralement être remplacées par d'autres instructions pour modifier le programme d'applications spécifiques autres que le contrôle direct des bits de sortie.

5-1	Notation	133
5-2	Format	133
5-3	Zones de données, données d'opérande et drapeaux	133
5-4	Instructions sur changement d'état	135
5-5	Codage des instructions de droite	136
5-6	Tableaux d'instructions	139
5-6-1	Codes fonction	139
5-6-2	Liste alphabétique des codes mnémoniques	140
5-7	Instructions en schéma à relais	143
5-7-1	LOAD, LOAD NOT, AND, AND NOT, OR et OR NOT	143
5-7-2	AND LOAD et OR LOAD	144
5-8	Instructions de contrôle de bit	144
5-8-1	OUTPUT et OUTPUT NOT – OUT et OUT NOT	145
5-8-2	SET et RESET – SET et RSET	145
5-8-3	KEEP – KEEP(11)	146
5-8-4	DIFFERENTIATE UP et DOWN – DIFU(13) et DIFD(14)	147
5-9	NO OPERATION – NOP(00)	148
5-10	END – END(01)	148
5-11	INTERLOCK et INTERLOCK CLEAR – IL(02) et ILC(03)	149
5-12	JUMP et JUMP END – JMP(04) et JME(05)	151
5-13	Instructions d'erreur utilisateur : FAILURE ALARM ET RESET – FAL(06) et SEVERE FAILURE ALARM – FALS(07)	152
5-14	Instructions d'étape : STEP DEFINE et STEP START–STEP(08)/SNXT(09)	153
5-15	Instructions de temporisation/compteur	155
5-15-1	TIMER – TIM	156
5-15-2	COUNTER – CNT	157
5-15-3	REVERSIBLE COUNTER – CNTR(12)	158
5-15-4	HIGH-SPEED TIMER – TIMH(15)	159
5-15-5	INTERVAL TIMER – STIM(—)	161
5-15-6	REGISTER COMPARISON TABLE – CTBL(—)	162
5-15-7	MODE CONTROL – INI(—)	167
5-15-8	HIGH-SPEED COUNTER PV READ – PRV(—)	169
5-16	Registre à décalage	171
5-16-1	SHIFT REGISTER – SFT(10)	171
5-16-2	WORD SHIFT – WSFT(16)	172
5-16-3	ARITHMETIC SHIFT LEFT – ASL(25)	173
5-16-4	ARITHMETIC SHIFT RIGHT – ASR(26)	173
5-16-5	ROTATE LEFT – ROL(27)	174
5-16-6	ROTATE RIGHT – ROR(28)	174

5-16-7	ONE DIGIT SHIFT LEFT – SLD(74)	175
5-16-8	ONE DIGIT SHIFT RIGHT – SRD(75)	176
5-16-9	REVERSIBLE SHIFT REGISTER – SFTR(84)	177
5-16-10	ASYNCHRONOUS SHIFT REGISTER – ASFT(—)	178
5-17	Instructions de déplacement de données	180
5-17-1	MOVE – MOV(21)	180
5-17-2	MOVE NOT – MVN(22)	180
5-17-3	BLOCK TRANSFER – XFER(70)	181
5-17-4	BLOCK SET – BSET(71)	182
5-17-5	DATA EXCHANGE – XCHG(73)	183
5-17-6	SINGLE WORD DISTRIBUTE – DIST(80)	184
5-17-7	DATA COLLECT – COLL(81)	185
5-17-8	MOVE BIT – MOVB(82)	188
5-17-9	MOVE DIGIT – MOVD(83)	189
5-17-10	TRANSFER BITS – XFRB(—)	190
5-18	Instructions de comparaison	191
5-18-1	COMPARE – CMP(20)	191
5-18-2	TABLE COMPARE – TCMP(85)	192
5-18-3	BLOCK COMPARE – BCMP(—)	193
5-18-4	DOUBLE COMPARE – CMPL(—)	195
5-18-5	MULTI-WORD COMPARE – MCMP(—)	196
5-18-6	SIGNED BINARY COMPARE – CPS(—)	197
5-18-7	DOUBLE SIGNED BINARY COMPARE – CPSL(—)	199
5-18-8	AREA RANGE COMPARE – ZCP(—)	200
5-18-9	DOUBLE AREA RANGE COMPARE – ZCPL(—)	201
5-19	Instructions de conversion	202
5-19-1	BCD-TO-BINARY – BIN(23)	202
5-19-2	BINARY-TO-BCD – BCD(24)	203
5-19-3	DOUBLE BCD-TO-DOUBLE BINARY – BINL(58)	204
5-19-4	DOUBLE BINARY-TO-DOUBLE BCD – BCDL(59)	204
5-19-5	4-TO-16 DECODER – MLPX(76)	205
5-19-6	16-TO-4 ENCODER – DMPX(77)	207
5-19-7	7-SEGMENT DECODER – SDEC(78)	209
5-19-8	ASCII CONVERT – ASC(86)	212
5-19-9	ASCII-TO-HEXADECIMAL – HEX(—)	214
5-19-10	SCALING – SCL(—)	216
5-19-11	SIGNED BINARY TO BCD SCALING – SCL2(—)	218
5-19-12	BCD TO SIGNED BINARY SCALING – SCL3(—)	220
5-19-13	HOURS-TO-SECONDS – SEC(—)	222
5-19-14	SECONDS-TO-HOURS – HMS(—)	223
5-19-15	COLUMN-TO-LINE – LINE(—)	224
5-19-16	LINE-TO-COLUMN – COLM(—)	225
5-19-17	2’S COMPLEMENT – NEG(—)	226
5-19-18	DOUBLE 2’S COMPLEMENT – NEGL(—)	227
5-20	Instructions de calcul BCD	229
5-20-1	SET CARRY – STC(40)	229
5-20-2	CLEAR CARRY – CLC(41)	229
5-20-3	BCD ADD – ADD(30)	229
5-20-4	BCD SUBTRACT – SUB(31)	230
5-20-5	BCD MULTIPLY – MUL(32)	232
5-20-6	BCD DIVIDE – DIV(33)	233
5-20-7	DOUBLE BCD ADD – ADDL(54)	234
5-20-8	DOUBLE BCD SUBTRACT – SUBL(55)	236

5-20-9	DOUBLE BCD MULTIPLY – MULL(56)	238
5-20-10	DOUBLE BCD DIVIDE – DIVL(57)	238
5-20-11	SQUARE ROOT – ROOT(72)	239
5-21	Instructions de calcul binaire	240
5-21-1	BINARY ADD – ADB(50)	240
5-21-2	BINARY SUBTRACT – SBB(51)	241
5-21-3	BINARY MULTIPLY – MLB(52)	243
5-21-4	BINARY DIVIDE – DVB(53)	243
5-21-5	DOUBLE BINARY ADD – ADBL(—)	244
5-21-6	DOUBLE BINARY SUBTRACT – SBBL(—)	245
5-21-7	SIGNED BINARY MULTIPLY – MBS(—)	247
5-21-8	DOUBLE SIGNED BINARY MULTIPLY – MBSL(—)	248
5-21-9	SIGNED BINARY DIVIDE – DBS(—)	249
5-21-10	DOUBLE SIGNED BINARY DIVIDE – DBSL(—)	250
5-22	Instructions mathématiques spéciales	251
5-22-1	FIND MAXIMUM – MAX(—)	251
5-22-2	FIND MINIMUM – MIN(—)	252
5-22-3	AVERAGE VALUE – AVG(—)	253
5-22-4	SUM – SUM(—)	255
5-22-5	ARITHMETIC PROCESS – APR(—)	257
5-23	Instructions logiques	260
5-23-1	COMPLEMENT – COM(29)	260
5-23-2	LOGICAL AND – ANDW(34)	261
5-23-3	LOGICAL OR – ORW(35)	262
5-23-4	EXCLUSIVE OR – XORW(36)	262
5-23-5	EXCLUSIVE NOR – XNRW(37)	263
5-24	Instructions d'incrémentation/décrémentation	264
5-24-1	BCD INCREMENT – INC(38)	264
5-24-2	BCD DECREMENT – DEC(39)	264
5-25	Instructions de sous-programme	265
5-25-1	SUBROUTINE ENTER – SBS(91)	265
5-25-2	SUBROUTINE DEFINE and RETURN – SBN(92)/RET(93)	267
5-26	Instructions spéciales	267
5-26-1	TRACE MEMORY SAMPLING – TRSM(45)	267
5-26-2	MESSAGE DISPLAY – MSG(46)	269
5-26-3	I/O REFRESH – IORF(97)	270
5-26-4	MACRO – MCRO(99)	271
5-26-5	BIT COUNTER – BCNT(—)	272
5-26-6	FRAME CHECKSUM – FCS(—)	273
5-26-7	FAILURE POINT DETECTION – FPD(—)	275
5-26-8	INTERRUPT CONTROL – INT(—)	279
5-26-9	SET PULSES – PULS(—)	280
5-26-10	SPEED OUTPUT – SPED(—)	282
5-26-11	PULSE OUTPUT – PLS2(—)	284
5-26-12	ACCELERATION CONTROL – ACC(—)	285
5-26-13	PULSE WITH VARIABLE DUTY RATIO – PWM(—)	288
5-26-14	DATA SEARCH – SRCH(—)	289
5-26-15	PID CONTROL – PID(—)	290
5-27	Instructions de communication	292
5-27-1	RECEIVE – RXD(—)	292
5-27-2	TRANSMIT – TXD(—)	293
5-28	Instructions d'E/S avancées	295
5-28-1	7-SEGMENT DISPLAY OUTPUT – 7SEG(—)	295

5-28-2	DIGITAL SWITCH INPUT – DSW(—)	296
5-28-3	HEXADECIMAL KEY INPUT – HKY(—)	297
5-28-4	TEN KEY INPUT – TKY(—)	298

5-1 Notation

Toutes les instructions sont nommées ici sous leur code mnémonique : l'instruction OUTPUT est donc appelée "OUT" et the AND LOAD, AND LD... Si vous avez des doutes, reportez-vous à la table des matières de ce chapitre.

Si un code de fonction est attribué à l'instruction, il est donné entre parenthèses à la suite du code mnémonique. Ces codes de fonction qui sont des nombres décimaux à deux chiffres, s'utilisent pour entrer la plupart des instructions dans l'UC. La liste des instructions se trouve dans le chapitre 5–6.

Le signe @ précédant le code mnémonique indique un seul changement d'état de cette instruction. Les instructions sur changement d'état sont décrites dans le chapitre 5–4.

5-2 Format

Les plupart des instructions ont au moins une opérande associée. Les opérandes indiquent ou fournissent des données pour l'exécution de l'instruction. Elles sont parfois entrées comme des valeurs numériques réelles (c'est-à-dire des constantes) mais sont généralement des adresses de canaux de zones ou de bits contenant les données à utiliser. Un bit dont l'adresse est désignée comme opérande s'appelle bit d'opérande ; un canal dont l'adresse est désignée comme opérande s'appelle canal d'opérande. Dans certaines instructions, l'adresse de canal désignée dans une instruction indique le premier canal multiple contenant la donnée désirée.

Chaque instruction demande un ou plusieurs mots de mémoire programme. Le premier mot est le mot d'instruction, spécifiant l'instruction et contenant les données d'opérande ou bits d'opérande requis par l'instruction. Les autres opérandes nécessaires pour l'instruction sont contenues dans les mots suivants, à raison d'une opérande par mot. Certaines instructions nécessitent jusqu'à 4 canaux.

Une donnée d'opérande est une opérande associée à une instruction et contenue dans le même mot que l'instruction elle-même. Ces opérandes définissent l'instruction plutôt qu'elles n'indiquent la donnée à utiliser. Les données d'opérande sont par exemple les numéros de TC (tempo./compteurs), utilisés dans les instructions de TC pour créer des temporisations et des compteurs, ou les numéros de sauts (définissant quelle instruction Jump est associée à telle instruction Jump End). Les opérandes de bits sont également contenues dans le même canal que l'instruction elle-même, bien que celles-ci ne soient pas considérées comme des données d'opérande.

5-3 Zones de données, données d'opérande et drapeaux

Dans ce chapitre, chaque description d'instruction contient son symbole en schéma à relais, les zones de données qui peuvent être utilisées par ses opérandes et les valeurs qui peuvent être utilisées comme données d'opérande. Les détails concernant la zone de données sont également fournis par le nom d'opérande et le type de donnée nécessaire à chaque opérande (c'est-à-dire canal ou bit et hexadécimal ou BCD pour les canaux).

Toutes les adresses de la zone de données spécifiée sont possibles pour une opérande, c'est-à-dire que si une opérande requiert deux canaux, le dernier canal d'une zone de données ne peut être désigné comme premier canal de l'opérande car tous les canaux d'une même opérande doivent appartenir à la même zone de données. Les autres limites sont décrites dans le chapitre "Limites". Reportez-vous au chapitre 3 pour les conventions d'adressage, les adresses de drapeaux et de bits de contrôle.

Attention : Les zones IR et SR sont considérées comme des zones de données séparées. Si une opérande a accès à l'une de ces zones, cela ne signifie pas forcément qu'elle a accès à l'autre zone. La limite entre ces deux zones peut cependant être franchie par une seule opérande : il faut spécifier le dernier bit de la zone IR pour une opérande nécessitant plusieurs canaux, dans la mesure où la zone SR est également possible pour cette opérande.

Le chapitre concernant les drapeaux établit la liste des drapeaux affectés par l'exécution d'une instruction. Ces drapeaux comprennent les drapeaux de la zone SR suivants :

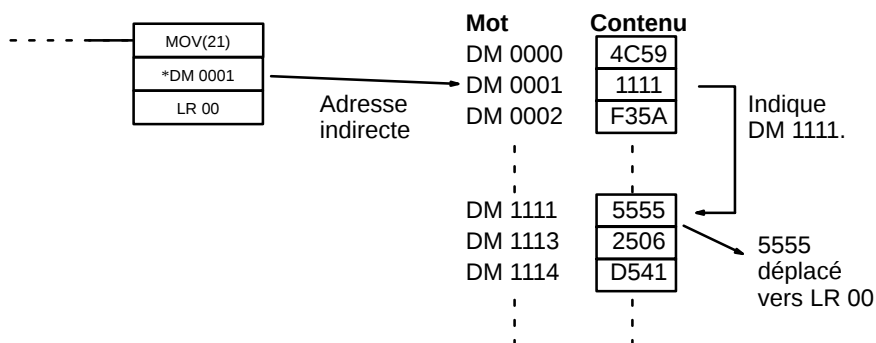
Abréviation	Dénomination	Bit
ER	Drapeau d'erreur d'exécution de l'instruction	25503
CY	Drapeau de retenue	25504
GR	Drapeau de supériorité	25505
EQ	Drapeau d'égalité	25506
LE	Drapeau d'infériorité	25507

ER est le drapeau le plus communément utilisé pour la surveillance de l'exécution d'une instruction. Lorsque ER passe à ON, cela indique qu'une erreur s'est produite pendant l'exécution de l'instruction en cours. Le chapitre drapeaux envisage les raisons possibles de cette erreur. ER passe à ON si les opérandes ne sont pas entrées correctement ; les instructions ne sont pas exécutées si ER est à ON.

Adressage indirect

Lorsque la zone DM est spécifiée pour une opérande, on peut utiliser une adresse indirecte. L'adressage DM indirect se spécifie en plaçant un astérisque avant "DM" : *DM.

Lorsque l'on spécifie une adresse DM indirecte, le mot DM désigné contient l'adresse du mot DM contenant lui-même les données à utiliser comme opérande de l'instruction. Si, par exemple, *DM 0001 est désigné comme première opérande et LR 00 comme deuxième opérande de MOV(21), si le contenu de DM 0001 est 1111 et si DM 1111 contient 5555, la valeur 5555 doit être déplacée vers LR 00.



Lorsque l'on utilise l'adressage indirect, l'adresse du mot désiré doit être en BCD et doit spécifier un mot appartenant à la zone DM. Dans l'exemple ci-dessus, le contenu de *DM 0000 doit être en BCD entre 0000 et 1999.

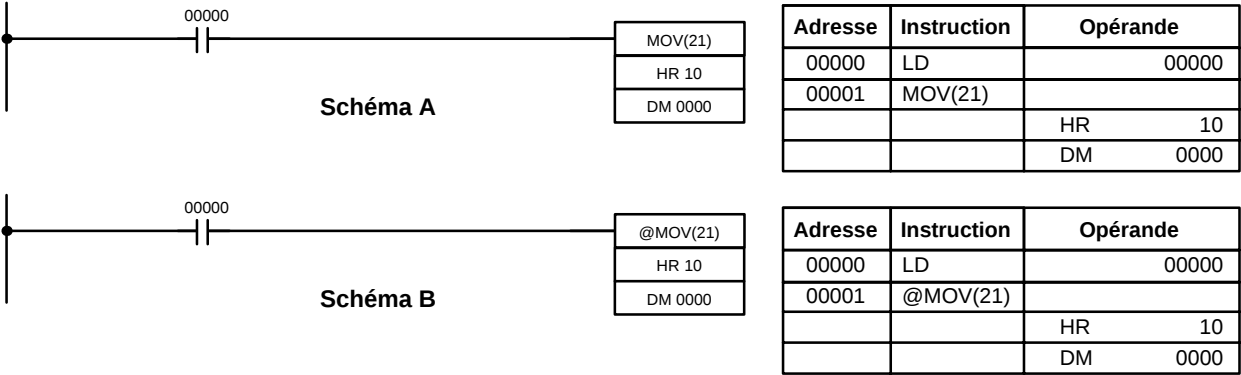
Désignation des constantes

Bien que les adresses des zones de données soient le plus souvent données sous forme d'opérandes, beaucoup d'opérandes et toutes les données d'opérandes sont entrées sous forme de constantes. La gamme de valeurs d'une donnée d'opérande ou d'une opérande donnée dépend de l'instruction particulière qui l'utilise. Les constantes doivent également être entrées sous la forme requise par l'instruction, c'est-à-dire en BCD ou en hexadécimal.

5-4 Instructions sur changement d'état

La plupart des instructions sont fournies sous leur forme sur changement d'état et sous leur forme simple. Les instructions sur changement d'état possèdent le signe @ figurant avant le code mnémonique de l'instruction.

Une instruction simple est exécutée à chaque scrutation et aussi longtemps que la condition d'exécution est à ON. Une instruction sur changement d'état n'est exécutée qu'une seule fois après que la condition d'exécution passe de OFF à ON. Si la condition n'a pas changé ou est passée de ON à OFF depuis la dernière scrutation, l'instruction n'est pas exécutée. Les deux exemples suivants montrent le fonctionnement de ce principe avec MOV(21) et @MOV(21), utilisés pour déplacer les données de l'adresse désignée par la première opérande vers l'adresse désignée par la deuxième opérande.



Adresse	Instruction	Opérande
00000	LD	00000
00001	@MOV(21)	
		HR 10
		DM 0000

Dans le schéma A, l'instruction simple MOV(21) déplace le contenu de HR 10 vers DM 0000 à chaque scrutation avec 00000. Si le temps de cycle est de 80 ms et que 00000 reste à ON pendant 2,0 secondes, ce déplacement est effectué 25 fois et seule la dernière valeur déplacée vers DM 0000 est conservée à cet endroit.

Dans le schéma B, l'instruction sur changement d'état @MOV(21) déplace le contenu de HR 10 vers DM 0000 une seule fois après le passage à ON de 00000. Même si 00000 reste à ON pendant 2,0 secondes avec un temps de cycle identique de 80 ms, le déplacement est effectuée une seule fois pendant le premier tour de scrutation qui a vu le passage de 00000 de OFF à ON. Le contenu de HR 10 pouvant très bien changer durant les 2 secondes pendant lesquelles 00000 est à ON, le contenu final du DM 0000 après 2 secondes peut être différent si l'on a utilisé MOV(21) ou @MOV(21).

Toutes les opérandes, tous les symboles des schémas à relais et autres caractéristiques des instructions sont identiques, qu'elles soient sous forme de changement d'état ou non. Lors de l'entrée, on utilise les mêmes codes de fonction mais on entre NOT après le code pour indiquer que l'instruction est sur changement d'état. La plupart des instructions (mais pas toutes) ont des formes sur changement d'état.

Reportez-vous au chapitre 5 *INTERLOCK* et *INTERLOCK CLEAR – IL(02) et IL(03)* pour de plus amples détails sur les branchements effectués avec des instructions sur changement d'état.

Le CQM1 possède ses instructions de changement d'état : DIFU(13) et DIFD(14). DIFU(13) fonctionne comme une instruction sur changement d'état mais s'utilise pour placer à ON un bit pour un tour de scrutation. DIFD(14) place également à ON un bit pour un tour de scrutation mais seulement lorsque la condition d'exécution est passée de ON à OFF. Reportez-vous au chapitre 5 *DIFFERENTIATE UP* et *DOWN - DIFU(13) et DIFD(14)* pour de plus amples détails.

5-5 Codage des instructions de droite

L'écriture du code mnémonique des instructions en schéma à relais est décrite dans le chapitre 4. La conversion des informations en schéma à relais suit le même principe pour toutes les instructions (cf ci-dessous) et n'est pas décrite individuellement pour chaque instruction.

Le premier canal d'une instruction définit l'instruction et fournit les données d'opérandes. Si l'instruction ne nécessite qu'une opérande de bit de signal sans donnée d'opérande, l'opérande de bit est également placée sur la même ligne que le mnémonique. Toutes les autres opérandes sont placées sur les lignes qui suivent la ligne d'instruction à raison d'une opérande par ligne et dans l'ordre de leur apparition sur le schéma à relais de l'instruction.

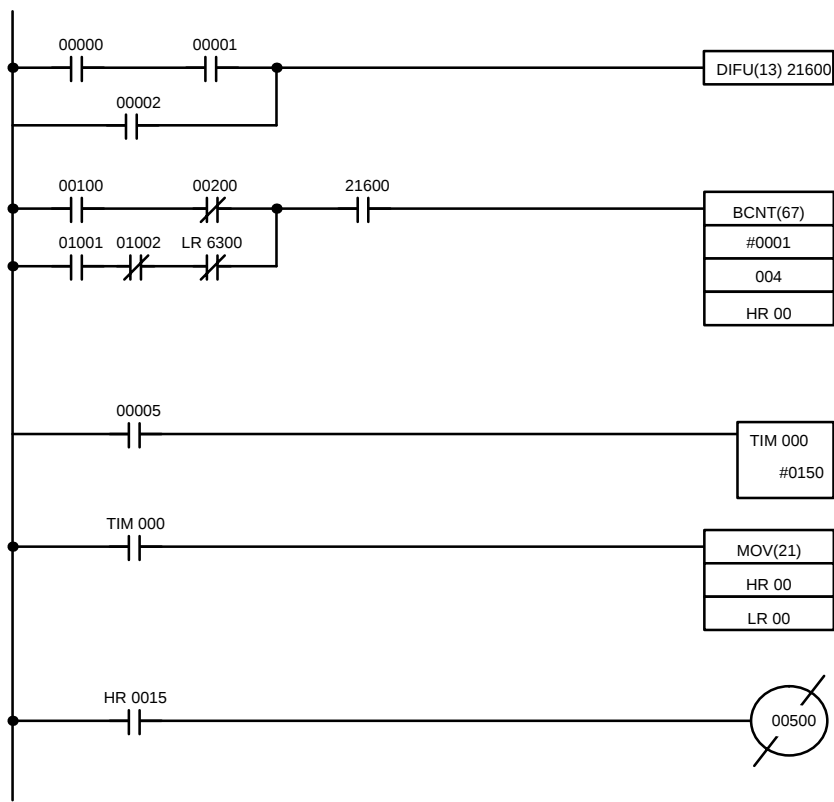
Les colonnes d'adresse et d'instruction du tableau de code mnémonique sont remplies pour le canal d'instruction seulement. Pour les autres lignes, les deux autres colonnes restent vides. Si l'instruction ne demande pas de donnée d'opérande ou d'opérande de bit, la colonne de données reste vide pour la première ligne. On peut faire des croix dans les colonnes vides (pour les instructions ne demandant pas de données) de façon à pouvoir rapidement repérer les adresses qu'on pourrait avoir oublié.

Si une adresse IR ou SR est utilisée dans la colonne de données, la partie gauche de la colonne reste vide. Si l'on utilise une autre zone de données, l'abréviation de la zone est placée à gauche et l'adresse est à droite. Si l'on doit entrer une constante, le symbole # (numéro) est placée à gauche de la colonne de données et le nombre à entrer est placé à droite. Tout nombre entré comme donnée d'opérande dans le canal d'instruction ne nécessite pas ce symbole sur la droite. Les bits TC, une fois définis comme temporisation ou compteur, prennent le préfixe TIM (temporisation) ou CNT (compteur).

Lors du codage d'une instruction ayant un code de fonction, assurez-vous de bien écrire le code de fonction qui sera nécessaire lors de l'entrée de l'instruction la console de programmation. Veillez à bien désigner les instructions sur changement d'état avec le symbole @.

Rem. : les codes mnémoniques des instructions étendues sont suivis de "(—)" comme code de fonction indiquant que l'utilisateur doit leur attribuer des codes de fonction dans le tableau des instructions avant de les utiliser en programmation.

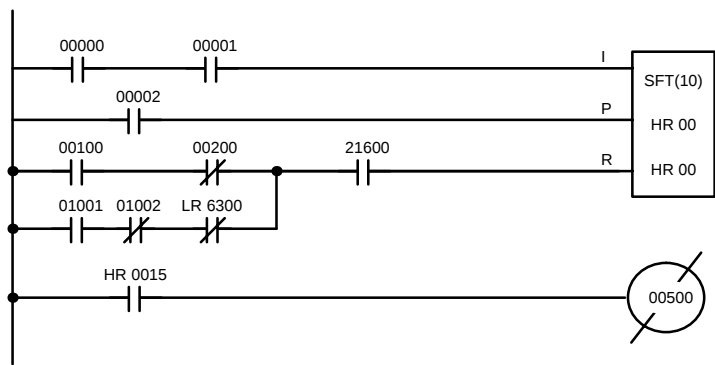
Voici une illustration de ce qui précède sous la forme d'un schéma et de codes mnémoniques correspondants :



Adresse	Instruction	Donnée
00000	LD	00000
00001	AND	00001
00002	OR	00002
00003	DIFU(13)	21600
00004	LD	00100
00005	AND NOT	00200
00006	LD	01001
00007	AND NOT	01002
00008	AND NOT	LR 6300
00009	OR LD	—
00010	AND	21600
00011	BCNT(67)	—
		# 0001
		004
		HR 00
00012	LD	00005
00013	TIM	000
		# 0150
00014	LD	TIM 000
00015	MOV(21)	—
		HR 00
		LR 00
00016	LD	HR 0015
00017	OUT NOT	00500

Lignes d'instructions multiples

Si une instruction de droite demande plusieurs lignes d'instruction (comme KEEP(11)), toutes ces lignes sont entrées avant l'instruction de droite. Chaque ligne est codée, en commençant par LD ou LD NOT, pour former des "blocs logiques" combinés par l'instruction de droite. Voici un exemple avec SFT(10) :



Adresse	Instruction	Donnée
00000	LD	00000
00001	AND	00001
00002	LD	00002
00003	LD	00100
00004	AND NOT	00200
00005	LD	01001
00006	AND NOT	01002
00007	AND NOT	LR 6300
00008	OR LD	—
00009	AND	21600
00010	SFT(10)	—
		HR 00
		HR 00
00011	LD	HR 0015
00012	OUT NOT	00500

END(01)

Une fois le codage du programme fini, assurez-vous que vous avez placé une instruction END(01) à la dernière adresse.

5-6 Tableaux d'instructions

Voici les tableaux d'instructions du CQM1. Le premier tableau peut être utilisé pour trouver une instruction grâce au code de fonction ; le second tableau sert à trouver une instruction par son code mnémotechnique. Dans les deux tableaux, le symbole @ indique les instructions sur changement d'état.

5-6-1 Codes fonction

Le tableau suivant donne une liste des instructions ayant des codes de fonction fixes. Chaque instruction figure par son code mnémotechnique et par son nom. Utilisez les nombres de la colonne d'extrême gauche comme chiffre de gauche et ceux de la colonne dite "Chiffre de droite" comme chiffre de droite du code de fonction.

Chiffre de gauche	Chiffre de droite									
	0	1	2	3	4	5	6	7	8	9
0	NOP NO OPERATION	END END	IL INTERLOCK	ILC INTERLOCK CLEAR	JMP JUMP	JME JUMP END	(@) FAL FAILURE ALARM AND RESET	FALS SEVERE FAILURE ALARM	STEP STEP DEFINE	SNXT STEP START
1	SFT SHIFT REGISTER	KEEP KEEP	CNTR REVERSIBLE COUNTER	DIFU DIFFERENTIATE UP	DIFD DIFFERENTIATE DOWN	TIMH HIGH-SPEED TIMER	(@) WSFT WORD SHIFT	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N
2	CMP COMPARE	(@) MOV MOVE	(@) MVN MOVE NOT	(@) BIN BCD TO BINARY	(@) BCD BINARY TO BCD	(@) ASL SHIFT LEFT	(@) ASR SHIFT RIGHT	(@) ROL ROTATE LEFT	(@) ROR ROTATE RIGHT	(@) COM COMPLEMENT
3	(@) ADD BCD ADD	(@) SUB BCD SUBTRACT	(@) MUL BCD MULTIPLY	(@) DIV BCD DIVIDE	(@) ANDW LOGICAL AND	(@) ORW LOGICAL OR	(@) XORW EXCLUSIVE OR	(@) XNRW EXCLUSIVE NOR	(@) INC INCREMENT	(@) DEC DECREMENT
4	(@) STC SET CARRY	(@) CLC CLEAR CARRY	---	---	---	TRSM TRACE MEMORY SAMPLE (SEE NOTE)	(@) MSG MESSAGE DISPLAY	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	---
5	(@) ADB BINARY ADD	(@) SBB BINARY SUBTRACT	(@) MLB BINARY MULTIPLY	(@) DVB BINARY DIVIDE	(@) ADDL DOUBLE BCD ADD	(@) SUBL DOUBLE BCD SUBTRACT	(@) MULL DOUBLE BCD MULTIPLY	(@) DIVL DOUBLE BCD DIVIDE	(@) BINL DOUBLE BCD-TO-DOUBLE BINARY	(@) BCDL DOUBLE BINARY-TO-DOUBLE BCD
6	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N
7	(@) XFER BLOCK TRANSFER	(@) BSET BLOCK SET	(@) ROOT SQUARE ROOT	(@) XCHG DATA EXCHANGE	(@) SLD ONE DIGIT SHIFT LEFT	(@) SRD ONE DIGIT SHIFT RIGHT	(@) MLPX 4-TO-16 DECODER	(@) DMPX 16-TO-4 ENCODER	(@) SDEC 7-SEGMENT DECODER	---
8	(@) DIST SINGLE WORD DISTRIBUTE	(@) COLL DATA COLLECT	(@) MOVb MOVE BIT	(@) MOVD MOVE DIGIT	(@) SFTR REVERSIBLE SHIFT REGISTER	(@) TCMP TABLE COMPARE	(@) ASC ASCII CONVERT	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N	EXPANSION INSTRUCT'N
9	---	(@) SBS SUBROUTINE ENTRY	SBN SUBROUTINE DEFINE	RET SUBROUTINE RETURN	---	---	---	(@) IORF I/O REFRESH	---	(@) MCRO MACRO

Rem. : TRSM(45) ne peut être utilisée avec les UC CQM1-CPU11/21-E.

5-6-2 Liste alphabétique des codes mnémoniques

Mnémonique	Code	Canaux	Nom	UC	Page
7SEG	—	4	7-SEGMENT DISPLAY OUTPUT	Toutes	295
ACC (@)	—	4	ACCELERATION CONTROL	CQM1-CPU43-E	285
ADB (@)	50	4	BINARY ADD	Toutes	240
ADBL (@)	—	4	DOUBLE BINARY ADD	CQM1-CPU4 [-E]	244
ADD (@)	30	4	BCD ADD	Toutes	229
ADDL (@)	54	4	DOUBLE BCD ADD	Toutes	234
AND	Aucun	1	AND	Toutes	143
AND LD	Aucun	1	AND LOAD	Toutes	144
AND NOT	Aucun	1	AND NOT	Toutes	143
ANDW (@)	34	4	LOGICAL AND	Toutes	261
APR (@)	—	4	ARITHMETIC PROCESS	Toutes	257
ASC (@)	86	4	ASCII CONVERT	Toutes	212
ASFT(@)	—	4	ASYNCHRONOUS SHIFT REGISTER	Toutes	178
ASL (@)	25	2	ARITHMETIC SHIFT LEFT	Toutes	173
ASR (@)	26	2	ARITHMETIC SHIFT RIGHT	Toutes	173
AVG	—	4	AVERAGE VALUE	Toutes	253
BCD (@)	24	3	BINARY TO BCD	Toutes	203
BCDL (@)	59	3	DOUBLE BINARY-TO-DOUBLE BCD	Toutes	204
BCMP (@)	—	4	BLOCK COMPARE	Toutes	193
BCNT (@)	—	4	BIT COUNTER	Toutes	272
BIN (@)	23	3	BCD-TO-BINARY	Toutes	202
BINL (@)	58	3	DOUBLE BCD-TO-DOUBLE BINARY	Toutes	204
BSET (@)	71	4	BLOCK SET	Toutes	182
CLC (@)	41	1	CLEAR CARRY	Toutes	229
CMP	20	3	COMPARE	Toutes	191
CMPL	—	4	DOUBLE COMPARE	Toutes	195
CNT	Aucun	2	COUNTER	Toutes	157
CNTR	12	3	REVERSIBLE COUNTER	Toutes	158
COLL (@)	81	4	DATA COLLECT	Toutes	185
COLM(@)	—	4	LINE TO COLUMN	Toutes	225
COM (@)	29	2	COMPLEMENT	Toutes	260
CPS	—	4	SIGNED BINARY COMPARE	CQM1-CPU4 [-E]	197
CPSL	—	4	DOUBLE SIGNED BINARY COMPARE	CQM1-CPU4 [-E]	199
CTBL(@)	—	4	COMPARISON TABLE LOAD	Toutes	162
DBS (@)	—	4	SIGNED BINARY DIVIDE	CQM1-CPU4 [-E]	249
DBSL (@)	—	4	DOUBLE SIGNED BINARY DIVIDE	CQM1-CPU4 [-E]	250
DEC (@)	39	2	BCD DECREMENT	Toutes	264
DIFD	14	2	DIFFERENTIATE DOWN	Toutes	147
DIFU	13	2	DIFFERENTIATE UP	Toutes	147
DIST (@)	80	4	SINGLE WORD DISTRIBUTE	Toutes	184

Mnémonique	Code	Canaux	Nom	UC	Page
DIV (@)	33	4	BCD DIVIDE	Toutes	233
DIVL (@)	57	4	DOUBLE BCD DIVIDE	Toutes	238
DMPX (@)	77	4	16-TO-4 ENCODER	Toutes	207
DSW	—	4	DIGITAL SWITCH	Toutes	296
DVB (@)	53	4	BINARY DIVIDE	Toutes	243
END	01	1	END	Toutes	148
FAL (@)	06	2	FAILURE ALARM AND RESET	Toutes	152
FALS	07	2	SEVERE FAILURE ALARM	Toutes	152
FCS (@)	—	4	FCS CALCULATE	Toutes	273
FPD	—	4	FAILURE POINT DETECT	Toutes	275
HEX (@)	—	4	ASCII-TO-HEXADECIMAL	Toutes	214
HKY	—	4	HEXADECIMAL KEY INPUT	Toutes	297
HMS	—	4	SECONDS TO HOURS	Toutes	223
IL	02	1	INTERLOCK	Toutes	149
ILC	03	1	INTERLOCK CLEAR	Toutes	149
INC (@)	38	2	INCREMENT	Toutes	264
INI (@)	—	4	MODE CONTROL	Toutes	167
INT (@)	—	4	INTERRUPT CONTROL	Toutes	279
IORF (@)	97	3	I/O REFRESH	Toutes	270
JME	05	2	JUMP END	Toutes	151
JMP	04	2	JUMP	Toutes	151
KEEP	11	2	KEEP	Toutes	146
LD	Aucun	1	LOAD	Toutes	143
LD NOT	Aucun	1	LOAD NOT	Toutes	143
LINE	—	4	LINE	Toutes	224
MAX (@)	—	4	FIND MAXIMUM	Toutes	251
MBS (@)	—	4	SIGNED BINARY MULTIPLY	CQM1-CPU4 [-E]	247
MBSL (@)	—	4	DOUBLE SIGNED BINARY MULTIPLY	CQM1-CPU4 [-E]	248
MCMP (@)	—	4	MULTI-WORD COMPARE	Toutes	196
MCRO (@)	99	4	MACRO	Toutes	271
MIN (@)	—	4	FIND MINIMUM	Toutes	252
MLB (@)	52	4	BINARY MULTIPLY	Toutes	243
MLPX (@)	76	4	4-TO-16 DECODER	Toutes	205
MOV (@)	21	3	MOVE	Toutes	180
MOVB (@)	82	4	MOVE BIT	Toutes	188
MOVD (@)	83	4	MOVE DIGIT	Toutes	189
MSG (@)	46	2	MESSAGE	Toutes	269
MUL (@)	32	4	BCD MULTIPLY	Toutes	232
MULL (@)	56	4	DOUBLE BCD MULTIPLY	Toutes	238
MVN (@)	22	3	MOVE NOT	Toutes	180
NEG (@)	—	4	2'S COMPLEMENT	CQM1-CPU4 [-E]	226

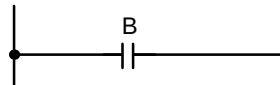
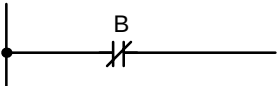
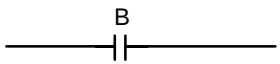
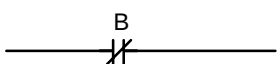
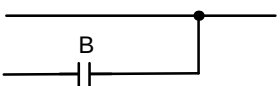
Mnémonique	Code	Canaux	Nom	UC	Page
NEGL (@)	—	4	DOUBLE 2'S COMPLEMENT	CQM1-CPU4 [-E]	227
NOP	00	1	NO OPERATION	Toutes	148
OR	Aucun	1	OR	Toutes	143
OR LD	Aucun	1	OR LOAD	Toutes	144
OR NOT	Aucun	1	OR NOT	Toutes	143
ORW (@)	35	4	LOGICAL OR	Toutes	262
OUT	Aucun	2	OUTPUT	Toutes	145
OUT NOT	Aucun	2	OUTPUT NOT	Toutes	145
PID	—	4	PID CONTROL	CQM1-CPU4 [-E]	290
PLS2 (@)	—	4	PULSE OUTPUT	CQM1-CPU43-E	284
PRV (@)	—	4	HIGH-SPEED COUNTER PV READ	Toutes	169
PULS (@)	—	4	SET PULSES	Toutes	280
PWM (@)	—	4	PULSE WITH VARIABLE DUTY RATIO	CQM1-CPU43-E	288
RET	93	1	SUBROUTINE RETURN	Toutes	267
ROL (@)	27	2	ROTATE LEFT	Toutes	174
ROOT (@)	72	3	SQUARE ROOT	Toutes	239
ROR (@)	28	2	ROTATE RIGHT	Toutes	174
RSET	Aucun	2	RESET	Toutes	239
RXD (@)	—	4	RECEIVE	Toutes	292
SBB (@)	51	4	BINARY SUBTRACT	Toutes	241
SBBL (@)	—	4	DOUBLE BINARY SUBTRACT	CQM1-CPU4 [-E]	245
SBN	92	2	SUBROUTINE DEFINE	Toutes	267
SBS (@)	91	2	SUBROUTINE ENTRY	Toutes	265
SCL (@)	—	4	SCALING	Toutes	216
SCL2 (@)	—	4	SIGNED BINARY TO BCD SCALING	CQM1-CPU4 [-E]	218
SCL3 (@)	—	4	BCD TO SIGNED BINARY SCALING	CQM1-CPU4 [-E]	220
SDEC (@)	78	4	7-SEGMENT DECODER	Toutes	209
SEC	—	4	HOURS TO SECONDS	Toutes	222
SET	Aucun	2	SET	Toutes	145
SFT	10	3	SHIFT REGISTER	Toutes	171
SFTR (@)	84	4	REVERSIBLE SHIFT REGISTER	Toutes	177
SLD (@)	74	3	ONE DIGIT SHIFT LEFT	Toutes	175
SNXT	09	2	STEP START	Toutes	153
SPED (@)	—	4	SPEED OUTPUT	Toutes	282
SRCH (@)	—	4	DATA SEARCH	Toutes	289
SRD (@)	75	3	ONE DIGIT SHIFT RIGHT	Toutes	176
STC (@)	40	1	SET CARRY	Toutes	229
STEP	08	2	STEP DEFINE	Toutes	153
STIM (@)	—	4	INTERVAL TIMER	Toutes	161
SUB (@)	31	4	BCD SUBTRACT	Toutes	230
SUBL (@)	55	4	DOUBLE BCD SUBTRACT	Toutes	236

Mnémonique	Code	Canaux	Nom	UC	Page
TCMP (@)	85	4	TABLE COMPARE	Toutes	192
TIM	Aucun	2	TIMER	Toutes	156
TIMH	15	3	HIGH-SPEED TIMER	Toutes	159
TKY (@)	—	4	TEN KEY INPUT	Toutes	298
TRSM	45	1	TRACE MEMORY SAMPLE	CQM1-CPU4 [-E]	267
TXD (@)	—	4	TRANSMIT	Toutes	293
WSFT (@)	16	3	WORD SHIFT	Toutes	172
XCHG (@)	73	3	DATA EXCHANGE	Toutes	183
XFER (@)	70	4	BLOCK TRANSFER	Toutes	181
XFRB (@)	—	4	TRANSFER BITS	CQM1-CPU4 [-E]	190
XNRW (@)	37	4	EXCLUSIVE NOR	Toutes	264
XORW (@)	36	4	EXCLUSIVE OR	Toutes	262
ZCP	—	4	AREA RANGE COMPARE	CQM1-CPU4 [-E]	200
ZCPL	—	4	DOUBLE AREA RANGE COMPARE	CQM1-CPU4 [-E]	201

5-7 Instructions en schéma à relais

Les instructions en schéma à relais comprennent des instructions par bloc logique et correspondent aux conditions du schéma. Les instructions par bloc logique s'utilisent pour relier entre elles certaines parties complexes du schéma.

5-7-1 LOAD, LOAD NOT, AND, AND NOT, OR et OR NOT

	Symbole	Zones de données d'opérande		
LOAD – LD		<table><tr><td>B: Bit</td></tr><tr><td>IR, SR, AR, HR, TC, LR, TR</td></tr></table>	B: Bit	IR, SR, AR, HR, TC, LR, TR
B: Bit				
IR, SR, AR, HR, TC, LR, TR				
LOAD NOT – LD NOT		<table><tr><td>B: Bit</td></tr><tr><td>IR, SR, AR, HR, TC, LR</td></tr></table>	B: Bit	IR, SR, AR, HR, TC, LR
B: Bit				
IR, SR, AR, HR, TC, LR				
AND – AND		<table><tr><td>B: Bit</td></tr><tr><td>IR, SR, AR, HR, TC, LR</td></tr></table>	B: Bit	IR, SR, AR, HR, TC, LR
B: Bit				
IR, SR, AR, HR, TC, LR				
AND NOT – AND NOT		<table><tr><td>B: Bit</td></tr><tr><td>IR, SR, AR, HR, TC, LR</td></tr></table>	B: Bit	IR, SR, AR, HR, TC, LR
B: Bit				
IR, SR, AR, HR, TC, LR				
OR – OR		<table><tr><td>B: Bit</td></tr><tr><td>IR, SR, AR, HR, TC, LR</td></tr></table>	B: Bit	IR, SR, AR, HR, TC, LR
B: Bit				
IR, SR, AR, HR, TC, LR				
OR NOT – OR NOT		<table><tr><td>B: Bit</td></tr><tr><td>IR, SR, AR, HR, TC, LR</td></tr></table>	B: Bit	IR, SR, AR, HR, TC, LR
B: Bit				
IR, SR, AR, HR, TC, LR				

Limites

Ces instructions ne sont pas limitées en nombre et peuvent être utilisées dans n'importe quel ordre, du moment que la capacité mémoire de l'API n'est pas dépassée.

Description

Ces six instructions de base correspondent aux conditions du schéma à relais. Comme il est dit dans le chapitre 4, l'état des bits attribués à chaque instruction

détermine la condition d'exécution de toutes les autres instructions ayant de l'équation logique. Chacune de ces instructions et chaque adresse de bit peut être utilisé autant de fois qu'on le désire.

L'état de l'opérande de bit (B) attribuée à LD ou LD NOT détermine la première condition d'exécution (départ de l'équation logique). AND établit un "et" logique entre la condition d'exécution et l'état de son opérande de bit ; AND NOT établit un "et" entre la condition d'exécution et l'inverse de l'état de son opérande de bit. OR établit un "ou" logique entre la condition d'exécution et l'état de son opérande de bit ; OR NOT établit un "ou" logique entre la condition d'exécution et l'inverse de l'état de son opérande de bit.

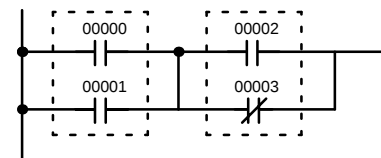
Drapeaux

Aucun drapeau n'est affecté par ces instructions.

5-7-2 AND LOAD et OR LOAD

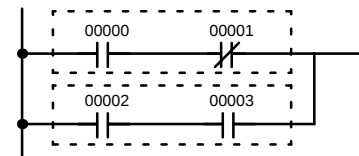
AND LOAD – AND LD

Symbole



OR LOAD – OR LD

Symbole



Description

Lorsque certaines instructions sont combinées dans des blocs qui ne peuvent être associés logiquement à l'aide de OR et AND uniquement, on utilise AND LD et OR LD. AND et OR combinent logiquement l'état d'un bit et une condition d'exécution alors que AND LD et OR LD combinent logiquement deux conditions d'exécution : celle en cours et la dernière non utilisée.

Pour dessiner des schémas à relais, il n'est pas nécessaire d'utiliser AND LD et OR LD (ni pour entrer un schéma directement, comme on le fait avec le LSS). Ces instructions sont toutefois indispensables pour convertir le programme et l'entrer en mnémonique.

Pour réduire le nombre d'instructions de programmation nécessaires, il faut avoir quelques notions de ce que sont les instructions à bloc logique. Reportez-vous au chapitre 4-3-6.

Drapeaux

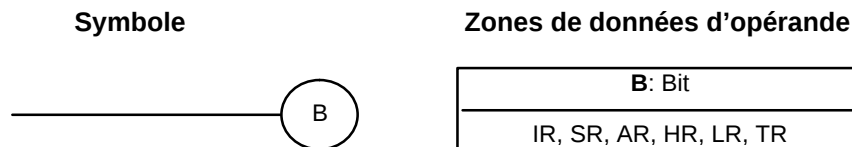
Aucun drapeau n'est affecté par ces instructions.

5-8 Instructions de contrôle de bit

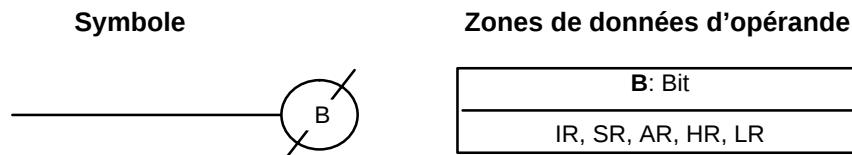
Il existe sept instructions qui peuvent généralement être utilisées pour contrôler l'état individuel des bits : OUT, OUT NOT, DIFU(13), DIFD(14), SET, RSET et KEEP(11). Ces instructions s'utilisent pour placer les bits à ON ou OFF de différentes manières.

5-8-1 OUTPUT et OUTPUT NOT – OUT et OUT NOT

OUTPUT – OUT



OUTPUT NOT – OUT NOT



Limites

Tout bit de sortie peut généralement être utilisé dans une seule instruction contrôlant son état.

Description

OUT et OUT NOT s'utilisent pour contrôler l'état du bit désigné selon la condition d'exécution.

OUT place à ON le bit désigné en cas de condition à ON et place le bit à OFF pour une condition à OFF. Avec un bit TR, OUT apparaît à un point d'embranchement plutôt qu'à la fin d'une ligne d'instruction. Reportez-vous au chapitre 4–3–8 pour de plus amples détails.

OUT NOT place à ON le bit désigné en cas de condition à OFF et place le bit à OFF en cas de condition à ON.

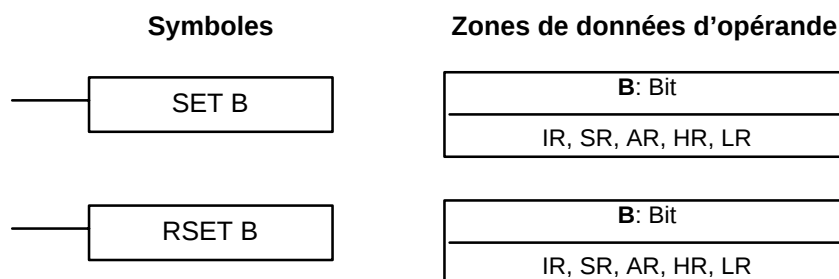
OUT et OUT NOT peuvent être utilisées pour contrôler l'exécution en plaçant à ON ou OFF les bits attribués aux conditions du schéma. Les conditions d'autres instructions peuvent ainsi être déterminées. Cela est particulièrement utile pour le contrôle de l'état d'un seul bit de travail et permet d'utiliser un ensemble de conditions complexes ; ce bit de travail peut ensuite être utilisé pour contrôler d'autres instructions.

Le temps pendant lequel un bit reste à ON ou OFF peut être contrôlé en combinant OUT ou OUT NOT avec TIM. Reportez-vous aux exemples du chapitre 5–15–1 pour de plus amples détails.

Drapeaux

Aucun drapeau n'est affecté par ces instructions.

5-8-2 SET et RESET – SET et RSET



Description

SET place le bit d'opérande à ON lorsque la condition d'exécution est à ON et y reste lorsque la condition est à OFF. RSET place le bit d'opérande à OFF lorsque la condition d'exécution est à ON et n'affecte pas l'état du bit d'opérande lorsque la condition d'exécution est à OFF.

SET ne fonctionne pas comme OUT car celle-ci place le bit d'opérande à OFF lorsque sa condition d'exécution est à OFF. De la même façon, RSET ne fonc-

tionne pas comme OUT NOT car OUT NOT place le bit d'opérande à ON lorsque sa condition est à OFF.

Les instructions SET et RESET sont les verrouillages/déverrouillages d'un bistable pouvant être programmé n'importe où dans le programme et non pas comme l'instruction KEEP qui demande une programmation structurée : VER. + DEV. + KEEP. En fait, ces 2 instructions sont un KEEP éclaté.

Précautions

L'état des bits d'opérande de SET et RSET programmées entre IL(002) et ILC(003) ou entre JMP(004) et JME(005) ne change pas en cas de condition de branchement ou de saut (lorsque IL(002) ou JMP(004) sont exécutées avec une condition à OFF).

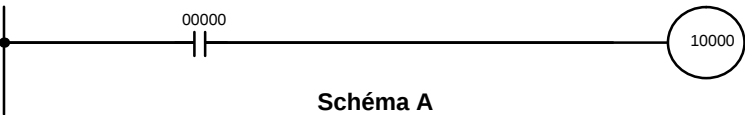
Drapeaux

Aucun drapeau n'est affecté par ces instructions.

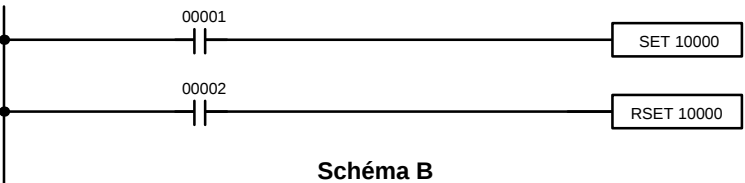
Exemples

Les exemples suivants illustrent la différence entre OUT et SET/RSET. Dans le premier exemple (schéma A), IR 10000 est place à ON ou OFF quand IR 00000 passe à ON ou à OFF.

Dans le deuxième exemple (schéma B), IR 10000 est placé à ON lorsque IR 00001 passe à ON et reste à ON (même si IR 00001 passe à OFF) jusqu'à ce que IR 00002 passe à ON.

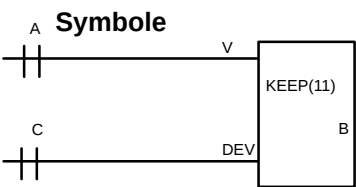


Adresse	Instruction	Opérande
00000	LD	00000
00001	OUT	10000



Adresse	Instruction	Opérande
00000	LD	00001
00001	SET	10000
00002	LD	00002
00003	RSET	10000

5-8-3 KEEP – KEEP(11)



Zones de données d'opérande

B: Bit
IR, SR, AR, HR, LR

Limites

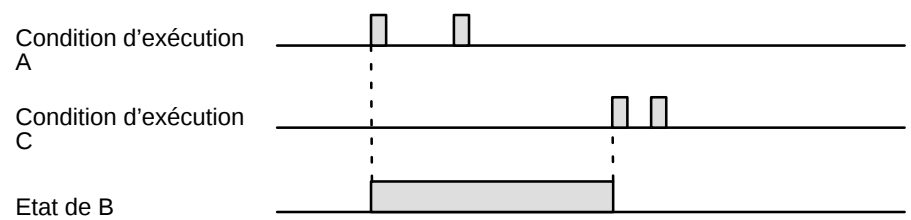
Tout bit de sortie peut généralement être utilisé dans une seule et unique instruction contrôlant son état.

Description

KEEP(11) est utilisée pour maintenir l'état du bit désigné provenant de deux conditions d'exécution. Ces conditions sont appelées V et DEV. V est l'entrée de sélection et DEV celle de remise à zéro. KEEP(11) fonctionne comme relais bistable sélectionné par V et remis à zéro par DEV.

Lorsque A passe à ON, le bit désigné passe à ON et y reste jusqu'à remise à zéro, quel que soit l'état ON ou OFF de V. Lorsque C passe à ON, le bit désigné B passe à OFF et y reste jusqu'à remise à zéro, quel que soit l'état ON ou OFF de C. Quand le verrouillage est ON et le déverrouillage ON, KEEP est à OFF : cela

signifie que le déverrouillage est prioritaire sur le verrouillage du KEEP.
Voici la relation entre les conditions d'exécution et l'état du bit de KEEP(11) :

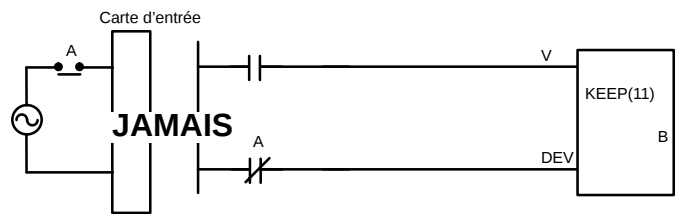


Drapeaux

Aucun drapeau n'est affecté par cette instruction.

Précautions

Faites attention lorsque vous utilisez une ligne de remise à zéro KEEP contrôlée par un appareil externe normalement fermé. N'utilisez jamais de bit d'entrée dans une condition inverse sur une remise à zéro (DEV) de KEEP(11) lorsque l'appareil d'entrée utilise une alimentation c.a. Le retard à la coupure d'alimentation c.c. de l'API (dû à l'alimentation c.a. de l'appareil d'entrée) peut causer la remise à zéro du bit désigné de KEEP(11) :



Les bits utilisés dans KEEP ne sont pas remis à zéro dans les branchements. Reportez-vous au chapitre 5–11 *INTERLOCK – et INTERLOCK CLEAR IL(02) et ILC(03)* pour de plus amples détails.

5-8-4 DIFFERENTIATE UP et DOWN – DIFU(13) et DIFD(14)

Symboles	Zones de données d'opérande
DIFU(13) B Front montant	B: Bit IR, SR, AR, HR, LR
DIFD(14) B Front descendant	B: Bit IR, SR, AR, HR, LR

Limites

Tout bit de sortie peut généralement être utilisé dans une seule et unique instruction contrôlant son état.

Description

DIFU(13) et DIFD(14) sont utilisées pour placer à ON le bit désigné pour un seul tour de scrutation.

A chaque exécution de DIFU(13), celle-ci compare son exécution en cours avec la précédente. Si la précédente était à OFF et que celle en cours est à ON, DIFU(13) place à ON le bit désigné. Si la condition d'exécution précédente était à ON et que la condition en cours est à ON ou OFF, DIFU(13) place le bit désigné à OFF ou le laisse à OFF selon le cas (c'est-à-dire s'il est déjà à OFF). Le bit désigné ne sera donc jamais à ON pendant plus d'un tour de scrutation si l'on part du principe qu'il est exécuté à chaque tour (cf *Précautions* ci-dessous).

A chaque exécution de DIFD(14), celle-ci compare son exécution en cours avec la précédente. Si la précédente était à ON et que celle en cours est à OFF, DIFD(14) place à ON le bit désigné. Si la condition d'exécution précédente était à OFF et que la condition en cours est à ON ou OFF, DIFD(14) place le bit désigné à OFF ou le laisse à OFF selon le cas. Le bit désigné ne sera donc jamais à ON pendant plus d'un tour de scrutation si l'on part du principe qu'il est exécuté à chaque tour (cf *Précautions* ci-dessous).

Ces instructions sont utilisées lorsque les instructions sur changement d'état (précédées de "@") ne sont pas disponibles et que l'exécution en un seul tour de scrutation d'une instruction particulière est souhaitée. On peut également les utiliser avec des instructions qui ne sont sur changement d'état que lorsqu'elles sont utilisées pour simplifier la programmation. En voici un exemple ci-dessous.

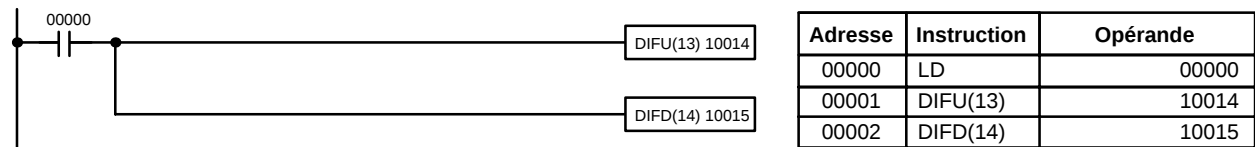
Drapeaux

Aucun drapeau n'est affecté par ces instructions.

Précautions

DIFU(13) et DIFD(14) peuvent ne pas fonctionner de manière très précise lorsqu'on les programme entre IL et ILC ou entre JMP et JME, ou dans des sous-programmes. Reportez-vous aux chapitres *AUCUN LIEN INTERLOCK* et *INTERLOCK CLEAR – IL(02) et ILC(03)*, *AUCUN LIEN JUMP* et *JUMP END – JMP(04) et JME(05)*, 5-25 et 5-26-8 *INTERRUPT CONTROL – INT(—)*.

Exemple : dans cet exemple, IR 10014 passe à ON (au front montant) pendant un tour de scrutation lorsque IR 00000 passe de OFF à ON. IR 10015 passe à ON (au front descendant) pendant un tour de scrutation lorsque IR 00000 passe de ON à OFF.



5-9 NO OPERATION – NOP(00)

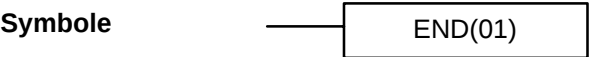
Description

NOP(00) n'est généralement pas nécessaire en programmation et il n'existe pas de symbole pour cette instruction. La présence de NOP(00) dans un programme n'entraîne aucune exécution et le programme passe à l'instruction suivante. Lorsque la mémoire est effacée avant programmation, NOP(00) s'inscrit dans toutes les adresses. NOP(00) peut être entrée par le code de fonction 00.

Drapeaux

Aucun drapeau n'est affecté par NOP(00).

5-10 END – END(01)



Description

END(01) est indispensable comme dernière instruction du programme principal. S'il y a des sous-programmes, END(01) est placée après le dernier. Aucune instruction écrite après END(01) n'est exécutée. END(01) peut être placée à tout endroit du programme pour exécuter toutes les instructions jusqu'à ce moment, comme on le fait parfois pour la mise au point du programme, mais elle doit être supprimée pour exécuter le reste du programme. Il peut y avoir plusieurs END dans un même programme pour aider à la mise au point d'une machine.

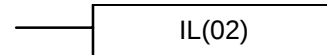
S'il n'y a pas d'instruction END(01) dans le programme, aucune instruction n'est exécutée et le message "NO END INST" apparaît.

Drapeaux

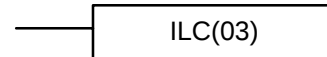
END(01) place à OFF les drapeaux ER, CY, GR, EQ et LE.

5-11 INTERLOCK et INTERLOCK CLEAR – IL(02) et ILC(03)

Symbole



Symbole



Description

IL(02) est toujours utilisée avec ILC(03) pour créer des branchements. Les branchements activent les embranchements de la même façon que les bits TR, mais le traitement des instructions entre IL(02) et ILC(03) diffère lorsque la condition d'exécution de IL(02) est à OFF. Si la condition d'exécution de IL(02) est à ON, le programme est exécuté comme il a été écrit avec une condition à ON pour commencer chaque ligne d'instruction à partir du point où IL(02) est située et jusqu'à l'instruction ILC(03) suivante.

Si la condition d'exécution de IL(02) est à OFF, la partie de l'embranchement comprise entre IL(02) et ILC(03) sera traitée comme dans le tableau suivant :

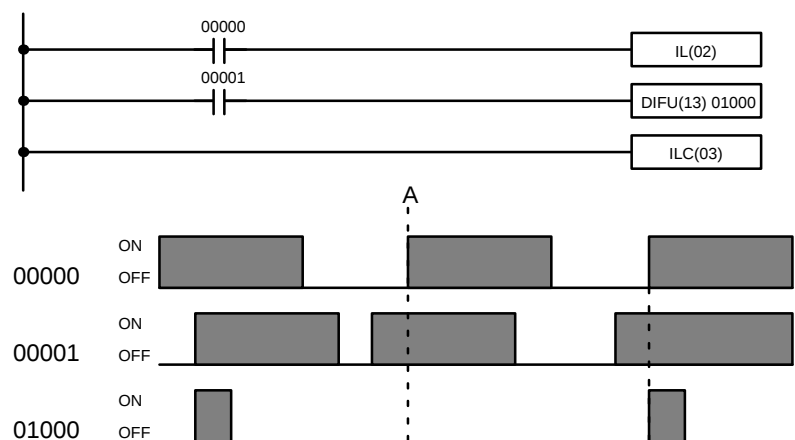
Instruction	Traitement
OUT et OUT NOT	Désignées par le bit placé à OFF
TIM et TIMH(15)	Remise à zéro
CNT, CNTR(12)	Valeur en cours maintenue
KEEP(11)	Etat du bit maintenu
DIFU(13) et DIFD(14)	Non exécutées (cf ci-dessous)
Autres instructions	Les instructions ne sont pas exécutées et tous les bits et canaux IR, AR, LR, HR et SR bits écrits comme opérandes dans les instructions sont placés à OFF.

IL(02) et ILC(03) ne doivent pas obligatoirement être utilisées en paires. IL(02) peut être utilisée plusieurs fois sur la même rangée, chaque IL(02) créant une partie en branchement jusqu'à l'instruction ILC(03) suivante. ILC(03) ne peut être utilisée s'il n'y a pas au moins une instruction IL(02) entre elle et l'ILC(03) précédente.

DIFU(13) et DIFD(14) en branchement

Les changements dans la condition d'exécution de DIFU(13) ou DIFD(14) ne sont pas enregistrés si l'instruction est dans une partie de programme en branchement et si la condition d'exécution de IL(02) est à OFF. Lorsque DIFU(13) ou DIFD(14) est exécutée dans un branchement tout de suite après le passage à ON de la condition de IL(02), la condition de DIFU(13) ou de DIFD(14) est comparée à la condition existant avant que le branchement ne devienne effectif (c'est-à-dire avant que la condition de IL(02) ne passe à OFF). Le schéma à relais et l'état du bit changent de la façon décrite ci-après. Le

branchement est effectif pendant que 00000 est à OFF. Remarquez que 01000 ne passe pas à ON au point "A" bien que 00001 soit passé à OFF puis à ON.



Adresse	Instruction	Opérande
00000	LD	00000
00001	IL(02)	
00002	LD	00001
00003	DIFU(13)	01000
00004	ILC(03)	

Précautions

Il doit y avoir une instruction ILC(03) après toute instruction IL(02). IL et ILC fonctionnent par paire.

Bien que l'on puisse utiliser autant d'instructions IL(02) qu'on le souhaite avec une seule ILC(03), les instruction ILC(03) ne s'utilisent pas à la suite sans au moins une instruction IL(02) entre elles : les emboîtements ne sont donc pas possibles. A chaque exécution d'une instruction ILC(03), tous les branchements situés entre l'instruction ILC(03) active et la précédente ILC(03) sont effacés.

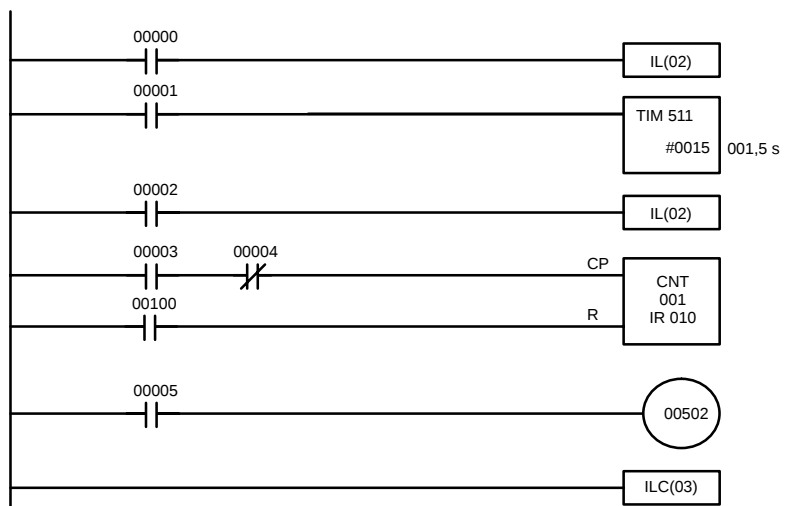
Lorsque l'on utilise plusieurs IL(02) avec une seule instruction ILC(03), un message d'erreur apparaît pendant la vérification du programme mais l'exécution s'effectue normalement.

Drapeaux

Aucun drapeau n'est affecté par ces instructions.

Exemple

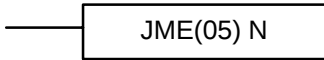
Le schéma suivant illustre l'utilisation de deux IL(02) avec une seule ILC(03) :



Adresse	Instruction	Opérande
00000	LD	00000
00001	IL(02)	
00002	LD	00001
00003	TIM	511
		# 0015
00004	LD	00002
00005	IL(02)	
00006	LD	00003
00007	AND NOT	00004
00008	LD	00100
00009	CNT	001
		010
00010	LD	00005
00011	OUT	00502
00012	ILC(03)	

Lorsque la condition d'exécution de la première IL(02) est à OFF, TIM 511 est désactivée à 1,5 s, CNT 001 reste inchangée et 00502 passe à OFF. Lorsque la condition de la première IL(02) est à ON et celle de la deuxième à IL(02) à OFF, TIM 511 est exécutée selon l'état de 00001, CNT 001 reste inchangée et 00502 passe à OFF. Lorsque la condition d'exécution des deux IL(02) sont à ON, la programme est exécuté comme il est écrit.

5-12 JUMP et JUMP END – JMP(04) et JME(05)

Symboles	Valeurs des données d'opérande		
	<table><tr><td>N: numéro de saut</td></tr><tr><td># (00 à 99)</td></tr></table>	N: numéro de saut	# (00 à 99)
N: numéro de saut			
# (00 à 99)			
	<table><tr><td>N: numéro de saut</td></tr><tr><td># (00 à 99)</td></tr></table>	N: numéro de saut	# (00 à 99)
N: numéro de saut			
# (00 à 99)			

Limites

Les numéros de saut 01 à 99 ne peuvent être utilisés qu'une seule fois dans JMP(04) et une seule fois également dans JME(05), c'est-à-dire que chacun de ces numéros ne peut être utilisé que pour définir un unique saut. Le numéro de saut 00 peut être utilisé autant de fois qu'on le désire.

Description

JMP(04) est toujours utilisé avec JME(05) pour créer des sauts, c'est-à-dire pour sauter d'un point à un autre dans le schéma. JMP(04) définit le point à partir duquel le saut doit être effectué et JME(05) définit la destination du saut. Lorsque la condition d'exécution de JMP(04) est à ON, aucun saut n'est effectué et le programme est ensuite exécuté comme il est écrit. Lorsque la condition de JMP(04) is OFF, un saut est exécuté vers JME(05) avec le même numéro de saut et l'instruction qui suit JME(05) est ensuite exécutée.

Si le numéro de saut de JMP(04) est situé entre 01 et 99, les sauts passent immédiatement à JME(05) lorsqu'ils sont exécutés, avec le même numéro et sans exécuter les instructions situées entre ces deux instructions. L'état des temporisations, compteurs et bits utilisés dans OUT ainsi que des bits utilisés dans OUT NOT et tous les autres bits d'état contrôlés par les instructions situées entre JMP(04) et JMP(05) restent inchangés. Cette instruction est importante car on peut l'utiliser pour figer un morceau du programme lors d'un arrêt d'urgence par exemple. Chaque numéro de saut ne peut être utilisé que pour définir un seul saut. Toutes les instructions entre JMP(04) et JME(05) étant sautées, les numéros de saut entre 01 et 99 peuvent être utilisés pour réduire le temps de scrutation.

Si le numéro de saut de JMP(04) est 00, L'UC cherche l'instruction JME(05) suivante ayant le numéro 00. Pour cela, l'UC inspecte le programme et le temps de scrutation s'en trouve allongé (lorsque la condition d'exécution est à OFF) par rapport aux autres sauts. L'état des temporisations, compteurs, bits utilisés dans OUT et dans OUT NOT et de tous les autres états contrôlés par les instructions situées entre JMP(04) 00 et JMP(05) 00 reste inchangé. Le numéro de saut 00 peut être utilisé autant de fois qu'on le souhaite. Un saut à partir de JMP(04) 00 passe toujours au JME(05) 00 suivant du programme ; il est donc possible d'utiliser plusieurs JMP(04) 00 à la suite et de les associer à un seul JME(05) 00. Il serait absurde d'utiliser plusieurs JME(05) 00 à la suite car tous les sauts se terminent de toute façon au premier JME(05) 00.

DIFU(13) et DIFD(14) dans les sauts

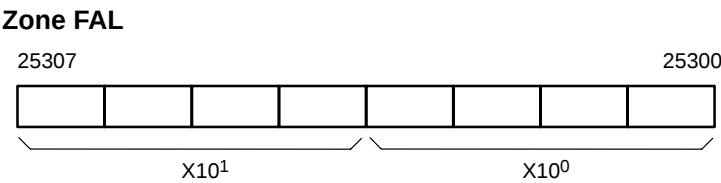
Bien que DIFU(13) et DIFD(14) soient conçues pour placer à ON le bit désigné pendant un tour de scrutation, cela n'est pas toujours le cas lorsqu'il est situé entre JMP(04) et JMP (05). Une fois que DIFU(13) ou DIFD(14) ont placé un bit à ON, il y reste jusqu'à la prochaine exécution de DIFU(13) ou DIFD(14) ; en programmation normale, cela signifie jusqu'au prochain tour de scrutation. Dans un saut, cela implique que la fois suivante où un saut de JMP(04) à JME(05) n'est pas effectué (c'est-à-dire si un bit est placé à ON par DIFU(13) ou DIFD(14) et si un saut est exécuté dans le tour de scrutation suivant pour que DIFU(13) ou DIFD(14) soient sautées), le bit désigné reste à ON jusqu'à l'occurrence sui-

	vante d'une condition d'exécution de l'instruction JMP(04) contrôlant le saut qui serait placée à ON.
Précautions	Lorsque JMP(04) et JME(05) ne sont pas utilisées en paire, un message d'erreur apparaît lors de la vérification du programme. Bien que ce message apparaisse également si JMP(04) 00 et JME(05) 00 ne sont pas utilisées en paire, le programme sera exécuté correctement, selon ce qui a été écrit.
Drapeaux	Aucun drapeau n'est affecté par ces instructions.
Exemples	Voir le chapitre <i>AUCUN LIEN</i> .

5-13 Instructions d'erreur utilisateur :
FAILURE ALARM AND RESET – FAL(06) et
SEVERE FAILURE ALARM – FALS(07)

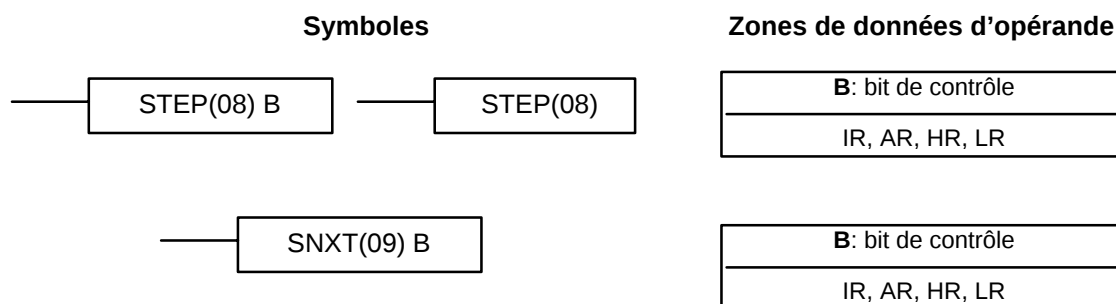
Symboles	Zones de données d'opérande
FAL(06) N @FAL(06) N	N: numéro de FAL # (00 à 99) N: numéro de FAL # (01 à 99)
FALS(07) N	

Description	FAL(06) et FALS(07) existent pour que le programmeur puisse sortir des numéros d'erreur pour le fonctionnement, la maintenance et la mise au point. Lors d'une exécution avec un condition à ON, l'une des ces instructions sort un numéro de FAL sur les bits 00 à 07 de SR 253. Le numéro de FAL sorti peut être situé entre 01 et 99 et est entré comme donnée d'opérande de FAL(06) ou FALS(07). FAL(06) avec une donnée d'opérande de 00 est utilisée pour remettre à zéro cette zone (cf ci-dessous) :
-------------	--



	FAL(06) produit une erreur non fatale et FALS(07) une erreur fatale. Lorsque FAL(06) est exécutée avec une condition d'exécution à ON, le voyant ALARM/ERROR à l'avant de l'UC clignote mais le fonctionnement de l'API continue. Lorsque FALS(07) est exécutée avec une condition à ON, le voyant ALARM/ERROR s'allume et le fonctionnement de l'API s'arrête.
	Le système génère également des codes d'erreur dans la zone FAL.
Remise à zéro des erreurs	Un maximum de trois codes d'erreur FAL sont retenus en mémoire, bien qu'un seul de ceux-ci soit disponible dans la zone FAL. Pour accéder aux autres codes FAL, remettez à zéro la zone FAL en exécutant FAL(06) 00. A chaque exécution d'une instruction FAL(06) 00, une autre erreur FAL est déplacée vers la zone FAL, effaçant la précédente. FAL(06) 00 s'utilise également pour effacer un message programmé avec une instruction, MSG(46). Si la zone FAL ne peut être effacée, comme c'est généralement le cas lorsque FALS(07) est exécutée, il faut d'abord remédier à la cause de l'erreur et effacer la zone FAL par la console de programmation ou le Sysmate LSS.

5-14 Instructions d'étape : STEP DEFINE et STEP START-STEP(08)/SNXT(09)



Limites

Tous les bits de contrôle doivent appartenir au même canal et doivent être consécutifs.

Description

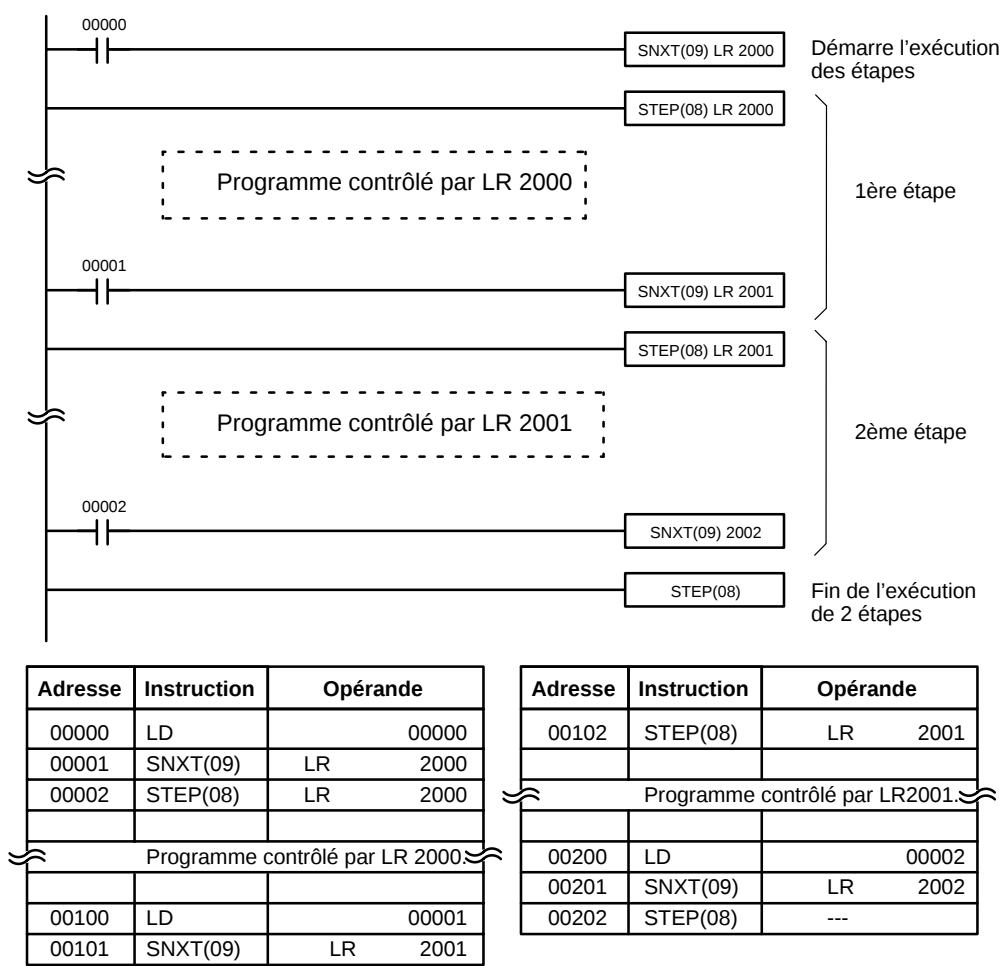
Les instructions STEP(08) et SNXT(09) sont utilisées ensemble pour configurer les arrêts de contrôle entre les différentes parties d'un grand programme de façon à ce que ces parties puissent être exécutées en tant qu'unités distinctes et remises à zéro à la fin. Une partie de programme est généralement définie pour correspondre à un processus réel d'application (cf exemples d'application ci-après). Une étape est comparable à un code de programmation normal, à ceci près que certaines instructions (END(01), IL(02)/ILC(03), JMP(04)/JME(05) et SBN(92)) ne doivent pas y figurer.

STEP(08) utilise un bit de contrôle dans les zones IR ou HR pour définir le début d'une partie de programme appelée étape. STEP(08) ne nécessite pas de condition d'exécution, c'est-à-dire que l'exécution est contrôlée par le bit de contrôle. Pour commencer l'exécution de l'étape, SNXT(09) est utilisée avec le même bit de contrôle que celui de STEP(08). Si SNXT(09) est exécutée avec une condition d'exécution à ON, l'étape contenant ce même bit de contrôle est exécutée. Si la condition d'exécution est à OFF, l'étape n'est pas exécutée. L'instruction SNXT(09) doit être écrite dans le programme de façon à ce qu'elle soit exécutée avant que le programme n'arrive à l'étape. Elle peut être utilisée en différents endroits avant l'étape pour la contrôler selon deux conditions d'exécution différentes (cf exemple 2 ci-dessous). Toute étape du programme qui ne commencerait pas avec SNXT(09) ne sera pas exécutée.

Une fois que l'on utilise SNXT(09) dans le programme, l'exécution de l'étape continue jusqu'à ce que STEP(08) soit exécutée sans bit de contrôle. STEP(08) sans bit de contrôle doit être précédée de SNXT(09) avec bit de contrôle factice. Le bit de contrôle factice peut être n'importe quel bit IR ou HR non utilisé. Ce peut être un bit utilisé dans l'une des instructions STEP(08).

L'exécution d'une étape se termine par l'exécution de l'instruction SNXT(09) suivante ou par le passage à OFF du bit de contrôle de l'étape (cf exemple 3 ci-dessous). Lorsque l'étape est terminée, tous les bits IR et HR de l'étape sont placés à OFF et toutes les temporisations de l'étape sont remplacées à leur valeur

sélectionnée. Les compteurs, enregistrements de décalage et bits utilisés dans KEEP(11) maintiennent leur état. Voici deux étapes simples :



Les étapes peuvent être programmées à la suite ; chaque étape commence par STEP(08) et se termine généralement par SNXT(09) (cf exemple 3, ci-dessous, comme exception). Lorsque les étapes sont programmées en série, trois types d'exécution sont possibles : à la suite, en branchement ou en parallèle. Les conditions d'exécution et le positionnement de SNXT(09) déterminent l'exécution des étapes. Les exemples ci-dessous illustrent les trois types d'exécution :

Précautions

Les branchements, sauts et instructions SBN(92) et END(01) ne peuvent être utilisées dans les programmes d'étape.

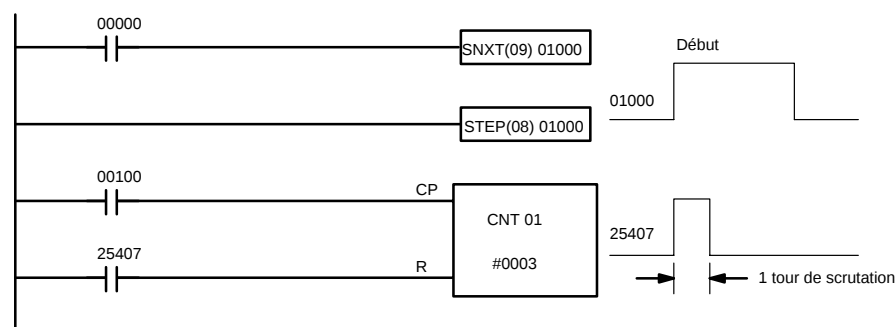
Les bits utilisés comme bits de contrôle ne doivent pas être utilisés ailleurs dans le programme, sauf pour contrôler le fonctionnement de l'étape (cf exemple 3, ci-dessous). Tous les bits de contrôle doivent appartenir au même canal et doivent être consécutifs.

Si les bits IR ou LR sont utilisés comme bits de contrôle, leur état est perdu en cas de coupure de courant. S'il faut en maintenir l'état pour reprendre l'exécution à la même étape, il convient d'utiliser HR.

Drapeaux

25407, drapeau de début d'étape : passe à ON pendant un tour de scrutation lorsque STEP(08) est exécutée et peut être utilisé par exemple pour

remettre à zéro les compteurs dans les étapes si nécessaire comme ci-dessous (le drapeau ne fonctionne qu'entre un STEP et un SNXT) :



Adresse	Instruction	Opérande
00000	LD	00000
00001	SNXT(09)	01000
00002	STEP(08)	01000
00003	LD	00100

Adresse	Instruction	Opérande
00004	LD	25407
00005	CNT	01
		# 0003

5-15 Instructions de temporisation/compteur

TIM et TIMH(15) sont des instructions de temporisation de retard au travail en décrémentation nécessitant un numéro de TC et une valeur sélectionnée (SV). STIM(—) s'utilise pour contrôler les tempo. cycliques qui servent à activer les sous-programmes d'interruption.

CNT est une instruction de compteur en décrémentation et CNTR est une instruction de compteur réversible. Toutes les deux nécessitent un numéro de TC et une SV ; elles sont connectées aux lignes d'instructions multiples et servent de signal d'entrée et de RAZ. CTBL(—), INT(—) et PRV(—) sont utilisées pour gérer le compteur rapide. INT(—) est également utilisée pour arrêter la sortie par impulsion.

Tout numéro de TC ne peut être défini deux fois, c'est-à-dire qu'une fois qu'il a été utilisé comme zone d'opérande dans une instruction de tempo. ou compteur, il ne peut être ré-utilisé. Une fois définis, les numéros de TC peuvent être utilisés autant de fois qu'on le souhaite comme opérande dans les instructions autres que tempo. ou compteur.

Les numéros de TC vont de 000 à 511. Aucun préfixe n'est nécessaire lorsque l'on utilise un numéro de TC comme donnée d'opérande dans une instruction de tempo. ou compteur. Une fois défini comme temporisation, un numéro de TC peut porter le préfixe TIM pour être utilisé comme opérande dans certaines instructions. Le préfixe TIM s'utilise sans tenir compte de l'instruction de temporisation utilisée pour définir la temporisation. Une fois défini comme compteur, un numéro de TC peut porter le préfixe CNT pour être utilisé comme opérande dans certaines instructions. CNT s'utilise également sans tenir compte de l'instruction compteur utilisée pour définir le compteur.

Les numéros de TC peuvent être désignés comme opérandes nécessitant une donnée de bit ou de canal. En cas de désignation d'une opérande nécessitant une donnée de bit, le numéro de TC donne accès à un bit fonctionnant comme "drapeau de fin" indiquant le moment où la tempo. ou le compteur sont écoulés : le bit qui est normalement à OFF passe à ON lorsque la SV désignée a expiré. En cas de désignation d'une opérande nécessitant une donnée de canal, le numéro de TC donne accès à un emplacement mémoire retenant la valeur en cours (PV) de la tempo. ou du compteur. Elle peut donc être utilisée par exem-

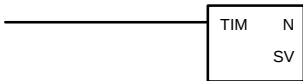
ple, comme opérande dans CMP(20), ou dans toute autre instruction admettant la zone TC. On désigne pour cela le numéro de TC utilisé pour définir la tempo. ou le compteur accédant à la mémoire retenant la PV.

Remarquez que "TIM 000" est utilisé pour désigner l'instruction TIMER définie avec le numéro de TC 000, pour désigner le drapeau de fin et la PV de cette tempo. Le premier terme est toujours une instruction, le deuxième toujours une opérande de bit et le troisième toujours une opérande de canal. Cela est également valable pour tous les autres numéros de TC portant le préfixe TIM ou CNT.

On peut entrer une SV comme constante ou comme adresse de canal dans une zone de données. Si un canal de zone IR attribué à une carte d'entrée est désigné comme adresse de canal, la carte d'entrée peut être câblée de façon à ce que la SV puisse être sélectionnée en externe par des interrupteurs de roue codeuse ou assimilé. Les tempo. et compteurs câblés de cette façon ne peuvent être sélectionnés que de façon externe en mode RUN ou MONITOR. Toutes les SV, y compris celles sélectionnées de façon externe, doivent être en BCD.

5-15-1 TIMER – TIM

Symbole



Valeurs des données d'opérande

N: n° de TC
(000 à 511)

Zones de données d'opérande

SV: valeur sélectionnée (canal, BCD)
IR, SR, AR, DM, HR, LR, #

Limites

SV est située entre 000,0 et 999,9. La virgule décimale n'est pas entrée.

Chaque numéro de TC peut être utilisé comme donnée d'opérande dans une seule instruction TIMER ou COUNTER.

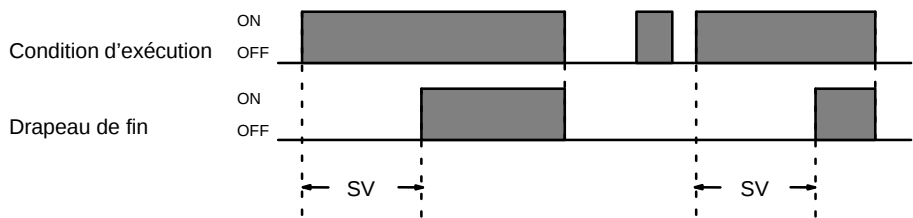
Les TC 000 à TC 015 ne doivent pas être utilisés dans TIM si ils sont nécessaires pour TIMH(15). Reportez-vous au chapitre *HIGH-SPEED TIMER – TIMH(15)* tempo. rapide, pour de plus amples détails.

Description

Une temporisation s'active lorsque sa condition d'exécution passe à ON et est remise à zéro (à la SV) lorsque la condition passe à OFF. Une fois activée, TIM mesure par unités de 0,1 seconde à partir de la SV.

Si la condition d'exécution reste suffisamment longtemps à ON pour que TIM atteigne zéro, le drapeau de fin du numéro de TC utilisé passe à ON et y reste jusqu'à ce que TIM soit remise à zéro (c'est-à-dire jusqu'à ce que la condition passe à OFF).

Le schéma suivant illustre le lien entre la condition de TIM et le drapeau de fin qui lui est assigné.



Précautions

Les tempo. des parties en branchement sont remises à zéro lorsque la condition de IL(02) est à OFF. Les coupures de courant remettent également à zéro les

tempo. Si l'on souhaite une tempo. qui ne soit pas remise à zéro dans ces conditions, les bits d'impulsion de l'horloge de la zone SR peuvent être comptés pour produire des tempo. utilisant CNT. Reportez-vous au chapitre suivant pour de plus amples détails.

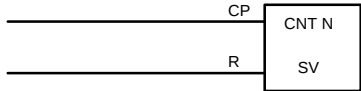
Drapeaux

ER:

La SV n'est pas en BCD.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-15-2 COUNTER – CNT

Symbole



Valeurs de données d'opérande

N: n° de TC
(000 à 511)

Zones de données d'opérande

SV: valeur sél. (canal, BCD)
IR, SR, AR, DM, HR, LR, #

Limites

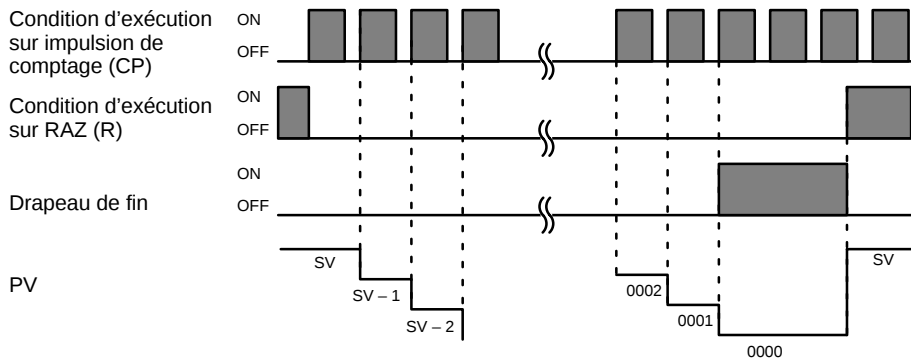
Chaque numéro de TC peut être utilisé comme donnée d'opérande dans une seule instruction TIMER ou COUNTER.

Description

CNT s'utilise pour décompter à partir de la SV lorsque la condition de l'impulsion de comptage, CP, passe de OFF à ON, c'est-à-dire que la valeur en cours (PV) est décrémentée de un à chaque fois que CNT est exécutée avec une condition à ON pour la CP et que la condition de la dernière exécution était à OFF. Si la condition d'exécution n'a pas changé ou est passée de ON à OFF, la PV de CNT reste inchangée. Le drapeau de fin d'un compteur passe à ON lorsque la PV atteint zéro et y reste jusqu'à remise à zéro du compteur.

CNT est remis à zéro à l'aide d'une entrée de RAZ, R. Lorsque R passe de OFF à ON, la PV est remise à la valeur de la SV. La PV n'est pas décrémentée lorsque R est à ON. Le décomptage à partir de SV recommence lorsque R passe à OFF. La CP de CNT n'est pas remise à zéro dans les branchements ni en cas de coupure de courant.

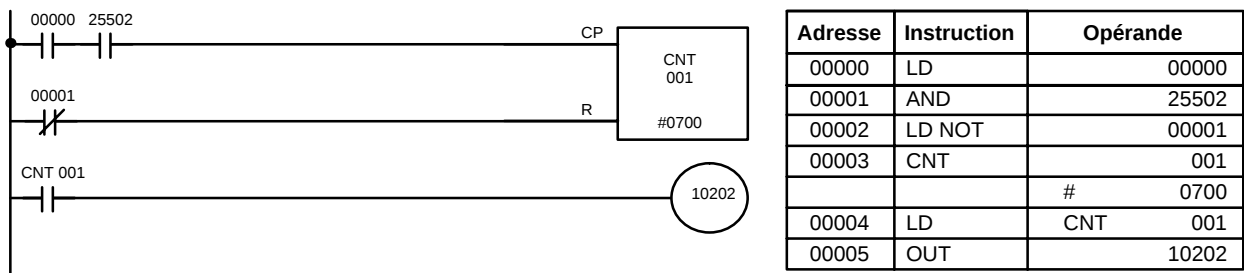
Les changements de conditions d'exécution, de drapeau de fin et de PV sont illustrés ci-dessous. La hauteur de la ligne de PV représente uniquement les modifications de la PV (valeur en cours).



Précautions L'exécution du programme continue même si l'on utilise une valeur qui n'est pas en BCD, mais la SV est alors faussée.

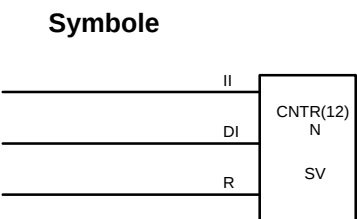
Drapeaux **ER:** La SV n'est pas en BCD.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple Dans l'exemple suivant, CNT est utilisé pour créer des temps sauvegardés de deux façons : en combinant TIM et CNT et en comptant les bits d'impulsion de l'horloge de la zone SR.
CNT 001 compte le nombre de fois où un bit d'impulsion d'horloge d'une seconde (SR 25502) passe de OFF à ON. A nouveau, IR 00000 est utilisé pour contrôler le temps de fonctionnement de CNT.
Comme dans cet exemple la SV de CNT 001 est 700, le drapeau de fin de CNT 001 passe à ON lorsque 1 seconde x 700 fois, ou 11 minutes et 40 secondes se sont écoulées. Cela a pour résultat de placer IR 10202 à ON.



Attention : les impulsions d'horloge les moins importantes risquent de produire des temporisations imprécises parce que leur temps ON est court et peut ne pas être lu très précisément pendant les tours de scrutations longs. En particulier, les impulsions d'horloge de 0,02 seconde et 0,01 seconde ne doivent pas être utilisées pour créer des temporisations avec les instructions CNT.

5-15-3 REVERSIBLE COUNTER – CNTR(12)



Valeurs de données d'opérande

N: n° de TC
(000 à 511)

Zones de données d'opérande

SV: valeur sél. (canal, BCD)
IR, SR, AR, DM, HR, LR, #

Limites Chaque numéro de TC peut être utilisé comme donnée d'opérande dans une seule instruction TIMER ou COUNTER.

Description CNTR(12) est un compteur rotatif incrémental/décrémental réversible, c'est-à-dire qu'il est utilisé pour effectuer un comptage entre zéro et la SV en fonction des modifications de deux conditions d'exécution dans l'entrée incrémentale (II) et dans l'entrée décrémentation (DI).
La valeur en cours (PV) est incrémentée de un à chaque exécution de CNTR(12) avec une condition d'exécution ON pour II et avec la dernière condi-

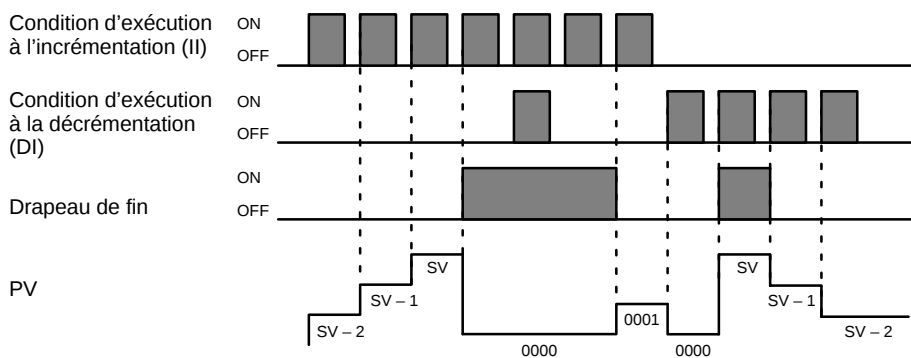
tion de II à OFF. Si des passages de OFF à ON se sont produits dans II et DI depuis la dernière exécution, la PV n'est pas modifiée.

Si les conditions d'exécution n'ont pas changé ou sont passées de ON à OFF pour II et DI, les PV de CNT restent inchangées.

Lors de la décrémentation à partir de 0000, la valeur en cours est placée à la valeur de la SV et le drapeau de fin passe à ON jusqu'à ce que la PV soit à nouveau décrémentée. Lorsque l'incrémenta-tion dépasse la SV, la PV est placée à 0000 et le drapeau de fin est placé à ON jusqu'à l'incrémenta-tion suivante de PV.

CNTR(12) est remise à zéro à l'aide d'une entrée de RAZ, R. Lorsque R passe de OFF à ON, la PV est remise à zéro. La PV n'est ni incrémentée ni décrémentée lorsque R est à ON. Le comptage reprend lorsque R passe à OFF. La PV de CNTR(12) n'est pas remise à zéro dans les branchements ni en cas de coupure de courant. R est prioritaire sur II et DI.

Les changements dans les conditions d'exécution II et DI, le drapeau de fin et la PV sont illustrés ci-dessous. Le schéma représente une partie du fonctionnement de CNTR(12) : à la remise à zéro, le comptage commence à partir de zéro. La hauteur de la ligne de la PV indique uniquement les changements de la PV.



Précautions

L'exécution du programme continue même en cas d'utilisation d'une SV qui n'est pas en BCD, mais cette SV n'est pas valide.

Drapeaux

ER:

La SV n'est pas en BCD.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-15-4 HIGH-SPEED TIMER – TIMH(15)

Symbole

TIMH(15) N

SV

Valeurs de données d'opérande

N: n° de TC

(000 à 015 de préférence)

Zones de données d'opérande

SV: valeur sél. (canal, BCD)

IR, SR, AR, DM, HR, LR, #

Limites

SV est située entre 00,00 et 99,99 (bien qu'on puisse effectuer la sélection entre 00,00 et 00,01, une sélection de 00,00 désactive la temporisation : elle place

immédiatement le drapeau de fin à ON et la scrutation de 00,01 n'est plus assurée de façon fiable). La virgule décimale n'est pas entrée.

Chaque numéro de TC peut être utilisé comme donnée d'opérande dans une seule instruction TIMER ou COUNTER.

Les tempo. rapides portant les numéros de tempo. TC 016 à TC 511 ne doivent pas être utilisés si le temps de scrutation dépasse 10 ms.

Description

TIMH(15) fonctionne de la même façon que TIM, à ceci près que TIMH mesure par unités de 0,01 seconde. Reportez-vous au chapitre *TIMER – TIM* pour de plus amples détails.

Précautions

Les temporisations des branchements sont remises à zéro lorsque la condition d'exécution de IL(02) est à OFF. Les coupures de courant remettent également à zéro les temporisations. Si l'on souhaite une tempo. qui ne soit pas remise à zéro dans ces conditions, les bits d'impulsion de l'horloge de la zone SR peuvent être comptés pour produire des tempo. en utilisant un CNT. Reportez-vous au chapitre *COUNTER – CNT* pour de plus amples détails.

Les tempo. des branchements ne sont pas remis à zéro lorsque la condition d'exécution de JMP(04) est à OFF, mais la tempo. ne fonctionne plus si l'on utilise le numéro de saut 00. Les tempo. continuent de fonctionner si l'on utilise les numéros de saut 01 à 99.

Les tempo. rapides portant les numéros TC 000 à TC 015 deviennent imprécises lorsque la configuration de l'API (DM 6629) est sélectionnée pour effectuer une interruption sur ces temporisations.

Les tempo. rapides portant les numéros TC 016 à TC 511 deviennent imprécises si le temps de scrutation dépasse 10 ms. Si le temps de scrutation dépasse 10 ms, utilisez TC 000 à TC 015 et sélectionnez DM 6629 pour les interruptions des numéros de tempo. utilisés.

Drapeaux

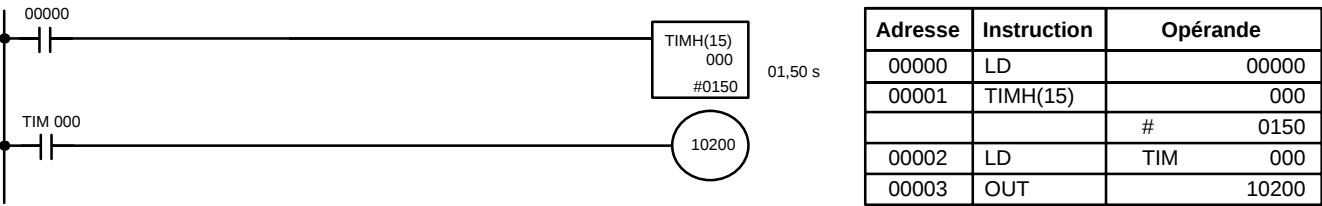
ER:

La SV n'est pas en BCD.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple

L'exemple suivant illustre une temporisation sélectionnée avec une constante : 10200 passe à ON après le passage de 00000 à ON et y reste pendant au moins 1,5 secondes. Lorsque 00000 passe à OFF, la tempo. est remise à zéro et 10200 passe à OFF.



5-15-5 INTERVAL TIMER – STIM(—)

Symboles

—	STIM(—)	—	@STIM(—)
	C1		C1
	C2		C2
	C3		C3

Zones de données d'opérande

C1: donnée de contrôle n° 1
000 à 008, 010 à 012
C2: donnée de contrôle n° 2
IR, SR, AR, DM, HR, TC, LR, #
C3: donnée de contrôle n° 3
IR, SR, AR, DM, HR, TC, LR, #

Limites

C1 doit être située entre 000 et 008 ou entre 010 et 012.

Si C1 est située entre 006 et 008, les DM 6143 à 6655 ne peuvent être utilisés pour C2 ou C3.

Description

STIM(—) est utilisée pour contrôler les tempo. cycliques en exécutant quatre fonctions de base : démarrage de la tempo. pour une interruption non-impulsionnelle, démarrage de la tempo. pour les interruptions programmées, arrêt de la tempo. et lecture de la PV de la tempo. Sélectionnez la valeur de C1 pour spécifier laquelle de ces fonctions est exécutée et sur quelle tempo. cyclique elle est effectuée (parmi les trois tempo. cycliques possibles), comme sur le tableau suivant. Reportez-vous à la page 32 pour de plus amples détails sur l'utilisation des interruptions de tempo. cyclique. STIM(—) est décrite plus précisément à la suite de ce tableau :

Fonction	Tempo.	Valeur de C1
Démarrage des tempo.	0	000
	1	001
	2	002
Démarrage des tempo. programmées	0	003
	1	004
	2	005
Lecture de la valeur en cours (PV) de la tempo.	0	006
	1	007
	2	008
Arrêt des tempo.	0	010
	1	011
	2	012

Rem. :1. La tempo. cyclique 0 ne peut être utilisée lorsqu'une sortie par impulsion est produite par l'instruction SPED(—).

2. La tempo. cyclique 2 ne peut être utilisée lorsque le fonctionnement du compteur rapide 0 a été activé dans le DM 6642 de la configuration API.

Démarrage des interruptions

Effectuez la sélection de C1 entre 000 et 002 pour lancer les tempo. 0 à 2 si vous voulez activer une interruption à une impulsion. Effectuez la sélection de C1 entre 003 et 005 pour lancer les interruptions programmées utilisant les tempo. 0 à 2.

C2, qui spécifie la SV de la tempo., peut être égal à une constante ou être le premier des deux canaux contenant la SV. Les sélections sont légèrement différentes selon la méthode utilisée.

Si C2 est une constante, il spécifie la valeur initiale du compteur décrémental (BCD, 0000 à 9999). L'intervalle de temps en décrémentation est de 1 ms.

Si C2 est une adresse de canal, il spécifie la valeur initiale du compteur décrémental (BCD, 0000 à 9999) et C2+1 spécifie l'intervalle du temps de décrémentation (BCD, 0005 à 0320) par unités de 0,1 ms. L'intervalle du temps de décrémentation peut donc être situé entre 0,5 et 32 ms.

C3 spécifie le sous-programme numéro 0000 à 0255 (0000 à 0127 avec le CQM1-CPU11/21-E).

Rem. : temps écoulé entre le lancement de la tempo. cyclique et le moment où le temps est atteint : $(\text{contenu de C2}) \times (\text{contenu de C2}+1) \times 0,1 \text{ ms}$

Lecture des PV de la tempo. Effectuez la sélection de C1 entre 006 et 008 pour lire les PV des tempo. 0 à 2. C2 spécifie le premier des deux canaux de destination qui reçoit la PV de la tempo. C2 reçoit le nombre de fois où le compteur décrémental a été décrémental (BCD, 0000 à 9999) et C2+1 reçoit l'intervalle du temps de décrémentation (BCD par unités de 0,1 ms).

C3 spécifie le canal de destination qui reçoit le temps écoulé depuis la dernière décrémentation de la tempo. (BCD par unités de 0,1 ms).

Rem. : le temps écoulé depuis le lancement de la tempo. peut être calculé comme suit : $[(\text{contenu de C2}) \times (\text{contenu de C2}+1) + (\text{contenu de C3})] \times 0,1 \text{ ms}$

Arrêt des tempo. Effectuez la sélection de C1 entre 010 et 012 pour arrêter les tempo. 0 à 2. C2 et C3 n'ont pas de fonction et ne doivent pas être tous les deux égaux à 000.

Drapeaux

ER: La tempo. cyclique 0 est lancée alors qu'une sortie par impulsion est en fonctionnement (C1=000 uniquement).

La tempo. cyclique 2 est lancée alors que le compteur rapide 0 est désactivé (C1=002 uniquement).

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Une des limites de zone de données a été dépassée.

5-15-6 REGISTER COMPARISON TABLE – CTBL(—)

Symboles

CTBL(—)

P

C

TB

@CTBL(—)

P

C

TB

Zones de données d'opérande

P: numéro du port

000, 001 ou 002

C: donnée de contrôle

000 à 003

TB: 1er canal de la table de comparaison

IR, SR, AR, DM, HR, LR

Limites

Les premiers et derniers canaux des tables de comparaison doivent appartenir à la même zone de données (la longueur de la table de comparaison varie selon les sélections).

Avec le CQM1-CPU43-E, CTBL(—) ne peut être utilisé si la configuration de l'API (DM 6611) est sélectionnée en mode de sortie par impulsion.

Description

Lorsque la condition d'exécution est à OFF, CTBL(—) n'est pas exécutée. Lorsque la condition est à ON, CTBL(—) enregistre une table de comparaison pour

utilisation avec la PV du compteur rapide. Selon la valeur de C, la comparaison avec la PV du compteur rapide peut commencer immédiatement ou séparément avec INI(—).

La donnée de spécification du port (P) spécifie le compteur rapide qui sera utilisé dans la comparaison :

P	Fonction
000	Spécifie le compteur rapide 0.
001	Spécifie le compteur rapide 1.
002	Spécifie le compteur rapide 2.

La fonction de CTBL(—) est déterminée par la donnée de contrôle C, comme dans le tableau suivant. Les fonctions sont décrites après le tableau.

C	Fonction CTBL(—)
000	Enregistre une table de comparaison des valeurs spécifiées et lance la comparaison.
001	Enregistre une table de comparaison des gammes et lance la comparaison.
002	Enregistre une table de comparaison des valeurs spécifiées et lance la comparaison avec INI(—).
003	Enregistre une table de comparaison des gammes et lance la comparaison avec INI(—).

Lorsque la PV est égale à la valeur spécifiée ou est située dans une gamme spécifiée, le sous-programme spécifié est appelé et exécuté.

Si le compteur rapide est activé dans la configuration API (DM 6642), il commence le comptage à partir de zéro lorsque le CQM1 entre en fonctionnement. La PV n'est pas comparée à la table avant que celle-ci soit enregistrée. La comparaison commence avec INI(—) ou CTBL(—). La comparaison peut être lancée ou arrêtée ou la PV peut être remise à zéro avec INI(—).

Une fois une table enregistrée, elle reste valable jusqu'à l'arrêt du CQM1 ou jusqu'à l'occurrence d'une erreur en cas de tentative d'enregistrement d'une nouvelle table. La forme sur changement d'état de CTBL(—) est recommandée lorsqu'il est possible de réduire le temps de scrutation.

Comparaison des valeurs spécifiées

Une table de comparaison des valeurs spécifiées contient jusqu'à seize valeurs spécifiées et un numéro de sous-programme correspondant à chaque valeur spécifiée. Le sous-programme correspondant est appelé et exécuté lorsque la PV correspond à une valeur spécifiée (lorsque l'interruption n'est pas souhaitée, un numéro de sous-programme non défini peut être entré).

Le tableau suivant illustre la structure d'une table de comparaison des valeurs spécifiées utilisé avec le compteur rapide 0 ou les compteurs rapides 1 ou 2 en mode linéaire.

TB	Nombre de valeurs spécifiées (BCD)	0001 à 0016 → Sél. d'une valeur spécifiée
TB+1	Valeur spécif. 1, 4 chiffres de poids faible (BCD)	
TB+2	Valeur spécif. 1, 4 chiffres de poids fort (BCD)	
TB+3	N° de sous-programme (cf Rem.)	
—	—	

La table suivante illustre la structure d'une table de comparaison des valeurs spécifiées utilisée avec le compteur rapide 1 ou 2 en anneau. Entrez les valeurs spécifiées par ordre croissant ou décroissant.

La valeur de l'anneau spécifie la valeur de comptage maximum (valeur de l'anneau = valeur de comptage max.+1). Ne changez pas la valeur de l'anneau pendant une comparaison.

TB	V. de l'anneau, 4 ch. de p. faible BCD	→ Sélection de la valeur d'anneau
TB+1	V. de l'anneau, 4 ch. de poids fort BCD	
TB+2	Nombre de valeurs spécifiées (BCD)	→ 0001 à 0016
TB+3	Valeur n°1, 4 ch. de poids faible (BCD)	
TB+4	Valeur n°1, 4 ch. de poids fort (BCD)	→ Une sélection d'une valeur spécifiée
TB+5	Nombre de ss-programme (cf Rem.)	
⋮	⋮	
⋮	⋮	

Le tableau suivant illustre la structure d'une table de comparaison des valeurs spécifiées utilisée avec les compteurs rapides absolus 1 et 2 (CQM1-CPU44-E uniquement). Entrez les valeurs spécifiées par ordre croissant ou décroissant.

TB	Nombre de valeurs spécifiées (BCD)	→ 0001 à 0016
TB+1	Valeur spécifiée n° 1 (BCD)	
TB+2	N° de sous-programme (cf Rem.)	→ Sélection d'une valeur spécifiée
⋮	⋮	
⋮	⋮	

Rem. : 1. Le numéro de sous-programme peut être situé entre F000 et F255 (F000 à F127 pour le CQM1-CPU11/21-E) pour activer le sous-programme lors de la décrémentation et entre 0000 et 0255 (0000 à 0127 pour le CQM1-CPU11/21-E) pour le sous-programme lors de l'incrémement (il y a 256 sous-programmes possibles).

2. Laissez un intervalle d'au moins 0,2 ms pour les interruptions lors de la sélection de la valeur spécifiée pour les compteurs rapides 1 et 2.

Comparaison de gammes

Une table de comparaison de gammes contient huit niveaux de deux seuils (définis par une limite basse à 8 chiffres et une limite haute à 8 chiffres) et leurs numéros de sous-programme correspondant. Le sous-programme correspondant est appelé et exécuté lorsque la PV est située dans une gamme donnée (lorsque l'interruption n'est pas souhaitée, un numéro de sous-programme non défini peut être entré).

Sélectionnez toujours huit niveaux de deux seuils ; si vous avez besoin de moins de huit gammes, placez les numéros de sous-programmes restants à FFFF. Si vous avez besoin de plus de huit niveaux, on peut utiliser une autre instruction de comparaison comme BCMP(—) pour comparer les niveaux avec les PV du compteur rapide dans les IR 230 à IR 235. Gardez à l'esprit que ces canaux sont rafraîchis une seule fois par tour de scrutation.

Certains drapeaux de la zone AR indiquent le fait qu'une PV appartienne à une ou plusieurs des huit niveaux. Les drapeaux passent à ON lorsqu'une PV appartient à la zone correspondante.

Compteur	Drapeaux des zones AR
Compteur rapide 0	AR 1100 à AR 1107 : gammes 1 à 8
Compteur rapide 1	AR 0500 à AR 0507 : gammes 1 à 8
Compteur rapide 2	AR 0600 à AR 0607 : gammes 1 à 8

Le tableau suivant illustre la structure d'un tableau de comparaison par niveau pour utilisation avec le compteur rapide 0 ou les compteurs rapides 1 ou 2 en mode linéaire.

TB	Limite basse 1, 4 ch. de p. faible BCD	→ Sélection de la première gamme TB et TB+1 : seuil bas TB+2 et TB+3 : seuil haut
TB+1	Limite basse 1, 4 ch. de p. fort BCD	
TB+2	Limite haute 1, 4 ch. de p. faible BCD	
TB+3	Limite haute 1, 4 ch. de p. fort BCD	
TB+4	N° de ss-programme (cf Rem. 1)	
⋮	⋮	
TB+35	Limite basse 8, 4 ch. de p. faible BCD	→ Sélection de la huitième gamme TB+35 et TB+36 : seuil bas TB+37 et TB+38 : seuil haut
TB+36	Limite basse 8, 4 ch. de p. fort BCD	
TB+37	Limite haute 8, 4 ch. de p. faible BCD	
TB+38	Limite haute 8, 4 ch. de p. fort BCD	
TB+39	N° de ss-programme (cf Rem. 1)	

Le tableau suivant illustre la structure d'une table de comparaison par gamme pour utilisation les compteurs rapides 1 ou 2 en mode par anneau. La valeur d'anneau spécifie le nombre de points de l'anneau et la valeur max. de comptage (valeur d'anneau = valeur max. de comptage + 1). Ne changez pas la valeur d'anneau pendant une comparaison.

TB	Val. d'anneau, 4 ch. de p. faible BCD	→ Sélection de la valeur d'anneau
TB+1	Val. d'anneau, 4 ch. de p. fort BCD	
TB+3	Limite basse 1, 4 ch. de p. faible BCD	→ Sélection de la première gamme
TB+4	Limite basse 1, 4 ch. de p. fort BCD	
TB+5	Limite haute 1, 4 ch. de p. faible BCD	
TB+6	Limite haute 1, 4 ch. de p. fort BCD	
TB+7	N° de ss-programme (cf Rem. 1)	
⋮	⋮	
TB+37	Limite basse 8, 4 ch. de p. faible BCD	→ Sélection de la huitième gamme
TB+38	Limite basse 8, 4 ch. de p. fort BCD	
TB+39	Limite haute 8, 4 ch. de p. faible BCD	
TB+40	Limite haute 8, 4 ch. de p. fort BCD	
TB+41	N° de ss-programme (cf Rem. 1)	

Le tableau suivant illustre la structure d'une table de comparaison par gamme pour utilisation avec les compteurs rapides absolus 1 et 2 (CQM1-CPU44-E uniquement).

TB	Limite basse 1 (BCD)	→ Sélection de la première gamme
TB+2	Limite haute 1 (BCD)	
TB+4	N° de ss-programme (cf Rem. 2)	
⋮	⋮	
TB+21	Limite basse 8 (BCD)	→ Sélection de la huitième gamme
TB+22	Limite haute 8 (BCD)	
TB+23	N° de ss-programme (cf Rem. 2)	

- Rem. : 1. Le numéro de sous-programme peut être situé entre 0000 et 0255 (0000 à 0127 pour le CQM1-CPU11/21-E) et le sous-programme est exécuté aussi longtemps que la PV du compteur est située dans la gamme spécifiée. Une valeur de FFFF indique qu'aucun sous-programme ne sera exécuté.
2. Le numéro de sous-programme peut être situé entre F000 et F255 (F000 à F127 pour le CQM1-CPU11/21-E) pour activer le sous-programme lors

d'une décrémentation et entre 0000 et 0255 (0000 à 0127 pour le CQM1-CPU11/21-E) pour activer le sous-programme lors d'une incrémentation.

3. Laissez un laps de temps d'au moins 0,2 ms entre les limites haute et basse (limite haute – limite basse > $0,002 \times \text{fréquence d'impulsion d'entrée}$) dans les comparaisons de gammes avec les compteurs rapides 1 et 2.

Le tableau suivant illustre les valeurs possibles pour les valeurs spécifiées, les valeurs de limite basse et de limite haute. La valeur hexadécimale F dans le chiffre de poids le plus fort indique que la valeur est négative.

Compteur	Valeurs possibles
Compteur rapide 0	Mode incrémental/décrémental : F003 2767 à 0003 2767 Mode incrémental : 0000 0000 à 0006 5535
Compteurs rapides 1 et 2	Mode linéaire : F838 8607 à 0838 8607 Mode en anneau : 0000 0000 à 0006 4999
Compteurs rapides absolus 1 et 2	Mode BCD : 0000 à 4095 Mode 360° : 0000 à 0355 (unités de 5°)

En mode 360°, les valeurs angulaires du compteur rapide absolu sont converties de façon interne en valeurs binaires. La valeur binaire après conversion dépend de la résolution choisie dans la configuration API (DM 6643 et/ou DM 6644). Le tableau suivant illustre les valeurs converties entre 5° et 45°.

Résolution	Valeur convertie								
	5°	10°	15°	20°	25°	30°	35°	40°	45°
8 bits (0 à 255)	4	7	11	14	18	21	25	28	32
10 bits (0 à 1023)	14	28	43	57	71	85	100	114	128
12 bits (0 à 4095)	57	114	171	228	284	341	398	455	512

Pour les valeurs plus importantes, prenez la valeur convertie de 45° et multipliez-la par le nombre de fois nécessaires, en ajoutant la valeur restante, dont vous trouverez la conversion dans le tableau. Par exemple, pour convertir 145° en résolution à 8 bits, il faut procéder comme suit :
 32×3 (pour 135°) + 7 (pour 10°) = 103.

Attention : pour les résolutions à 10 bits et 12 bits, l'interruption peut ne pas être déclenchée lorsque la valeur angulaire correspond à la valeur de comparaison car les valeurs converties ne sont pas exactement identiques.

Drapeaux

ER: Erreur dans les sélections du compteur rapide
 Port et fonction spécifiés non compatibles

Instruction CTBL(—) dans le sous-programme appelé par une autre instruction CTBL(—).

Une instruction CTBL(—) utilisant un format de comparaison différent est exécutée pendant la comparaison.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Le tableau de comparaison dépasse la limite de zone de données ou il existe une erreur dans les sélections du tableau de comparaison.

CTBL(—) est exécutée dans un sous-programme d'interruption pendant qu'une E/S par impulsion ou une instruction de compteur rapide est exécutée dans le programme principal.

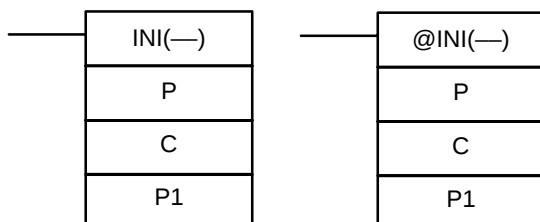
AR 05: les drapeaux AR 0500 à AR 0507 passent à ON pour indiquer que la PV du compteur rapide 1 appartient aux gammes 1 à 8.

AR 06: les drapeaux AR 0600 à AR 0607 passent à ON pour indiquer que la PV du compteur rapide 2 appartient aux gammes 1 à 8.

AR 11: les drapeaux AR 1100 à AR 1107 passent à ON pour indiquer que la PV du compteur rapide 0 appartient aux gammes 1 à 8.

5-15-7 MODE CONTROL – INI(—)

Symboles



Zones de données d'opérande

P: donnée de spécification du port
000, 001 ou 002
C: donnée de contrôle
000 à 003
P1: premier canal de PV
IR, SR, AR, DM, HR, LR

Limites

P doit être égal à 000, 001 ou 002 et C doit être situé entre 000 et 003.

P1 doit être égal à 000, sauf si C est égal à 002.

P1 et P1+1 doivent appartenir à la même zone de données.

Les DM 6143 à DM 6655 ne peuvent être utilisés pour P1.

Description

Lorsque la condition d'exécution est à OFF, INI(—) n'est pas exécutée. Lorsque la condition est à ON, INI(—) est utilisée pour contrôler le fonctionnement du compteur rapide et pour arrêter la sortie d'impulsion.

La valeur de spécification du port (P) spécifie le compteur rapide ou la sortie par impulsion qui sont contrôlés.

P	Fonction
000	Spécifie le compteur rapide 0 ou une sortie par impulsion à partir d'un bit.
001	Spécifie le compteur rapide 1 ou une sortie par impulsion à partir du port 1.
002	Spécifie le compteur rapide 2 ou une sortie par impulsion à partir du port 2.

La fonction de INI(—) est déterminée par la donnée de contrôle C (P1 et P1+1 contiennent la nouvelle PV du compteur rapide lors d'une modification de PV).

C	P1	Fonction INI(—)
000	000	Lance la comparaison de table CTBL(—).
001	000	Arrête la comparaison de table CTBL(—).
002	Nouvelle PV du compteur rapide	Modifie la PV du compteur rapide.
003	000	Arrête la sortie par impulsion.

Comparaison du tableau CTBL(—)

Si C est égal à 000 ou à 001, INI(—) lance ou arrête la comparaison de la PV du compteur rapide et du tableau enregistré avec CTBL(—).

Modification de PV

Si C est égal à 002, INI(—) modifie la PV du compteur rapide en valeur à 8 chiffres dans P1 et P1+1.

Avec le compteur rapide 0, la PV peut être comprise entre F003 2767 et 0003 2767 en mode incrémental/décroissant ou entre 0000 0000 et 0006 5535 en

mode incrémental. La valeur hexadécimale F du chiffre de poids le plus fort de PV indique que la PV est négative.

4 chiffres d'extrême gauche	4 chiffres d'extrême droite	Mode incrémental/ décrémental F0032767 à 00032767	Mode incrémental 00000000 à 00065535
P1+1	P1		

Avec les compteurs rapides 1 et 2, la PV peut être comprise entre F838 8607 et 0838 8607 en mode linéaire ou entre 0000 0000 et 0006 4999 en mode par anneau. La valeur hexadécimale F dans le chiffre de poids le plus fort de PV indique que la PV est négative.

4 chiffres d'extrême gauche	4 chiffres d'extrême droite	Mode linéaire F8388607 à 08388607 (-8 388 607 à 8 388 607)	Mode en anneau 00000000 à 00064999
P1+1	P1		

Rem. : la PV des compteurs rapides absolus 1 et 2 (CQM1-CPU44-E uniquement) ne peut être modifiée.

Arrêt de sortie par impulsion Si C est égal à 003, INI(—) arrête la sortie par impulsion.

Drapeaux

ER: Port et fonction spécifiés non compatibles

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

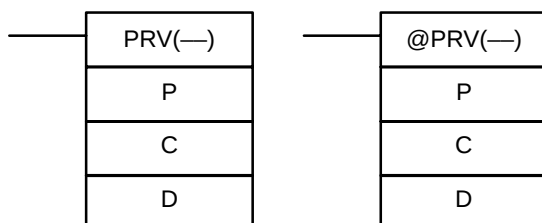
P1+1 dépasse la limite de zone de données (C=002)

Erreur dans les sélections d'opérande

INI(—) est exécutée sans un sous-programme d'interruption pendant l'exécution d'une E/S par impulsion ou d'une instruction de compteur rapide dans le programme principal.

5-15-8 HIGH-SPEED COUNTER PV READ – PRV(—)

Symboles



Zones de données d'opérande

P: valeur de spécification du port
000, 001 ou 002
C: donnée de contrôle
000, 001 ou 002
D: premier canal de destination
IR, SR, AR, DM, HR, LR

Limites

P et C doivent être égaux à 000, 001 ou 002.

D et D+1 doivent appartenir à la même zone de données.

Les DM 6143 à DM 6655 ne peuvent être utilisés pour D.

Description

Lorsque la condition est à OFF, PRV(—) n'est pas exécutée. Lorsque la condition est à ON, PRV(—) lit les données spécifiées par P et C et les écrit dans D ou D et D+1.

La valeur de spécification du port (P) spécifie le compteur rapide ou la sortie par impulsion.

P	Fonction
000	Spécifie le compteur rapide 0 ou une sortie par impulsion à partir d'un bit.
001	Spécifie le compteur rapide 1 ou une sortie par impulsion à partir du port 1.
002	Spécifie le compteur rapide 2 ou une sortie par impulsion à partir du port 2.

La donnée de contrôle C détermine le type de données auquel on a accès.

C	Donnée	Canaux de destination
000	PV du compteur rapide	D et D+1
001	Etat du compteur rapide et de la sortie par impulsion	D
002	Résultats de la comparaison par gamme	D

PV du compteur rapide (C=000)

Si C est égal à 000, PRV(—) lit la PV du compteur rapide spécifié et écrit la valeur à 8 chiffres dans D et D+1.

Avec le compteur rapide 0, la PV peut être comprise entre F003 2767 et 0003 2767 en mode incrémental/décrémental ou entre 0000 0000 et 0006 5535 en mode incrémental. La valeur hexadécimale F du chiffre de poids le plus fort de PV indique que la PV est négative.

4 chiffres d'extrême gauche	4 chiffres d'extrême droite	Mode incrémental/décrémental	Mode incrémental
D+1	D	F0032767 à 00032767	00000000 à 00065535

Avec les compteurs rapides 1 et 2, la PV peut être comprise entre F838 8607 et 0838 8607 en mode linéaire ou entre 0000 0000 et 0006 4999 en mode par anneau. La valeur hexadécimale F dans le chiffre de poids le plus fort de PV indique que la PV est négative.

4 chiffres d'extrême gauche	4 chiffres d'extrême droite	Mode linéaire	Mode en anneau
D+1	D	F8388607 à 08388607 (-8 388 607 à 8 388 607)	00000000 à 00064999

Avec les compteurs rapides absolus 1 et 2, la PV peut être située entre 0000 0000 et 0000 4095 en mode BCD ou entre 0000 0000 et 0000 0359 en mode 360 °.

4 chiffres d'extrême gauche	4 chiffres d'extrême droite	Mode BCD	Mode 360 °
D+1	D	0000 0000 à 0000 4095	0000 0000 à 0000 0359

Etat du compteur rapide ou de la sortie par impulsion (C=001)

Si C est égal à 001, PRV(—) lit l'état de fonctionnement du compteur rapide ou de la sortie par impulsion spécifié et écrit les données dans D.

Le tableau suivant illustre la fonction des bits de D pour les compteurs rapides 1 et 2 et les sorties par impulsion des ports 1 et 2 (CQM1-CPU43-E uniquement). Les bits non représentés ici ne sont pas utilisés et sont constamment à 0.

Bit	Fonction
00	Etat de la comparaison des compteurs rapide (0: arrêt ; 1: comparaison)
01	Non-respect de la gamme du compteur rapide (0: normal ; 1: gamme non respectée)
04	Décélération de la fréquence d'impulsion (0: pas de spécification ; 1: spécification)
05	Nombre total d'impulsions spécifiées (0: pas de spécification ; 1: spécification)
06	Fin de la sortie par impulsion (0 : non terminée ; 1 : terminée)
07	Etat de la sortie par impulsion (0 : terminée ; 1: sortie en cours)

Pour les compteurs rapides absolus 1 et 2 (CQM1-CPU44-E uniquement), le bit 00 de D indique l'état de la comparaison (0 : terminée ; 1: non terminée). Les autres bits de D (01 à 15) ne sont pas utilisés et restent constamment à 0.

Rem. : ces drapeaux sont dans AR 05 et AR 06, mais ces canaux ne sont normalement rafraîchis qu'une fois par tour de scrutation ; la donnée obtenue par PRV(—) est donc plus à jour.

**Résultat de la comparaison
par gamme (C=002)**

Si C est égal à 002, PRV(—) lit les résultats de la comparaison de la PV sur les huit gammes définies par CTBL(—) et écrit ces données dans D. Les bits 00 à 07 de D contiennent les drapeaux des résultat de comparaison des gammes 1 à 8 (0 : gamme non respectée ; 1 : gamme respectée).

Rem. : Ces drapeaux sont dans AR 05 et AR 06, mais ces canaux ne sont normalement rafraîchis qu'une fois par tour de scrutation ; la donnée obtenue par PRV(—) est donc plus à jour.

Drapeaux

ER: Port et fonction spécifiés non compatibles

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

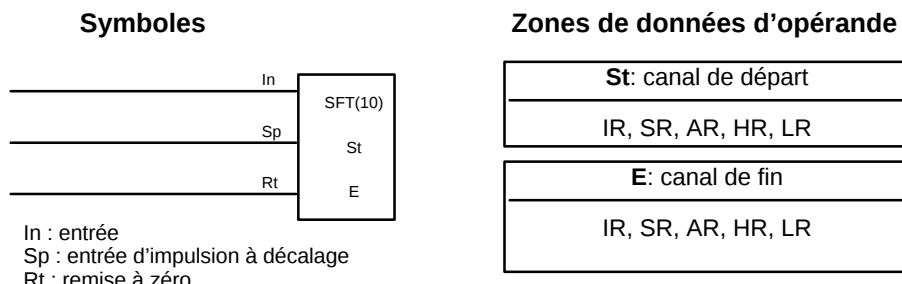
D+1 dépasse la limite de zone de données (C=000)

Erreur dans les sélections d'opérande

PRV(—) est exécutée dans un sous-programme d'interruption alors qu'une instruction d'E/S par impulsion ou de compteur rapide est exécutée dans le programme principal.

5-16 Registre à décalage

5-16-1 SHIFT REGISTER – SFT(10)



Limites

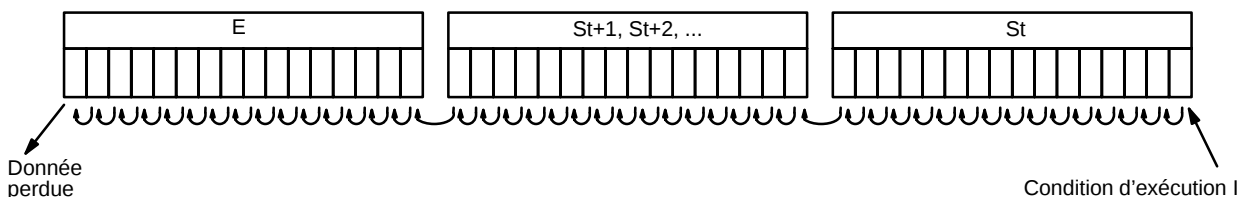
Fonctionne comme relais sériel de décalage d'entrées. Décale une donnée vers la gauche en unité de bits.

E doit être supérieur ou égal à St et St et E doivent appartenir à la même zone de données.

Si une adresse de bit de l'un des canaux utilisés dans le registre de décalage est également utilisée dans une instruction contrôlant l'état individuel des bits (par ex. OUT, KEEP(11)), une erreur ("COIL/OUT DUPL") est générée lors de la vérification de la syntaxe du programme sur la console de programmation ou sur le logiciel Sysmate LSS. Le programme sera toutefois exécuté comme il est écrit. Cf exemple 2 : "Contrôle des bits dans les registres à décalage" pour illustrer ce type de programmation.

Description

SFT(10) est contrôlé par trois conditions d'exécution : In, Sp et Rt. SFT(10) est exécutée si la condition des Sp est à ON alors qu'elle était à OFF à la dernière exécution et si Rt est à OFF, alors, la condition In est décalée dans le bit d'extrême droite d'un registre de décalage défini entre St et E, c'est-à-dire que si In est à ON, une valeur de 1 est décalée dans le registre ; si In est à OFF, un "0" est décalé. Lorsque In est décalé dans le registre, tous les bits précédemment présents dans le registre sont décalés vers la gauche et le bit d'extrême gauche du registre est perdu.



La condition d'exécution de P fonctionne comme une instruction sur changement d'état, c'est-à-dire que I est décalé dans le registre seulement lorsque P est à ON et était à OFF à la dernière exécution de SFT(10). Si la condition de P n'a pas été modifiée ou est passée de ON à OFF, le registre à décalage reste inchangé.

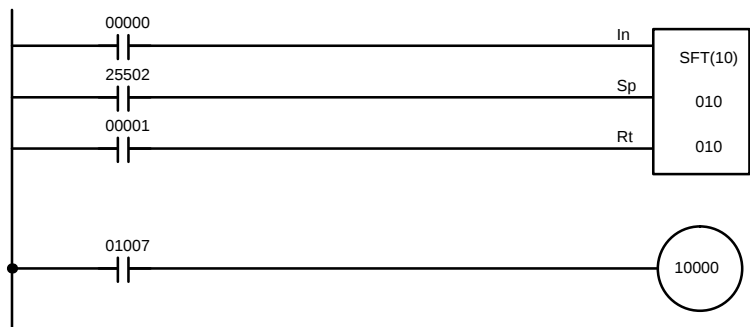
St désigne le canal de départ ; E désigne le canal d'extrême gauche (dernier canal du registre). Le registre à décalage comprend ces deux canaux et tous ceux qui sont situés entre eux. Un même canal peut être désigné pour St et E afin de créer un unique registre à décalage à 16 bits.

Lorsque la condition d'exécution Rt passe à ON, tous les bits du registre à décalage passent à OFF (c'est-à-dire à 0) et le registre à décalage ne fonctionne pas jusqu'à ce que Rt passe à OFF.

Drapeaux

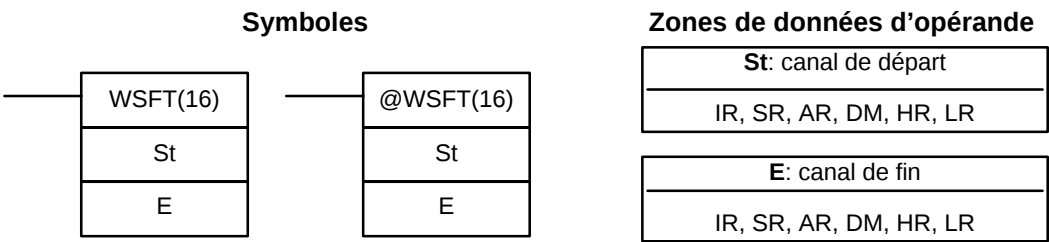
Aucun drapeau n'est affecté par SFT(10).

Exemple L'exemple suivant utilise un bit d'impulsion d'horloge d'une seconde, Sp, (25502) de façon à ce que la condition produite par 00000 (In) soit décalée dans IR 010 toutes les secondes. La sortie 10000 passe à ON à chaque fois qu'un "1" est décalé dans 01007.



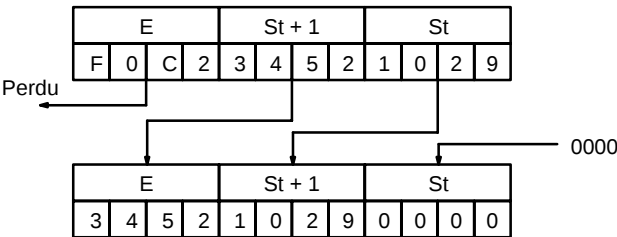
Adresse	Instruction	Opérande
00000	LD	In 00000
00001	LD	Sp 25502
00002	LD	Rt 00001
00003	SFT(10)	
		St 010
		E 010
00004	LD	01007
00005	OUT	10000

5-16-2 WORD SHIFT – DECALAGE DE MOTS - WSFT(16)



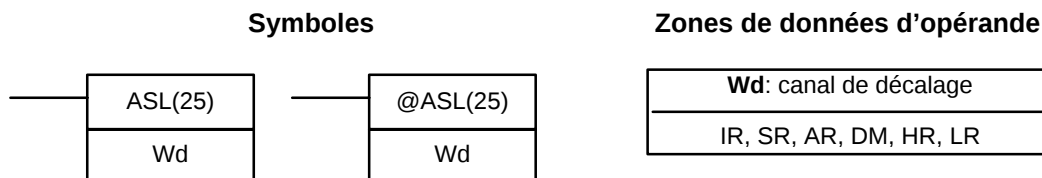
Limites Registre à décalage de mot de 16 bits. St et E doivent appartenir à la même zone de données et E doit être supérieur ou égal à St.
Les DM 6144 à DM 6655 ne peuvent être utilisés pour St ni E.

Description Lorsque la condition d'exécution est à OFF, WSFT(16) n'est pas exécutée. Lorsque la condition est à ON, WSFT(16) décale les données entre St et E par unités de canal (16 bits). Les zéros sont écrits dans St et le contenu de E est perdu.



Drapeaux **ER:** Les canaux St et E appartiennent à ces zones différentes ou St est supérieur à E.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-16-3 ARITHMETIC SHIFT LEFT – DECALAGE ARITHMETIQUE VERS LA GAUCHE – ASL(25)

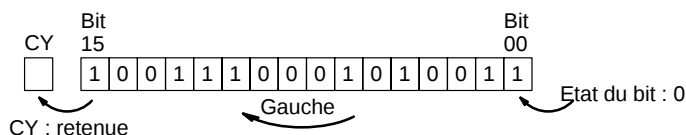


Limites

Décale une donnée de canal d'un bit vers la gauche avec retenue.
Les DM 6144 à 6655 ne peuvent être utilisés pour Wd.

Description

Lorsque la condition d'exécution est à OFF, ASL(25) n'est pas exécutée. Lorsque la condition est à ON, ASL(25) décale un 0 dans le bit 00 de Wd, décale les bits de Wd d'un bit vers la gauche et décale l'état du bit 15 dans CY.



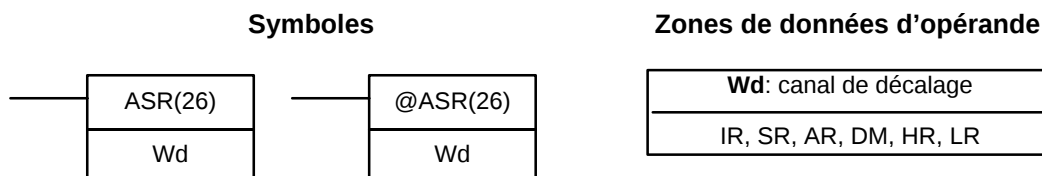
Précautions

Un 0 est décalé dans le bit 00 à chaque tour de scrutation si la forme simple de ASL(25) est utilisée. Utilisez la forme sur changement d'état (@ASL(25)) ou combinez ASL(25) et DIFU(13) ou DIFD(14) pour effectuer un seul décalage.

Drapeaux

- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- CY:** Reçoit l'état du bit 15.
- EQ:** ON lorsque le contenu de Wd est égal à zéro ; reste à OFF dans les autres cas.

5-16-4 ARITHMETIC SHIFT RIGHT – DECALAGE ARITHMETIQUE VERS LA DROITE – ASR(26)

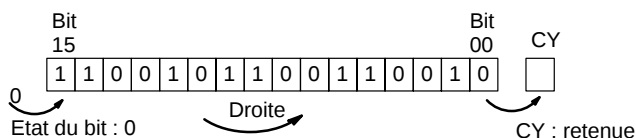


Limites

Décale une donnée de canal d'un bit vers la droite avec retenue. Les DM 6144 à 6655 ne peuvent être utilisés pour Wd.

Description

Lorsque la condition est à OFF, ASR(25) n'est pas exécutée. Lorsque la condition est à ON, ASR(25) décale un 0 dans le bit 15 de Wd, décale les bits de Wd d'un bit vers la droite et décale l'état du bit 00 dans CY.



Précautions

Un 0 est décalé dans le bit 15 à chaque tour de scrutation si la forme simple de ASR(26) est utilisée. Utilisez la forme sur changement d'état (@ASL(26)) ou combinez ASL(26) et DIFU(13) ou DIFD(14) pour effectuer un seul décalage.

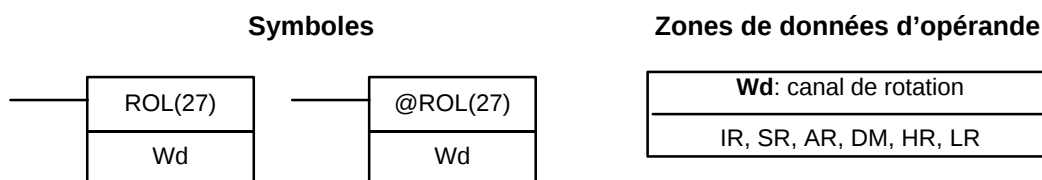
Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

CY: Reçoit les données du bit 00.

EQ: ON lorsque le contenu de Wd est égal à zéro ; reste à OFF dans les autres cas.

5-16-5 ROTATE LEFT – PERMUTATION CIRCULAIRE VERS LA GAUCHE – ROL(27)

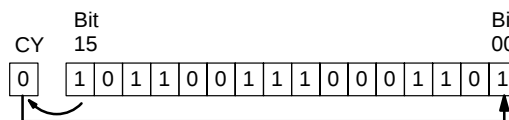
**Limites**

Effectue une permutation circulaire d'un bit vers la gauche pour une donnée à 16 bits, retenue comprise.

Les DM 6144 à 6655 ne peuvent être utilisés pour Wd.

Description

Lorsque la condition est à OFF, ROL(27) n'est pas exécutée. Lorsque la condition est à ON, ROL(27) décale tous les bits Wd d'un bit vers la gauche, décale CY dans le bit 00 de Wd et décale le bit 15 de Wd dans CY.

**Précautions**

Utilisez STC(41) pour sélectionner l'état de CY ou CLC(41) pour effacer l'état de CY avant d'effectuer une rotation afin de s'assurer que CY contient l'état qui convient avant l'exécution de ROL(27).

CY est décalé dans le bit 00 à chaque tour de scrutation si la forme simple de ROL(27) est utilisée. Utilisez la forme sur changement d'état (@ROL(27)) ou combinez ROL(27) et DIFU(13) ou DIFD(14) pour effectuer un seul décalage.

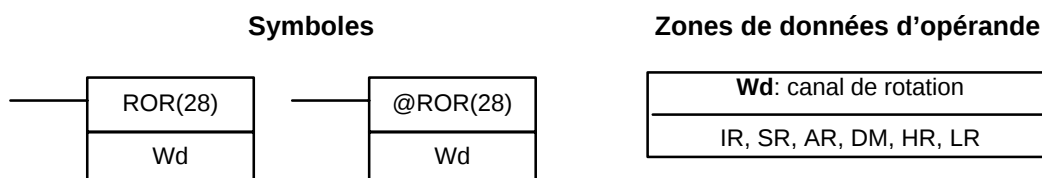
Flags

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

CY: Reçoit les données du bit 15.

EQ: ON lorsque le contenu de Wd est égal à zéro ; reste à OFF dans les autres cas.

5-16-6 ROTATE RIGHT – PERMUTATION CIRCULAIRE VERS LA DROITE – ROR(28)

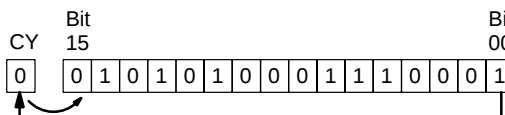


Limites

Effectue la permutation circulaire d'un bit vers la droite d'une donnée de 16 bits, retenue comprise. Les DM 6144 à 6655 ne peuvent être utilisées pour Wd.

Description

Lorsque la condition d'exécution est à OFF, ROR(28) n'est pas exécutée. Lorsque la condition est à ON, ROR(28) décale tous les bits de Wd d'un bit vers la droite, décale CY dans le bit 15 de Wd et décale le bit 00 de Wd dans CY.

**Précautions**

Utilisez STC(41) pour sélectionner l'état de CY ou CLC(41) pour effacer l'état de CY avant d'effectuer une rotation afin de s'assurer que CY contient l'état qui convient avant l'exécution de ROR(28).

CY est décalé dans le bit 15 à chaque tour de scrutation si la forme simple de ROR(28) est utilisée. Utilisez la forme sur changement d'état (@ROR(28)) ou combinez ROR(28) et DIFU(13) ou DIFD(14) pour effectuer un seul décalage.

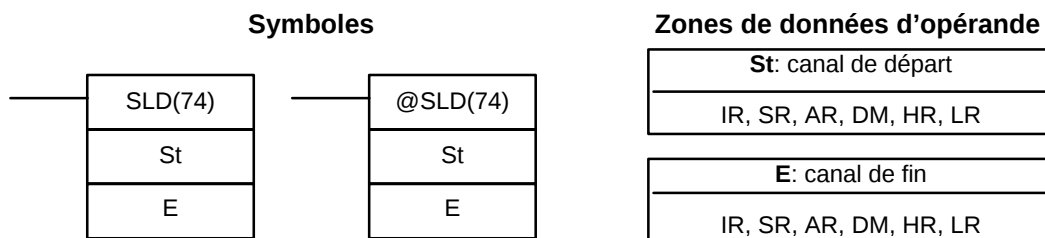
Flags

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

CY: Reçoit les données du bit 00.

EQ: ON lorsque le contenu de Wd est égal à zéro ; reste à OFF dans les autres cas.

5-16-7 ON DIGIT SHIFT LEFT – DECALAGE D'UN DIGIT VERS LA GAUCHE – SLD(74)

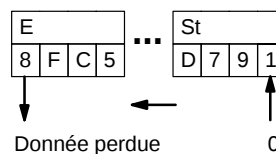
**Limites**

Décalage de 4 bits vers la gauche des données comprises entre les canaux de début et de fin. St et E doivent appartenir à la même zone de données et E doit être supérieur ou égal à St.

Les DM 6144 à 6655 ne peuvent être utilisés pour St ni pour E.

Description

Lorsque la condition d'exécution est à OFF, SLD(74) n'est pas exécutée. Lorsque la condition est à ON, SLD(74) décale les données entre St et E (E compris) d'un chiffre ou "digit" (4 bits) vers la gauche. 0 est inscrit dans le digit d'extrême droite de St et le contenu du digit d'extrême gauche de E est perdu.

**Précautions**

Si une coupure de courant se produit pendant un décalage de plus de 50 canaux, il se peut que le décalage ne soit effectué que partiellement.

Un 0 est décalé dans le bit de poids le plus faible de St à chaque tour de scrutation si la forme simple de SLD(74) est utilisée. Utilisez la forme sur changement

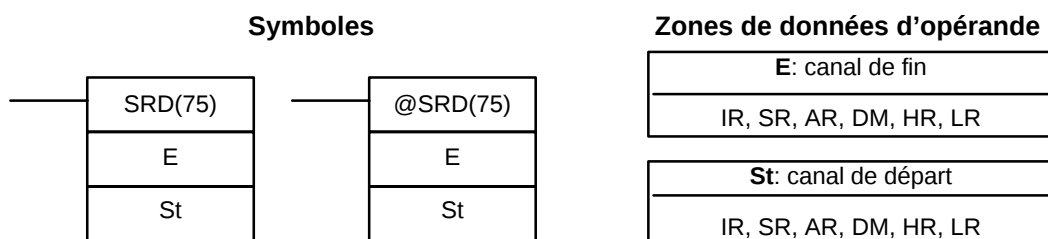
d'état (@SLD(74)) ou combinez SLD(74) et DIFU(13) ou DIFD(14) pour effectuer un seul décalage.

Drapeaux

ER: Les canaux St et E appartiennent à des zones différentes ou St est supérieur à E.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-16-8 ONE DIGIT SHIFT RIGHT – DECALAGE D'UN DIGIT VERS LA DROITE – SRD(75)



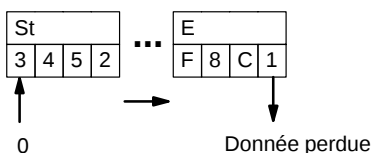
Limites

Décale d'un bit vers la droite les données comprises entre les canaux de début et de fin. St et E doivent appartenir à la même zone de données et E doit être supérieur ou égal à St.

Les DM 6144 à 6655 ne peuvent être utilisés pour St ni pour E.

Description

Lorsque la condition d'exécution est à OFF, SRD(75) n'est pas exécutée. Lorsque la condition est à ON, SRD(75) décale les données entre St et E (E compris) d'un digit (4 bits) vers la droite. 0 est inscrit dans le digit d'extrême gauche de St et le digit d'extrême droite de E est perdu.



Précautions

Si une coupure de courant se produit pendant un décalage de plus de 50 canaux, il se peut que le décalage ne soit effectué que partiellement.

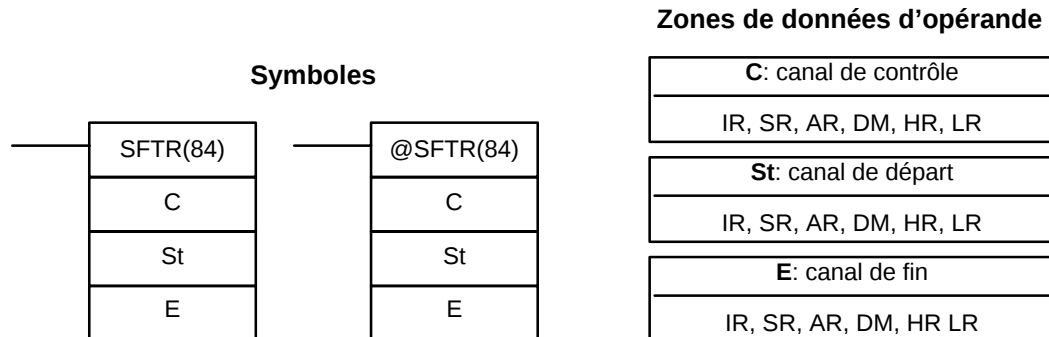
Un 0 est décalé dans le bit de poids le plus fort de St à chaque tour de scrutation si la forme simple de SRD(75) est utilisée. Utilisez la forme sur changement d'état (@SRD(75)) ou combinez SRD(75) et DIFU(13) ou DIFD(14) pour effectuer un seul décalage.

Drapeaux

ER: Les canaux St et E appartiennent à des zones différentes ou St est inférieur à E.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-16-9 REVERSIBLE SHIFT REGISTER – REGISTRE A DECALAGE REVERSIBLE – SFTR(84)



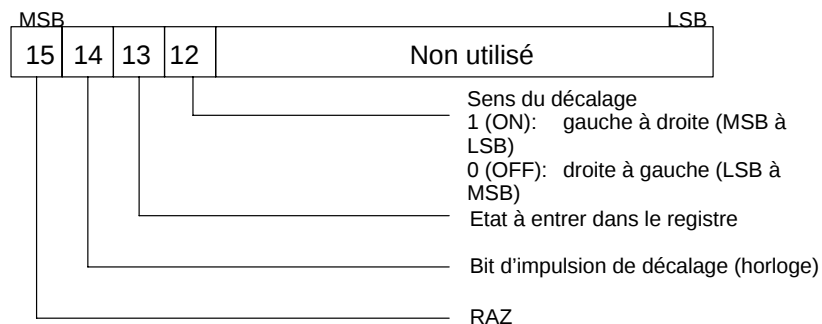
Limites

Décale une donnée de 16 bits par unité de 1 bit vers la droite ou vers la gauche. Réserver un canal de contrôle et un ou plusieurs canaux qui formeront le registre à décalage réversible. St et E doivent appartenir à la même zone de données et St doit être supérieur ou égal à E.

Les DM 6144 à 6655 ne peuvent être utilisés pour C, ni pour St, ni pour E.

Description

SFTR(84) s'utilise pour créer un registre de décalage à canal unique ou à plusieurs canaux capable de décaler les données vers la droite ou vers la gauche. Pour créer un registre à canal unique, désignez le même canal pour St et E. Le canal de contrôle produit le sens du décalage, l'état à entrer dans le registre, l'impulsion de décalage et l'entrée de RAZ. Le canal de contrôle s'attribue de la façon suivante :



Les données du registre à décalage sont décalées d'un bit dans le sens indiqué par le bit 12, décalant ainsi un bit dans CY et l'état du bit 13 à l'autre extrémité à chaque fois que SFTR(84) est exécutée avec une condition ON aussi longtemps que le bit de RAZ est à OFF et le bit 14 à ON. Si SFTR(84) est exécutée avec une condition OFF ou si SFTR(84) est exécutée avec le bit 14 à OFF, le registre à décalage reste inchangé. Si SFTR(84) est exécutée avec une condition à ON et le bit de RAZ (bit 15) à OFF, le registre à décalage entier et CY sont remis à zéro.

Drapeaux

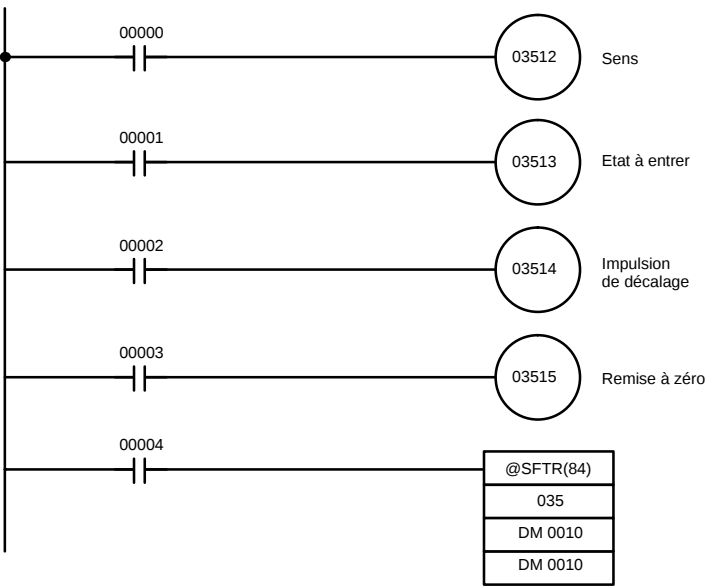
ER: St et E n'appartiennent pas à la même zone de données ou ST est supérieur à E.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

CY: Reçoit l'état du bit 00 de St ou du bit 15 de E, selon la direction du décalage.

Exemple

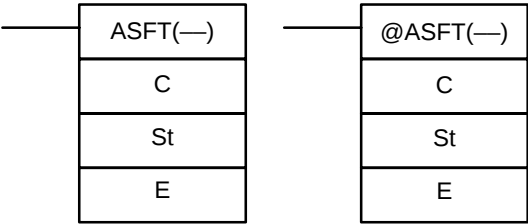
Dans l'exemple suivant, IR 00000, IR 00001, IR 00002 et IR 00003 sont utilisés pour contrôler les bits de C utilisés dans @SFTR(84). Le registre à décalage est dans DM 0010 et il est contrôlé par IR 00004.



Adresse	Instruction	Opérande
00000	LD	00000
00001	OUT	03512
00002	LD	00001
00003	OUT	03513
00004	LD	00002
00005	OUT	03514
00006	LD	00003
00007	OUT	03515
00008	LD	00004
00009	@SFTR(84)	
		035
		DM 0010
		DM 0010

5-16-10 ASYNCHRONOUS SHIFT REGISTER – REGISTRE A DECALAGE REVERSIBLE ET ASYNCHRONE – ASFT(—)

Symboles



Zones de données d'opérande

C: canal de contrôle
IR, SR, AR, DM, HR, LR, #
St: canal de départ
IR, SR, AR, DM, HR, LR
E: canal de fin
IR, SR, AR, DM, HR, LR

Limites

St et E doivent appartenir à la même zone de données et E doit être supérieur ou égal à St.

Les DM 6144 à 6655 ne peuvent être utilisés pour St ni pour E.

Description

Lorsque la condition d'exécution est à OFF, ASFT(—) n'effectue aucune opération et le programme passe à l'instruction suivante. Lorsque la condition est à ON, ASFT(—) est utilisée pour créer et contrôler un registre à décalage de canal réversible et asynchrone entre St et E. Ce registre ne décale les canaux que lorsque le canal suivant dans le registre est égal à zéro (par ex., si aucun canal du registre ne contient 0, aucun décalage n'est effectué). Un seul canal est décalé pour chaque canal du registre contenant zéro. Lorsque le contenu d'un canal est décalé vers le canal suivant, le contenu du canal original est remis à zéro. Lorsque le registre est décalé, chaque canal à zéro du registre prend la place du canal suivant (cf exemple ci-dessous).

Le sens du décalage (le "canal suivant" est le canal le plus important ou le moins important qui suit) est désigné dans C. C est également utilisé pour remettre à

zéro le registre. N'importe quelle partie du registre peut être remise à zéro en désignant la partie concernée à l'aide de St et E.

Canal de contrôle

Les bits 00 à 12 de C ne sont pas utilisés. Le bit 13 est le sens du décalage : placez le bit 13 à ON pour décaler vers le bas (vers les canaux portant des adresses moins importantes) et à OFF pour décaler vers le haut (vers les canaux portant des adresses plus importantes). Le bit 14 est le bit d'activation du décalage : placez le bit 14 à ON pour activer le décalage en fonction de l'état du bit 13 ou à OFF pour désactiver le registre. Le bit 15 est le bit de RAZ : le registre est remis à zéro entre St et E lorsque ASFT(—) est exécuté avec le bit 15 à ON. Placez le bit 15 à OFF pour obtenir un fonctionnement normal.

Rem. : si l'on utilise la forme simple de ASFT(—) les données seront décalées à chaque tour de scrutation si la condition est à ON. Utilisez la forme sur changement d'état de l'instruction pour éviter cela.

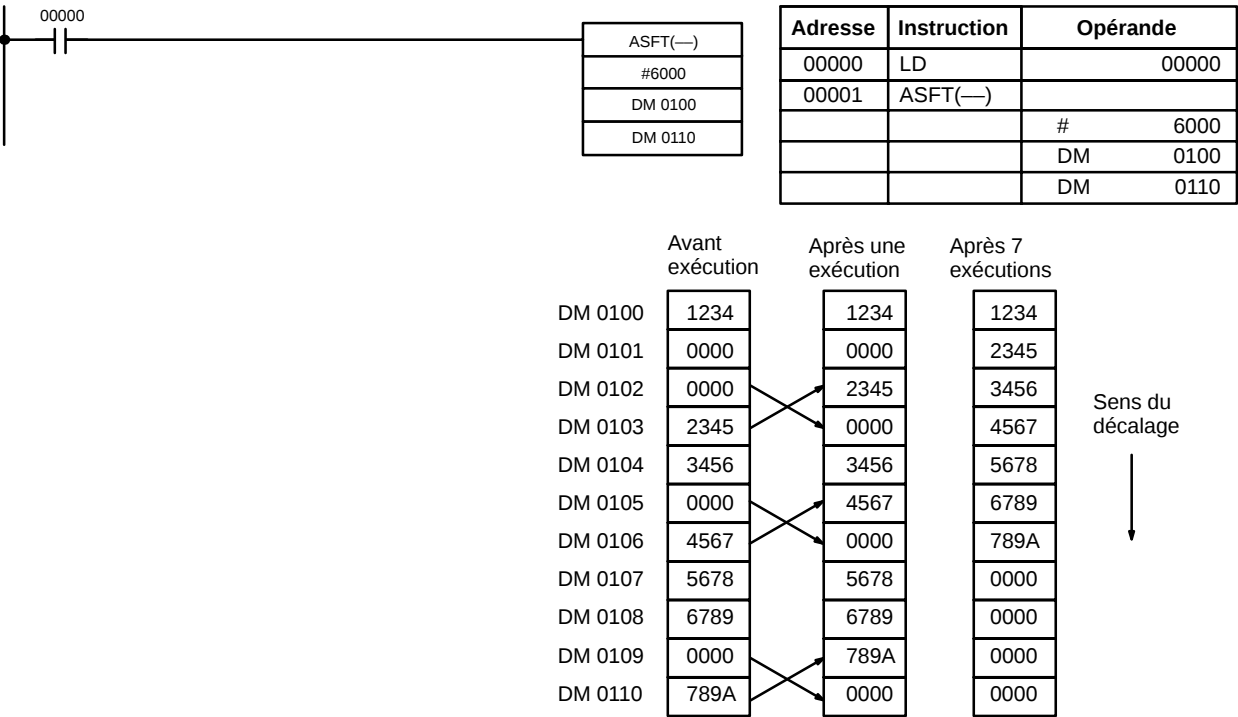
Drapeaux

ER: Les canaux St et E sont situés dans des zones différentes ou St est supérieur à E.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple

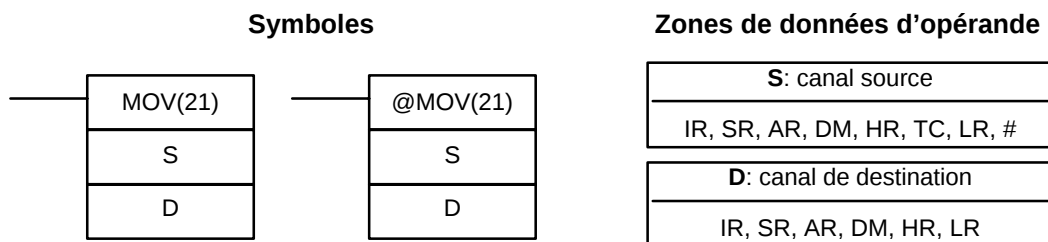
L'exemple suivant montre l'instruction ASFT(—) utilisée pour décaler des canaux dans un registre à décalage de 11 canaux créé entre les DM 0100 et 0110 avec C=#6000. Les données qui ne sont pas des zéros sont décalées vers St (DM 0110).



Les zéros sont décalés vers le haut si C=4000 et le registre à décalage entier est remis à zéro si C=8000.

5-17 Instructions de déplacement de données

5-17-1 MOVE – TRANSFERT – MOV(21)

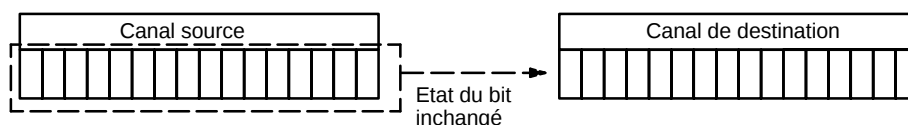


Limites

L'instruction MOV copie une donnée du canal ou une constante à 4 chiffres (16 bits) vers un canal spécifié. Les DM 6144 à 6655 ne peuvent être utilisés pour D.

Description

Lorsque la condition est à OFF, MOV(21) n'est pas exécutée. Lorsque la condition est à ON, MOV(21) copie le contenu de S dans D.



Précautions

Les numéros de TC/CNT ne peuvent être désignés comme valeur D pour changer la PV (valeur en cours) de la tempo. ou du compteur. On peut facilement y parvenir en utilisant BSET(71).

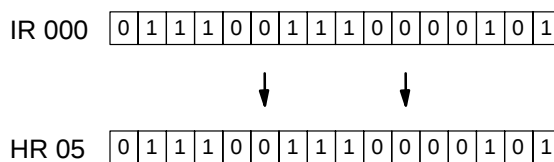
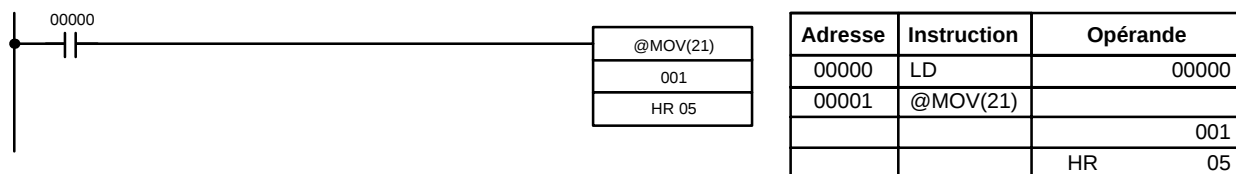
Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

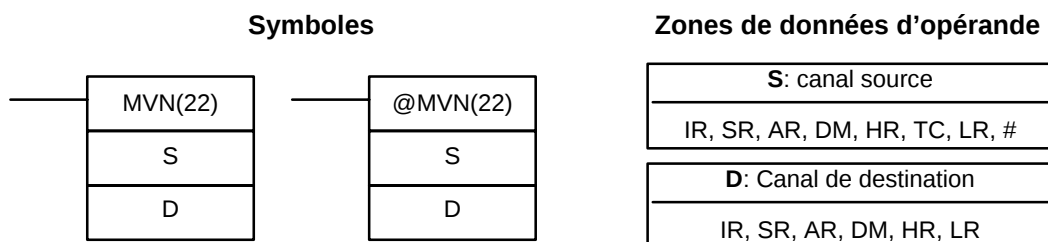
EQ: ON lorsque tous les zéros sont transférés vers D.

Exemple

L'exemple suivant montre l'utilisation de @MOV(21) pour copier le contenu de IR 001 dans HR 05 lorsque IR 00000 passe de OFF à ON.



5-17-2 MOVE NOT – TRANSFERT COMPLEMENTE – MVN(22)

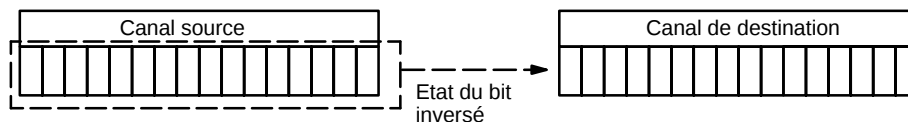


Limites

L'instruction MVN inverse une donnée de canal ou une constante à 4 chiffres et la copie vers un canal spécifié. Les DM 6144 à 6655 ne peuvent être utilisés pour D.

Description

Lorsque la condition d'exécution est à OFF, MVN(22) n'est pas exécutée. Lorsque la condition est à ON, MVN(22) transfère le contenu inversé de S (canal spécifié ou constante à 4 chiffres hexadécimaux) dans D, c'est-à-dire que pour chaque bit de S à ON, le bit correspondant de D est placé à OFF et que pour chaque bit de S à OFF, le bit correspondant de D est placé à ON.

**Précautions**

Les numéros de TC/CNT ne peuvent être désignés comme valeur D pour changer la valeur en cours (PV) de la tempo. ou du compteur. On peut facilement y parvenir en utilisant BSET(71).

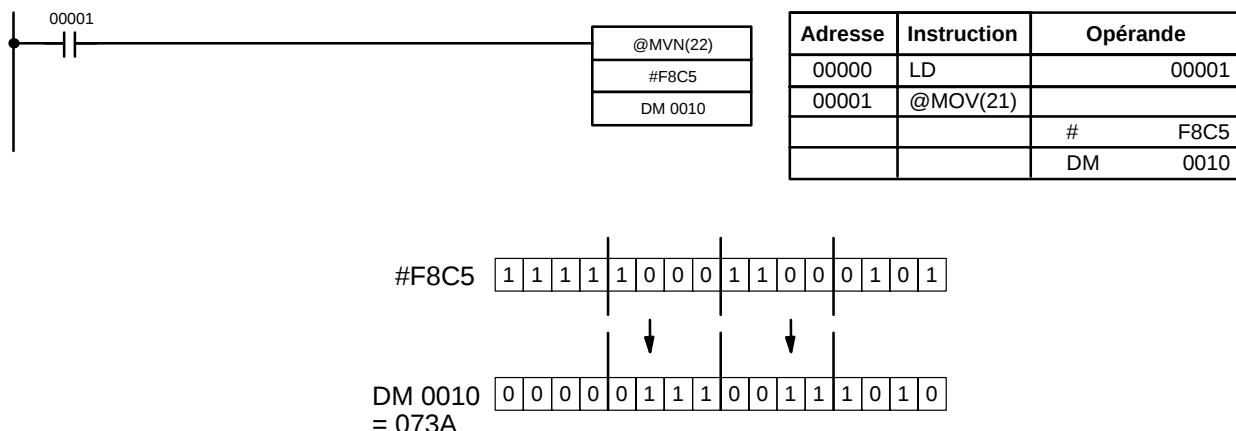
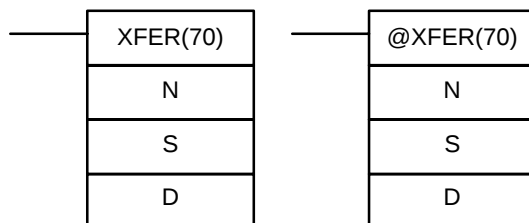
Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque tous les zéros sont transférés vers D.

Exemple

L'exemple suivant montre @MVN(22) utilisé pour copier le complément de #F8C5 dans DM 0010 lorsque IR 00001 passe de OFF à ON.

**5-17-3 BLOCK TRANSFER – TRANSFERT DE BLOC – XFER(70)****Symboles****Zones de données d'opérande**

N: nombre de canaux (BCD)
IR, SR, AR, DM, HR, TC, LR, #
S: canal source de départ
IR, SR, AR, DM, HR, TC, LR
D: canal de destination de départ
IR, SR, AR, DM, HR, TC, LR

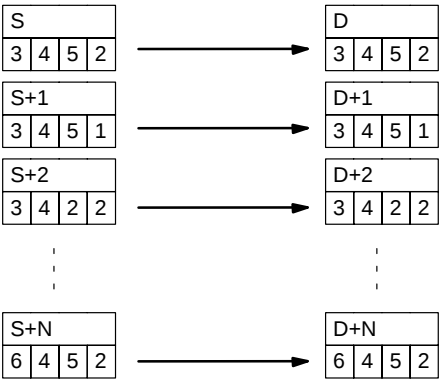
Limites

Transfert de données de canaux consécutifs vers d'autres canaux consécutifs en une seule fois. S et S+N doivent appartenir à la même zone de données, ainsi que D et D+N.

Les DM 6144 à 6655 ne peuvent être utilisés pour D.

Description

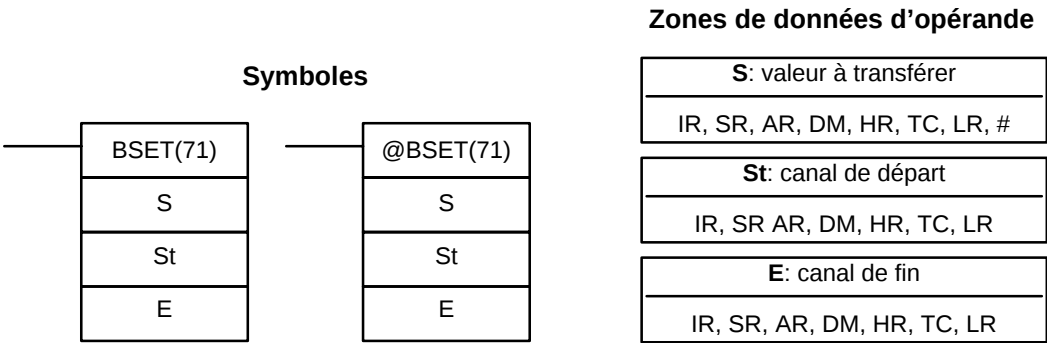
Lorsque la condition d'exécution est à OFF, XFER(70) n'est pas exécutée. Lorsque la condition est à ON, XFER(70) copie le contenu de S, S+1, ..., S+N dans D, D+1, ..., D+N.



Drapeaux

ER: N n'est pas en BCD.
S et S+N ou D et D+N n'appartiennent pas à la même zone de données.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-17-4 BLOCK SET – ACTIVATION DE BLOC – BSET(71)

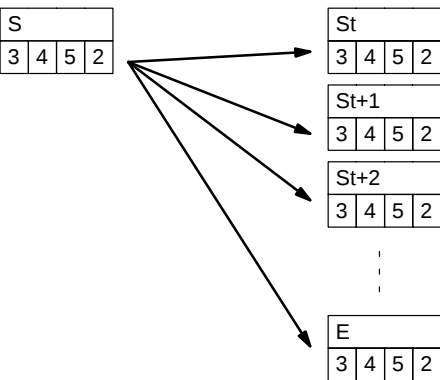


Limites

Initialise une suite de canaux consécutifs avec une valeur identique. St doit être inférieur ou égal à E et St et E doivent appartenir à la même zone de données. Les DM 6144 à 6655 ne peuvent être utilisés pour St ni pour E.

Description

Lorsque la condition d'exécution est à OFF, BSET(71) n'est pas exécutée. Lorsque la condition est à ON, BSET(71) copie le contenu de S dans tous les canaux, de St à E.



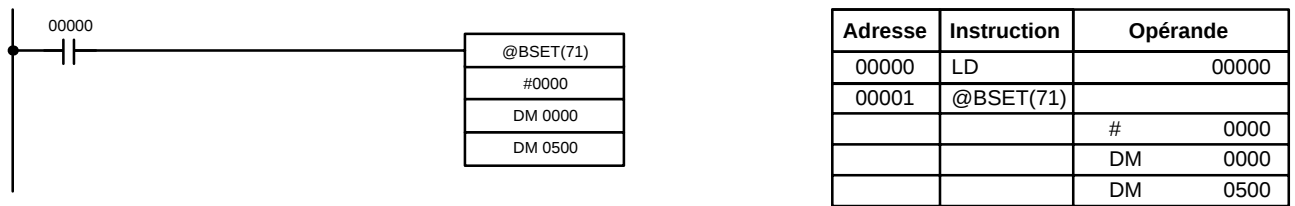
BSET(71) peut être utilisée pour changer la PV des tempo./compteurs (cela ne peut être fait avec MOV(21) ni avec MVN(22)) et également pour effacer certaines parties de zones de données, par ex. la zone DM, afin de préparer l'exécution d'autres instructions. BSET(71) peut également être utilisée pour effacer des canaux en transférant tous les zéros.

Drapeaux

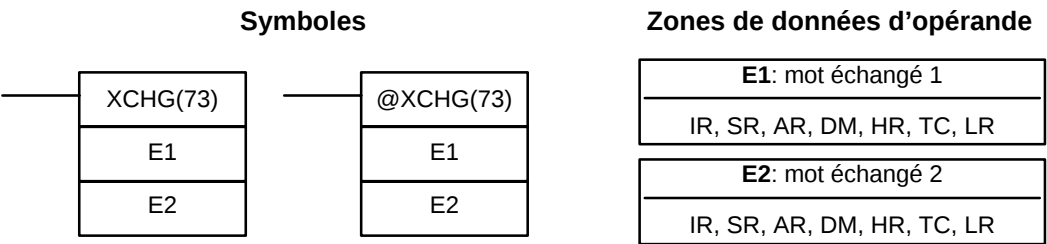
ER: St et E n'appartiennent pas à la même zone de données ou St est supérieur à E.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple

L'exemple suivant montre l'utilisation de BSET(71) pour copier une constante (#0000) dans un bloc de la zone DM (DM 0000 à 0500) lorsque IR 00000 est à ON.



5-17-5 DATA EXCHANGE – ECHANGE DE DONNEES – XCHG(73)

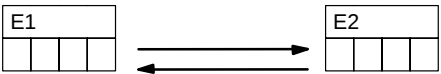


Limites

Echange deux canaux de données à 16 bits entre eux.
Les DM 6144 à 6655 ne peuvent être utilisés pour E1 ni pour E2.

Description

Lorsque la condition d'exécution est à OFF, XCHG(73) n'est pas exécutée. Lorsque la condition est à ON, XCHG(73) échange les contenus de E1 et E2.

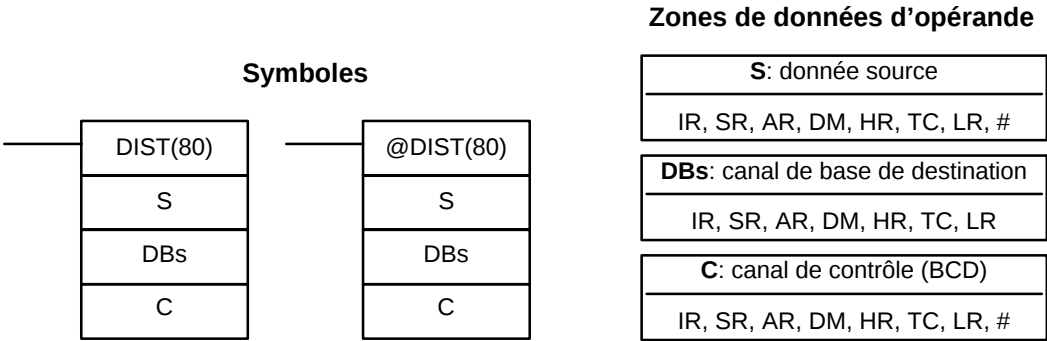


Si l'on désire effectuer un échange de blocs dont la taille dépasse 1 canal, il faut utiliser des canaux de travail comme mémoire tampon intermédiaire en utilisant XFER(70) 3 fois.

Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-17-6 SINGLE WORD DISTRIBUTE – ECRITURE D'UNE TABLE DE DONNEES – DIST(80)



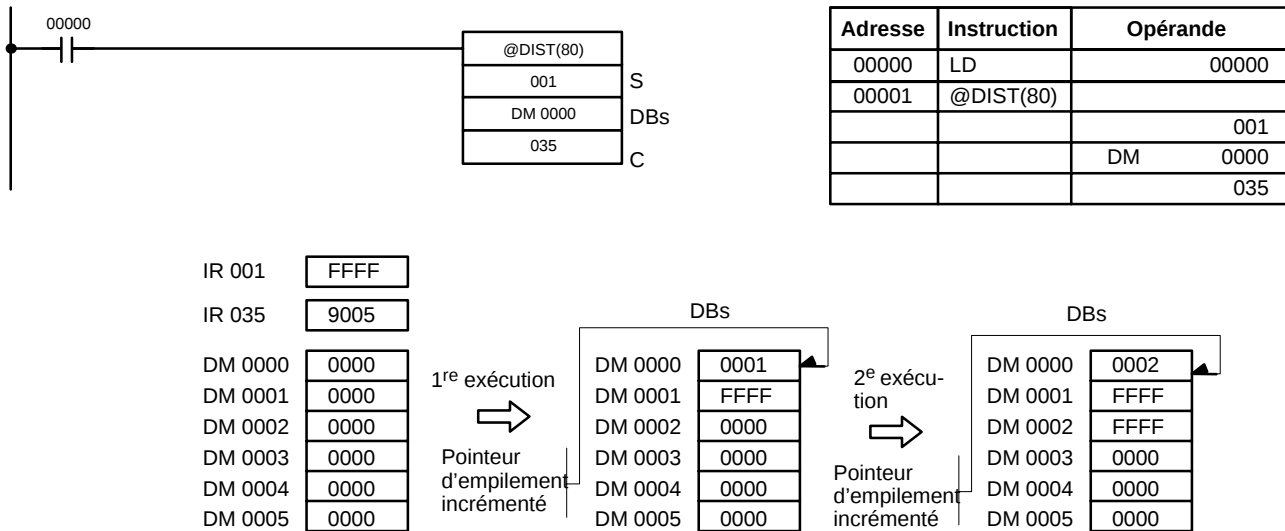
tenu de DBs. En d'autres termes, on ajoute 1 et le contenu de DBs pour obtenir le canal de destination. Le contenu de DBs est alors incrémenté de 1.

Rem. : 1. DIST(80) est exécutée à chaque tour de scrutation sauf si sa forme sur changement d'état est utilisée (@DIST(80)) ou si DIST(80) est utilisée avec DIFU(13) ou DIFD(14).

2. Veillez à bien initialiser le pointeur avant d'utiliser DIST(80) en empilement.

Exemple

L'exemple suivant illustre l'utilisation de DIST(80) pour créer un empilement entre DM 0001 et DM 0005. DM 0000 représente le pointeur d'empilement.



Drapeaux

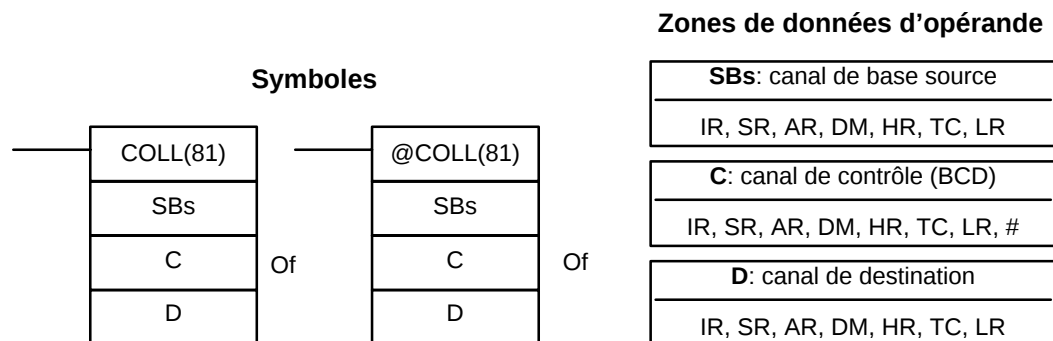
ER: La longueur du décalage ou de l'empilement dans le canal de contrôle n'est pas en BCD.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Pendant l'empilement, la valeur du pointeur d'empilement + 1 dépasse la longueur de l'empilement.

EQ: ON lorsque le contenu de S est égal à zéro ; reste à OFF dans les autres cas.

5-17-7 DATA COLLECT – LECTURE D'UNE TABLE DE DONNEES – COLL(81)



Limites

Permet d'extraire une donnée d'un canal de source et de la transférer dans un canal de destination. Le canal de destination est déterminé par l'addition du canal de source et du canal de décalage. C doit être en BCD.

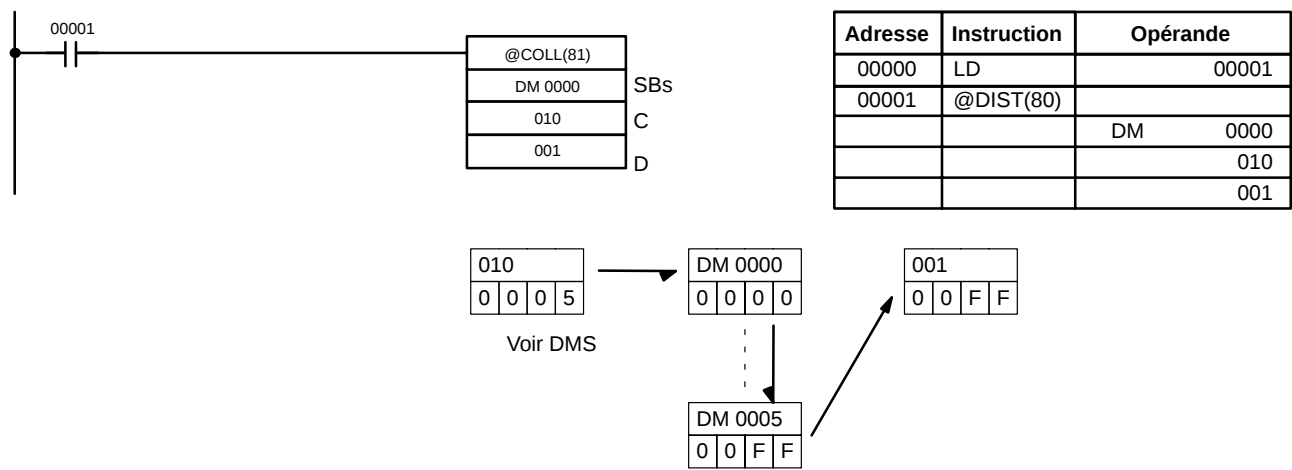
Les DM 6144 à 6655 ne peuvent être utilisées pour D.

Description COLL(81) peut être utilisée pour la collecte des données, une opération d'empilement en FIFO ou en LIFO en fonction du contenu du canal de contrôle C.

Collecte des données Lorsque les bits 12 à 15 de C=0 à 7, COLL(81) est utilisée pour la collecte des données. Le contenu entier de C spécifie un décalage, Of. Lorsque la condition d'exécution est à OFF, COLL(81) n'est pas exécutée. Lorsque la condition est à ON, COLL(81) copie le contenu de SBs + Of dans D, c'est-à-dire que Of est ajoutée à SBs pour déterminer le canal source.

Rem. : SBs et SBs+Of doivent appartenir à la même zone de données.

Exemple L'exemple suivant montre l'utilisation de COLL(81) pour copier le contenu de DM 0000+Of dans IR 001. Le contenu de 010 est égal à #0005 ; le contenu de DM 0005 (DM 0000 + 5) est copié dans IR 001 lorsque IR 00001 est à ON.

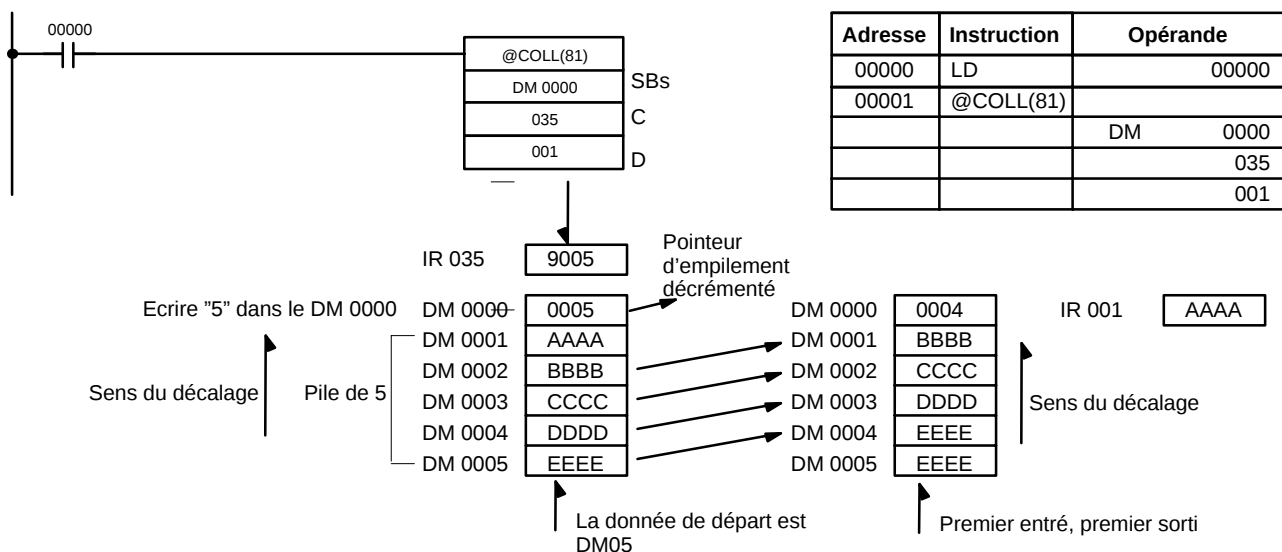


Empilement en FIFO Lorsque les bits 12 à 15 de C=9, COLL(81) peut être utilisée en empilement "FIFO" (premier entré, premier sorti). Les 3 autres chiffres de C spécifient le nombre de canaux de l'empilement (000 à 999). Le contenu de SBs représente le pointeur d'empilement. Lorsque la condition d'exécution est à ON, COLL(81) décale vers le haut le contenu de chaque canal de l'empilement d'une adresse, et décale à la fin les données de SBs+1 (première valeur écrite dans l'empilement) dans le canal de destination (D). Le contenu du pointeur d'empilement (SBs) est alors décrémenté de 1.

Rem. : COLL(81) est exécutée à chaque tour de scrutation, sauf si la forme sur changement d'état (@COLL(81)) ou COLL(81) et DIFU(13) ou DIFD(14) sont utilisées.

Exemple L'exemple suivant illustre l'utilisation de COLL(81) pour créer un empilement entre DM 0001 et DM 0005. DM 0000 représente le pointeur d'empilement. Lorsque IR 00000 passe de OFF à ON, COLL(81) déplace vers le haut le contenu des DM 0002 à 0005 d'une adresse et décale les données de DM 0001

dans IR 001. Le contenu du pointeur d'empilement (DM 0000) est alors décrémenté de 1.



Empilement en LIFO

Lorsque les bits 12 à 15 de C=8, COLL(81) peut être utilisée en empilement "LIFO" (dernier entré, premier sorti). Les 3 autres chiffres de C spécifient le nombre de canaux de l'empilement (000 à 999). Le contenu de SBs représente le pointeur d'empilement.

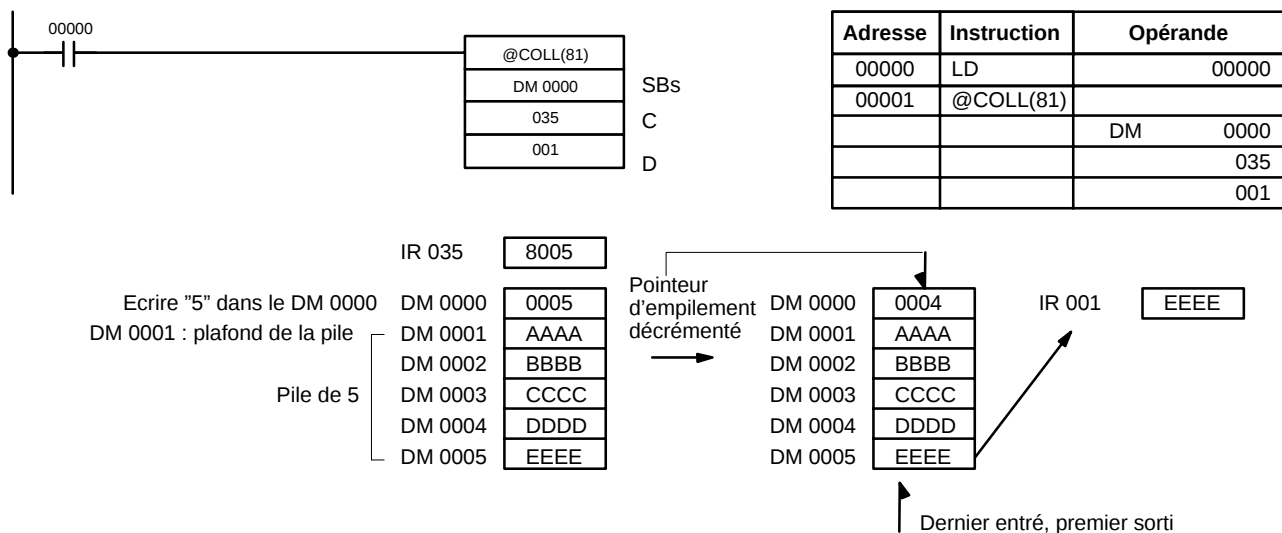
Lorsque la condition d'exécution est à ON, COLL(81) copie les données du canal indiqué par le pointeur d'empilement (SBs + contenu de SBs) dans le canal de destination D. Le contenu du pointeur d'empilement (SBs) est alors décrémenté de 1. Le pointeur est le seul canal modifié de l'empilement.

Rem. : COLL(81) est exécutée à chaque tour de scrutation, sauf si la forme sur changement d'état (@DIST(80)) ou DIST(80) et DIFU(13) ou DIFD(14) sont utilisées.

Exemple

L'exemple suivant illustre l'utilisation de COLL(81) pour créer un empilement entre DM 0001 et DM 0005. DM 0000 représente le pointeur d'empilement.

Lorsque IR 00000 passe de OFF à ON, COLL(81) copie le contenu du DM 0005 (DM 0000 + 5) dans IR 001. Le contenu du pointeur d'empilement (DM 0000) est alors décrémenté de 1.



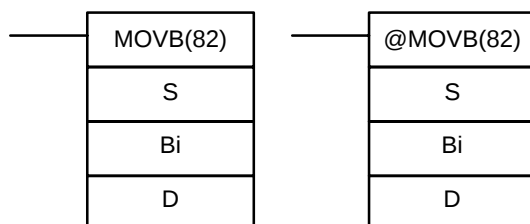
Drapeaux

ER: La longueur du décalage ou de l'empilement dans le canal de contrôle n'est pas en BCD.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Pendant l'empilement, la valeur du pointeur dépasse la longueur de l'empilement. Essai d'écriture dans un canal situé au-delà du canal.

EQ: ON lorsque le contenu de S est égal à zéro ; reste à OFF dans les autres cas.

5-17-8 MOVE BIT – TRANSFERT DE BIT – MOVB(82)**Symboles****Zones de données d'opérande**

S: canal source
IR, SR, AR, DM, HR, LR, #
Bi: désignation du bit (BCD)
IR, SR, AR, DM, HR, TC, LR, #
D: canal de destination
IR, SR, AR, DM, HR, LR

Limites

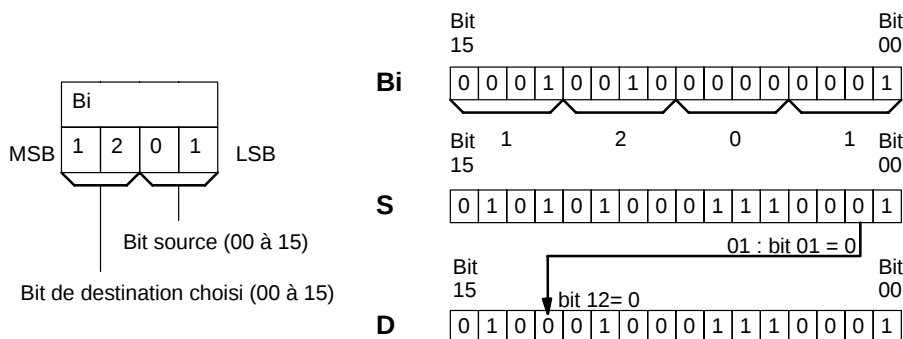
Transfert de bit spécifié d'un canal de source vers un autre bit spécifié dans un canal de destination. Les deux chiffres d'extrême gauche et d'extrême droite de Bi doivent être situés entre 00 et 15.

Les DM 6144 à 6655 ne peuvent être utilisés pour Bi ou D.

Description

Lorsque la condition est à OFF, MOVB(82) n'est pas exécutée. Lorsque la condition est à ON, MOVB(82) copie le bit spécifié de S dans le bit spécifié de D. Les bits de S et D sont spécifiés par Bi.

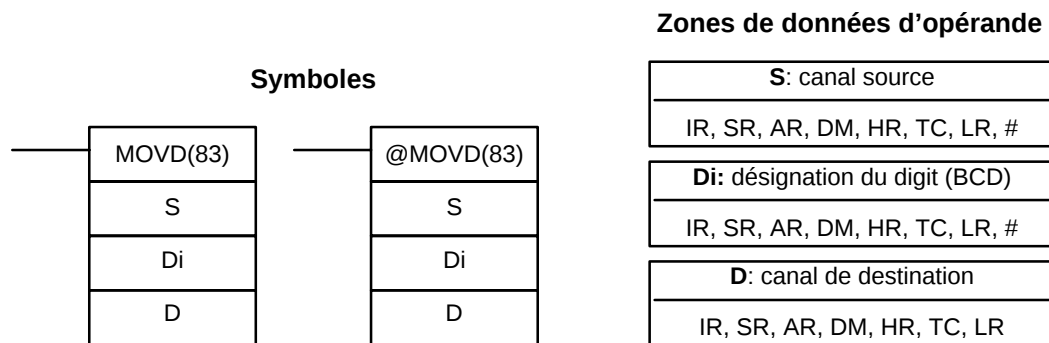
Les 2 chiffres d'extrême droite de Bi désignent le bit source ; les 2 chiffres d'extrême gauche désignent le bit de destination.

**Drapeaux**

ER: Bi n'est pas en BCD ou spécifie un bit qui n'existe pas (le bit spécifié doit être situé entre 00 et 15).

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-17-9 MOVE DIGIT – TRANSFERT DE DIGIT – MOVD(83)

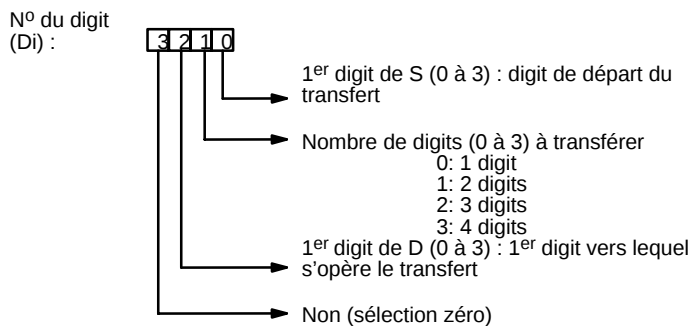
**Limites**

Transfère le contenu hexadécimal d'un digit de source vers un digit de destination. Les trois digits d'extrême droite de Di doivent être situés entre 0 et 3.

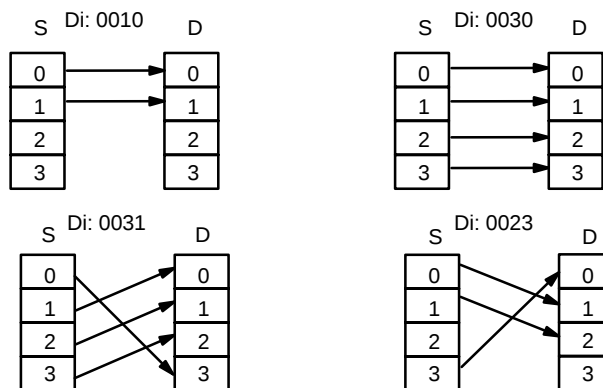
Les DM 6144 à 6655 ne peuvent être utilisés pour Di ni pour D.

Description

Lorsque la condition d'exécution est à OFF, MOVD(83) n'est pas exécutée. Lorsque la condition est à ON, MOVD(83) copie le contenu des digits (ou du digit) spécifié(s) dans S dans le(s) digit(s) spécifié(s) de D. On peut transférer jusqu'à quatre digits à la fois. Le premier digit à copier, le nombre de digits à copier et le premier digit qui reçoit la copie sont désignés dans Di comme ci-dessous. Les digits de S sont copiés dans les digits consécutifs de D en partant du premier digit désigné et jusqu'à ce que le nombre de digits désigné soit atteint. Si le dernier digit est atteint dans S ou dans D, les autres digits sont utilisés en recommençant à partir de 0.

**Désignation de chiffres**

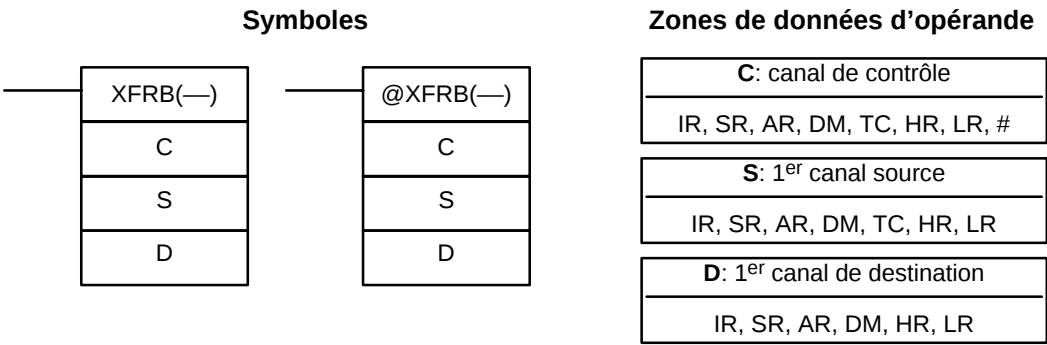
Exemples de mouvements de données pour diverses valeurs de Di :

**Drapeaux**

ER: L'un des 3 chiffres d'extrême droite au moins n'est pas compris entre 0 et 3.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-17-10 TRANSFER BITS – XFRB(—)



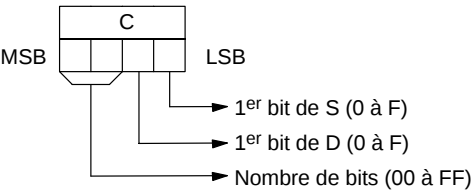
Limites Instruction disponible avec les UC **CQM1-CPU4 [-E.]**

Les bits de source spécifiés doivent appartenir à la même zone de données.

Les bits de destination spécifiés doivent appartenir à la même zone de données.

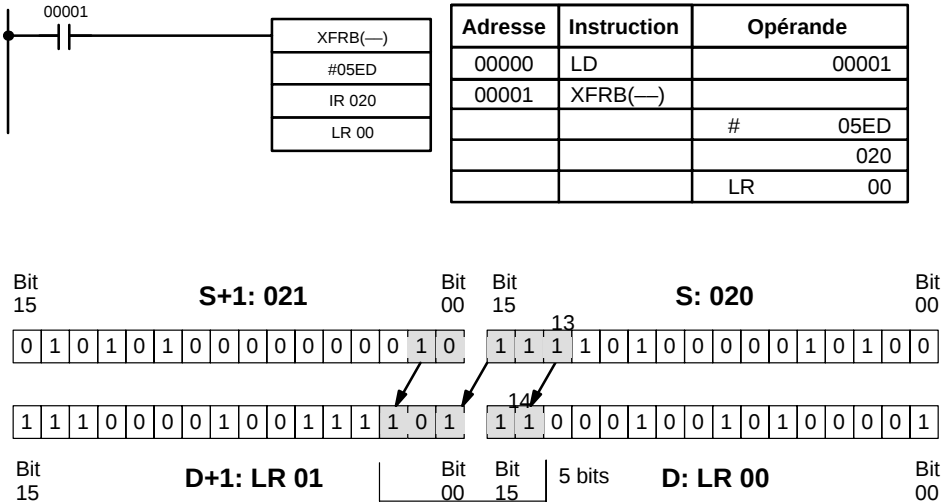
Les DM 6144 à 6655 ne peuvent être utilisés pour D.

Description Lorsque la condition d'exécution est à OFF, XFRB(—) n'est pas exécutée. Lorsque la condition est à ON, XFRB(—) copie les bits de source spécifiés dans les bits de destination spécifiés. Les deux chiffres d'extrême droite de C spécifient les bits de départ de S et D et les deux chiffres d'extrême gauche indiquent le nombre de bits à copier.



On peut copier jusqu'à 255 bits (FF) à la fois ou 16 canaux.

Exemple Dans l'exemple suivant, XFRB(—) est utilisée pour transférer 5 bits situés entre IR 020 et IR 021 dans les LR 00 à 01. Le bit de départ de IR 020 est D (bit13) et le bit de départ dans LR 00 est E (bit14) : les IR 02013 à 02101 sont donc copiés dans les LR 0014 à 0102.



Drapeaux

ER: Les bits de source spécifiés n'appartiennent pas à la même zone de données.
 Les bits de destination spécifiés n'appartiennent pas à la même zone de données.
 Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-18 Instructions de comparaison

5-18-1 COMPARE – COMPARAISON – CMP(20)

Symboles	Zones de données d'opérande
CMP(20) Cp1 Cp2	Cp1: 1^{er} canal de comparaison IR, SR, AR, DM, HR, TC, LR, # Cp2: 2^e canal de comparaison IR, SR, AR, DM, HR, TC, LR, #

Limites

Compare un canal de données ou une constante à 4 chiffres à un autre canal de données. En cas de comparaison d'une valeur à la PV d'une tempo. ou d'un compteur, la valeur doit être en BCD.

Description

Lorsque la condition d'exécution est à OFF, CMP(20) n'est pas exécutée. Lorsque la condition est à ON, CMP(20) compare les valeurs Cp1 et Cp2 et sort le résultat dans les drapeaux GR, EQ et LE de la zone SR.

Précautions

Le fait de placer d'autres instructions entre CMP(20) et l'accès aux drapeaux EQ, LE et GR peut modifier l'état de ces drapeaux. Programmez ces drapeaux immédiatement après l'instruction CMP.

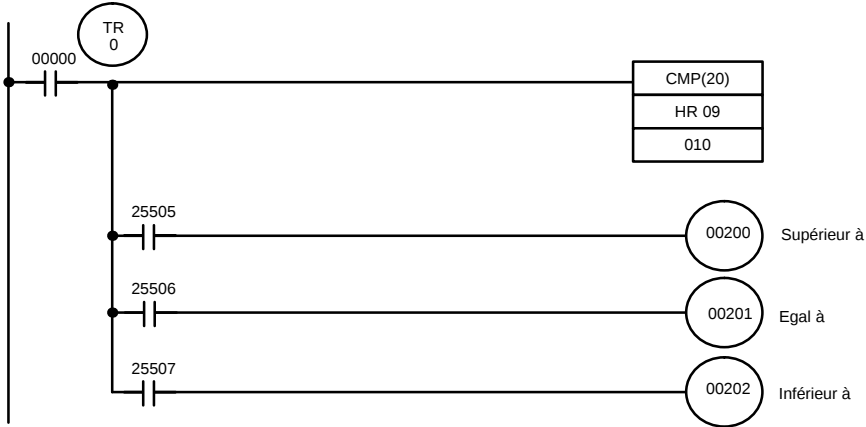
Flags

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
EQ: ON si Cp1 égale Cp2.
LE: ON si Cp1 est inférieur à Cp2.
GR: ON si Cp1 est supérieur à Cp2.

Drapeau	Adresse	C1 < C2	C1 = C2	C1 > C2
GR	25505	OFF	OFF	ON
EQ	25506	OFF	ON	OFF
LE	25507	ON	OFF	OFF

Exemple:
Sauvegarde des résultats
de CMP(20)

L'exemple suivant montre comment sauvegarder immédiatement les résultats d'une comparaison. Si le contenu de HR 09 est supérieur à celui de 010, 00200 passe à ON ; si les deux contenus sont égaux, 00201 passe à ON ; si le contenu de HR 09 est inférieur à celui de 010, 00202 passe à ON. Dans certaines applications, seul un des trois OUT est nécessaire, ce qui rend facultatif l'emploi de TR 0. Dans ce type de programmation, 00200, 00201 et 00202 ne sont modifiés que lorsque CMP(20) est exécutée.



Adresse	Instruction	Opérande
00000	LD	00000
00001	OUT	TR 0
00002	CMP(20)	
		010
		HR 09
00003	LD	TR 0
00004	AND	25505

Adresse	Instruction	Opérande
00005	OUT	00200
00006	LD	TR 0
00007	AND	25506
00008	OUT	00201
00009	LD	TR 0
00010	AND	25507
00011	OUT	00202

5-18-2 TABLE COMPARE – COMPARAISON SUR TABLE – TCMP(85)

Zones de données d'opérande

Symboles

TCMP(85)	@TCMP(85)
CD	CD
TB	TB
R	R

CD:données de comparaison
IR, SR, DM, HR, TC, LR, #
TB: 1 ^{er} canal du tableau de comp.
IR, SR, DM, HR, TC, LR
R: canal de résultat
IR, SR, DM, HR, TC, LR

Limites

Compare une donnée à 16 bits à 16 canaux d'une table et sort le résultat de la comparaison sur un canal spécifié. A chaque fois que la valeur source à comparer est égale à une des 16 valeurs de la table, le bit correspondant dans le canal de résultat sera mis à ON ("1").
Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, TCMP(85) n'est pas exécutée. Lorsque la condition est à ON, TCMP(85) compare CD au contenu de TB, TB+1, TB+2, ..., et TB+15. Si CD est égal au contenu d'un de ces canaux, le bit correspondant de R est sélectionné. Par ex., si CD est égal au contenu de TB, le bit 00 passe à ON, s'il est égal au contenu de TB+1, le bit 01 passe à ON, etc. Le reste des bits de R est placé à OFF.

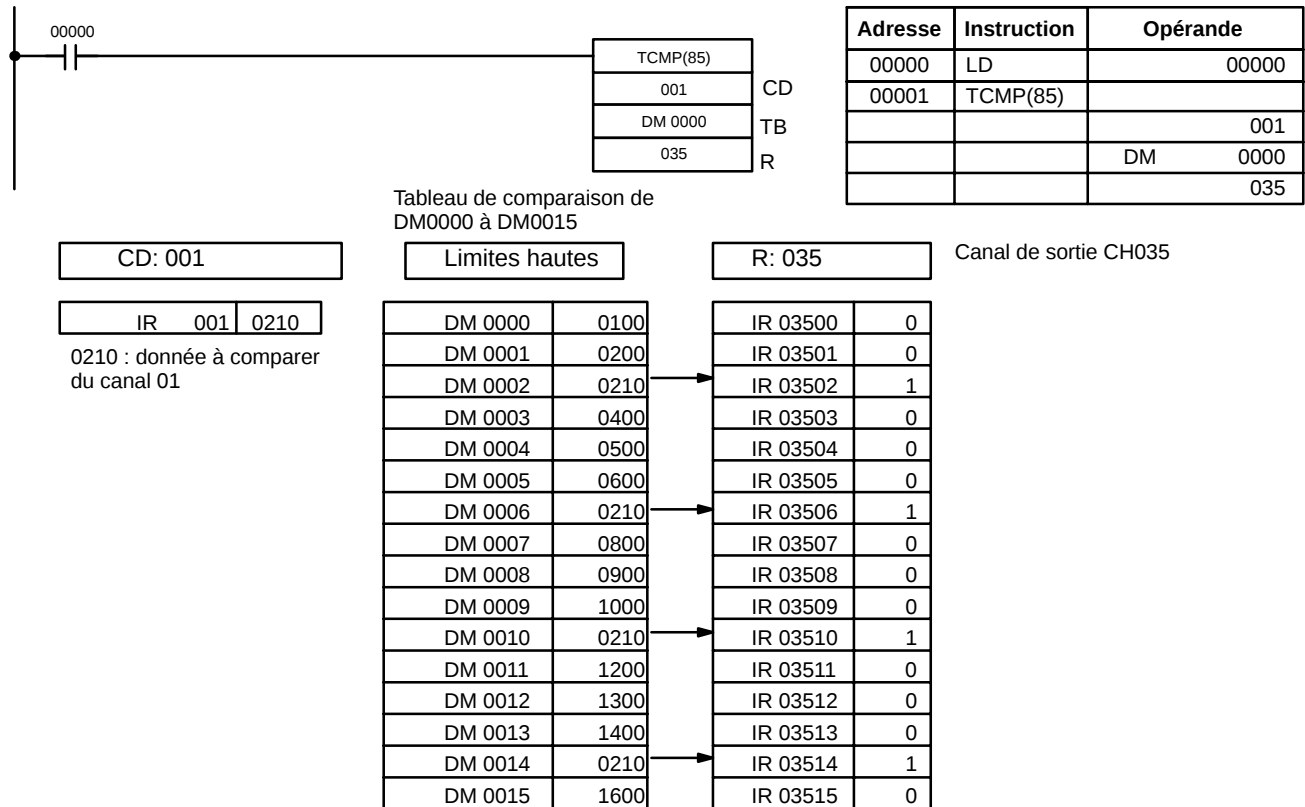
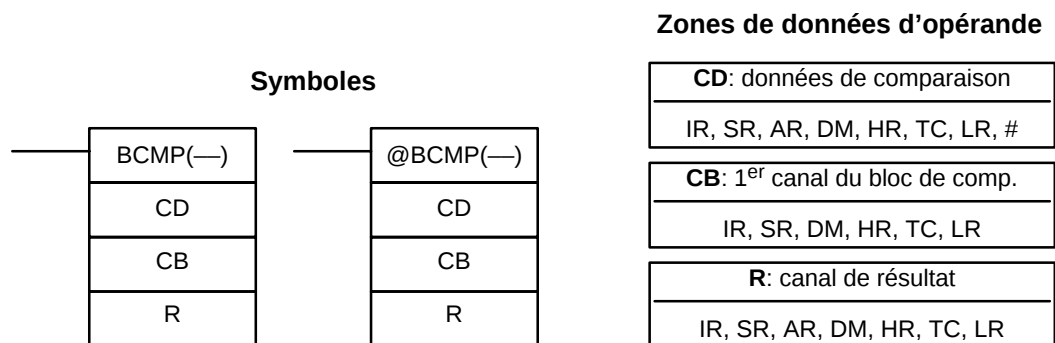
Drapeaux

ER: Le tableau de comparaison (c'est-à-dire TB à TB+15) dépasse la zone de données.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple

L'exemple suivant montre une comparaison et les résultats de TCMP(85). Ici la comparaison est effectuée à chaque tour de scrutation lorsque IR 00000 est à ON.

**5-18-3 BLOCK COMPARE – BCMP(—)****Limites**

BCMP (comparaison de blocs) compare les données à 4 chiffres spécifiques (source) à chacun des 16 niveaux d'une table de comparaison et sort les résultats sur le canal de destination.

Chaque canal de limite basse du bloc de comparaison doit être inférieur ou égal à la limite haute.

Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, BCMP(—) n'est pas exécutée. Lorsque la condition est à ON, BCMP(—) compare CD aux gammes définies par un

bloc composé de CB, CB+1, CB+2, ..., CB+31. Chaque gamme est définie par deux canaux : le premier donne la limite basse et le deuxième la limite haute. Si CD est située entre ces deux gammes (limites haute et basse incluses), le bit correspondant de R est sélectionné. Voici les comparaisons effectuées et le bit correspondant de R sélectionné pour chaque comparaison vraie. Les autres bits de R sont placés à OFF.

Canal de valeur min. : $CB \leq CD \leq CB+1$ (val. max.) Bit 00 : canal de résultat

$CB+2 \leq CD \leq CB+3$	Bit 01
$CB+4 \leq CD \leq CB+5$	Bit 02
$CB+6 \leq CD \leq CB+7$	Bit 03
$CB+8 \leq CD \leq CB+9$	Bit 04
$CB+10 \leq CD \leq CB+11$	Bit 05
$CB+12 \leq CD \leq CB+13$	Bit 06
$CB+14 \leq CD \leq CB+15$	Bit 07
$CB+16 \leq CD \leq CB+17$	Bit 08
$CB+18 \leq CD \leq CB+19$	Bit 09
$CB+20 \leq CD \leq CB+21$	Bit 10
$CB+22 \leq CD \leq CB+23$	Bit 12
$CB+24 \leq CD \leq CB+25$	Bit 13
$CB+26 \leq CD \leq CB+27$	Bit 14
$CB+28 \leq CD \leq CB+29$	Bit 15
$CB+30 \leq CD \leq CB+31$	Bit 16

CD : données à comparer

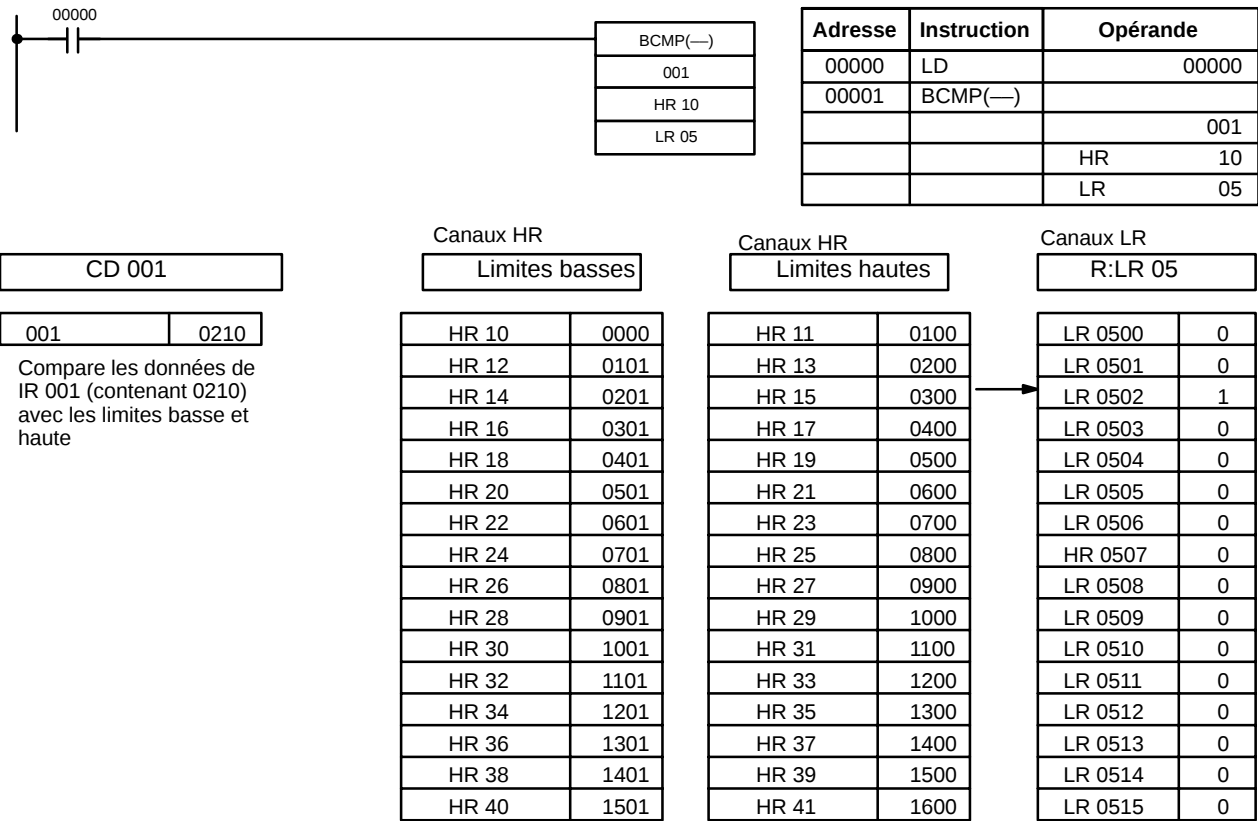
Drapeaux

ER: Le bloc de comparaison (c'est-à-dire CB à CB+31) dépasse la zone de données.

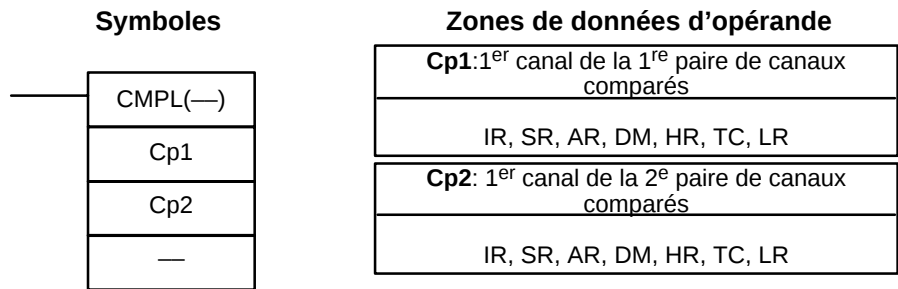
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple

L'exemple suivant illustre une comparaison entre les résultats de BCMP(—). La comparaison est effectuée à chaque tour de scrutation lorsque IR 00000 est à ON.



5-18-4 DOUBLE COMPARE – COMPARAISON A DOUBLE LONGUEUR – CMPL(—)



Limites

Cp1 et Cp1+1 doivent appartenir à la même zone de données (zone de 16 bits).
Cp2 et Cp2+1 doivent appartenir à la même zone de données (zones de 16 bits).

Description

CMPL compare une paire de canaux (16 bits) hexadécimaux à une autre paire de canaux hexadécimaux.
Lorsque la condition d'exécution est à OFF, CMPL(—) n'est pas exécutée. Lorsque la condition est à ON, CMPL(—) rapproche le contenu à 4 chiffres hexadécimaux de Cp1+1 et celui de Cp1, ainsi que le contenu de Cp2+1 et celui de Cp2 pour créer deux nombres à 8 chiffres hexadécimaux (Cp+1, Cp1 et Cp2+1, Cp2). Les deux nombres à 8 chiffres sont alors comparés et le résultat est sorti sur les drapeaux GR, EQ et LE et dans la zone SR.

Précautions

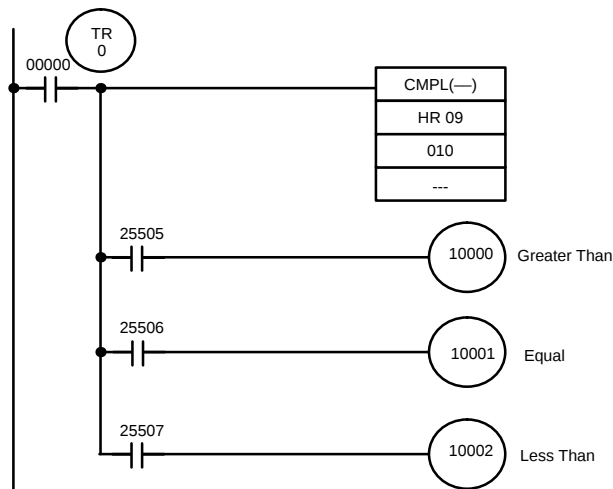
Le fait de placer d'autres instructions entre CMPL(—) et l'accès aux drapeaux EQ, LE et GR peut changer l'état de ces drapeaux. Programmez ces drapeaux jusqu'à l'instruction CMPL.

Drapeaux

- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- GR:** ON si Cp1+1,Cp1 est supérieur à Cp2+1,Cp2.
- EQ:** ON si Cp1+1,Cp1 égale Cp2+1,Cp2.
- LE:** ON si Cp1+1,Cp1 est inférieur à Cp2+1,Cp2.

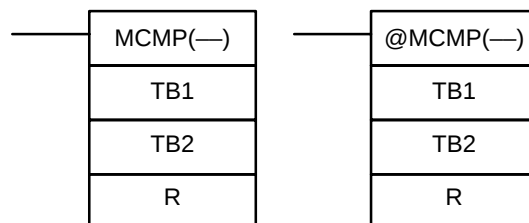
Exemple:
Sauvegardes des résultats
de CMPL(—)

L'exemple suivant illustre comment sauvegarder immédiatement le résultat de la comparaison. Si le contenu de HR 10, HR 09 est supérieur à celui de 011, 010, alors 10000 est placé à ON ; si les deux contenus sont égaux, 10001 est placé à ON ; si le contenu de HR 10, HR 09 est inférieur à celui de 011, 010, alors 10002 est placé à ON. Dans certaines applications, un seul des trois OUT pourrait être utilisé, ce qui dispense de l'usage de TR 0. Dans ce type de programmation, 10000, 10001 et 10002 ne sont modifiés que lorsque CMPL(—) est exécuté.



Adresse	Instruction	Opérande
00000	LD	00000
00001	OUT	TR 0
00002	CMPL(—)	
		HR 09
		010
00003	AND	25505
00004	OUT	10000
00005	LD	TR 0
00006	AND	25506
00007	OUT	10001
00008	LD	TR 0
00009	AND	25507
00010	OUT	10002

5-18-5 MULTI-WORD COMPARE – COMPARAISONS MULTIPLES – MCMP(—)

Symboles**Zones de données d'opérande**

TB1: 1 ^{er} canal du tableau 1
IR, SR, AR, DM, HR, TC, LR
TB2: 1 ^{er} canal du tableau 2
IR, SR, AR, DM, HR, TC, LR
R: canal de résultat
IR, SR, AR, DM, HR, TC, LR

Limites

MCMP compare 32 canaux consécutifs en une seule fois. Les bits comparés différents mettent le résultat à 1. TB1 et TB1+15 doivent appartenir à la même zone de données.

TB2 et TB2+15 doivent appartenir à la même zone de données.

Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, MCMP(—) n'est pas exécutée. Lorsque la condition est à ON, MCMP(—) compare le contenu de TB1 à TB2, TB1+1 à TB2+1, TB1+2 à TB2+2, ... et TB1+15 à TB2+15. Si les deux termes de la première paire sont égaux, le premier bit de R est à OFF, etc., c'est-à-dire que si le contenu de TB1 égale le contenu de TB2, le bit 00 est placé à OFF ; si le

contenu de TB1+1 égale celui de TB2+1, le bit 01 est placé à ON, etc. Les autres bits de R sont placés à ON.

Drapeaux

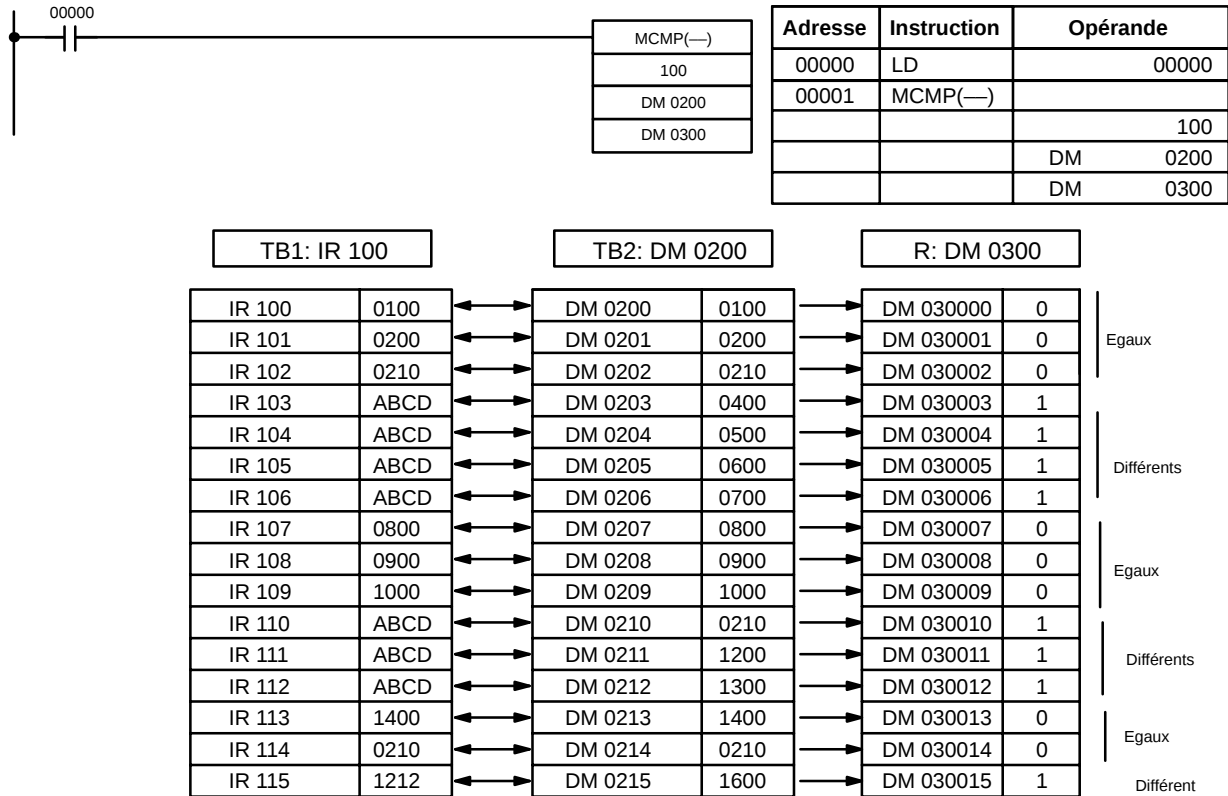
ER: L'un des tableaux (c'est-à-dire TB1 à TB1+15 ou TB2 à TB2+15) dépasse la zone de données.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

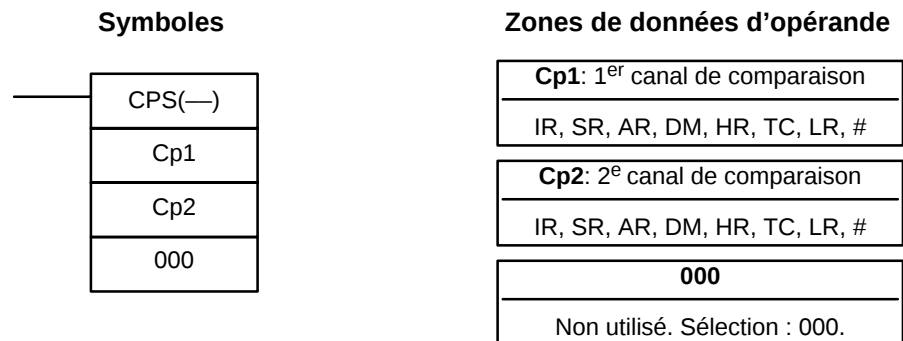
EQ: ON si les contenus des deux tableaux sont tous égaux et si R=0000.

Exemple

L'exemple suivant illustre une comparaison et les résultats de MCMP(—). la comparaison est ici effectuée à chaque tour de scrutation lorsque 00000 est à ON.



5-18-6 SIGNED BINARY COMPARE – COMPARAISON BINAIRE – CPS(—)

**Limites**

Cette instruction est disponible avec les UC du **CQM1-CPU4 [-E.]**

Description

Lorsque la condition d'exécution est à OFF, CPS(—) n'est pas exécutée. Lorsque la condition est à ON, CPS(—) compare le contenu binaire à 16 bits (4

chiffres) de Cp1 et Cp2 et sort le résultat dans les drapeaux GR, EQ et LE de la zone SR.

Précautions

Le fait de placer d'autres instructions entre CPS(—) et l'accès aux drapeaux EQ, LE et GR peut changer l'état de ces drapeaux. Accédez à ces drapeaux avant modification de l'état désiré.

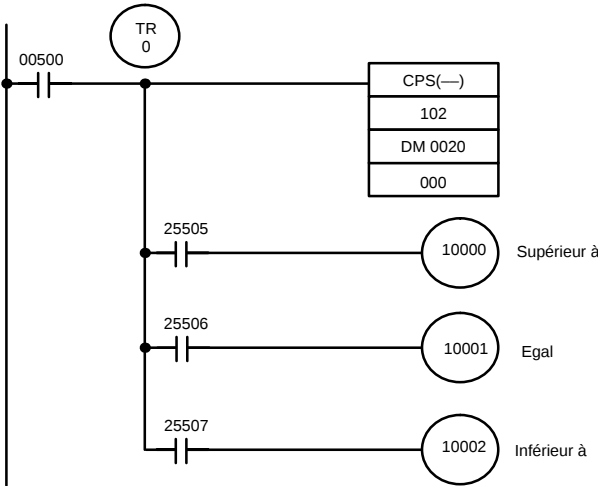
Drapeaux

- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- EQ:** ON si Cp1 égale Cp2.
- LE:** ON si Cp1 est inférieur à Cp2.
- GR:** ON si Cp1 est supérieur à Cp2.

Résultat de la comparaison	Etat du drapeau		
	GR (SR 25505)	EQ (SR 25506)	LE (SR 25507)
Cp1 < Cp2	0	0	1
Cp1 = Cp2	0	1	0
Cp1 > Cp2	1	0	0

Exemple

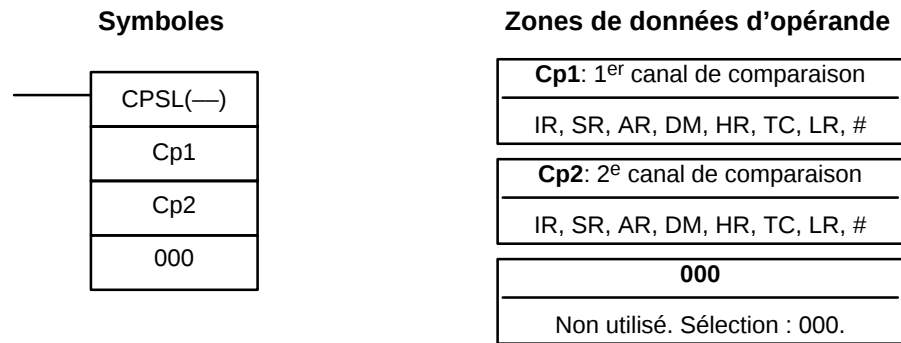
Dans l'exemple suivant, le contenu de 102 est supérieur à celui du DM 0020 : 10000 est donc placé à ON et les autres bits, 10001 et 10002, à OFF.



Adresse	Instruction	Opérande
00000	LD	00500
00001	OUT	TR 0
00002	CPS(—)	
		102
		DM 0020
		000
00003	AND	25505
00004	OUT	10000
00005	LD	TR 0
00006	AND	25506
00007	OUT	10001
00008	LD	TR 0
00009	AND	25507
00010	OUT	10002

Cp1: 102				>	Cp2: DM 0020			
6	F	A	4		A	E	3	5
(28 580 décimal)					(– 20 939 décimal)			

5-18-7 DOUBLE SIGNED BINARY COMPARE – COMPARAISON BINAIRE DOUBLE LONGUEUR – CPSL(—)



Limites

Cette instruction est disponible avec les UC du **CQM1-CPU4 [-E,]**

Description

Lorsque la condition d'exécution est à OFF, CPSL(—) n'est pas exécutée. Lorsque la condition est à ON, CPSL(—) compare le contenu binaire à 32 bits (8 chiffres) de Cp1+1, Cp1 et de Cp2+1, Cp2 et sort le résultat dans les drapeaux GR, EQ et LE de la zone SR.

Précautions

Le fait de placer d'autres instructions entre CPSL(—) et l'accès aux drapeaux EQ, LE et GR peut changer l'état de ces drapeaux. Accédez à ces drapeaux avant modification de l'état désiré.

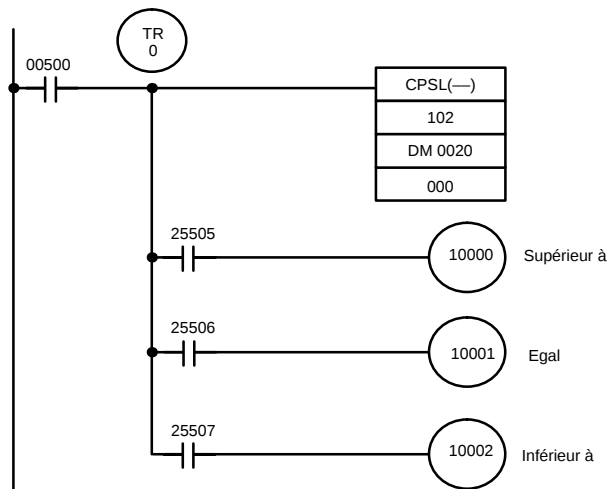
Drapeaux

- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- EQ:** ON si Cp1+1, Cp1 égale Cp2+1, Cp2.
- LE:** ON si Cp1+1, Cp1 est inférieur à Cp2+1, Cp2.
- GR:** ON si Cp1+1, Cp1 est supérieur à Cp2+1, Cp2.

Résultat de la comparaison	Etat du drapeau		
	GR (SR 25505)	EQ (SR 25506)	LE (SR 25507)
Cp1+1, Cp1 < Cp2+1, Cp2	0	0	1
Cp1+1, Cp1 = Cp2+1, Cp2	0	1	0
Cp1+1, Cp1 > Cp2+1, Cp2	1	0	0

Exemple

Dans l'exemple suivant, le contenu de 103, 102 est inférieur à celui du DM 0021, DM 0020 : 10002 est donc placé à ON et les autres bits, 10000 et 10001, à OFF.



Adresse	Instruction	Opérande
00000	LD	00500
00001	OUT	TR 0
00002	CPSL(---)	
		102
		DM 0020
		000
00003	AND	25505
00004	OUT	10000
00005	LD	TR 0
00006	AND	25506
00007	OUT	10001
00008	LD	TR 0
00009	AND	25507
00010	OUT	10002

Cp1+1: 103				Cp1: 102			
8	2	B	6	F	5	7	B

(- 2 101 938 823 décimal)

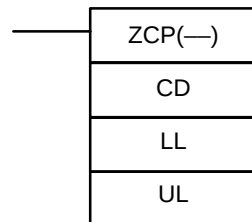
<

Cp2+1: DM 0021				Cp2: DM 0020			
0	5	6	A	9	9	D	B

(90 872 283 décimal)

5-18-8 AREA RANGE COMPARE – COMPARAISON ENTRE DEUX SEUILS BINAIRES – ZCP(---)

Symbole



Zones de données d'opérande

CD: donnée de comparaison
IR, SR, AR, DM, HR, TC, LR, #

LL: limite basse de la gamme
IR, SR, AR, DM, HR, TC, LR, #

UL: limite haute de la gamme
IR, SR, AR, DM, HR, TC, LR, #

Limites

Cette instruction est disponible avec les UC **CQM1-CPU4 [-E.]**

LL doit être inférieure ou égale à UL.

Description

Lorsque la condition d'exécution est à OFF, ZCP(---) n'est pas exécutée. Lorsque la condition est à ON, ZCP(---) compare CD à la gamme définie par la limite basse LL et la limite haute UL et sort le résultat dans les drapeaux GR, EQ et LE de la zone SR. Voici l'état du drapeau qui en résulte :

Résultat de la comparaison	Etat du drapeau		
	GR (SR 25505)	EQ (SR 25506)	LE (SR 25507)
CD < LL	0	0	1
LL ≤ CD ≤ UL	0	1	0
UL < CD	1	0	0

Précautions

Le fait de placer d'autres instructions entre ZCP(---) et l'accès aux drapeaux EQ, LE et GR peut changer l'état de ces drapeaux. Accédez à ces drapeaux avant modification de l'état désiré.

- Drapeaux

ER:

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
LL est supérieure à UL.

EQ:

ON si $LL \leq CD \leq UL$

LE:

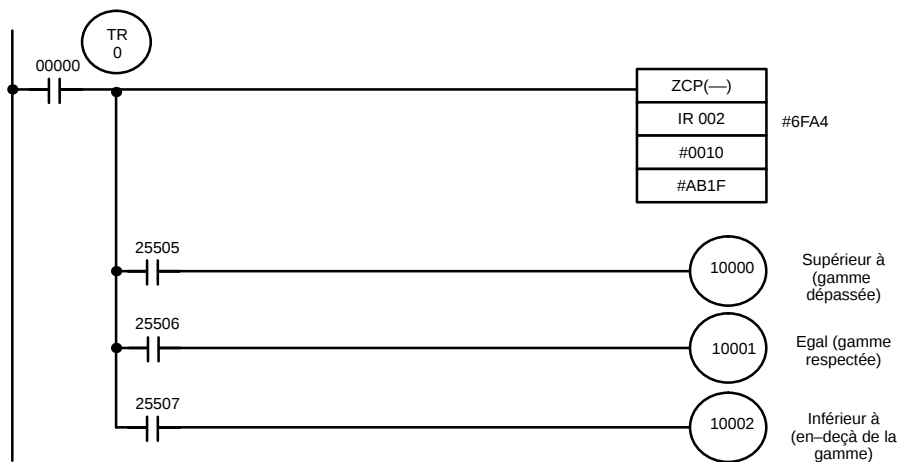
ON si $CD < LL$.

GR:

ON si $CD > UL$.

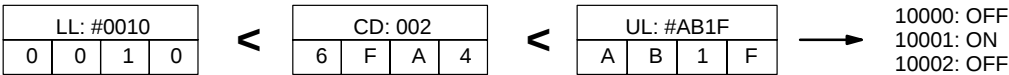
Exemple

Dans l'exemple suivant, le contenu de IR 002 (#6FA4) est comparé à la gamme #0010 à #AB1F. Comme $\#0010 \leq \#6FA4 \leq \#AB1F$, le drapeau EQ et IR 10001 sont placés à ON.



Adresse	Instruction	Opérande
00000	LD	00000
00001	OUT	TR 0
00002	ZCP(—)	
		002
		# 0010
00003	LD	# AB1F
00004	AND	25505

Adresse	Instruction	Opérande
00005	OUT	10000
00006	LD	TR 0
00007	AND	25506
00008	OUT	10001
00009	LD	TR 0
00010	AND	25507
00011	OUT	10002



5-18-9

DOUBLE AREA RANGE COMPARE – COMPARAISON DOUBLE LONGUEUR DE 2 SEUILS BINAIRES – ZCPL(—)

Symbole

ZCPL(—)

CD

LL

UL

Zones de données d'opérande

CD: donnée de comparaison

IR, SR, AR, DM, HR, LR

LL: limite basse de gamme

IR, SR, AR, DM, HR, LR

UL: limite haute de gamme

IR, SR, AR, DM, HR, LR

Limites

Cette instruction est disponible avec les UC du CQM1-CPU4

E

La valeur à 8 chiffres de LL+1,LL doit être inférieure ou égale à UL+1,UL.

Description

Lorsque la condition d'exécution est à OFF, ZCPL(—) n'est pas exécutée. Lorsque la condition est à ON, ZCPL(—) compare la valeur à 8 chiffres de CD, CD+1 à la gamme définie par la limite basse LL+1,LL et la limite haute UL+1,UL et sort le résultat dans les drapeaux GR, EQ et LE de la zone SR. Voici l'état du drapeau qui en résulte :

Résultat de la comparaison	Etat du drapeau		
	GR (SR 25505)	EQ (SR 25506)	LE (SR 25507)
CD , CD+1 < LL+1,LL	0	0	1
LL+1,LL ≤ CD, CD+1 ≤ UL+1,UL	0	1	0
UL+1,UL < CD, CD+1	1	0	0

Précautions

Le fait de placer d'autres instructions entre ZCPL(—) et l'accès aux drapeaux EQ, LE et GR peut changer l'état de ces drapeaux. Accédez à ces drapeaux avant modification de l'état désiré.

Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
LL+1,LL est supérieure à UL+1,UL.

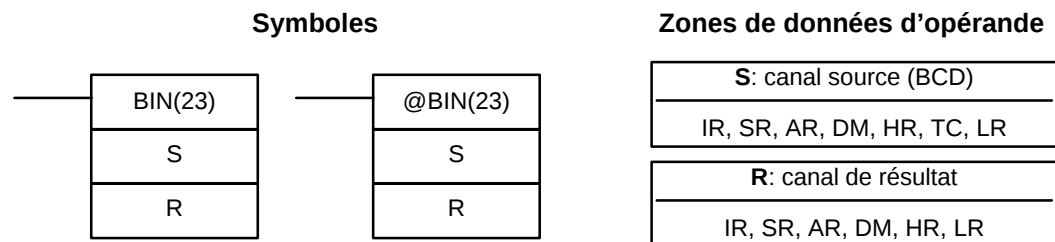
EQ: ON if LL+1,LL ≤ CD, CD+1 ≤ UL+1,UL

LE: ON if CD, CD+1 < LL+1,LL.

GR: ON if CD, CD+1 > UL+1,UL.

5-19 Instructions de conversion

5-19-1 BCD-TO-BINARY – BCD VERS BINAIRE – BIN(23)

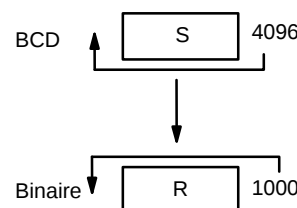


Limites

Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, BIN(23) n'est pas exécutée. Lorsque la condition est à ON, BIN(23) convertit le contenu BCD de S en bits binaires (16 bits) numériquement équivalents et sort la valeur binaire dans R. Seul le contenu de R est modifié ; le contenu de S demeure inchangé.



BIN(23) peut être utilisée pour la conversion de BCD en binaire pour que les affichages de la console de programmation ou d'autres appareils de programmation apparaissent en hexadécimal plutôt qu'en décimal. Elle peut également

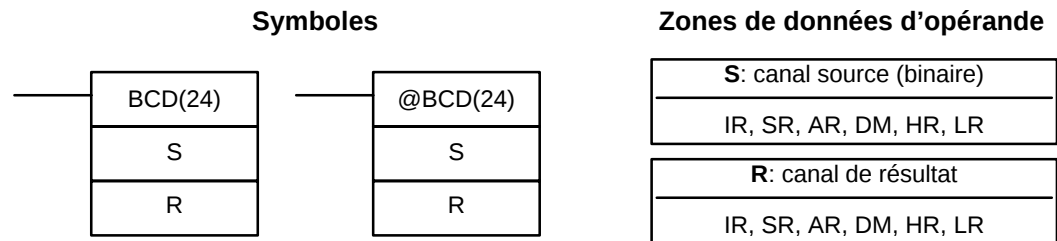
servir à convertir en binaire pour effectuer des opérations arithmétiques binaires plutôt qu'en BCD (par ex. lorsqu'il faut additionner des valeurs BCD et binaires).

Drapeaux

ER: le contenu de S n'est pas en BCD.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à zéro.

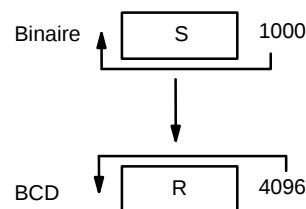
5-19-2 BINARY-TO-BCD – BINAIRE VERS BCD – BCD(24)**Limites**

Si le contenu de S dépasse 270F, le résultat converti dépasse 9999 et BCD(24) n'est pas exécutée. Lorsque l'instruction n'est pas exécutée, le contenu de R demeure inchangé.

Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

BCD(24) convertit le contenu binaire (hexadécimal) de S en bits BCD numériquement équivalents et sort les bits BCD dans R. Seul le contenu de R est modifié ; le contenu de S demeure inchangé.



BCD(24) peut être utilisée pour la conversion de binaire en BCD pour que les affichages de la console de programmation ou d'autres appareils de programmation apparaissent en décimal plutôt qu'en hexadécimal. Elle peut également servir à convertir en BCD pour effectuer des opérations arithmétiques en BCD plutôt qu'en binaire (par ex. lorsqu'il faut additionner des valeurs BCD et binaires).

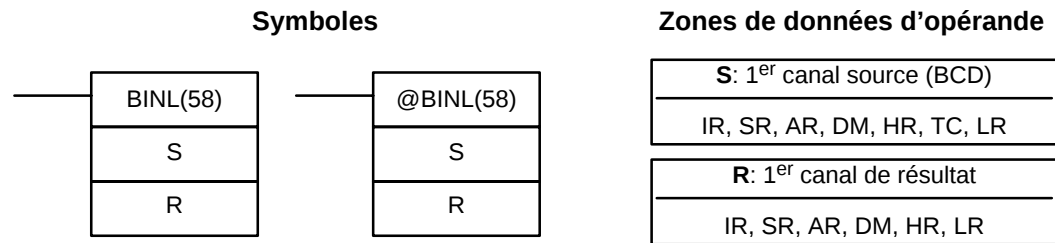
Drapeaux

ER: S est supérieure à 270F.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à zéro.

5-19-3 DOUBLE BCD-TO-DOUBLE BINARY –BCD EN BINAIRE DOUBLE LONGUEUR – BINL(58)

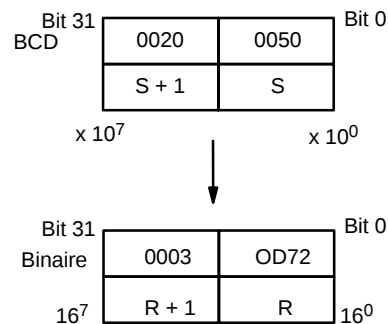


Limites

Les DM 6143 à 6655 ne peuvent être utilisés pour R.

Description

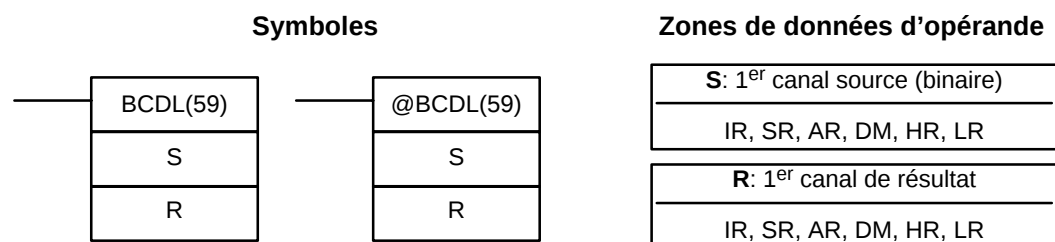
Lorsque la condition d'exécution est à OFF, BINL(58) n'est pas exécutée. Lorsque la condition est à ON, BINL(58) convertit un nombre à 8 chiffres de S et S+1 en donnée binaire de 32 bits et sort la donnée convertie dans deux canaux : R et R +1.



Drapeaux

- ER:** Le contenu des canaux S et/ou S+1 ne sont pas en BCD.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- EQ:** ON lorsque le résultat est égal à zéro.

5-19-4 DOUBLE BINARY-TO-DOUBLE BCD – BINAIRE EN BCD DOUBLE LONGUEUR – BCDL(59)

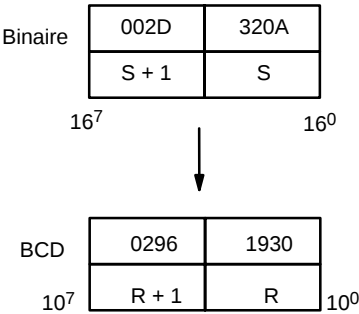


Limites

Si le contenu de S dépasse 05F5E0FF, le résultat converti dépasse 99999999 et BCDL(59) n'est pas exécutée. Lorsque l'instruction n'est pas exécutée, le contenu de R et R+1 demeure inchangé.

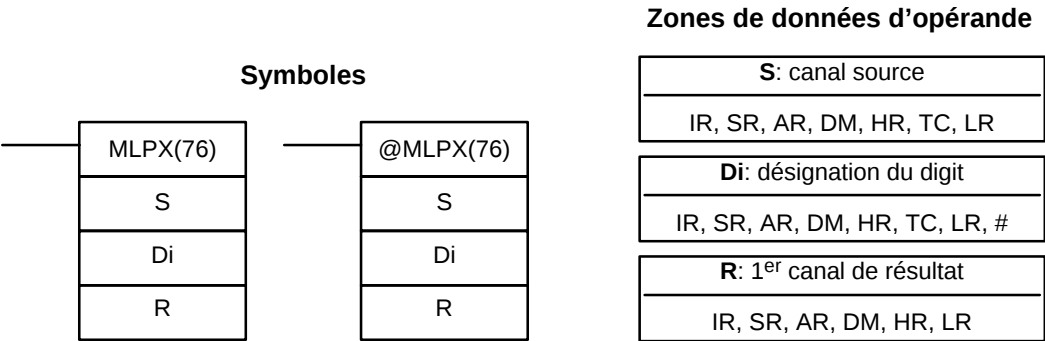
Les DM 6143 à 6655 ne peuvent être utilisés pour R.

Description BCDL(59) convertit le contenu binaire à 32 bits de S et S+1 en 8 chiffres BCD et sort la donnée convertie dans deux canaux : R et R+1.



Drapeaux **ER:** Le contenu de R et R+1 dépasse 99999999.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
EQ: ON lorsque le résultat est égal à zéro.

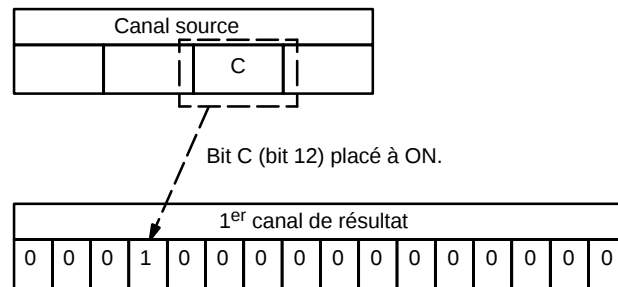
5-19-5 4-TO-16 DECODER – DECODEUR DE 1 DIGIT HEXADECIMAL EN BINAIRE – MLPX(76)



Limites Décode 1 à 4 valeurs hexadécimales de 4 bits (source) en une valeur décimale de destination de 16 bits.
Les deux chiffres d'extrême droite de Di doivent être situés entre 0 et 3.
Tous les canaux de résultat doivent appartenir à la même zone de données.
Les DM 6144 à 6655 ne peuvent être utilisés pour S, Di ni R.

Description Lorsque la condition d'exécution est à OFF, MLPX(76) n'est pas exécutée. Lorsque la condition est à ON, MLPX(76) convertit jusqu'à 4 digits hexadécimaux de 4 bits en valeurs décimales de 0 à 15, chacune étant utilisée pour indiquer une position de bit. Le bit correspondant à chaque valeur convertie est alors placé à ON dans un canal de résultat. Si l'on spécifie plusieurs digits, un bit est placé à ON dans les canaux suivants, en commençant par R (cf exemples ci-dessous).

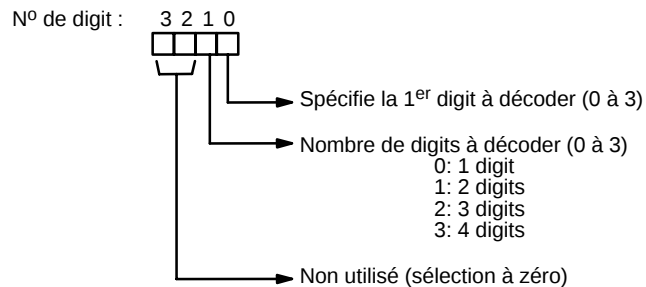
Voici un exemple d'opération de décodage à un digit à partir du chiffre 1 de S (c'est-à-dire que Di est égal à 0001).



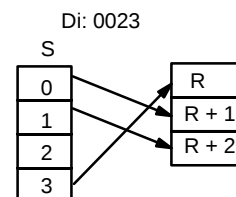
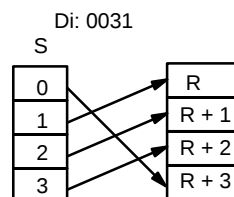
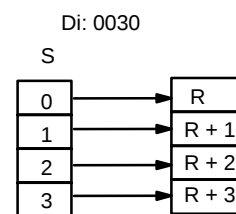
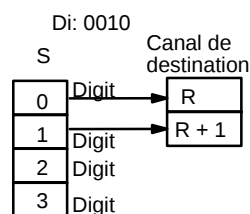
Les premiers digits et nombre de digits à être convertis sont désignés dans Di. Si l'on désigne plus de digits qu'il n'en reste dans S (à partir du 1^{er} digit désigné), les digits restants sont pris en compte à partir du début de S. Le dernier canal devant stocker le résultat converti (R plus le nombre de digits à convertir) doit appartenir à la même zone de données que R (c'est-à-dire que si deux digits sont convertis, la dernière adresse de canal d'une zone de donnée ne peut être désignée ; si trois digits sont convertis, les deux derniers canaux d'une zone de données ne peuvent être désignés).

Désignation du chiffre

Sélection des digits de Di :



Exemples de valeurs de Di et de conversions chiffre vers canal produites :



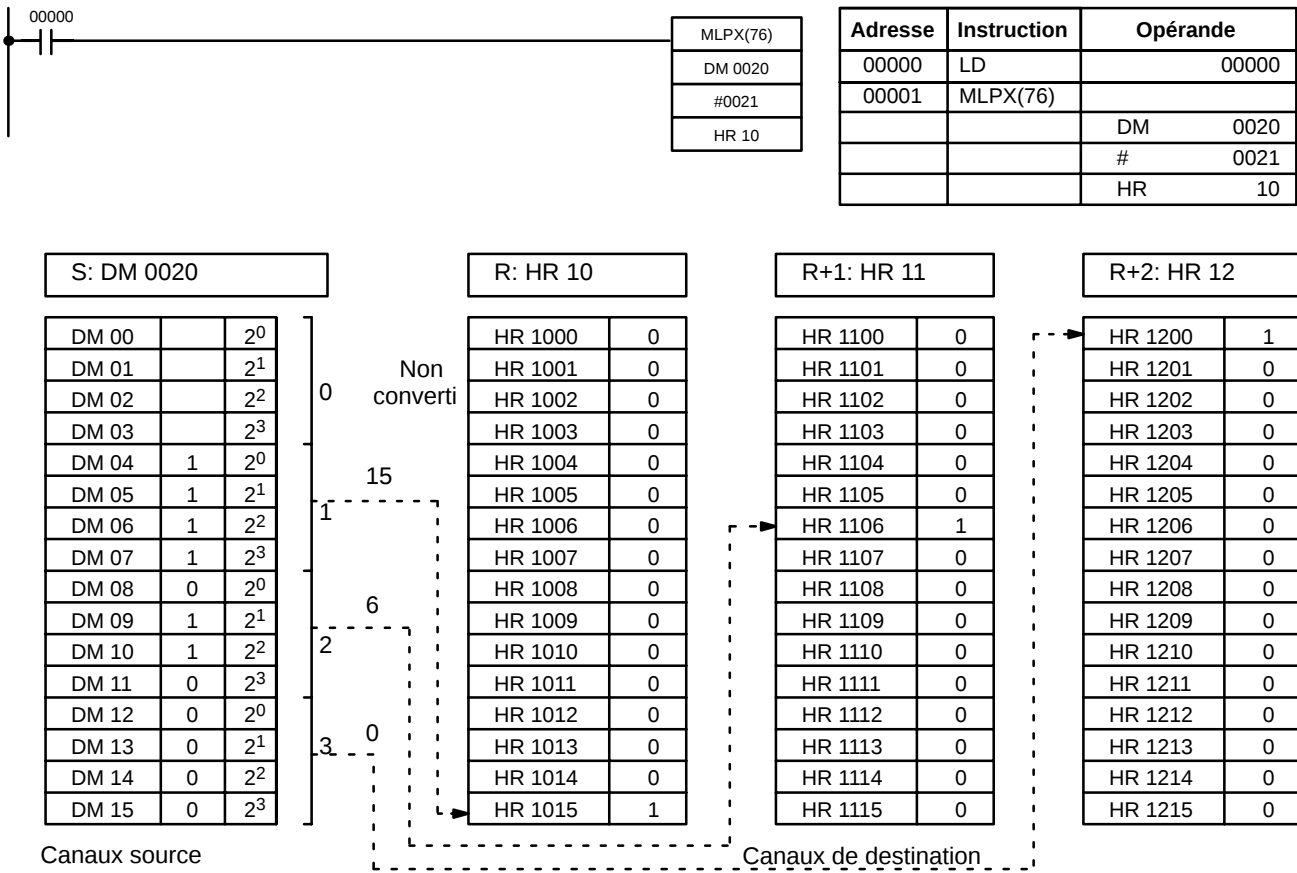
Drapeaux

ER: Chiffre de désignation non défini ou R plus le nombre de chiffres dépasse une zone de données.

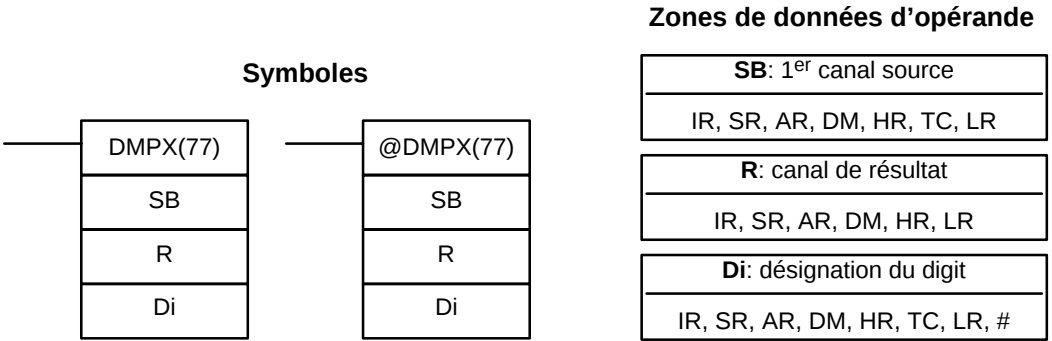
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple

Le programme suivant convertit les digits 1 à 3 des données à partir de DM 0020 en positions de bits et place à ON les bits correspondants dans trois canaux consécutifs en commençant par HR 10. Le digit 0 n'est pas converti.



5-19-6 16-TO-4 ENCODER- CODAGE DE BINAIRE EN 1 DIGIT HEXADECIMAL – DMPX(77)



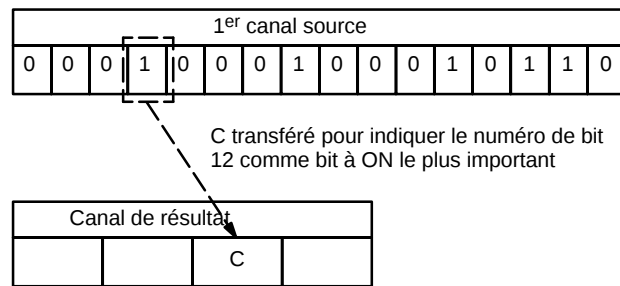
Limites

Les deux chiffres d'extrême droite de Di doivent être situés entre 0 et 3.
Tous les canaux source doivent appartenir à la même zone de données.
Les DM 6144 à 6655 ne peuvent être utilisés pour SB, R ni Di.

Description

Lorsque la condition d'exécution est à OFF, DMPX(77) n'est pas exécutée. Lorsque la condition est à ON, DMPX(77) détermine la position du bit à ON le plus important de S, le code en valeur hexadécimale à un digit correspondant au numéro du bit à ON le plus important et transfère ensuite la valeur hexadécimale dans le digit spécifié de R. Les digits qui reçoivent le résultat sont spécifiés dans Di qui spécifie également le nombre de digits à coder.

Voici un exemple de codage à un digit en nombre 1 de R, c'est-à que Di est égal à 0001.

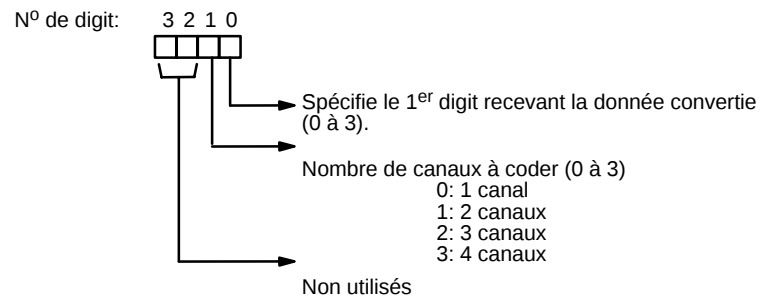


On peut coder jusqu'à 4 digits à partir de 4 canaux source consécutifs à partir de S et inscrire les digits dans R à partir du premier digit désigné. Si l'on spécifie plus de digits qu'il n'en reste dans R (en comptant à partir du premier digit désigné), les digits restants sont placés à partir du début de R.

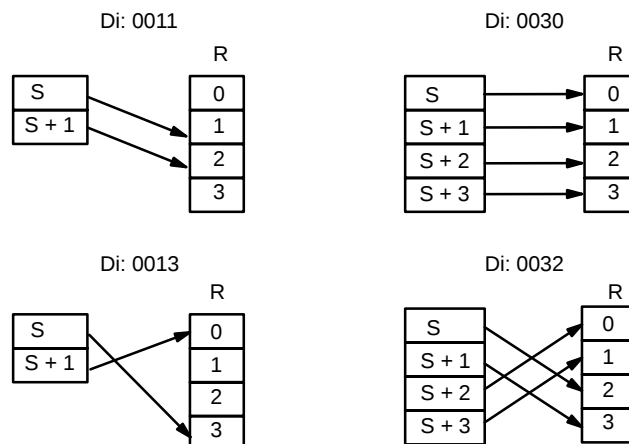
Le dernier canal à être converti (S plus le nombre de digits à convertir) doit appartenir à la même zone de données que SB.

Désignation du digit

Sélection des digits de Di : il faut désigner le nombre de canaux ou les données qui doivent être codés et le digit à partir duquel la valeur codée doit être stockée.



Exemples de valeurs de Di et de conversions de canal vers les digits produits :



Drapeaux

ER: Désignation de digit non définie ou S plus le nombre de digits dépasse une zone de données.

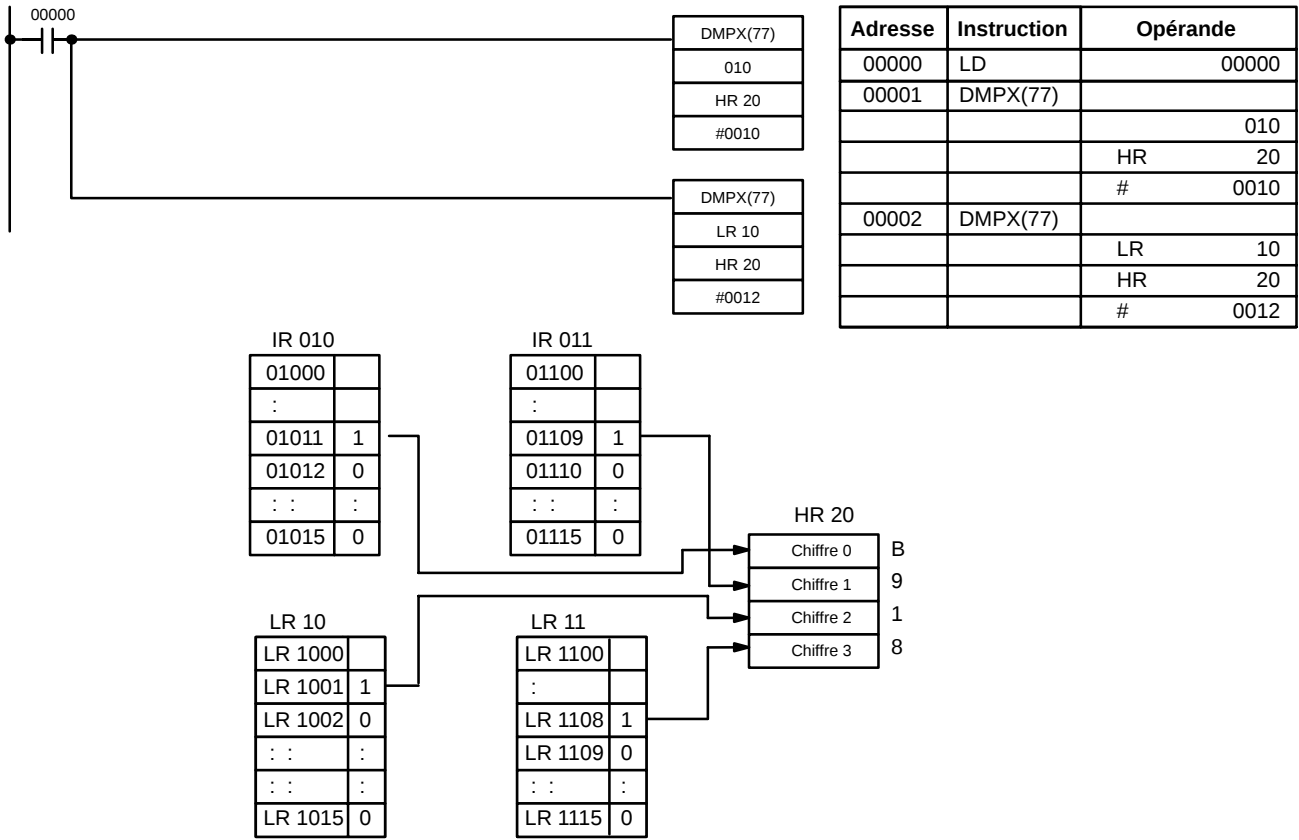
Le contenu d'un canal source est égal à zéro.

Le canal DM indirectement adressé n'existe pas (le contenu du canal

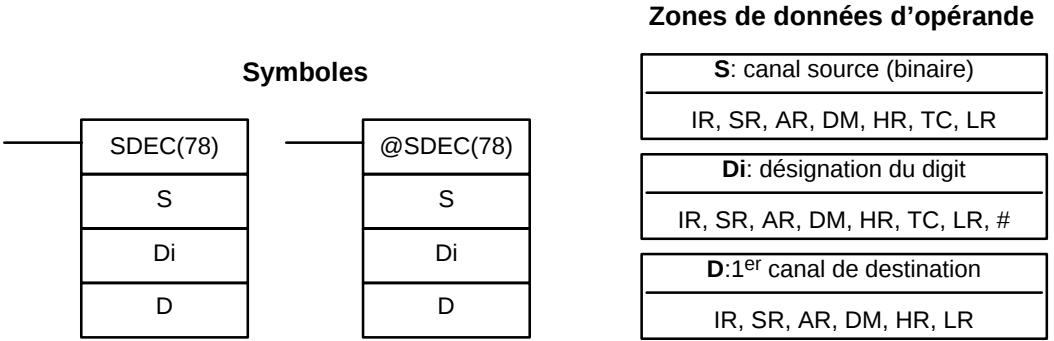
*DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple

Lorsque 00000 est à ON, le schéma suivant code les canaux IR 010 et 011 en deux premiers digits de HR 20 et les canaux LR 10 et 11 en deux derniers digits de HR 20. Bien que l'état de chaque canal source ne soit pas montré, on part du principe que le bit d'état 1 (ON) est le bit à ON le plus important du canal.



5-19-7 7-SEGMENT DECODER – DECODEUR 7 SEGMENTS – SDEC(78)



Limites

Convertit 4 bits d'une donnée à 16 bits en une valeur de 8 bits pour un affichage 7 segments.
Di doit être situé dans les gammes de valeurs suivantes.
Tous les canaux de destination doivent appartenir à la même zone de données.
Les DM 6144 à 6655 ne peuvent être utilisés pour D.

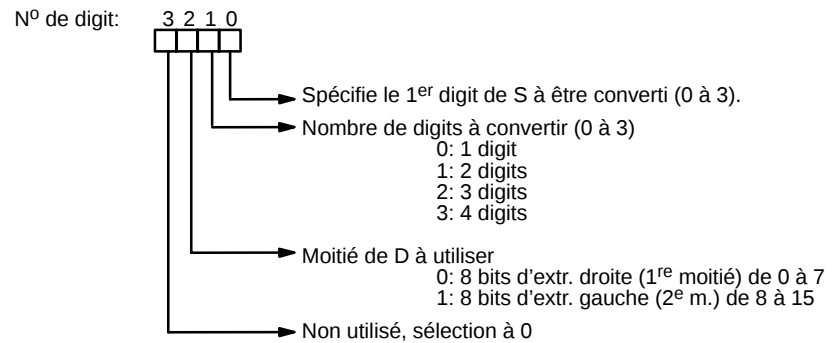
Description

Lorsque la condition d'exécution est à OFF, SDEC(78) n'est pas exécutée. Lorsque la condition est à ON, SDEC(78) convertit les chiffres (digits) désignés de S en valeur à 8 bits équivalente et en affichage à 7 segments et place la valeur dans les canaux de destination en commençant par D.

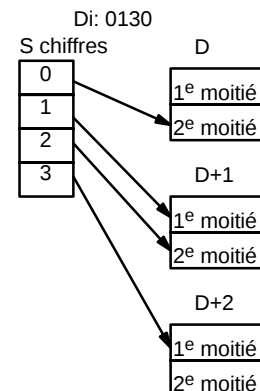
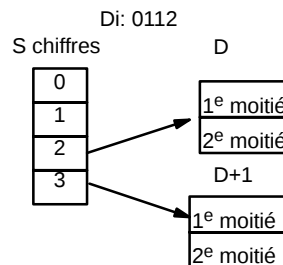
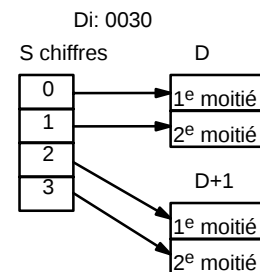
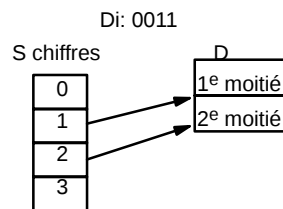
N'importe quel chiffre de S ou la totalité des digits de S peuvent être convertis à la suite à partir du premier digit désigné. Le premier digit, le nombre de digits à convertir et la moitié de D qui doit recevoir le premier code d'affichage à 7 segments (8 bits d'extrême droite ou d'extrême gauche) sont désignés dans Di. Si plusieurs digits sont désignés, ils sont placés par ordre en partant de la moitié désignée de D et chacun prend deux digits. Si l'on désigne plus de digits qu'il n'en reste dans S (en comptant à partir du premier digit désigné), les digits suivants sont utilisés en partant du début de S.

Désignation du chiffre

Sélection des chiffres de Di : designer la destination, le nombre de digits à décoder et le digit à partir duquel la valeur doit être transférée.

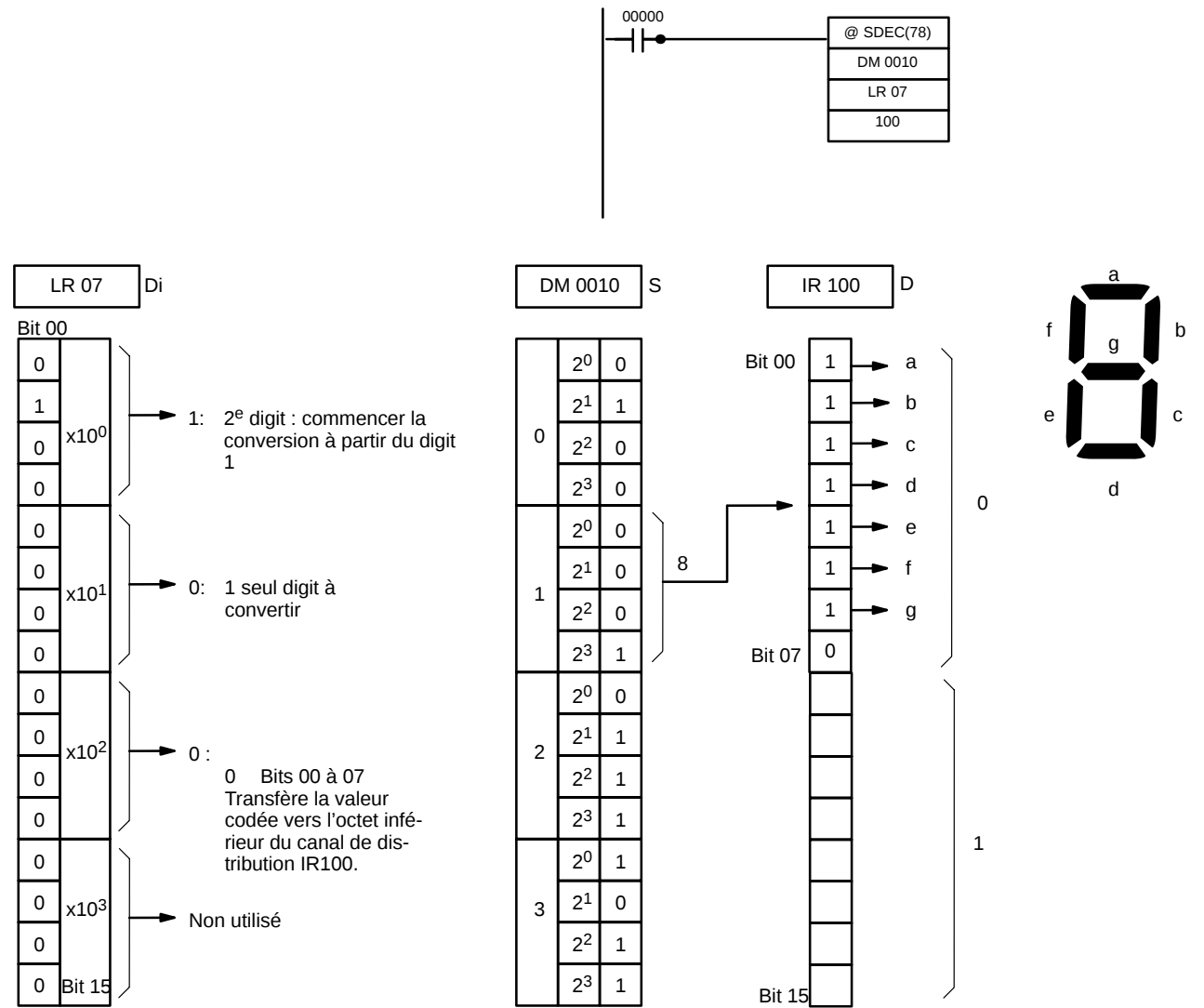


Voici quelques exemples de valeurs Di et des conversions de binaire à 4 bits vers l'affichage à 7 segments :



Exemple

Voici un exemple de données produisant un "8". Les lettres minuscules représentent les segments de l'affichage à 7 segments. Le tableau qui suit illustre les données originales et leur conversion pour tous les chiffres hexadécimaux :



Donnée d'entrée de conversion					Donnée de sortie de conversion 7 segments								Affi- chage
Hex.	Bits				-	g	f	e	d	c	b	a	
	2 ³	2 ²	2 ¹	2 ⁰									
0	0	0	0	0	0	0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1	1	0	0	0	0	1
2	0	0	1	0	0	0	1	1	0	0	1	1	2
3	0	0	1	1	0	0	1	1	0	0	1	1	3
4	0	1	0	0	0	0	1	1	0	1	0	0	4
5	0	1	0	1	0	0	1	1	0	1	0	1	5
6	0	1	1	0	0	0	1	1	0	1	0	1	6
7	0	1	1	1	0	0	1	1	0	1	1	1	7
8	1	0	0	0	0	0	1	1	1	0	0	0	8
9	1	0	0	1	0	0	1	1	1	0	0	1	9
A	1	0	1	0	0	1	0	0	0	0	0	1	A
B	1	0	1	1	0	1	0	0	0	0	1	0	b
C	1	1	0	0	0	1	0	0	0	0	1	1	c
D	1	1	0	1	0	1	0	0	0	1	0	0	d
E	1	1	1	0	0	1	0	0	0	1	0	1	e
F	1	1	1	1	0	1	0	0	0	1	1	0	F

Drapeaux

ER: Désignation du chiffre incorrecte ou zone de données de destination dépassée.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-19-8 ASCII CONVERT – CONVERSION EN CODE ASCII – ASC(86)

Zones de données d'opérande

Symboles	
ASC(86)	@ASC(86)
S	S
Di	Di
D	D

S: canal source
IR, SR, AR, DM, HR, TC, LR
Di: désignation du chiffre
IR, SR, AR, DM, HR, TC, LR, #
D: 1 ^{er} canal de destination
IR, SR, AR, DM, HR, LR

Limites

Convertit 4 bits (de 0à F) d'une donnée à 16 bits en code ASCII à 8 bits.

Di doit être situé dans la gamme de valeur ci-dessous.

Tous les canaux de destination doivent appartenir à la même zone de données.

Les DM 6144 à 6655 ne peuvent être utilisés pour D.

Description

Lorsque la condition d'exécution est à OFF, ASC(86) n'est pas exécutée. Lorsque la condition est à ON, ASC(86) convertit le(s) chiffre(s) désigné(s) de S en code ASCII à 8 bits équivalent et le(s) place dans le canal ou les canaux de destination en commençant par D.

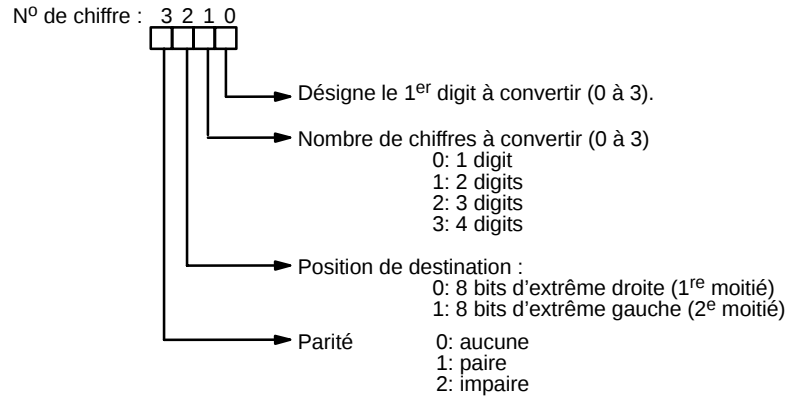
N'importe quel digit de S ou la totalité des digits de S peuvent être convertis à la suite à partir du premier digit désigné. Le premier digit, le nombre de digits à convertir et la moitié de D qui doit recevoir le premier code ASCII (8 bits d'extrême droite ou d'extrême gauche) sont désignés dans Di. Si plusieurs digits sont

désignés, ils sont placés par ordre en partant de la moitié désignée de D et chacun prend deux digits. Si l'on désigne plus de digits qu'il n'en reste dans S (en comptant à partir du premier digit désigné), les chiffres suivants sont utilisés en partant du début de S.

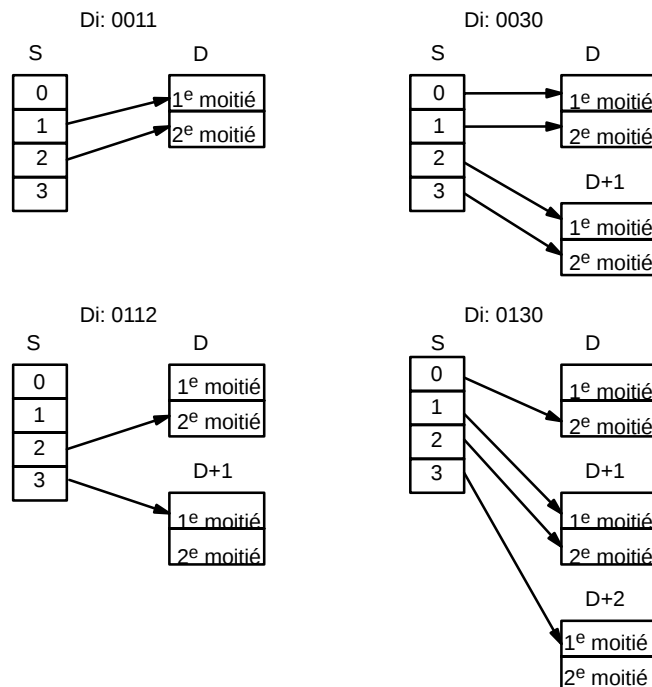
Un tableau des caractères ASCII se trouve dans l'annexe K.

Désignation du chiffre

Sélection des chiffres de Di :



Exemples de valeurs de Di et de conversions de binaire à 4 bits vers le code ASCII à 8 bits :



Parité

Le bit d'extrême gauche de chaque caractère ASCII (2 chiffres) peut être automatiquement réglé en parité paire ou impaire. Si aucune parité n'est désignée, le bit d'extrême gauche sera toujours à zéro.

Lorsque la parité paire est désignée, le bit d'extrême gauche est réglé de façon à ce que le nombre total de bits à ON soit pair (en parité paire, "31" en ASCII (00110001) est égal à "B1" (10110001: bit de parité à ON pour créer un nombre pair de bits à ON) ; "36" en ASCII (00110110) est égal à "36" (00110110: bit de parité à OFF car le nombre de bits à ON est déjà pair). L'état du bit de parité n'affecte pas la signification du code ASCII.

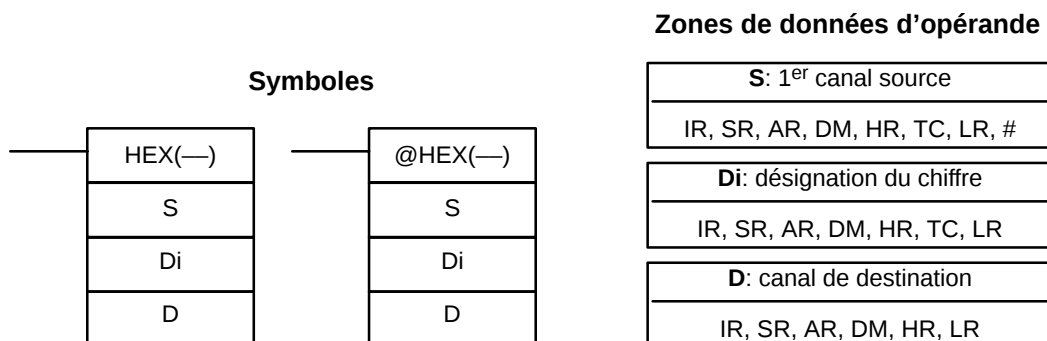
Lorsqu'une parité impaire est désignée, le bit d'extrême gauche de chaque caractère ASCII est réglé de façon à ce qu'il y ait un nombre impair de bits à ON.

Drapeaux

ER: Désignation du digit incorrecte ou zone de données de destination dépassée.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-19-9 ASCII-TO-HEXADECIMAL – CONVERSION ASCII EN HEXADECIMAL – HEX(—)



Limites

Di doit être situé dans la gamme de valeurs ci-dessous.

Tous les canaux source doivent appartenir à la même zone de données.

Les octets des canaux de source doivent contenir les codes ASCII équivalents aux valeurs hexadécimales, soient les valeurs 30 à 39 (0 à 9) ou 41 à 46 (A à F).

Les DM 6144 à 6655 ne peuvent être utilisés pour D.

Description

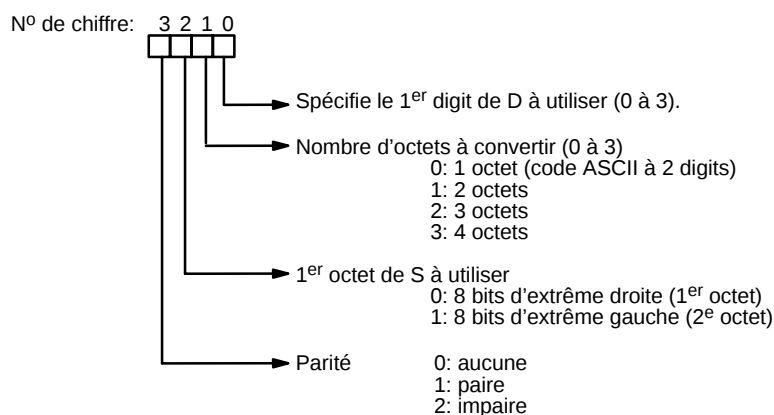
Lorsque la condition d'exécution est à OFF, HEX(—) n'est pas exécutée. Lorsque la condition est à ON, HEX(—) convertit les octets en ASCII désignés à partir du canal ou des canaux source en valeurs hexadécimales équivalentes et les place dans D.

On peut convertir jusqu'à 4 codes ASCII en commençant par le premier octet désigné de S. Les valeurs hexadécimales converties sont ensuite placées dans D par ordre à partir du digit désigné. Le premier octet (8 bits d'extrême droite ou d'extrême gauche), le nombre d'octets à convertir et le digit de D recevant la première valeur hexadécimale sont désignés dans Di. Si plusieurs octets sont désignés, ils sont convertis par ordre en partant de la moitié désignée de S et jusqu'à S+1 et S+2, si besoin est.

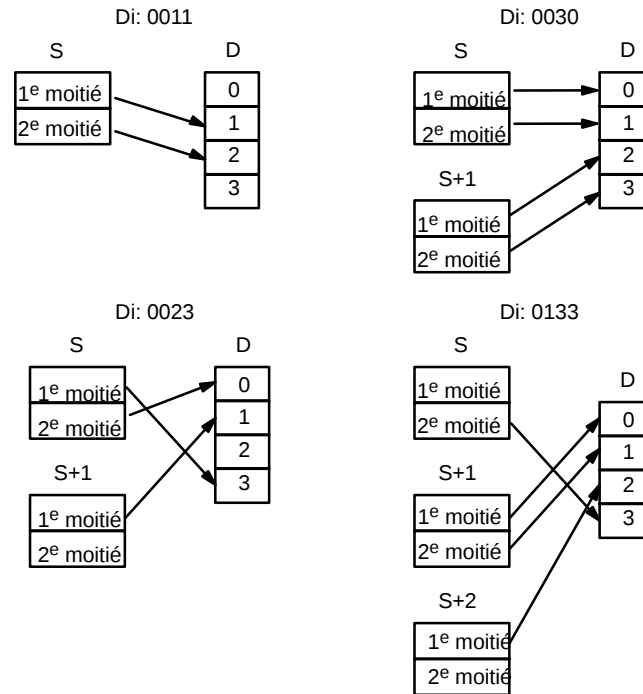
Si l'on désigne plus de digits qu'il n'en reste dans D (en comptant à partir du premier digit désigné), les digits suivants sont utilisés en commençant à partir du début de D. Les digits de D qui ne reçoivent pas de données converties demeurent inchangés.

Désignation du chiffre

Sélection des chiffres de Di :



Voici quelques exemples de valeurs de Di et de conversions de codes ASCII à 8 bits en valeurs hexadécimales :



Codes ASCII

Codes ASCII avant conversion et valeurs hexadécimales après conversion (cf annexe K pour le tableau des caractères ASCII) :

Donnée originale									Donnée convertie				
Code ASCII	Etat du bit (cf Rem.)								Digit	Bit			
30	*	0	1	1	0	0	0	0	0	0	0	0	0
31	*	0	1	1	0	0	0	1	1	0	0	0	1
32	*	0	1	1	0	0	1	0	2	0	0	1	0
33	*	0	1	1	0	0	1	1	3	0	0	1	1
34	*	0	1	1	0	1	0	0	4	0	1	0	0
35	*	0	1	1	0	1	0	1	5	0	1	0	1
36	*	0	1	1	0	1	1	0	6	0	1	1	0
37	*	0	1	1	0	1	1	1	7	0	1	1	1
38	*	0	1	1	1	0	0	0	8	1	0	0	0
39	*	0	1	1	1	0	0	1	9	1	0	0	1
41	*	1	0	1	0	0	0	1	A	1	0	1	0
42	*	1	0	1	0	0	1	0	B	1	0	1	1
43	*	1	0	1	0	0	1	1	C	1	1	0	0
44	*	1	0	1	0	1	0	0	D	1	1	0	1
45	*	1	0	1	0	1	0	1	E	1	1	1	0
46	*	1	0	1	0	1	1	0	F	1	1	1	1

Rem. : le bit d'extrême gauche de chaque code ASCII est réglé pour la parité.

Parité

Le bit d'extrême gauche de chaque caractère ASCII (2 digits) est automatiquement réglé pour la parité paire ou impaire.

Si aucune parité n'est désignée, le bit d'extrême gauche doit toujours être à zéro. En cas de parité paire ou impaire, le bit d'extrême gauche de chaque caractère ASCII doit être réglé de façon à ce qu'il y ait un nombre pair ou impair de bits à ON.

Si la parité du code ASCII de S ne correspond pas à celle spécifiée dans Di, le drapeau ER (SR 25503) est placé à ON et l'instruction n'est pas exécutée.

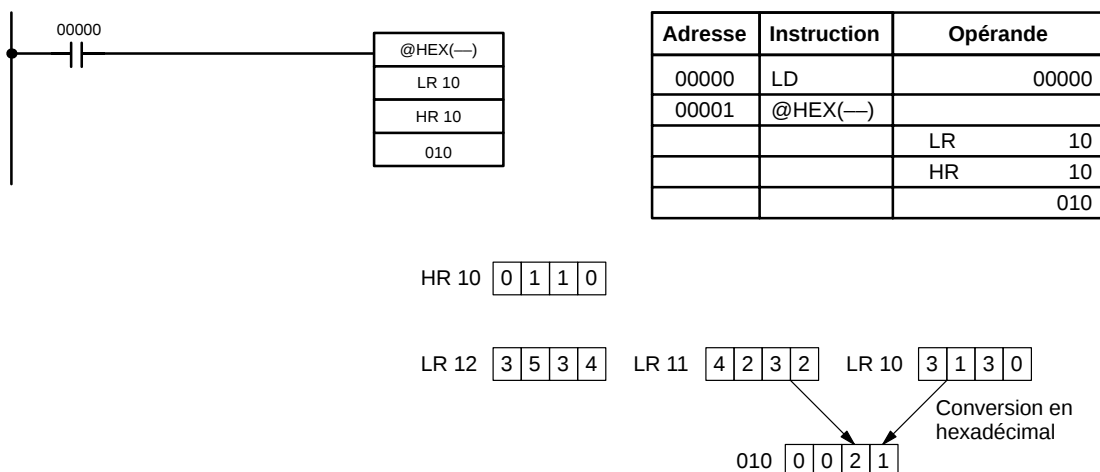
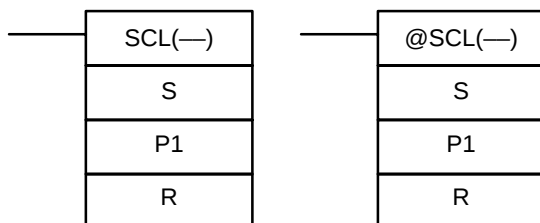
Drapeaux

ER: Désignation du digit incorrecte ou zone de données de destination dépassée.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple

Dans l'exemple suivant, le 2^e octet de LR 10 et le 1^{er} octet de LR 11 sont convertis en valeurs hexadécimales et ces valeurs sont inscrites dans les 1^{er} et 2^e octets de IR 010.

**5-19-10 SCALING – MISE A L'ECHELLE – SCL(---)****Symboles****Zones de données d'opérande**

S: canal source
IR, SR, AR, DM, HR, TC, LR, #
P1: 1 ^{er} canal de paramètre
IR, SR, AR, DM, HR, TC, LR
R: canal de résultat
IR, SR, AR, DM, HR, LR

Limites

S doit être en BCD.

P1 à P1+3 doivent appartenir à la même zone de données.

Les DM 6144 à 6655 ne peuvent être utilisés pour P1 à P1+3 ou R.

Description

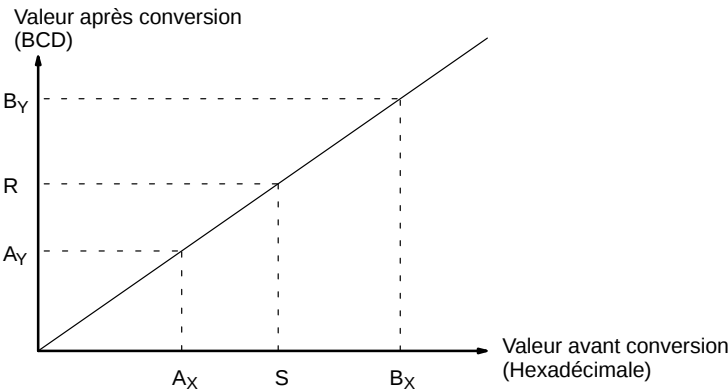
SCL(---) s'utilise pour convertir linéairement une valeur hexadécimale à 4 chiffres en valeur BCD à 4 chiffres. A la différence de BCD(24), qui convertit une valeur hexadécimale à 4 chiffres en équivalent BCD à 4 chiffres ($S_{\text{hex}} \rightarrow S_{\text{BCD}}$), SCL(---) convertit la valeur hexadécimale selon une relation linéaire spécifiée. La ligne de conversion est définie par 2 points spécifiés dans les canaux de paramètres P1 à P1+3.

Lorsque la condition d'exécution est à OFF, SCL(---) n'est pas exécutée. Lorsque la condition est à ON, SCL(---) convertit la valeur hexadécimale de S en valeur BCD à 4 chiffres sur la ligne définie par les points (P1, P1+1) et (P1+2, P1+3) et place le résultat dans R. Le résultat est arrondi au nombre entier le plus proche. Si le résultat est inférieur à 0000, "0000" est inscrit dans R et si le résultat est supérieur à 9999, "9999" est inscrit dans R.

Fonctions et gammes de canaux de paramètre :

Paramètre	Fonction	Gamme	Remarques
P1	Point BCD n° 1 (A _Y)	0000 à 9999	---
P1+1	Point hexa. n° 1 (A _X)	0000 à FFFF	Sélection interdite P1+1=P1+3
P1+2	Point BCD n° 2 (B _Y)	0000 à 9999	---
P1+3	Point hexa. n° 2 (B _X)	0000 à FFFF	Sélection interdite P1+3=P1+1

Voici le canal source S converti en D selon la ligne définie par les points (A_Y, A_X) et (B_Y, B_X) :



Les résultats peuvent être calculés en convertissant d'abord toutes les valeurs en BCD et en utilisant la formule suivante :

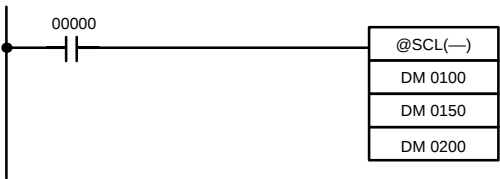
Résultats = B_Y – [(B_Y – A_Y)/(B_X – A_X) X (B_X – S)]

Drapeaux

- ER:** La valeur de P1+1 égale celle de P1+3.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
P1 et P1+3 n'appartiennent pas à la même zone de données ou autre erreur de sélection.
- EQ:** ON lorsque le résultat R est égal à 0000.

Exemple

Lorsque 00000 passe à ON dans l'exemple suivant, la donnée source BCD de DM 0100 (#0100) est convertie en hexadécimal selon les paramètres de DM 0150 à 0153. Le résultat (#0512) est alors inscrit dans DM 0200.

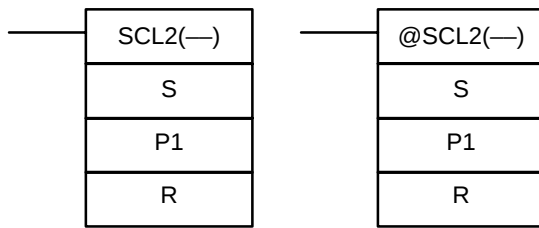


Adresse	Instruction	Opérande
00000	LD	00000
00001	@SCL(—)	
		Source : DM0100
		Param. : DM0150
		Résultat : DM0200

DM 0150	0010	Paramètres	DM 0100	0100
DM 0151	0005		Source	↓
DM 0152	0050			
DM 0153	0019			
Résultat			DM 0200	0512

5-19-11 SIGNED BINARY TO BCD SCALING – CONVERSION LINEAIRE D'UNE VALEUR BINAIRE AVEC SIGNE EN BCD – SCL2(--)

Symboles



Zones de données d'opérande

S: canal source
IR, SR, AR, DM, HR, LR
P1:1^{er} canal de paramètre
IR, SR, AR, DM, HR, LR
R: canal de résultat
IR, SR, AR, DM, HR, LR

Limites

Cette instruction est disponible avec les UC du **CQM1-CPU4 [-E.]**

S doit être en BCD.

P1 à P1+2 doivent appartenir à la même zone de données.

Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

SCL2(—) s'utilise pour convertir linéairement une valeur hexadécimale comportant un signe et ayant 4 chiffres en valeur BCD à 4 chiffres. A la différence de BCD(24), qui convertit une valeur hexadécimale à 4 chiffres en équivalent BCD à 4 chiffres ($S_{\text{hex}} \rightarrow S_{\text{BCD}}$), SCL2(—) convertit la valeur hexadécimale portant un signe selon une relation linéaire spécifiée. La ligne de conversion est définie par une intersection x et par la pente de la ligne spécifiée dans les canaux de paramètre P1 à P1+2.

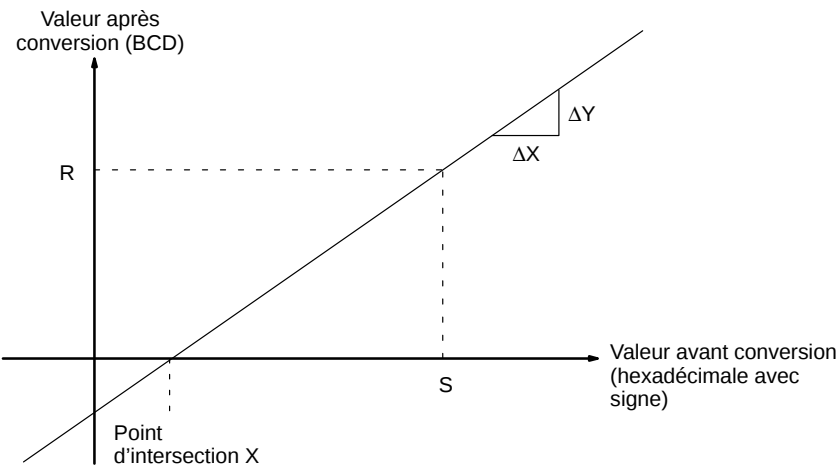
Lorsque la condition d'exécution est à OFF, SCL2(—) n'est pas exécutée. Lorsque la condition est à ON, SCL2(—) convertit la valeur de S à 4 chiffres hexadécimaux et portant un digit en valeur BCD à 4 chiffres sur la ligne définie par l'interception x (P1, 0) et par la pente ($(P1+2 \div P1+1)$) et place les résultats dans R. Le résultat est arrondi au nombre entier le plus proche.

Si le résultat est négatif, CY est placé à 1. Si le résultat est inférieur à - 9999, "- 9999" est inscrit dans R. Si le résultat est supérieur à 9999, "9999" est inscrit dans R.

Fonctions et gammes des canaux de paramètre :

Paramètre	Fonction	Gamme
P1	Interception x (hexa. avec signe)	8000 à 7FFF (- 32 768 à 32 767)
P1+1	ΔX (hexa. avec signe)	8000 à 7FFF (- 32 768 à 32 767)
P1+2	ΔY (BCD)	0000 à 9999

Voici le canal source S converti en R selon la ligne définie par le point (P1, 0) et la pente $\Delta Y/\Delta X$.



Le résultat peut être calculé en convertissant toutes les valeurs hexadécimales avec un signe en BCD et en utilisant la formule suivante :

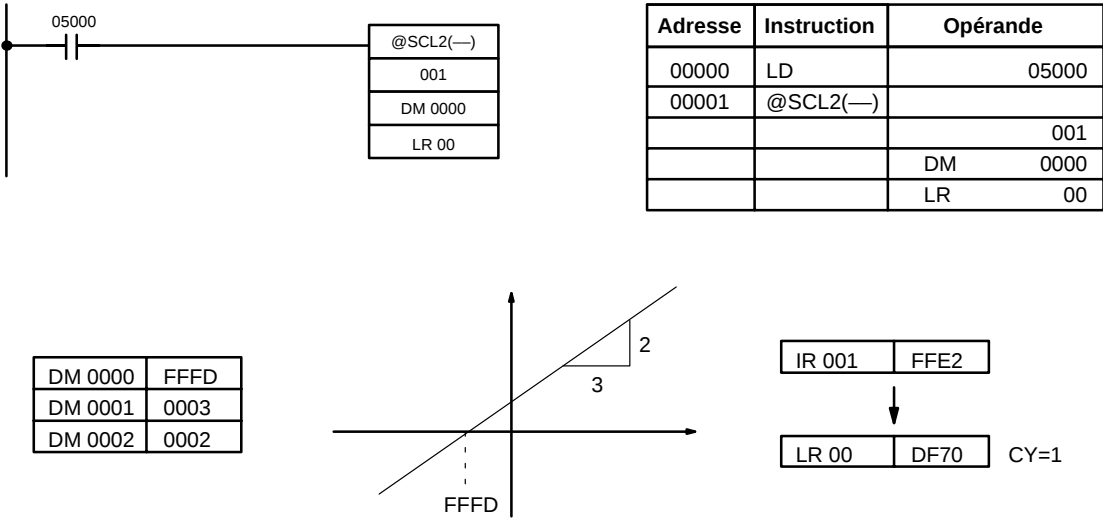
$$R = \frac{Y}{X} \times (P1 - S)$$

Drapeaux

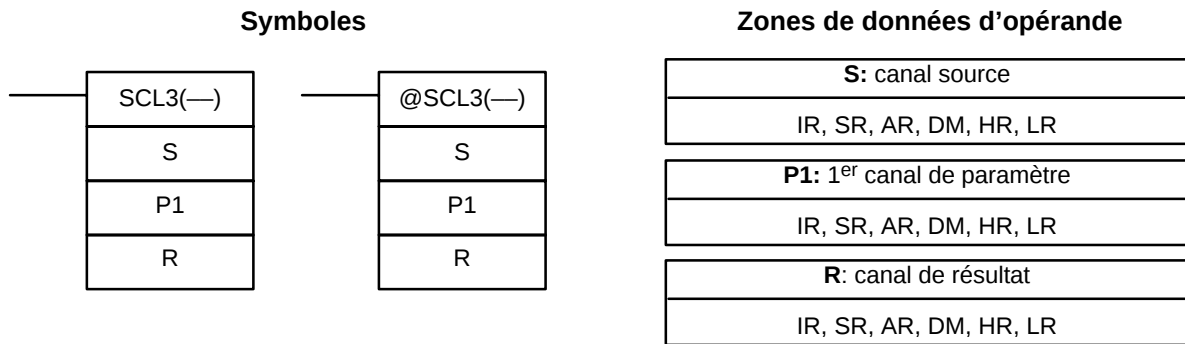
- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
P1 et P1+2 n'appartiennent pas à la même zone de données ou erreur de sélection.
- CY:** ON lorsque le résultat R est négatif.
- EQ:** ON lorsque le résultat R est égal à 0000.

Exemple

Lorsque 05000 est placé à ON dans l'exemple suivant, la donnée source binaire portant un signe dans 001 (#FFE2) est convertie en BCD selon les paramètres de DM 0000 à 0002. Le résultat (#0018) est alors inscrit dans LR 00 et CY est placée à ON car le résultat est négatif.



5-19-12 BCD TO SIGNED BINARY SCALING – CONVERSION LINEAIRE D'UNE VALEUR BCD EN BINAIRE AVEC SIGNE– SCL3(—)



Limites

Cette instruction est disponible avec les UC du **CQM1-CPU4 [-E.]**

P1+1 doit être en BCD.

P1 à P1+4 doivent appartenir à la même zone de données

Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

SCL3(—) s'utilise pour convertir linéairement une valeur BCD à 4 chiffres en valeur hexadécimale portant un signe et ayant 4 chiffres. SCL3(—) convertit la valeur BCD selon une relation linéaire spécifiée. La ligne de conversion est définie par une interception y et par la pente de la ligne spécifiée dans les canaux de paramètre P1 à P1+2.

Lorsque la condition d'exécution est à OFF, SCL3(—) n'est pas exécutée. Lorsque la condition est à ON, SCL3(—) convertit la valeur BCD à 4 chiffres de S en valeur hexadécimale portant un signe sur la ligne définie par l'intersection y (0, P1) et par la pente (P1+2 ÷ P1+1) et place ensuite le résultat dans R. Le résultat est arrondi au nombre entier le plus proche.

Le contenu de S peut être situé entre 0000 à 9999, mais S sera traité comme une valeur négative si CY=1 ; la gamme effective de S s'étend donc de -9999 à 9999. Assurez-vous de bien sélectionner le signe correct dans CY avec STC(40) ou CLC(41).

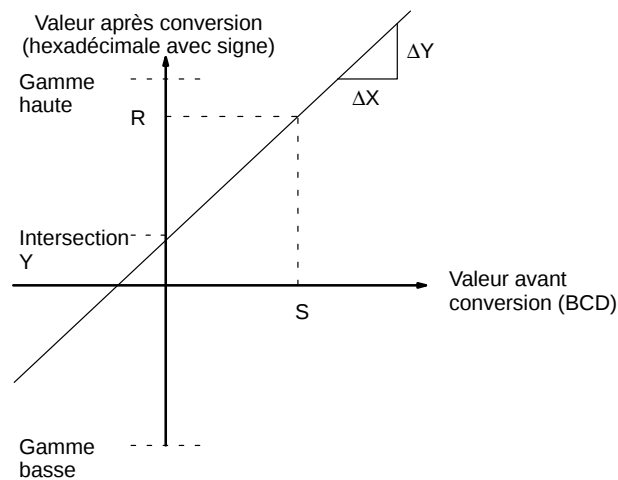
Les canaux de paramètre P1+3 et P1+4 définissent les gammes hautes et basses du résultat. Si le résultat est supérieur à la gamme haute de P1+3, la gamme haute est inscrite dans R. Si le résultat est inférieur à la gamme basse de P1+4, la gamme basse est inscrite dans R.

Rem. : les gammes haute et basse d'une carte d'entrée analogique à 12 bits sont 07FF et F800.

Fonctions et gammes des canaux de paramètre :

Paramètre	Fonction	Gamme
P1	Interception x (hexa. avec signe)	8000 à 7FFF (– 32 768 à 32 767)
P1+1	ΔX (BCD)	0000 à 9999
P1+2	ΔY (hexa. avec signe)	8000 à 7FFF (– 32 768 à 32 767)
P1+3	Gamme haute (hexa. avec signe)	8000 à 7FFF (– 32 768 à 32 767)
P1+4	Gamme basse (hexa. avec signe)	8000 à 7FFF (– 32 768 à 32 767)

Voici le canal source S converti en R selon la ligne définie par le point (0, P1) et par la pente $\Delta Y/\Delta X$.



Le résultat peut être calculé d'abord en convertissant toutes les valeurs BCD en binaires à signe et en utilisant la formule suivante :

$$R = \left(\left(S \times \frac{\Delta Y}{\Delta X} \right) + P1 \right) \text{ mod } 16$$

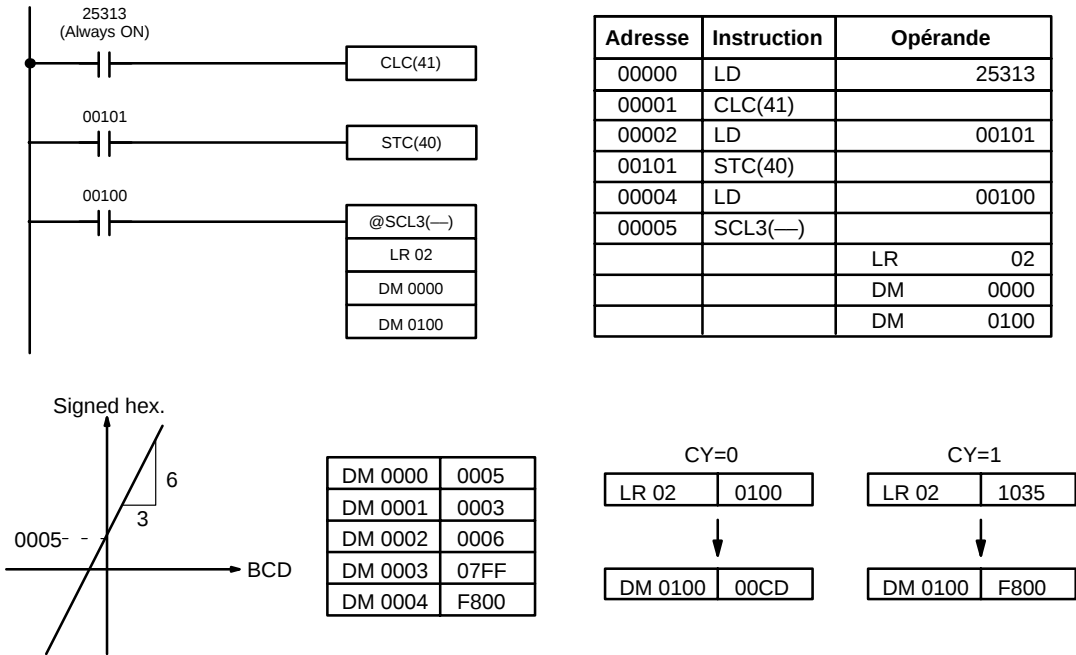
Drapeaux

- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
Le contenu de S n'est pas en BCD.
- CY:** CY n'est pas affecté par SCL3(—) (CY porte le signe de S avant l'exécution).
- EQ:** ON lorsque le résultat R est égal à 0000.

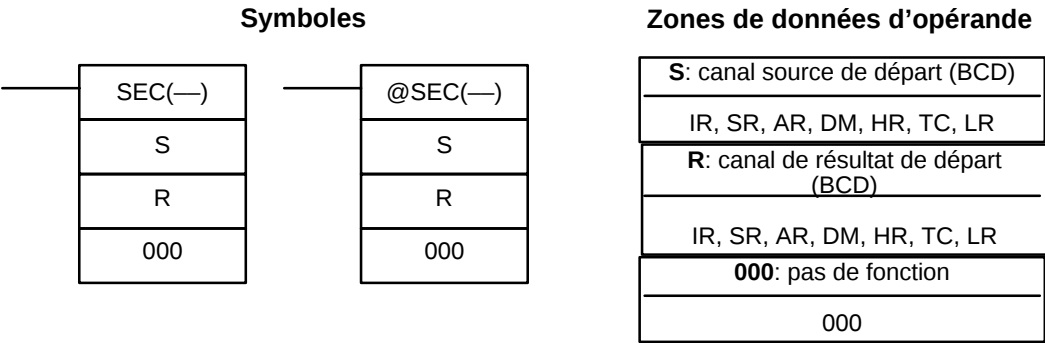
Exemple

L'état de 00101 détermine le signe du canal source BCD dans l'exemple suivant. Si 00101 est à ON, le canal source est négatif ; lorsque 00100 passe à ON, la donnée source BCD de LR 02 est convertie en binaire à signe selon les paramètres des DM 0000 à 0004. Le résultat est inscrit dans le DM 0100 (dans la deuxième conversion, l'équivalent binaire avec signe de – 1035 est

inférieur à la gamme basse spécifiée dans le DM 0004 ; la gamme basse est inscrite dans le DM 0100.)



5-19-13 HOURS-TO-SECONDS – HEURES EN SECONDES – SEC(—)



- Limites

S et S+1 doivent appartenir à la même zone de données. R et R+1 doivent appartenir à la même zone de données. S et S+1 doivent être en BCD et doivent être dans le bon format d'heure/minute/seconde.

Les DM 6143 à 6655 ne peuvent être utilisés pour R.
- Description

SEC(—) s'utilise pour convertir une notation de temps en heures/minutes/secondes en un équivalent en secondes uniquement.

Pour la donnée source, les secondes sont désignées dans les bits 00 à 07 et les minutes dans les bits 08 à 15 de S. Les heures sont désignées dans S+1. Le maximum est donc de 9 999 heures, 59 minutes et 59 secondes.

Le résultat est sorti dans R et R+1.

La valeur maximum est de 35 999 999 secondes.
- Drapeaux

ER:

S et S+1 ou R et R+1 n'appartiennent pas à la même zone de données.

S et/ou S+1 ne sont pas en BCD.

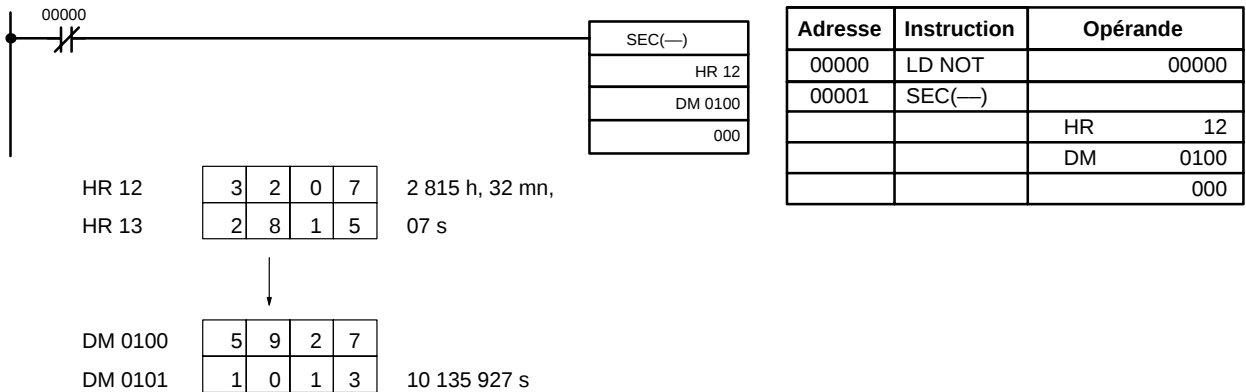
Le nombre de secondes et/ou de minutes dépasse 59.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

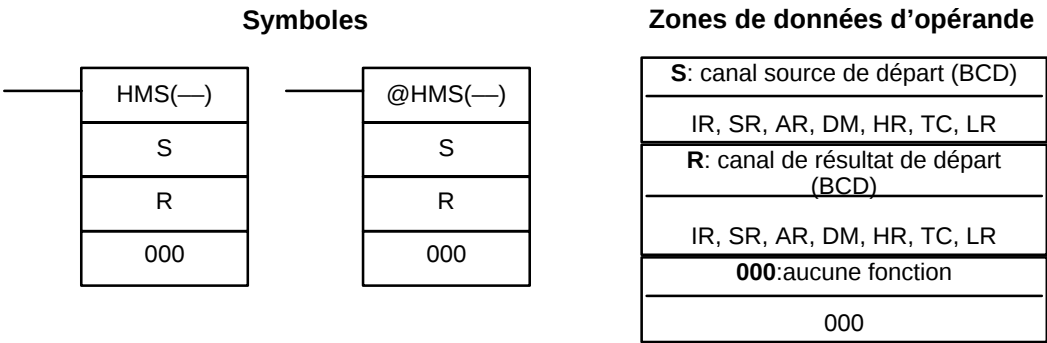
EQ: ON lorsque le résultat est égal à zero.

Exemple

Lorsque 00000 est à OFF (lorsque la condition d'exécution est à ON), l'instruction suivante convertit les heures, minutes et secondes de HR 12 et HR 13 en secondes et stocke les résultats dans DM 0100 et DM 0101 comme suit :

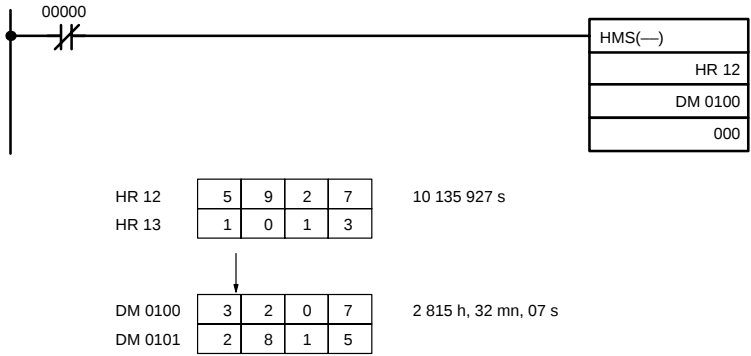


5-19-14 SECONDS-TO-HOURS – SECONDES EN HEURES – HMS(—)



Exemple

Lorsque 00000 est à OFF (lorsque la condition d'exécution est à ON), l'instruction suivante convertit les secondes de HR 12 et HR 13 en heures, minutes et secondes et stocke les résultats dans DM 0100 et DM 0101 comme suit :



Adresse	Instruction	Opérande
00000	LD NOT	00000
00001	HMS(—)	
		HR 12
		DM 0100
		000

5-19-15 COLUMN-TO-LINE – COLONNE EN LIGNE – LINE(—)

Symboles

LINE(—)	@LINE(—)
S	S
C	C
D	D

Zones de données d'opérande

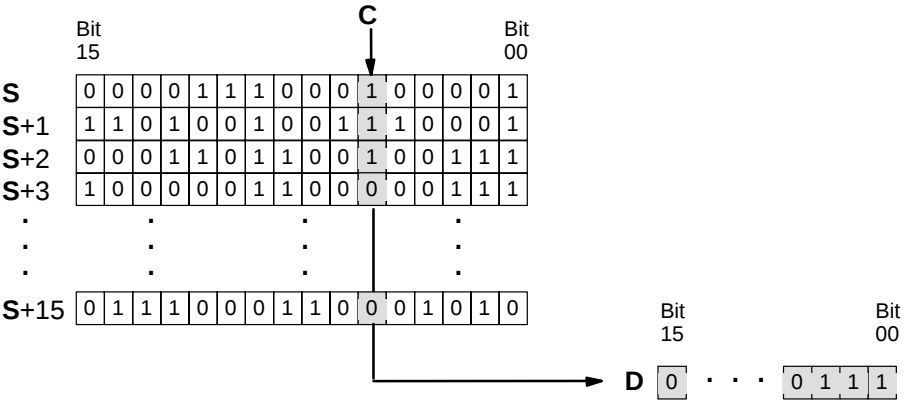
S: 1 ^{er} canal de la sélection des 16 canaux source
IR, SR, AR, DM, HR, TC, LR
C: désignation du bit de colonne (BCD)
IR, SR, AR, DM, HR, TC, LR, #
D: canal de destination
IR, SR, AR, DM, HR, TC, LR

Limites

S et S+15 doivent appartenir à la même zone de données.
C doit être en BCD entre #0000 et #0015.
Les DM 6144 à 6655 ne peuvent être utilisés pour D.

Description

Lorsque la condition d'exécution est à OFF, LINE(—) n'est pas exécutée. Lorsque la condition est à ON, LINE(—) copie la colonne du bit C dans une sélection de 16 canaux (S à S+15) et reconstitue les 16 bits dans le canal D (00 à 15). Le bit du canal S est le bit 0 du canal D et le bit du canal S + 15 est le bit 15 du canal D.



Drapeaux

ER: La désignation du bit de colonne C n'est pas en BCD ou spécifie un bit qui n'existe pas (c'est-à-dire que la spécification du bit doit être comprise entre 00 et 15).

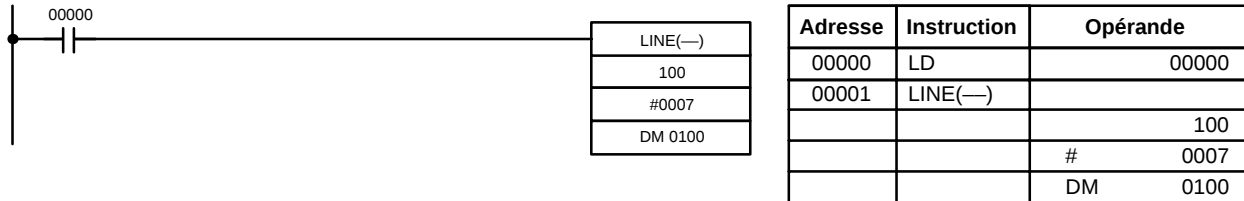
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

S et S+15 n'appartiennent pas à la même zone de données.

EQ: ON lorsque le contenu de D est égal à zéro ; reste à OFF dans les autres cas.

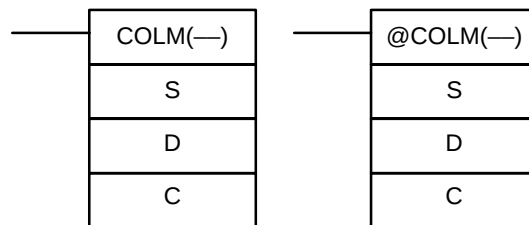
Exemple

L'exemple suivant montre l'utilisation de LINE(—) pour déplacer la colonne du bit 07 de la sélection (IR 100 à IR 115) vers DM 0100.



5-19-16 LINE-TO-COLUMN – LIGNE EN COLONNE – COLM(--)

Symboles



Zones de données d'opérande

S: canal source
IR, SR, AR, DM, HR, TC, LR
D: 1 ^{er} canal de la sélection de destination
IR, SR, AR, DM, HR, TC LR
C: désignation du bit de colonne (BCD)
IR, SR, AR, DM, HR, TC, LR, #

Limites

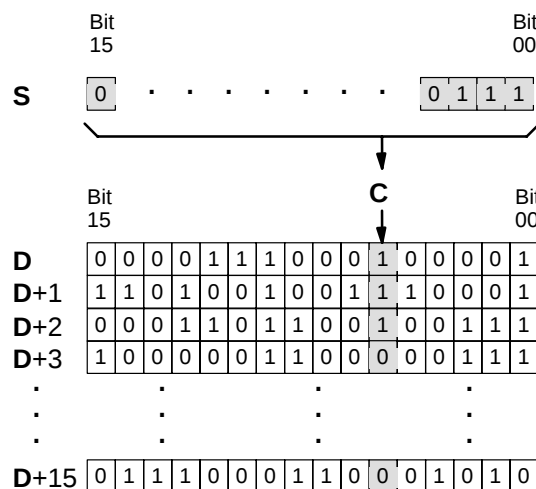
D et D+15 doivent appartenir à la même zone de données.

Les DM 6129 à 6655 ne peuvent être utilisés pour D.

C doit être en BCD entre #0000 et #0015.

Description

Lorsque la condition d'exécution est à OFF, COLM(—) n'est pas exécutée. Lorsque la condition est à ON, COLM(—) copie les16 bits du canal S (00 à 15) et les répartit dans16 canaux (D à D+15) à l'endroit où le bit de colonne C a été sélectionné.



Drapeaux

ER: La désignation du bit C n'est pas en BCD ou spécifie un bit qui n'existe pas (la spécification du bit doit être comprise entre 00 et 15).

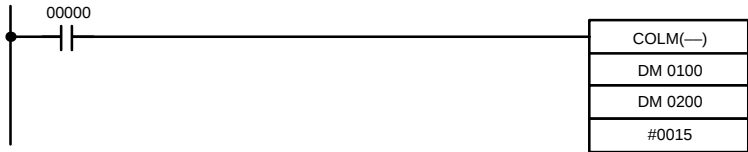
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

D et D+15 n'appartiennent pas à la même zone de données.

EQ: ON lorsque le contenu de S est égal à zéro ; reste à OFF dans les autres cas.

Exemple

Voici un exemple d'utilisation de COLM(—) pour le déplacement du contenu du canal DM 0100 (00 à 15) vers la colonne du bit 15 de la sélection (DM 0200 à 0215).



Adresse	Instruction	Opérande
00000	LD	00000
00001	COLM(—)	
		DM 0100
		DM 0200
		# 0015

5-19-17 2’S COMPLEMENT – NEG(—)

Symboles

NEG(—)	@NEG(—)
S	S
R	R
000	000

Zones de données d’opérande

S: canal source
IR, SR, AR, DM, HR, TC, LR, #
R: canal de résultat
IR, SR, AR, DM, HR, LR
000
Non utilisé. Sélection à 000.

Limites

Cette instruction est disponible avec les UC du **CQM1-CPU4 [-E.]**
Les DM 6144 à 6655 ne peuvent être utilisés pour R.

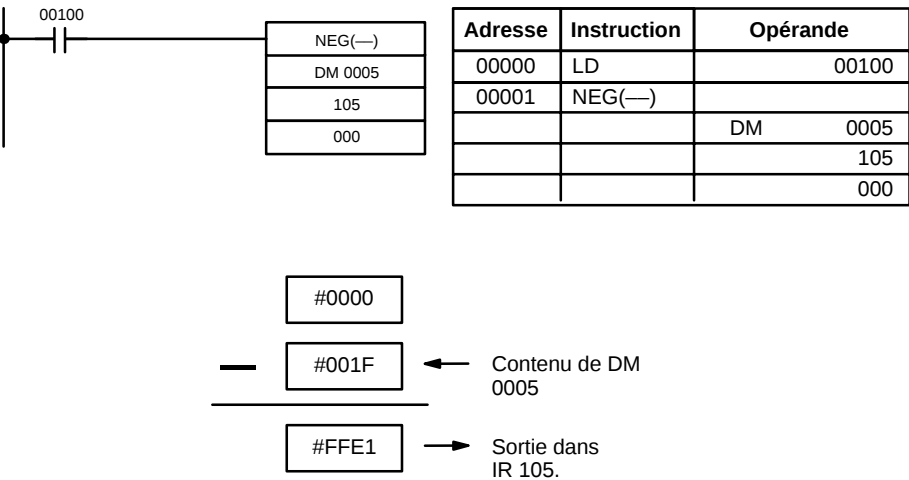
Description

Convertit le contenu hexadécimal à 4 digits du canal source (S) en son complément à 2 et sort le résultat dans le canal de résultat (R). Cette opération revient au même que la soustraction de S à 0000 avec sortie du résultat dans R ; elle calcule la valeur absolue de la donnée binaire en y ajoutant un signe négatif.
Si le contenu de S est égal à 0000, le contenu de R est égal à 0000 après exécution et EQ (SR 25506) passe à ON.
Si le contenu de S est égal à 8000, le contenu de R est égal à 8000 après exécution et UF (SR 25405) passe à ON.

Drapeaux

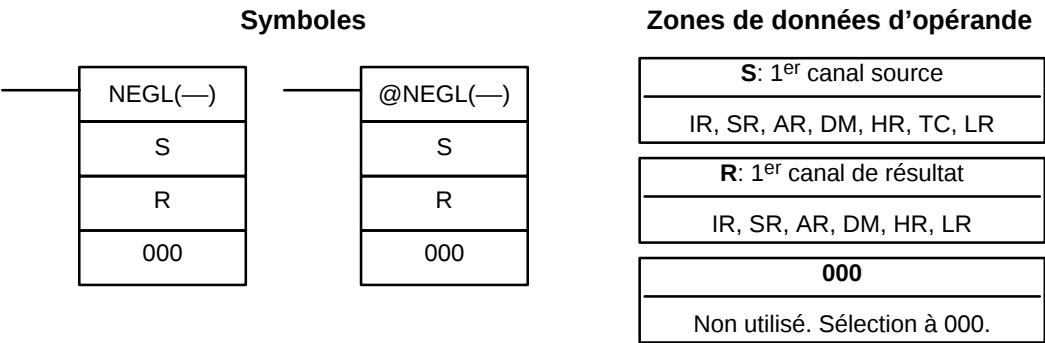
ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
EQ: ON lorsque le contenu de R est égal à zéro ; reste à OFF dans les autres cas.
UF: ON lorsque la contenu de S est égal à 8000 ; reste à OFF dans les autres cas.

Exemple L'exemple suivant montre l'utilisation de NEG(—) pour trouver le complément de 2 du contenu de DM 0005 et la sortie du résultat dans IR 105.

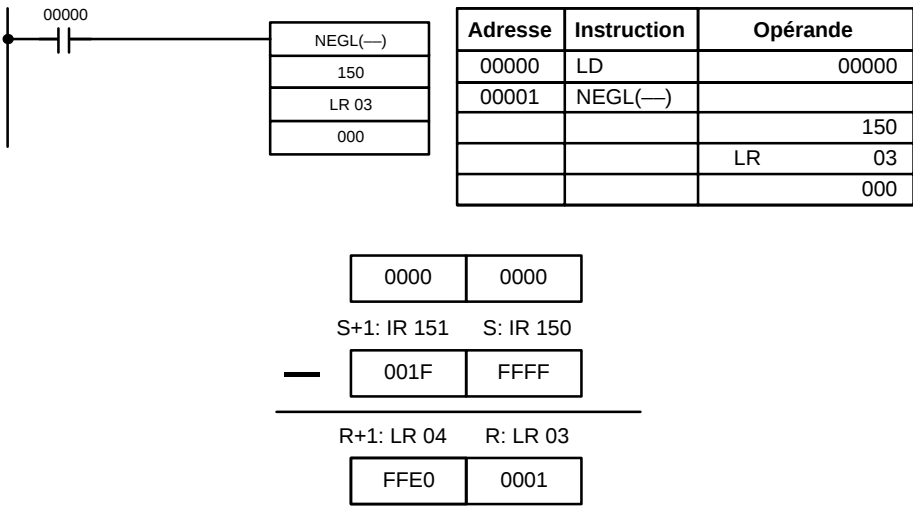


5-19-18

DOUBLE 2'S COMPLEMENT – NEGL(—)



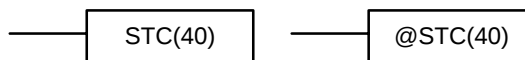
Exemple L'exemple suivant montre l'utilisation de NEGL(—) pour trouver le complément de 2 de la valeur hexadécimale de IR 151, IR 150 (001F FFFF) et la sortie du résultat dans HR 04, HR 03 :



5-20 Instructions de calcul BCD

5-20-1 SET CARRY – FORCAGE DE LA RETENUE – STC(40)

Symboles

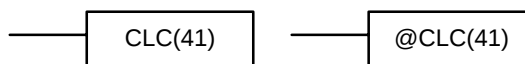


Lorsque la condition d'exécution est à OFF, STC(40) n'est pas exécutée. Lorsque la condition est à ON, STC(40) place le registre de retenue CY (SR 25504) à "1" (ON).

Rem. : reportez-vous à l'annexe C pour la liste des instructions affectant CY.

5-20-2 CLEAR CARRY – ANNULATION DE LA RETENUE – CLC(41)

Ladder Symboles

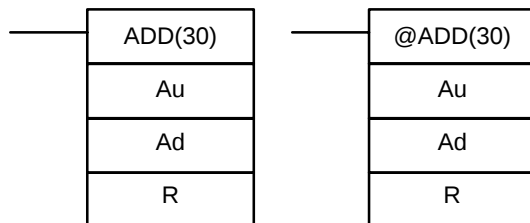


Lorsque la condition d'exécution est à OFF, CLC(41) n'est pas exécutée. Lorsque la condition est à ON, CLC(41) place le registre de retenue CY (SR 25504) à "0" (OFF).

Rem. : reportez-vous à l'annexe C pour la liste des instructions affectant CY.

5-20-3 BCD ADD – ADDITION BCD – ADD(30)

Symboles



Zones de données d'opérande

Au: 1 ^{er} canal de l'addition (BCD)
IR, SR, AR, DM, HR, TC, LR, #
Ad: 2 ^e canal de l'addition (BCD)
IR, SR, AR, DM, HR, TC, LR, #
R: canal de résultat
IR, SR, AR, DM, HR, LR

Limites

Effectue une addition entre deux données BCD à 4 digits.
Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, ADD(30) n'est pas exécutée. Lorsque la condition est à ON, ADD(30) additionne le contenu de Au, Ad et CY et place le résultat dans R. CY est sélectionné si le résultat est supérieur à 9999. Au et Ad ne peuvent être des constantes.

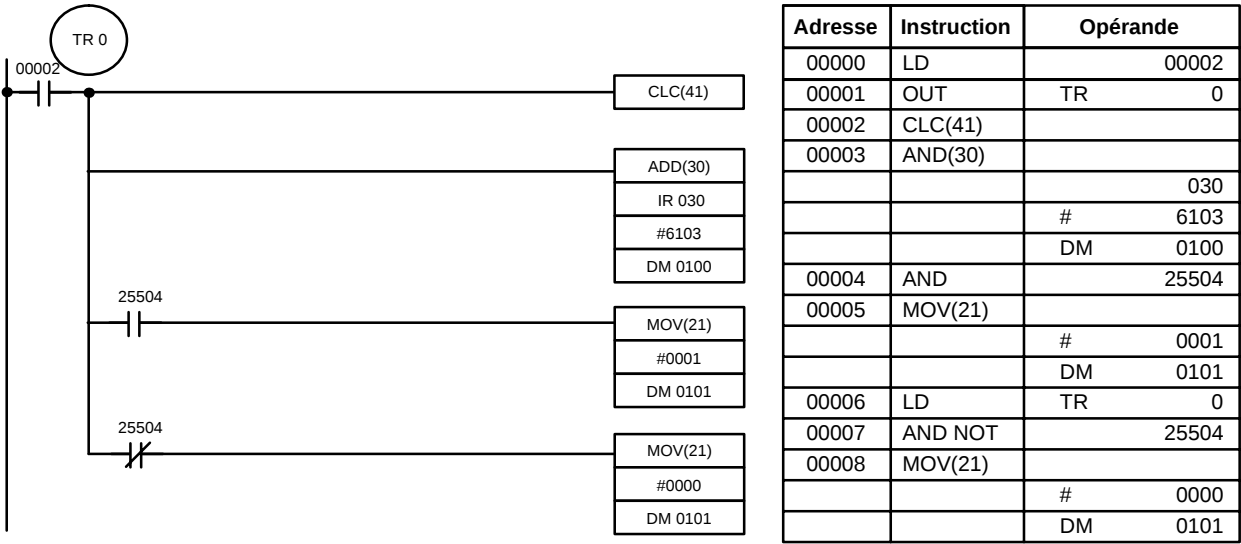
$$\boxed{\text{Au}} + \boxed{\text{Ad}} + \boxed{\text{CY}} \rightarrow \boxed{\text{CY}} \quad \boxed{\text{R}}$$

Drapeaux

- ER:** Au et/ou Ad n'est pas en BCD.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- CY:** ON lorsqu'il y a une retenue dans le résultat.
- EQ:** ON lorsque le résultat est égal à 0.

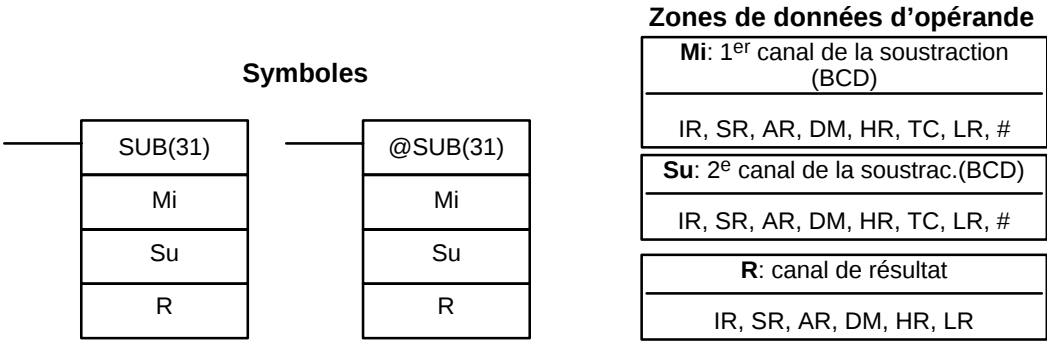
Exemple

Si 00002 est à ON, le programme représenté par le schéma suivant efface CY avec CLC(41), ajoute le contenu de IR 030 à une constante (6103), et place le résultat dans DM 0100. Le programme déplace ensuite tous les zéros ou bien 0001 dans DM 0101 selon l'état de CY (25504). Cela garantit la préservation de toutes les retenues à partir du dernier chiffre dans R+1 de façon à ce que le résultat entier puisse être manipulé plus tard comme une donnée à 8 chiffres.



Bien que l'on puisse utiliser deux ADD(30) pour une addition BCD à 8 chiffres, ADDL(54) sert spécialement à cet effet.
Avant d'exécuter l'instruction ADD, CLC doit être programmée pour annuler le drapeau de retenue.

5-20-4 BCD SUBTRACT – SOUSTRACTION BCD – SUB(31)



Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

CY: ON lorsque le résultat est négatif, c'est-à-dire que Mi est inférieur à Su plus CY.

EQ: ON lorsque le résultat est égal à 0.

Attention : veillez à effacer le drapeau de retenue avec CLC(41) avant d'exécuter SUB(31) si son état précédent n'est pas nécessaire et vérifiez l'état de CY après soustraction avec SUB(31). Si CY est à ON après exécution de SUB(31) (résultat négatif), le résultat est sorti en complément de 10 de la réponse réelle. Pour convertir le résultat sorti en valeur réelle, soustrayez la valeur de R de 0.

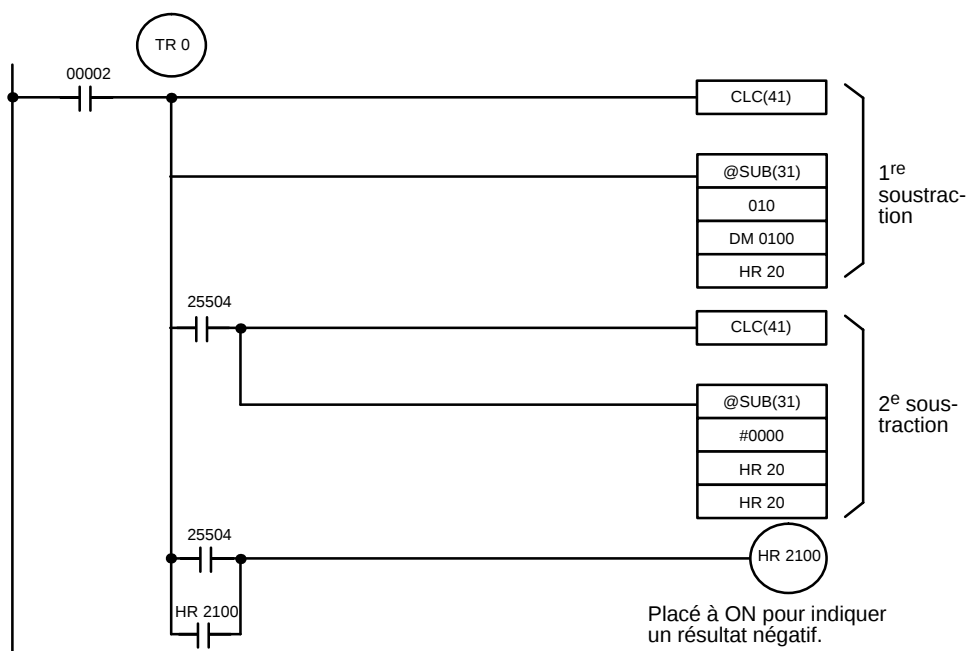
Exemple

Lorsque 00002 est à ON, le schéma à relais suivant efface CY, soustrait le contenu de DM 0100 et CY du contenu de 010 et place le résultat dans HR 20.

Si CY est sélectionné en exécutant SUB(31), le résultat de HR 20 est soustrait de zéro (CLC(41) est à nouveau nécessaire pour obtenir un résultat précis), le résultat est remplacé dans HR 20 et HR 2100 est placé à ON pour indiquer un résultat négatif.

Si CY n'est pas sélectionné en exécutant SUB(31), le résultat est positif, la deuxième soustraction n'est pas effectuée et HR 2100 n'est pas placée à ON. HR 2100 est programmé comme bit d'auto-maintien de façon à ce qu'une modification de l'état de CY ne le place pas à OFF au deuxième tour de scrutation du programme.

Dans cet exemple, les formes sur changement d'état de SUB(31) sont utilisées de façon à ce que la soustraction soit effectuée une seule fois à chaque passage à ON de 00002. Lors de l'exécution d'une autre soustraction, 00002 devra être placé à OFF pendant au moins un tour de scrutation (RAZ de HR 2100) puis remplacé à ON.



Adresse	Instruction	Opérande
00000	LD	00002
00001	OUT	TR 0
00002	CLC(41)	
00003	@SUB(31)	
		010
		DM 0100
		HR 20
00004	AND	25504
00005	CLC(41)	
00006	@SUB(31)	
		# 0000
		HR 20
		HR 20
00007	LD	TR 0
00008	AND	25504
00009	OR	HR 2100
00010	OUT	HR 2100

La 1^{re} et la 2^e soustraction de ce schéma sont montrées ci-dessous en utilisant une donnée d'exemple pour 010 et DM 0100.

Rem. : l'opération réelle de SUB(31) implique une soustraction de Su et CY de 10 000 plus Mi. Pour les résultats positifs, le chiffre d'extrême gauche est tronqué. Pour les résultats négatifs, on recherche le complément de 10. Voici la procédure d'obtention de la réponse correcte :

1^{re} soustraction

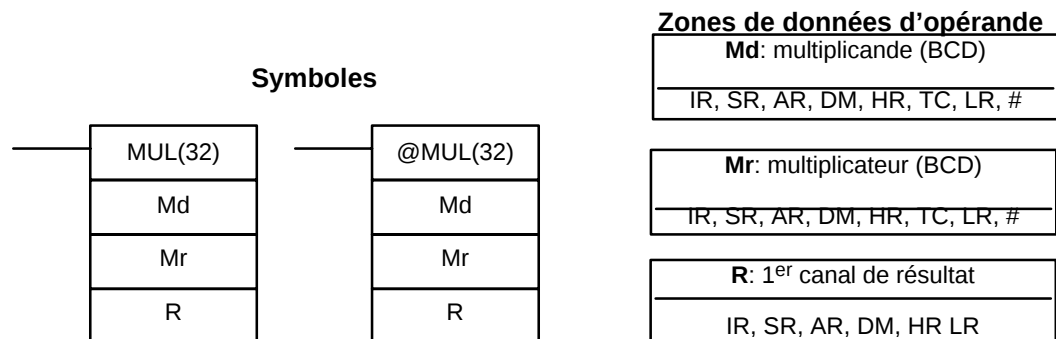
Mi	IR 010	1029	
Su	DM 0100	- 3452	
CY	CY	- 0	
R	HR 20	7577	(1029 + (10000 - 3452))
	CY	1	(résultat négatif)

2^e soustraction

Mi		0000	
Su	HR 20	-7577	
CY	CY	-0	
R	HR 20	2423	(0000 + (10000 - 7577))
	CY	1	(résultat négatif) : - 2423

Dans le cas ci-dessus, le programme doit placer HR 2100 à ON pour indiquer que la valeur de HR 20 est négative.

5-20-5 BCD MULTIPLY – MULTIPLICATION BCD – MUL(32)

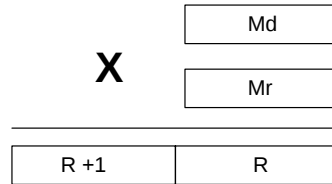


Limites

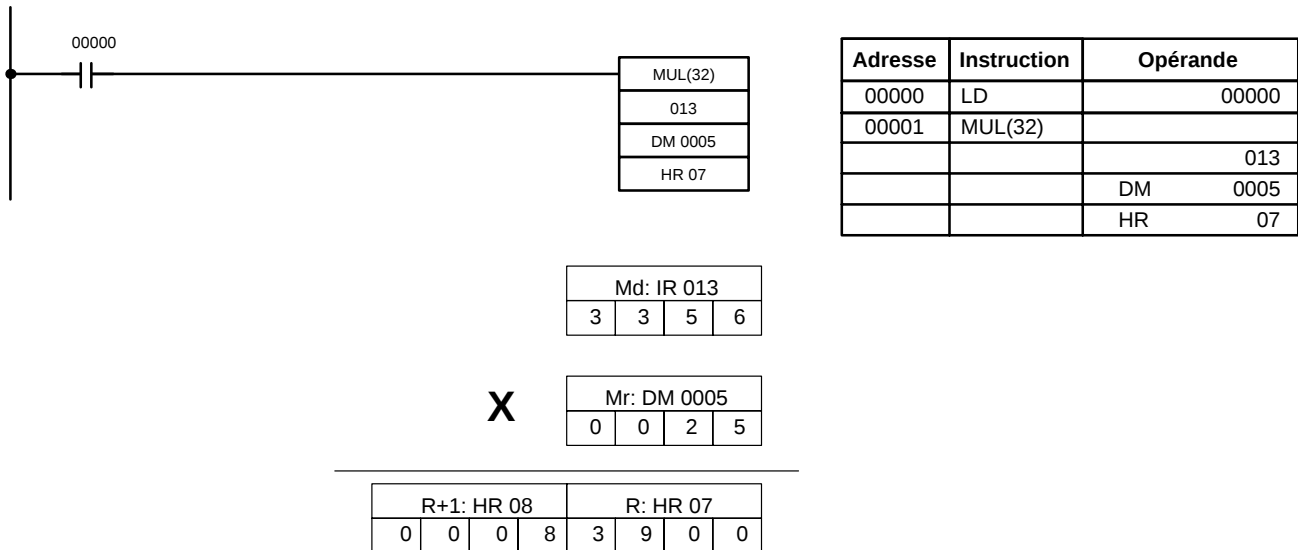
Effectue une multiplication entre deux données BCD à 4 chiffres.
Les DM 6143 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, MUL(32) n'est pas exécutée. Lorsque la condition est à ON, MUL(32) multiplie Md par le contenu de Mr et place le résultat dans R et R+1.

**Exemple**

Lorsque IR 00000 est à ON avec le programme suivant, les contenus de IR 013 et de DM 0005 sont multipliés et le résultat est placé dans HR 07 et HR 08. Voici quelques données et calculs montrés en exemple :

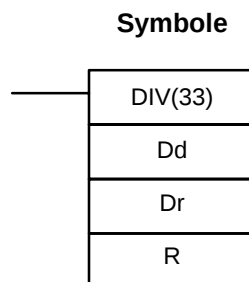
**Drapeaux**

ER: Md et/ou Mr ne sont pas en BCD.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

CY: ON lorsqu'il y a une retenue dans le résultat.

EQ: ON lorsque le résultat est égal à 0.

5-20-6 BCD DIVIDE – DIVISION BCD – DIV(33)**Zones de données d'opérande**

Dd: dividende (BCD)
IR, SR, AR, DM, HR, TC, LR, #
Dr: diviseur (BCD)
IR, SR, AR, DM, HR, TC, LR, #
R: 1 ^{er} canal de résultat (BCD)
IR, SR, AR, DM, HR, LR

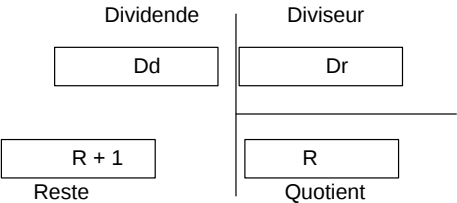
Limites

R et R+1 doivent appartenir à la même zone de données. Les DM 6143 à 6655 ne peuvent être utilisés pour R.

Description

Exécute une division entre deux données BCD à 4 chiffres. Lorsque la condition d'exécution est à OFF, DIV(33) n'est pas exécutée et le programme passe à

l'instruction suivante. Lorsque la condition est à ON, Dd est divisé par Dr et le résultat est placé dans R et R + 1 : le quotient est placé dans R et le reste dans R + 1.



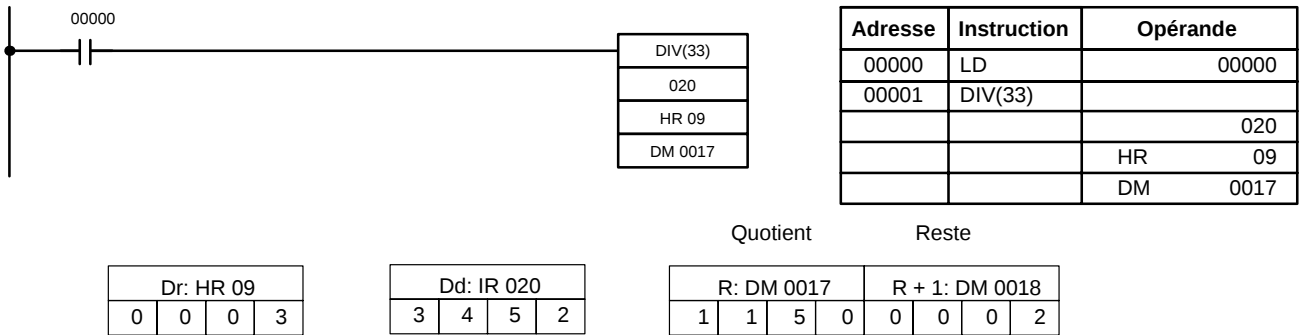
Drapeaux

- ER:** Dd ou Dr n'est pas en BCD.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- EQ:** ON lorsque le résultat est égal à 0.

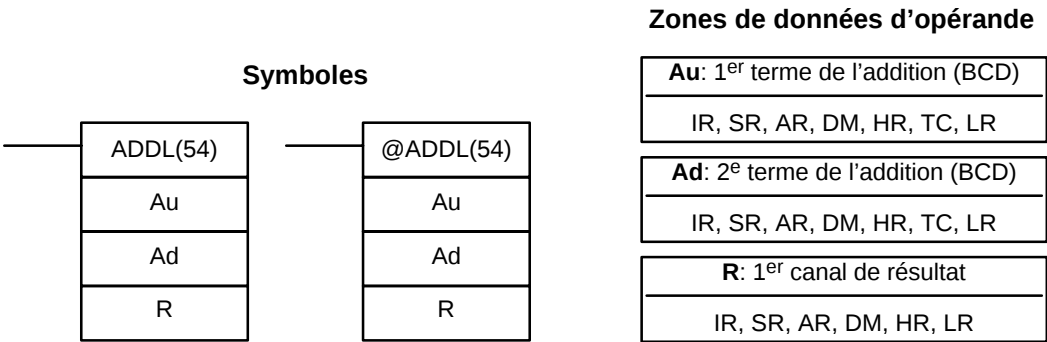
Rem. : Dd et Dr peuvent être des constantes mais pas simultanément.

Exemple

Lorsque IR 00000 est à ON avec le programme suivant, le contenu de IR 020 est divisé par HR 09 et le résultat est placé dans DM 0017 et DM 0018. Voici des exemples de données et calculs :



5-20-7 DOUBLE BCD ADD – ADDITION DOUBLE LONGUEUR – ADDL(54)



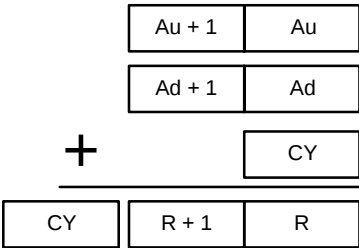
Limites

Effectue une addition entre 2 données BCD à 8 chiffres (2 canaux), retenue comprise, et sort le résultat sur les canaux spécifiés.
Les DM 6143 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, ADDL(54) n'est pas exécutée. Lorsque la condition est à ON, ADDL(54) ajoute le contenu de CY à la valeur à 8 chiffres de Au et Au+1 ainsi qu'à la valeur à 8 chiffres de Ad et Ad+1, et place le

résultat dans R et R+1. CY est activé (à ON) si le résultat est supérieur à 99999999.



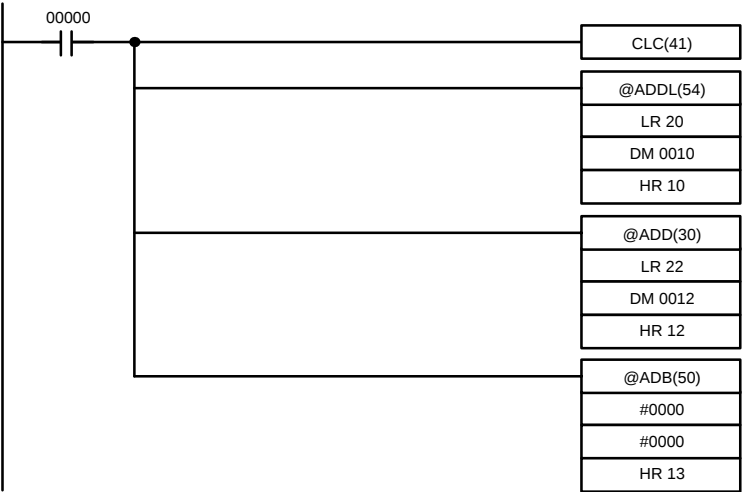
Drapeaux

- ER:** Au et/ou Ad ne sont pas en BCD.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- CY:** ON lorsque le résultat contient une retenue.
- EQ:** ON lorsque le résultat est égal à 0.

Exemple

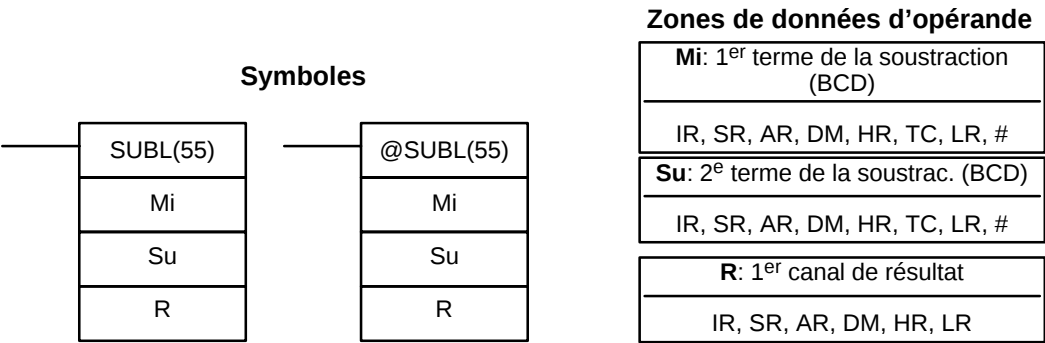
Lorsque 00000 est à ON, la partie de programme qui suit ajoute deux nombres à 12 chiffres : le premier de LR 20 à LR 22 et le deuxième de DM 0010 à DM 0012. Le résultat est placé dans HR 10 à HR 13.

Les 8 chiffres d'extrême droite des deux nombres sont additionnés avec ADDL(54) : le contenu de LR 20 à LR 21 est additionné à DM 0010 et DM 0011 et le résultat est placé dans HR 10 et HR 11. La deuxième addition ajoute les 4 chiffres d'extrême gauche de chaque nombre utilisant ADD(30) et inclut la retenue de la première addition. La dernière instruction, ADB(50) (cf 5-21-1 BINARY ADD – ADB(50)) ajoute deux constantes composées de zéros et place la retenue de la deuxième addition dans HR 13.



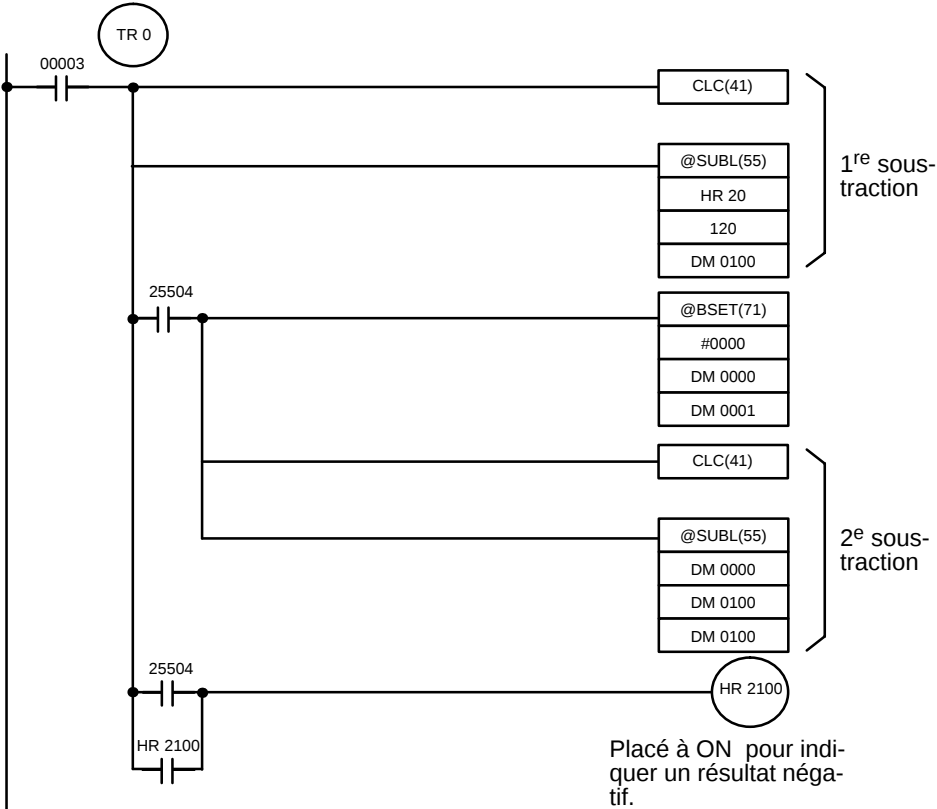
Adresse	Instruction	Opérande
00000	LD	00000
00001	CLC(41)	
00002	@ADDL(54)	
		LR 20
		DM 0010
		HR 10
00003	@ADD(30)	
		LR 22
		DM 0012
		HR 12
00004	@ADB(50)	
		# 0000
		# 0000
		HR 13

5-20-8 DOUBLE BCD SUBTRACT – SOUSTRACTION BCD A DOUBLE LONGUEUR – SUBL(55)



Exemple

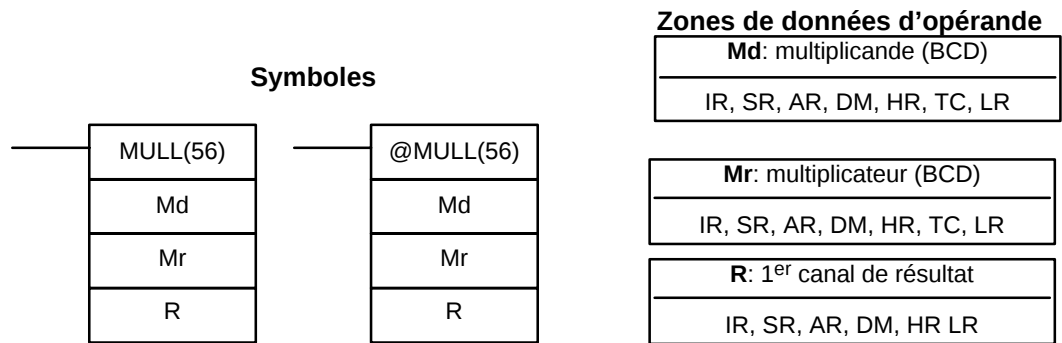
L'exemple suivant fonctionne pratiquement comme celui de la soustraction simple, mais BSET(71) est nécessaire pour effacer le contenu de DM 0000 et de DM 0001, de façon à ce qu'un résultat négatif puisse être soustrait de 0 (l'entrée d'une constante à 8 chiffres est impossible).



Adresse	Instruction	Opérande
00000	LD	00003
00001	OUT	TR 0
00002	CLC(41)	
00003	@SUBL(55)	
		HR 20
		120
		DM 0100
00004	AND	25504
00005	@BSET(71)	
		# 0000
		DM 0000
		DM 0001

Adresse	Instruction	Opérande
00006	CLC(41)	
00007	@SUBL(55)	
		DM 0000
		DM 0100
		DM 0100
00008	LD	TR 0
00009	AND	25504
00010	OR	HR 2100
00011	OUT	HR 2100

5-20-9 DOUBLE BCD MULTIPLY – MULTIPLICATION BCD A DOUBLE LONGUEUR – MULL(56)



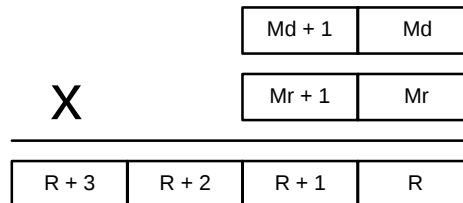
Limites

Effectue une multiplication entre deux données BCD à 8 chiffres et sort le résultat sur les canaux spécifiés.

Les DM 6141 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, MULL(56) n'est pas exécutée. Lorsque la condition est à ON, MULL(56) multiplie le contenu à 8 chiffres de Md et Md+1 par le contenu de Mr et Mr+1, et place la résultat dans R à R+3.



Drapeaux

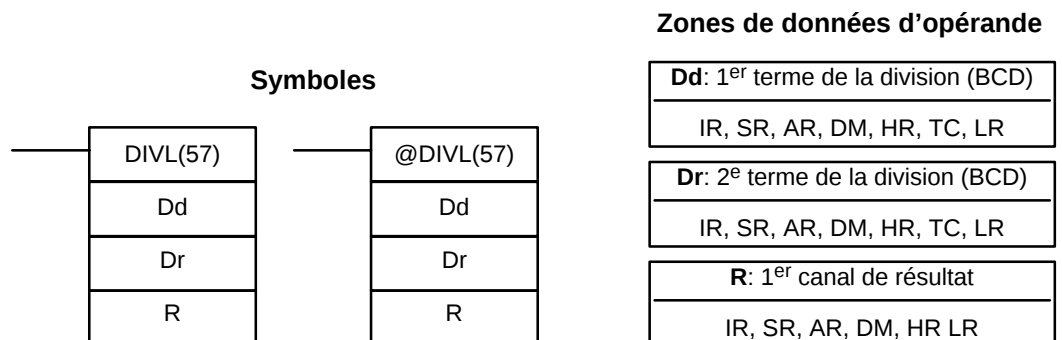
ER: Md, Md+1, Mr ou Mr+1 ne sont pas en BCD.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

CY: ON lorsqu'il y a une retenue dans le résultat.

EQ: ON lorsque le résultat est égal à 0.

5-20-10 DOUBLE BCD DIVIDE – DIVISION DOUBLE LONGUEUR – DIVL(57)



Limites

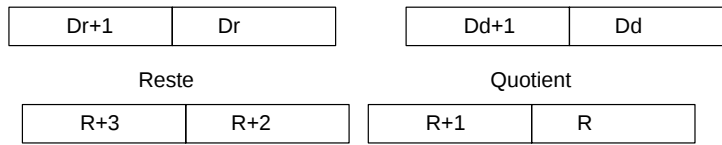
Effectue une division entre 2 données BCD à 8 chiffres sans virgule et sort le résultat sur les canaux spécifiés.

Les DM 6141 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, DIVL(57) n'est pas exécutée. Lorsque la condition est à ON, le contenu à 8 chiffres de Dd et D+1 est divisé par le

contenu de Dr et Dr+1 et le résultat est placé dans R à R+3 : le quotient est placé dans R et R+1, le reste dans R+2 et R+3.

**Drapeaux**

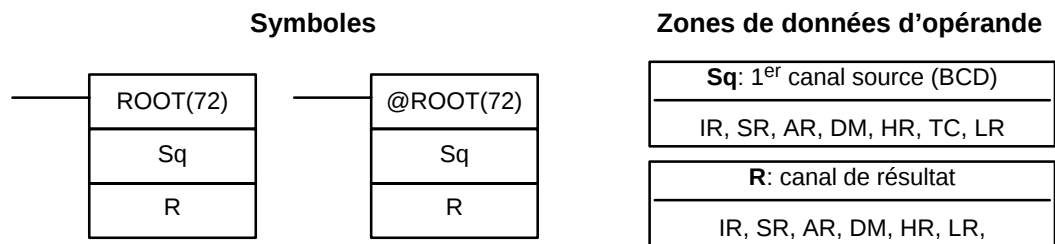
ER: Dr et Dr+1 ont 0 pour contenu.

Dd, Dd+1, Dr ou Dr+1 ne sont pas en BCD.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à 0.

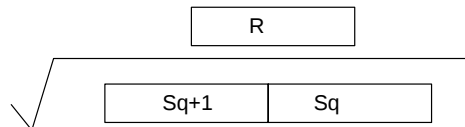
5-20-11 SQUARE ROOT – RACINE CARREE – ROOT(72)

**Limites**

Calcule la racine carrée d'une donnée BCD à 8 chiffres.
Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, ROOT(72) n'est pas exécutée.
Lorsque la condition est à ON, ROOT(72) calcule la racine carrée du contenu à 8 chiffres de Sq et Sq+1 et place le résultat dans R. Le résultat est arrondi sur la partie précédant la virgule.

**Drapeaux**

ER: Sq n'est pas en BCD.

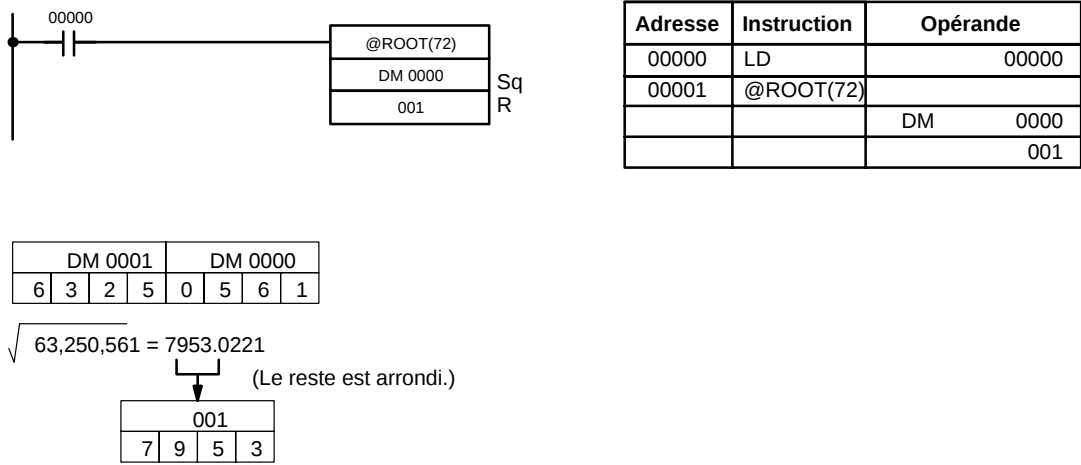
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à 0.

Exemple

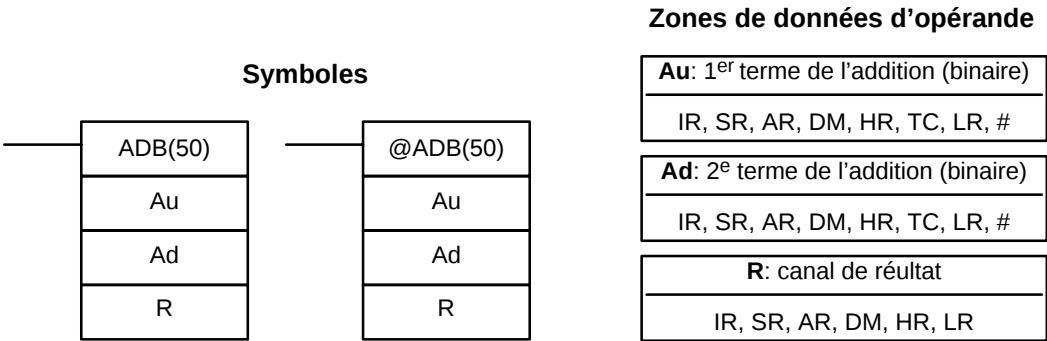
L'exemple suivant montre le calcul de la racine carrée d'un nombre à 8 chiffres. Le résultat est un nombre à 4 chiffres, le reste étant arrondi et le résultat également.

Dans cet exemple, $\sqrt{63250561} = 7953,0221\dots$, arrondi à 7953.



5-21 Instructions de calcul binaire

5-21-1 BINARY ADD – ADDITION BINAIRE – ADB(50)



- Limites**

Effectue une addition entre 2 données binaires à 16 bits.
Les DM 6144 à 6655 ne peuvent être utilisés pour R.
- Description**

Lorsque la condition d'exécution est à OFF, ADB(50) n'est pas exécutée. Lorsque la condition est à ON, ADB(50) ajoute le contenu de Au, Ad et CY et place le résultat dans R. CY est activé si le résultat est supérieur à FFFF.

Au

+

Ad

+

CY

→

CY

R

ADB(50) peut également être utilisé pour ajouter une donnée binaire portant un signe. Avec les UC CQM1-CPU4 ~~[-E, Ics]~~, les drapeaux de gamme non respectée (dépassement de niveau) SR 25404 et SR 25405 indiquent si le résultat dépasse les gammes haute et basse des données binaires à 16 bits avec signe.

- Drapeaux**
- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

CY: ON lorsque le résultat est supérieur à FFFF.

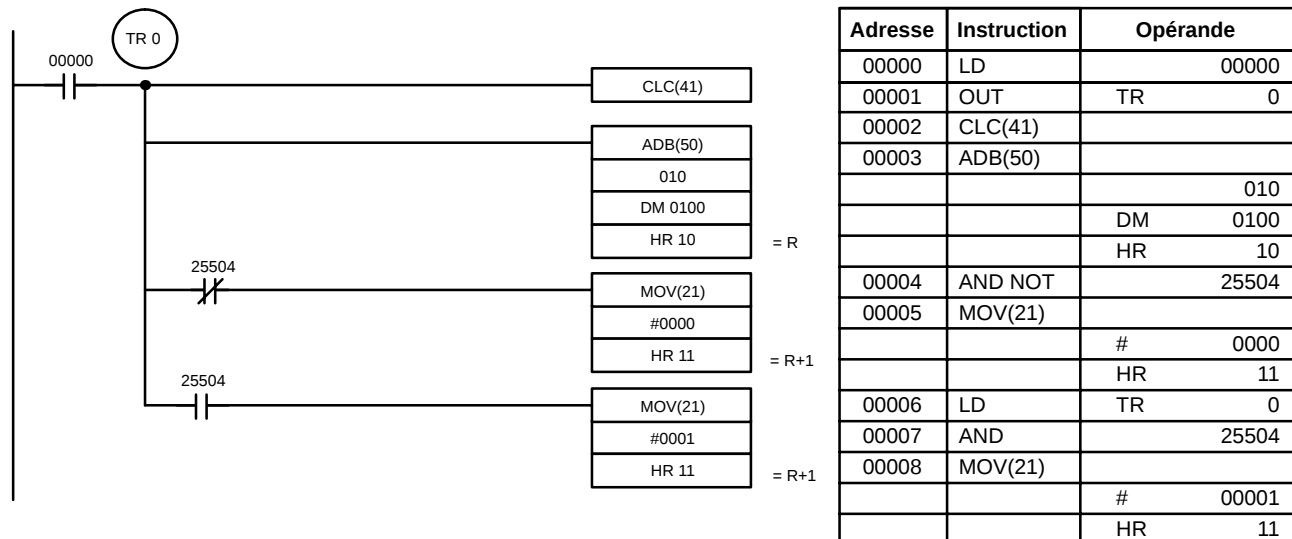
EQ: ON lorsque le résultat est égal à 0.

OF: ON lorsque le résultat dépasse + 32 767 (7FFF) (CQM1-CPU4 ~~[-E uniquement]~~).

UF: ON lorsque le résultat est inférieur à – 32 768 (8000) (CQM1-CPU4 ~~[-E uniquement]~~).

Exemple

L'exemple suivant montre une addition à 4 chiffres avec CY plaçant #0000 ou #0001 dans R+1 pour préserver les retenues :



Dans le cas ci-dessous, $A6E2 + 80C5 = 127A7$. Le résultat est un nombre à 5 chiffres ; donc CY (SR 25504) = 1 et le contenu de R + 1 passe à #0001.

CY = 0 : le drapeau de retenue a été remis à zéro par CLC

Au: IR 010			
A	6	E	2

+

Ad: DM 0100			
8	0	C	5

R+1: HR 11				R: HR 10			
0	0	0	1	2	7	A	7

puisque le résultat est supérieur à FFFF

Rem. : pour les calculs binaires avec signe, l'état des drapeaux UF et OF indique si le résultat dépasse la gamme des données binaires portant un signe : -32 768 (8000) à +32 767 (7FFF) (CQM1-CPU4 uniquement).

5-21-2 BINARY SUBTRACT – SOUSTRACTION BINAIRE – SBB(51)

Symboles

SBB(51)	@SBB(51)
Mi	Mi
Su	Su
R	R

Zones de données d'opérande

Mi: 1 ^{er} terme de la soustraction (binaire)
IR, SR, AR, DM, HR, TC, LR, #
Su: 2 ^e terme de la sous. (binaire)
IR, SR, AR, DM, HR, TC, LR, #
R: canal de résultat
IR, SR, AR, DM, HR, LR

Limites

Effectue une soustraction entre 2 données binaires à 16 bits.
Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, SBB(51) n'est pas exécutée. Lorsque la condition est à ON, SBB(51) soustrait le contenu de Su et CY de Mi et

place le résultat dans R. Si le résultat est négatif, CY est activé et le complément de 2 du résultat réel est placé dans R.

Mi

−

Su

−

CY

→

CY

R

SBB(51) peut également être utilisé pour soustraire les données binaires portant un signe. Avec les UC CQM1-CPU4 ~~[-E, les~~ drapeaux de dépassement de gamme SR 25404 et SR 25405 indiquent si le résultat a dépassé les limites des gammes haute et basse des données à 16 bits portant un signe.

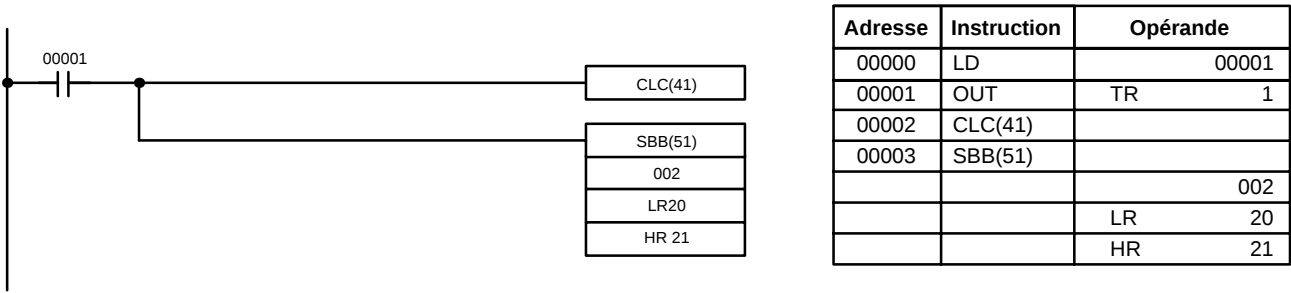
Drapeaux

- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- CY:** ON lorsque le résultat est négatif (Mi inférieur à Su plus CY).
- EQ:** ON lorsque le résultat est égal à 0.
- OF:** ON lorsque le résultat est supérieur à + 32 767 (7FFF) (CQM1-CPU4 ~~[-E uniquement~~).
- UF:** ON lorsque le résultat est inférieur à − 32 768 (8000) (CQM1-CPU4 ~~[-E uniquement~~).

Exemple

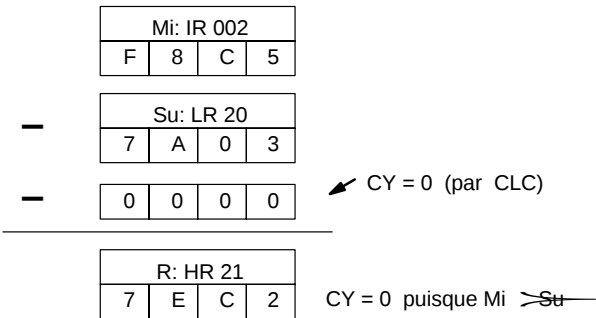
L'exemple suivant montre une soustraction à 4 chiffres. Lorsque IR 00001 est à ON, le contenu de LR 20 et de CY est soustrait du contenu de IR 002 et le résultat est écrit dans HR 21.

CY est placé à ON si le résultat est négatif. Si l'on utilise une donnée normale, un résultat négatif (binaire avec signe) doit être converti en donnée normale en utilisant NEG(—). Reportez-vous au chapitre 5-19-17 2's COMPLEMENT – NEG(—) pour de plus amples détails.



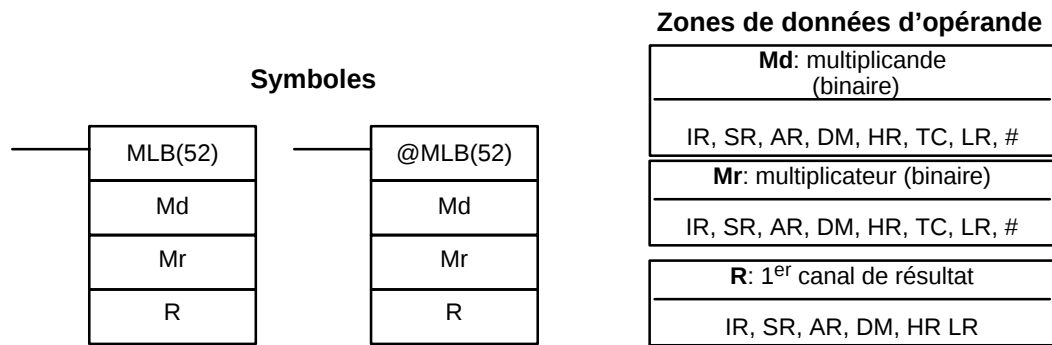
Dans l'exemple ci-dessous, le contenu de LR 20 (#7A03) et de CY est soustrait de IR 002 (#F8C5). Depuis que le résultat est positif, CY est égal à 0.

Si le résultat était négatif, CY serait placé à 1. Pour les données normales (sans signe), le résultat doit être converti en complément de 2.



Rem. : pour les calculs binaires avec signe, l'état des drapeaux UF et OF indique si le résultat dépasse la gamme des données binaires avec signe : − 32 768 (8000) à + 32 767 (7FFF) (CQM1-CPU4 ~~[-E uniquement~~).

5-21-3 BINARY MULTIPLY – MULTIPLICATION BINAIRE – MLB(52)

**Limites**

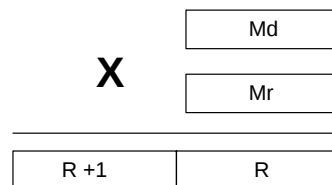
Effectue une multiplication entre deux données binaires à 16 bits.

Les DM 6143 à 6655 ne peuvent être utilisés pour R.

MLB(52) ne peut être utilisée pour multiplier les données binaires avec signe. Utilisez MBS(—) à la place et reportez-vous au chapitre 5-21-7 *SIGNED BINARY MULTIPLY – MBS(—)* pour de plus amples détails.

Description

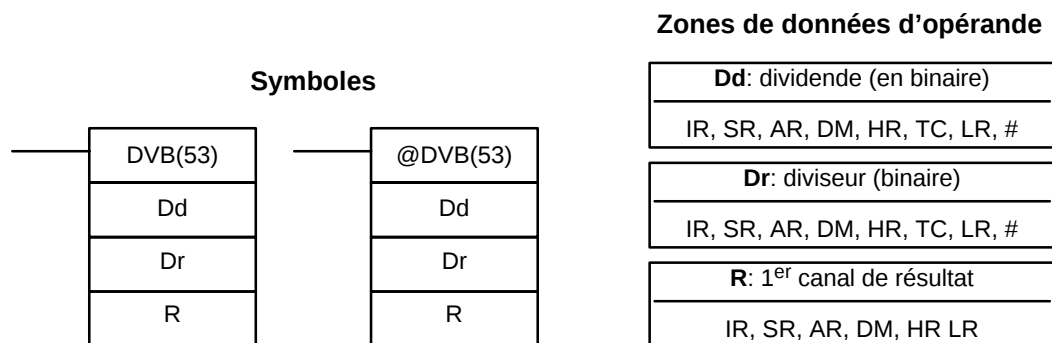
Lorsque la condition d'exécution est à OFF, MLB(52) n'est pas exécutée. Lorsque la condition est à ON, MLB(52) multiplie le contenu de Md par celui de Mr, place les 4 chiffres d'extrême droite du résultat dans R et les 4 chiffres d'extrême gauche dans R+1. Le résultat de l'opération comporte 32 bits.

**Drapeaux**

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à 0.

5-21-4 BINARY DIVIDE – DIVISION BINAIRE – DVB(53)

**Limites**

Effectue une division entre deux données binaires à 16 bits et sort le quotient et le reste sur les canaux spécifiés.

Les DM 6143 à 6655 ne peuvent être utilisés pour R.

DVB(53) ne peut être utilisée pour diviser une donnée binaire portant un signe. Utilisez DBS(—) à la place et reportez-vous au chapitre 5-21-9 *SIGNED BINARY DIVIDE – DBS(—)* pour de plus amples détails.

Description

Lorsque la condition d'exécution est à OFF, DVB(53) n'est pas exécutée. Lorsque la condition est à ON, DVB(53) divise le contenu de Dd par le contenu de Dr

et le résultat est placé dans R et R+1 : le quotient est placé dans R et le reste dans R+1.

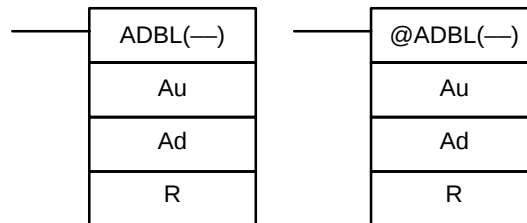
**Drapeaux**

ER: Dr a 0 pour contenu.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à 0.

5-21-5 DOUBLE BINARY ADD – ADDITION DOUBLE LONGUEUR BINAIRE – ADBL(--)

Symboles**Zones de données d'opérande**

Au: 1 ^{er} terme de l'addition (binaire)
IR, SR, AR, DM, HR, TC, LR, #
Ad: 2 ^e terme de l'addition (binaire)
IR, SR, AR, DM, HR, TC, LR, #
R: 1 ^{er} canal de résultat
IR, SR, AR, DM, HR, LR

Limites

Effectue une addition entre 2 données binaires à 8 chiffres (2 canaux), retenue comprise, et sort le résultat sur les canaux spécifiés.

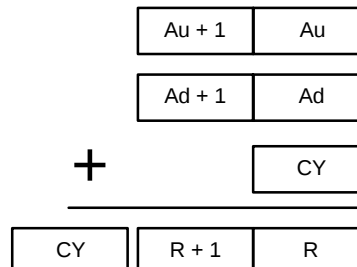
Cette instruction est disponible avec les UC du **CQM1-CPU4** E.

Au et Au+1 doivent appartenir à la même zone de données, ainsi que Ad et Ad+1, et R et R+1.

Les DM 6142 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, ADBL(--)) n'est pas exécutée. Lorsque la condition est à ON, ADBL(--)) ajoute le contenu à 8 chiffres de Au+1 et Au, le contenu à 8 chiffres de Ad+1 et Ad et CY et place le résultat dans R. CY est activé si le résultat est supérieur à FFFF FFFF.



ADBL(--)) peut également être utilisée pour ajouter des données binaires portant un signe. Les drapeaux de gamme non respectée (dépassement de niveau) SR 25404 et SR 25405 indiquent si le résultat a dépassé la gamme haute ou basse des données binaires à 32 bits.

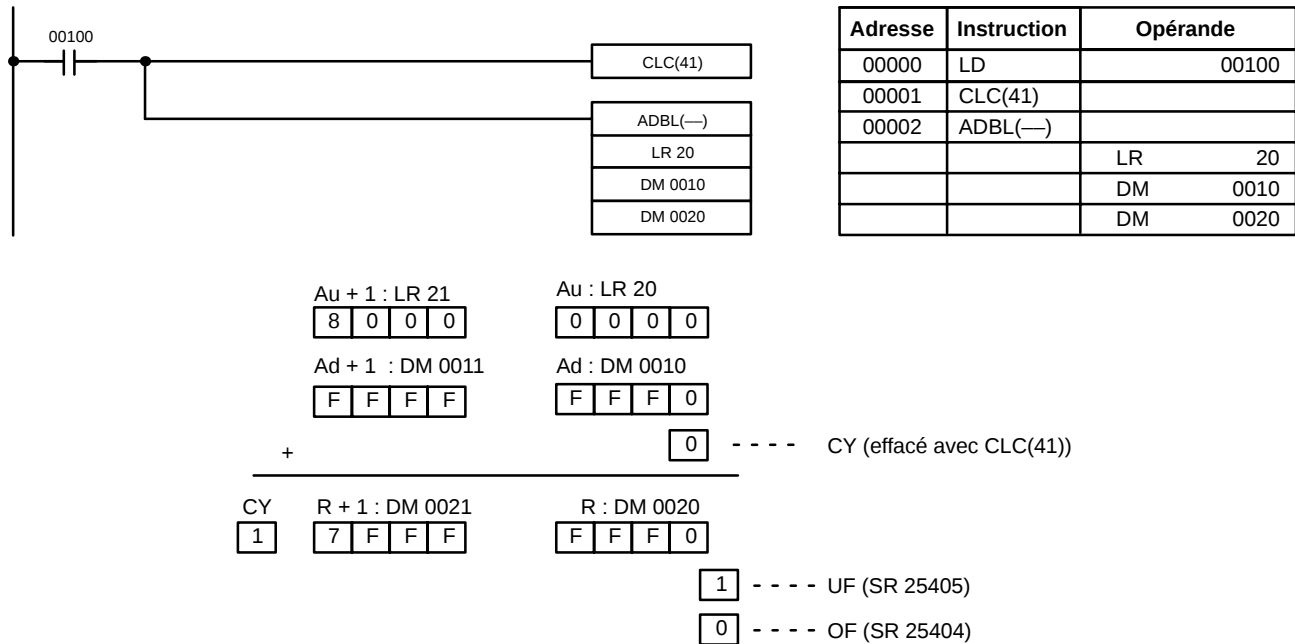
Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

CY: ON lorsque le résultat est supérieur à FFFF FFFF.
EQ: ON lorsque le résultat est égal à 0.
OF: ON lorsque le résultat dépasse + 2 147 483 647 (7FFF FFFF).
UF: ON lorsque le résultat est inférieur à – 2 147 483 648 (8000 0000).

Exemple

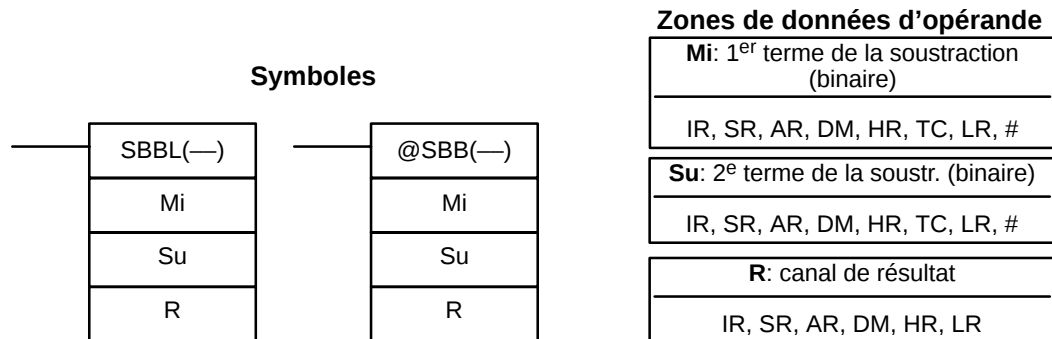
L'exemple suivant montre une addition à 8 chiffres avec CY (SR 25504) utilisée pour représenter l'état du 9^e chiffre. L'état des drapeaux UF et OF indique si le résultat a dépassé la gamme des données binaires portant un signe :
 – 2 147 483 648 (8000 0000) à + 2 147 483 647 (7FFF FFFF).



Rem. : 1. Pour l'addition binaire sans signe, CY indique que la somme des deux valeurs dépasse FFFF FFFF (UF et OF peuvent être ignorés).

2. Pour l'addition portant un signe, le drapeau UF indique que la somme des deux valeurs est inférieure à – 2 147 483 648 (8000 0000). CY peut être ignoré.

5-21-6 DOUBLE BINARY SUBTRACT – SOUSTRACTION BINAIRE DOUBLE LONGUEUR – SBBL(—)

**Limites**

Effectue une soustraction binaire entre 2 données à 8 chiffres avec retenue et sort le résultat sur les canaux spécifiés.

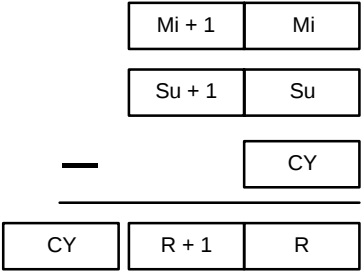
Cette instruction est disponible avec les UC du **CQM1-CPU4 [-E.]**

Mi et Mi+1 doivent appartenir à la même zone de données, ainsi que Su et Su+1, et R et R+1.

Les DM 6142 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, SBBL(—) n'est pas exécutée. Lorsque la condition est à ON, SBBL(—) soustrait CY et la valeur à 8 chiffres de Su et Su+1 de la valeur à 8 chiffres de Mi et Mi+1 et place le résultat dans R et R+1. Si le résultat est négatif, CY est sélectionné et le complément de 2 du résultat réel est placé dans R+1 et R. Utilisez NEGL(—) pour convertir le complément de 2 en résultat réel.



SBBL(—) peut également être utilisée pour soustraire des données binaires portant un signe. Les drapeaux de dépassement de niveau (gamme non respectée) SR 25404 et SR 25405 indiquent si le résultat a dépassé les gammes haute ou basse des données binaires à 32 bits portant un signe.

Drapeaux

- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- CY:** ON lorsque le résultat est négatif (Mi inférieur à Su plus CY).
- EQ:** ON lorsque le résultat est égal à 0.
- OF:** ON lorsque le résultat dépasse + 2 147 483 647 (7FFF FFFF).
- UF:** ON lorsque le résultat est inférieur à – 2 147 483 648 (8000 0000).

Exemple

L'exemple suivant montre une soustraction à 8 chiffres avec CY (SR 25504) utilisé pour indiquer un résultat négatif (avec une donnée sans signe). L'état des drapeaux UF et OF indique si le résultat a dépassé la gamme des données binaires portant un signe :
– 2 147 483 648 (8000 0000) à + 2 147 483 647 (7FFF FFFF).

Adresse	Instruction	Opérande
00000	LD	00101
00001	CLC(41)	
00002	SBBL(—)	
		LR 22
		DM 0012
		DM 0022

Mi + 1 : LR 23

7 F F F

Su + 1 : DM 0023

F F F F

–

–

CY

1

Mi : LR 22

F F F 0

Su : DM 0022

F F F 0

–

–

R + 1 : LR 23

8 0 0 0

R : LR 22

0 0 0 0

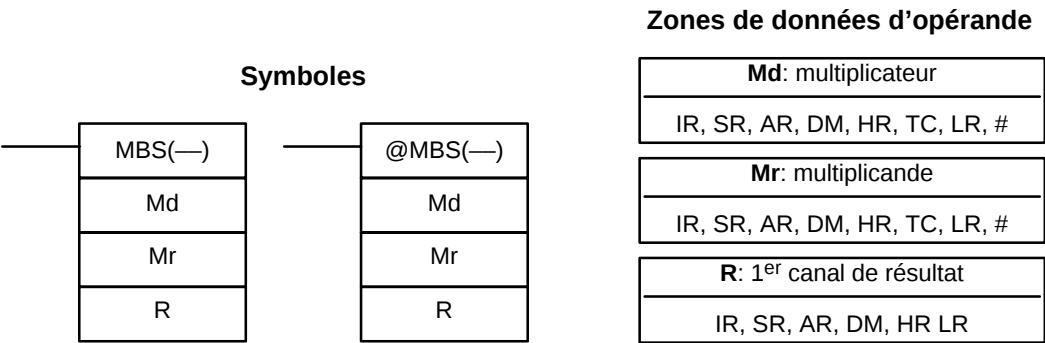
0 – – – UF (SR 25405)

1 – – – OF (SR 25404)

0 – – – CY (effacé avec CLC(41))

- Rem. : 1. Pour les données binaires sans signe, CY indique que le résultat est négatif. Prenez le complément de 2 avec NEGL(—) pour obtenir la valeur absolue du résultat réel (UF et OF peuvent être ignorés).
2. Pour les données portant un signe, le drapeau OF indique que le résultat dépasse + 2 147 483 647 (7FFF FFFF). CY peut être ignoré.

5-21-7 SIGNED BINARY MULTIPLY – MULTIPLICATION BINAIRE AVEC SIGNE – MBS(—)



Limites

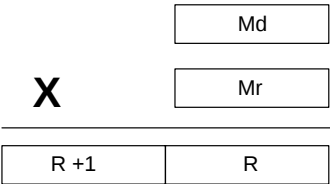
Cette instruction est disponible avec les UC du CQM1-CPU4 [-E.]

Description

Les DM 6143 à 6655 ne peuvent être utilisés pour R.

Description

MBS(—) multiplie le contenu binaire de deux canaux portant un signe et sort le résultat à 8 chiffres binaire et portant un signe dans R+1 et R. Les 4 chiffres d'extrême droite du résultat sont placés dans R et les 4 chiffres d'extrême gauche dans R+1.



- Drapeaux

ER:

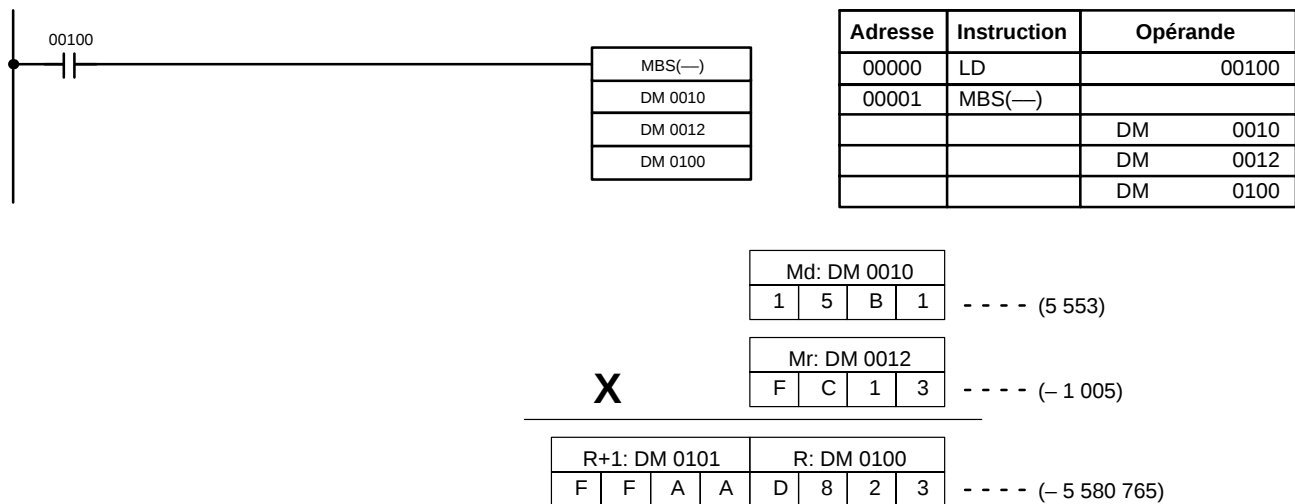
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ:

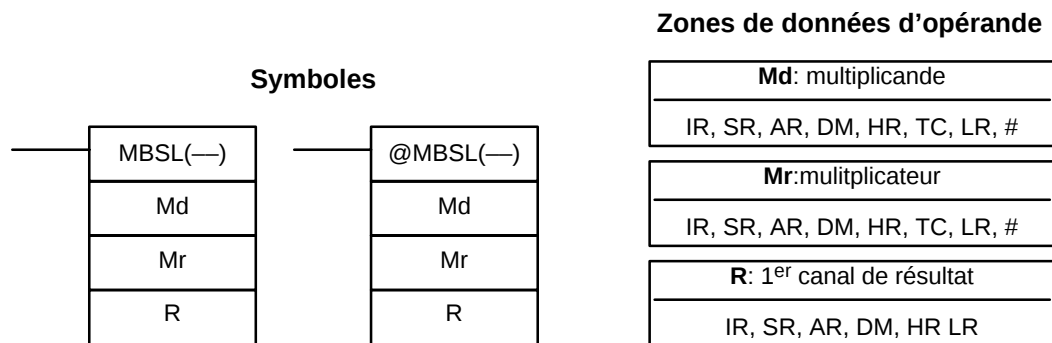
ON lorsque le résultat est égal à 0000 0000 ; reste à OFF dans les autres cas.

Exemple

Dans l'exemple suivant, MBS(—) est utilisée pour multiplier le contenu binaire de DM 0010 portant un signe par le contenu binaire de DM 0012 avec signe et sort le résultat dans DM 0100 et DM 0101.



5-21-8 DOUBLE SIGNED BINARY MULTIPLY– DOUBLE MULTIPLICATION BINAIRE AVEC SIGNE – MBSL(—)

**Limites**

Cette instruction est disponible avec les UC du **CQM1-CPU4** E.

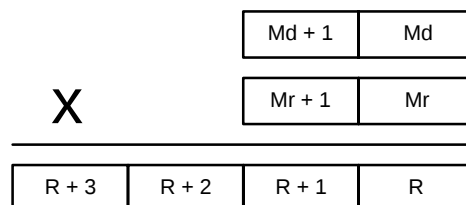
Md et Md+1 doivent appartenir à la même zone de données, ainsi que Mr et Mr+1.

R et R+3 doivent appartenir à la même zone de données.

Les DM 6143 à 6655 ne peuvent être utilisés pour R.

Description

MBSL(—) multiplie les données binaires à signe à 32 bits (8 chiffres) de Md+1 et Md par les données binaires à signe à 32 bits de Mr+1 et Mr, et sort le résultat binaire à signe à 16 chiffres dans R+3 à R.

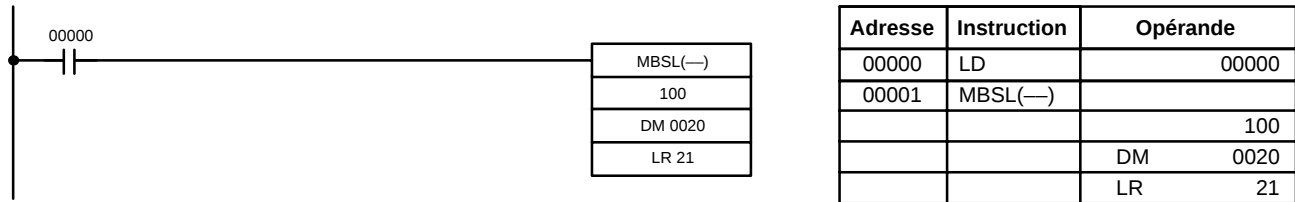
**Drapeaux**

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à 0 (contenu de R+3 à R = zéro) ; reste à OFF dans les autres cas.

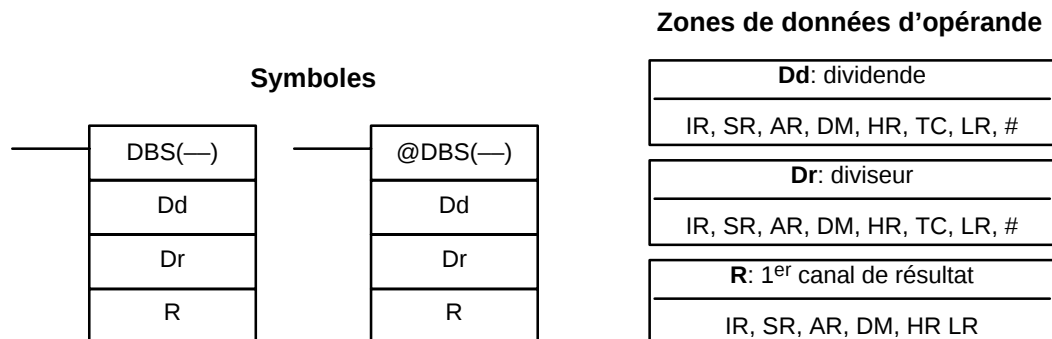
Exemple

Dans l'exemple suivant, MBSL(—) est utilisée pour multiplier le contenu binaire à signe de IR 101 et IR 100 par le contenu binaire à signe de DM 0021 et DM 0020 et sort le résultat dans LR 24 à LR 21.

**X**

Md+1: IR 101				Md: IR 100				
0	0	0	8	7	9	3	8	----- (555 320)
Mr+1: DM 0021				Mr: DM 0020				
F	F	F	0	A	8	1	2	----- (- 1 005 550)
R+3: LR 24				R+2: LR 23				
F	F	F	F	F	F	7	D	
R+1: LR 22				R: LR 21				
F	C	A	5	4	5	F	0	----- (- 55 840 206 000)

5-21-9 SIGNED BINARY DIVIDE – DIVISION BINAIRE AVEC SIGNE – DBS(—)

**Limites**

Cette instruction est disponible avec les UC du **CQM1-CPU4 [-E.]**

Les DM 6143 à 6655 ne peuvent être utilisés pour R.

Description

DBS(—) divise le contenu binaire à signe de Dd par le contenu binaire à signe de Dr et sort le résultat à 8 chiffres binaire à signe dans R+1 et R. Le quotient est placé dans R et le reste dans R+1.

**Drapeaux**

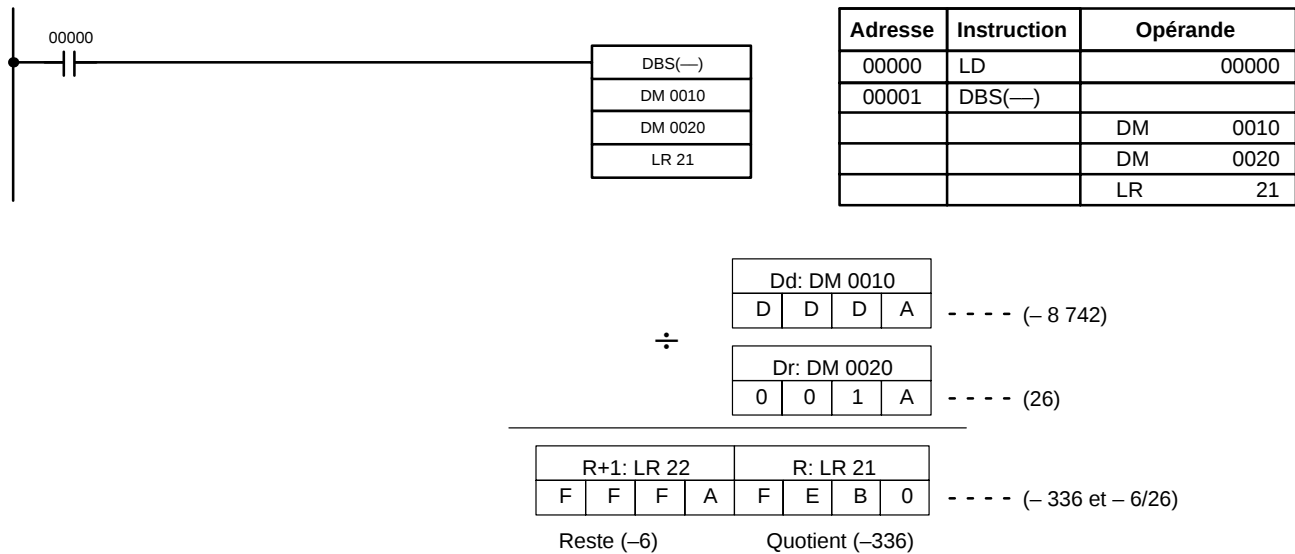
ER: Dr a 0 pour contenu.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

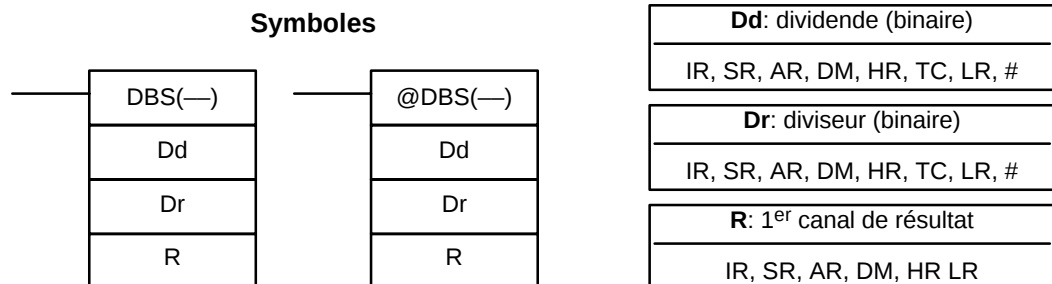
EQ: ON lorsque le contenu de R (quotient) est égal à 0000, reste à OFF dans les autres cas.

Exemple

Dans l'exemple suivant, DBS(—) est utilisée pour diviser le contenu binaire à signe de DM 0010 par le contenu binaire à signe de DM 0020 et pour sortir le résultat dans LR 21 et LR 22.



5-21-10 DOUBLE SIGNED BINARY DIVIDE – DOUBLE DIVISION BINAIRE AVEC SIGNE – DBSL(—)

Zones de données d'opérande**Limites**

Cette instruction est disponible avec les UC du **CQM1-CPU4 [-E.]**

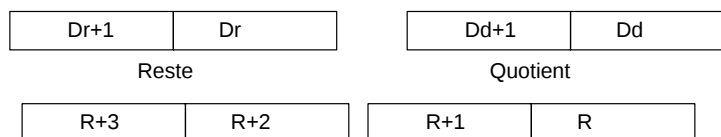
Dd et Dd+1 doivent appartenir à la même zone de données, ainsi que Dr et Dr+1.

R et R+3 doivent appartenir à la même zone.

Les DM 6143 à 6655 ne peuvent être utilisés pour R.

Description

DBS(—) divise les données binaires à 32 bits (8 chiffres) binaires à signe de Dd+1 et Dd par les données à 32 bits binaires à signe de Dr+1 et Dr et sort le résultat à 16 chiffres binaire à signe dans R+3 à R. Le quotient est placé dans R+1 et R et le reste dans R+3 et R+2.

**Drapeaux**

ER: Dr+1 et Dr ont 0 pour contenu.

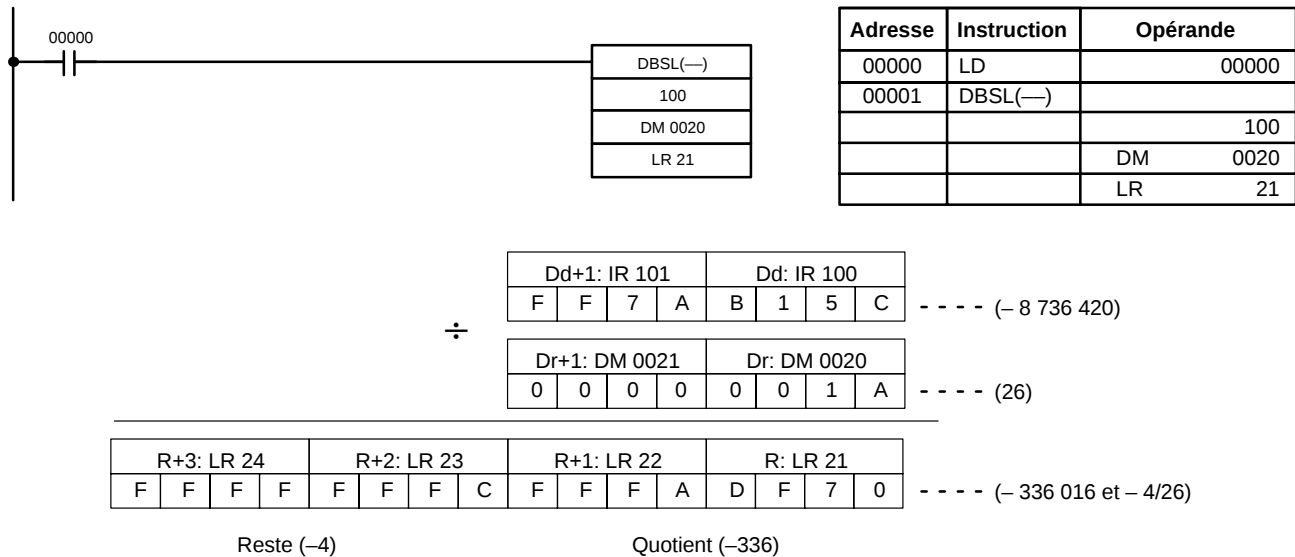
Le canal DM indirectement adressé n'existe pas (le contenu du canal

*DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le contenu de R+1 et R (quotient) est égal à 0, reste à OFF dans les autres cas.

Example

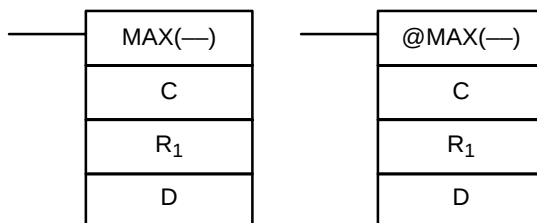
Dans l'exemple suivant, DBSL(—) est utilisée pour diviser le contenu binaire à signe de IR 101 et IR 100 par le contenu binaire à signe de DM 0021 et DM 0020 et sort le résultat dans LR 24 à LR 21.



5-22 Instructions mathématiques spéciales

5-22-1 FIND MAXIMUM – RECHERCHE DE LA VALEUR MAXIMUM DANS UNE ZONE DEFINIE – MAX(--)

Symboles



Zones de données d'opérande

C: données de contrôle
IR, SR, AR, DM, HR, TC, LR, #
R₁: 1 ^{er} canal de gamme
IR, SR, AR, DM, HR, TC, LR
D: canal de destination
IR, SR, AR, DM, HR, LR

Limits

N doit être en BCD entre 0001 et 9999.

R_1 et R_1+N-1 doivent appartenir à la même zone de données.

Les DM 6144 à 6655 ne peuvent être utilisés pour D.

Description

Lorsque la condition d'exécution est à OFF, MAX(—) n'est pas exécutée. Lorsque la condition est à ON, MAX(—) recherche dans la gamme de mémoire de R_1 à R_1+N-1 l'adresse contenant la valeur maximum et sort cette dernière sur le canal de (D).

Si le bit 15 de C est à ON, MAX(—) identifie l'adresse du canal contenant la valeur maximum de D+1. L'adresse porte une identification différente pour la zone DM :

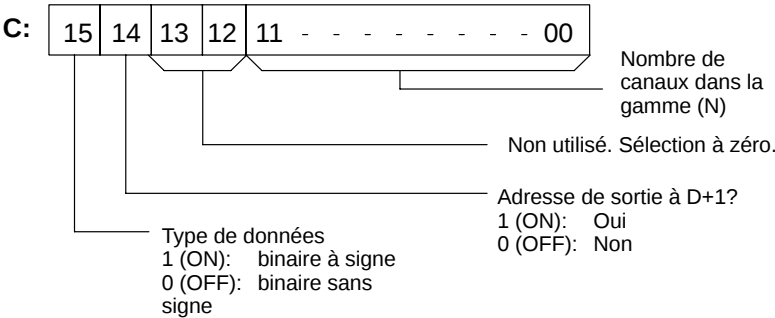
- 1, 2, 3...** 1. Pour une adresse de la zone DM, l'adresse du canal est écrite dans C+1. Par exemple, si l'adresse contenant la valeur maximum est DM 0114, alors #0114 est écrit dans D+1.

2. Pour une adresse dans une autre zone, le nombre d'adresses à partir du début de la recherche est écrit dans D+1. Par ex., si l'adresse contenant la valeur maximum est IR 114 et que le premier canal de la gamme de recherche est IR 014, alors #0100 est écrit dans D+1.

Si le bit 14 de C est à ON et que plusieurs adresses contiennent la même valeur maximum, la position de la plus basse des adresses est sortie dans D+1. La position est sortie comme l'adresse DM de la zone DM, mais en tant que position absolue relative au premier canal de gamme pour toutes les autres zones.

Le nombre de canaux de la gamme (N) est dans les trois chiffres d'extrême droite de C, qui doivent être en BCD entre 001 et 999.

Lorsque le bit 15 de C est à OFF, les données comprises dans la gamme sont traitées comme données binaires sans signe et comme données binaires avec signe lorsque le bit est à ON.



Attention : si le bit 14 de C est à ON, les valeurs supérieures à #8000 sont traitées comme des nombres négatifs et les résultats diffèrent selon le type de donnée spécifiée. Veuillez à spécifier le bon type de donnée.

Drapeaux

- ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
R₁ et R₁+N-1 n'appartiennent pas à la même zone de données.
- EQ: ON lorsque la valeur maximum est égale à #0000.

5-22-2 FIND MINIMUM – RECHERCHE DE LA VALEUR MINIMUM DANS UNE ZONE DEFINIE – MIN(--)

Symboles

MIN(—)

C

R₁

D

@MIN(—)

C

R₁

D

Zones de données d'opérande

C: donnée de contrôle

IR, SR, AR, DM, HR, TC, LR, #

R₁: 1^{er} canal de la gamme

IR, SR, AR, DM, HR, TC, LR

D: canal de destination

IR, SR, AR, DM, HR, LR

Limites

- N doit être en BCD entre 0001 et 9999.
- R₁ et R₁+N-1 doivent appartenir à la même zone de données.
- Les DM 6144 à 6655 ne peuvent être utilisés pour D.

Description

Lorsque la condition d'exécution est à OFF, MIN(—) n'est pas exécutée. Lorsque la condition est à ON, MIN(—) recherche dans la gamme de mémoire de R₁

à R_1+N-1 l'adresse contenant la valeur minimum et sort cette dernière sur le canal de (D).

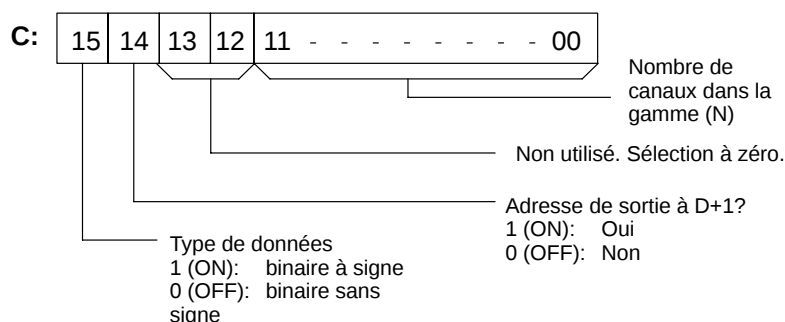
Si le bit 15 de C est à ON, MIN(—) identifie l'adresse du canal contenant la valeur minimum de D+1. L'adresse porte une identification différente pour la zone DM :

- 1, 2, 3...**
1. Pour une adresse de la zone DM, l'adresse du canal est écrite dans C+1. Par exemple, si l'adresse contenant la valeur minimum est DM 0114, alors #0114 est écrit dans D+1.
 2. Pour une adresse dans une autre zone, le nombre d'adresses à partir du début de la recherche est écrit dans D+1. Par ex., si l'adresse contenant la valeur minimum est IR 114 et que le premier canal de la gamme de recherche est IR 014, alors #0100 est écrit dans D+1.

Si le bit 14 de C est à ON et que plusieurs adresses contiennent la même valeur minimum, la position de la plus basse des adresses est sortie dans D+1. La position est sortie comme l'adresse DM de la zone DM, mais en tant que position absolue relative au premier canal de gamme pour toutes les autres zones.

Le nombre de canaux de la gamme (N) est dans les trois chiffres d'extrême droite de C, qui doivent être en BCD entre 001 et 999.

Lorsque le bit 15 de C est à OFF, les données comprises dans la gamme sont traitées comme données binaires sans signe et comme données binaires avec signe lorsque le bit est à ON.



Attention : si le bit 14 de C est à ON, les valeurs supérieures à #8000 sont traitées comme des nombres négatifs et les résultats diffèrent selon le type de donnée spécifiée. Veuillez à spécifier le bon type de donnée.

Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

R_1 et R_1+N-1 n'appartiennent pas à la même zone de données.

EQ: ON lorsque la valeur maximum est égale à #0000.

5-22-3 AVERAGE VALUE – VALEUR MOYENNE – AVG(—)

Symboles

—	AVG(—)	—	@AVG(—)
	S		S
	N		N
	D		D

Zones de données d'opérande

S: canal source
IR, SR, AR, DM, HR, TC, LR
N: nombre de cycles
IR, SR, AR, DM, HR, TC, LR, #
D: 1 ^{er} canal de destination
IR, SR, AR, DM, HR, LR

Limites

S doit être en hexadécimal.

N doit être en BCD de #0001 à #0064.

D et D+N+1 doivent appartenir à la même zone de données.

Les DM 6144 à 6655 ne peuvent être utilisés pour S, N, ni pour D à D+N+1.

Description

AVG(—) est utilisée pour calculer la valeur moyenne de S sur N cycles.

Lorsque la condition est à OFF, AVG(—) n'est pas exécutée.

A chaque exécution de AVG(—), le contenu de S est stocké dans les canaux D+2 à D+N+1. A la première exécution, AVG(—) écrit le contenu de S dans D+2; à la deuxième exécution, elle écrit le contenu de S dans D+3, etc. A la N^e exécution, AVG(—) écrit le contenu de S dans D+N+1, calcule la valeur moyenne des valeurs stockées dans D+2 à D+N+1 et l'inscrit dans D.

Cette instruction peut être utilisée par la carte d'entrée analogique CQM1-AD.

Fonction des canaux D à D+N+1 :

D	Valeur moyenne (après N exécutions min.)
D+1	Utilisé par le système.
D+2	Contenu de S à partir de la 1 ^e exéc. de AVG(—)
D+3	Contenu de S à partir de la 2 ^e exéc. de AVG(—)
⋮	⋮
D+N+1	Contenu de S à partir de la N ^e exéc. de AVG(—)

Précautions

La valeur moyenne est calculée en binaire. Veillez à ce que le contenu de S soit binaire.

N doit être en BCD de #0001 à #0064. Si le contenu de N $\geq$ #0065, AVG(—) fonctionne avec N=64.

La valeur moyenne est arrondie au nombre entier le plus proche (0,5 est arrondi à 1).

Laissez le contenu de D+1 à #0000 après la 1^e exécution d'AVG(—).

Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

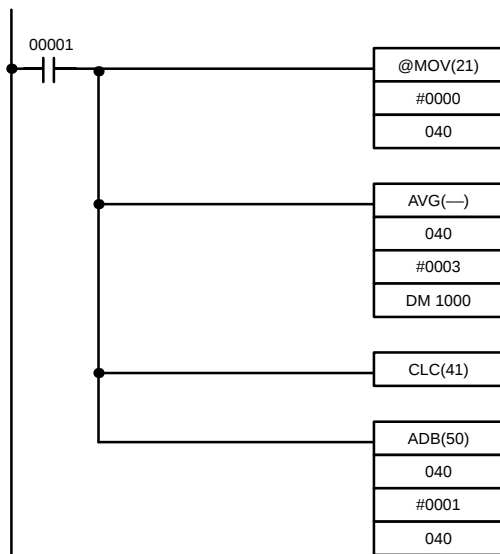
Une ou plusieurs opérandes ont été incorrectement sélectionnées.

D et D+N+1 n'appartiennent pas à la même zone de données.

Exemple

Dans l'exemple suivant, le contenu de IR 040 est à #0000, puis incrémenté de 1 à chaque tour de scrutation. Pour les deux premiers tours, AVG(—) déplace le contenu de IR 040 vers DM 1002 et DM 1003. A partir du 3^e tour, AVG(—) cal-

cule la valeur moyenne du contenu de DM 1002 à DM 1004 et écrit cette valeur dans DM 1000.



Adresse	Instruction	Opérande
00000	LD	00001
00001	@MOV(21)	
		# 0000
		040
00002	AVG(—)	
		040
		# 0003
		DM 1000
00003	CLC(41)	
00004	ADB(50)	
		040
		# 0001
		040

	1 ^{er} tour	2 ^e tour	3 ^e tour	4 ^e tour
IR 040	0000	0001	0002	0003

	1 ^{er} tour	2 ^e tour	3 ^e tour	4 ^e tour	
DM 1000	0000	0001	0001	0002	Moyenne Utilisé par le système Précédentes valeurs de IR 40
DM 1001					
DM 1002	0000	0000	0000	0003	
DM 1003	---	0001	0001	0001	
DM 1004	---	---	0002	0002	

5-22-4 SUM – SOMME– SUM(--)

Symboles

SUM(—)
C
R ₁
D

@SUM(—)
C
R ₁
D

Zones de données d'opérande

C: donnée de contrôle
IR, SR, AR, DM, HR, LR, #
R₁: 1 ^{er} canal de la gamme
IR, SR, AR, DM, HR, TC, LR
D: 1 ^{er} canal de destination
IR, SR, AR, DM, HR, LR

Limites

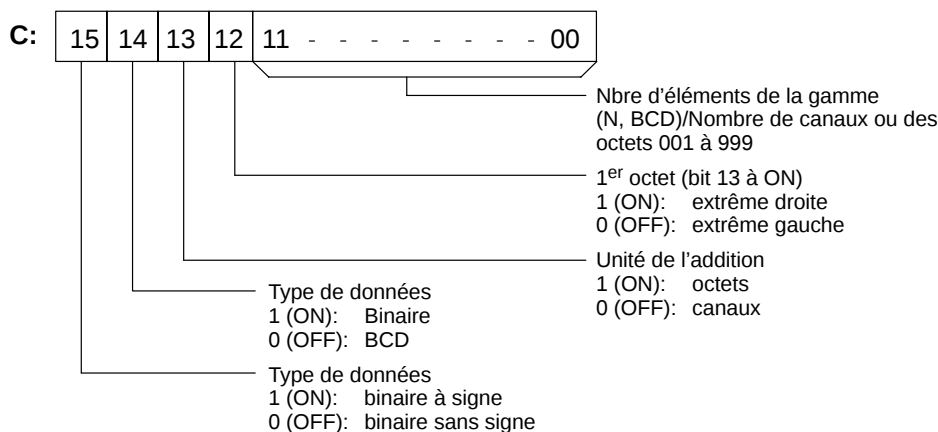
Les trois chiffres d'extrême droite de C doivent être en BCD entre 001 et 999.
Les DM 6143 à 6655 ne peuvent être utilisés pour D.

Si le bit 14 de C est à OFF (sélection des données BCD), toutes les données de la gamme R_1 à R_1+N-1 doivent être en BCD.

Description

Lorsque la condition d'exécution est à OFF, SUM(—) n'est pas exécutée. Lorsque la condition est à ON, SUM(—) ajoute le contenu des canaux R_1 à R_1+N-1 ou les octets des canaux R_1 à $R_1+N/2-1$ et sort cette valeur dans les canaux de destination (D et D+1). Les données peuvent être additionnées en binaire ou BCD et sorties sous cette même forme. Les données binaires peuvent porter ou non un signe.

Fonction des bits de C :

**Nombre d'éléments de la gamme**

Ce nombre N est contenu dans les 3 chiffres d'extrême droite de C, qui doit être en BCD entre 001 et 999. Ce nombre indique le nombre de canaux ou d'octets selon l'unité à additionner.

Unité de l'addition

Des canaux sont additionnés si le bit 13 est à OFF et ce sont des octets si le bit 13 est à ON.

Si les octets sont spécifiés, la gamme peut commencer par l'octet d'extrême droite ou d'extrême gauche de R_1 . L'octet d'extrême gauche de R_1 n'est pas ajouté si le bit 12 est à ON.

	MSB	LSB
R_1	1	2
R_1+1	3	4
R_1+2	5	6
R_1+3	7	8
⋮	⋮	⋮
⋮	⋮	⋮

Des octets sont ajoutés dans cet ordre si le bit 12 est à OFF: 1+2+3+4....

Des octets sont ajoutés dans cet ordre si le bit 12 est à ON: 2+3+4....

Type de données

Les données de la gamme sont traitées comme binaires sans signe lorsque le bit 14 de C est à ON et que le bit 15 est à OFF et comme binaires à signe lorsque les bits 14 et 15 sont à ON.

Les données de la gamme sont traitées comme BCD lorsque le bit 14 de C est à OFF, quel que soit l'état du bit 15.

Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

R_1 et R_1+N-1 n'appartiennent pas à la même zone de données.

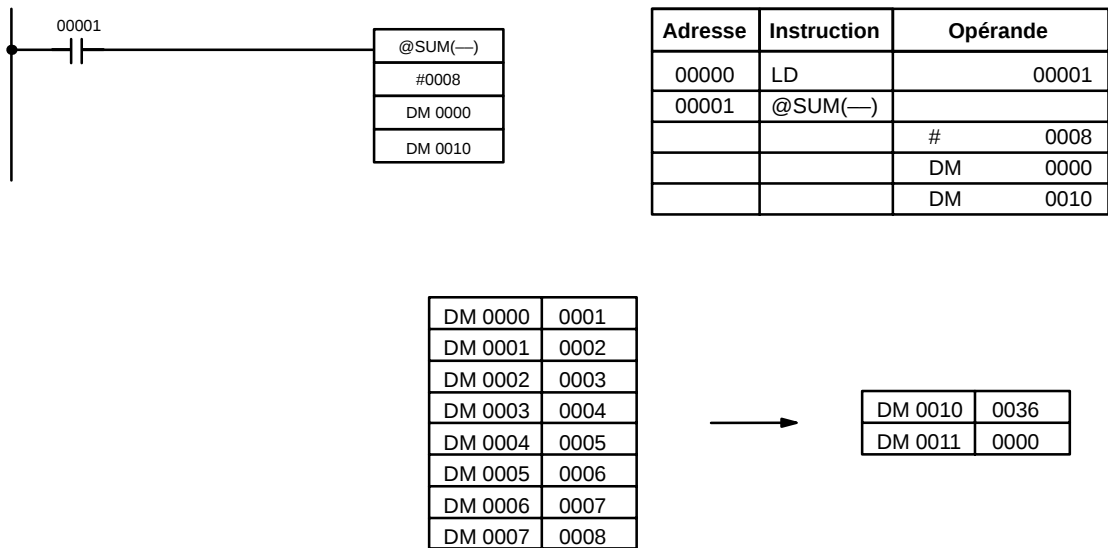
Le nombre d'éléments de C n'est pas en BCD entre 001 et 999.

Les données additionnées ne sont pas en BCD alors que BCD a été désigné.

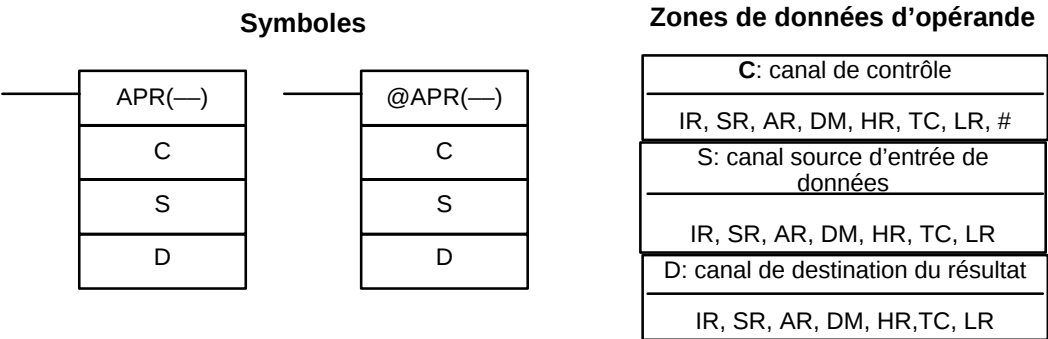
EQ: ON lorsque le résultat est égal à 0.

Exemple

Dans l'exemple suivant, les contenus BCD des 8 canaux de DM 0000 à 0007 sont ajoutés lorsque IR 00001 est à ON et le résultat est écrit dans DM 0010 et DM 0011.



5-22-5 ARITHMETIC PROCESS – FONCTION TRIGONOMETRIQUE – APR(—)



tDrapeaux

- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
Pour les fonctions trigonométriques, $x > 0900$ (x = contenu de S).
Une constante autre que #0000 ou #0001 a été désignée pour C.
La donnée d'approximation linéaire n'est pas lisible.
- EQ:** Le résultat est égal à 0000.

Exemples

Fonction sinus

Voici un exemple d'utilisation de la fonction sinus d'APR(—) pour calculer le sinus de 30°. La fonction sinus est spécifiée lorsque C = #0000.



Adresse	Instruction	Opérande
00000	LD	00000
00001	APR(—)	
		# 0000
		DM 0000
		DM 0100

Donnée d'entrée, x

S: DM 0000			
0	10 ¹	10 ⁰	10 ⁻¹
0	3	0	0

La donnée d'entrée de doit pas dépasser #0900 en BCD.

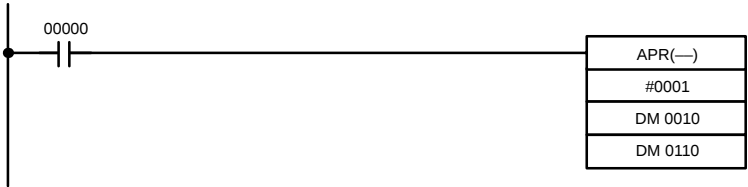
Donnée de résultat

D: DM 0100			
10 ⁻¹	10 ⁻²	10 ⁻³	10 ⁻⁴
5	0	0	0

La donnée de résultat a 4 chiffres significatifs, le 5^e et les suivants sont ignorés. Résultat de sin(90) = 0,9999, et non 1.

Fonction cosinus

L'exemple suivantTmontre l'utilisation de la fonction cosinus de APR(—) pour calculer le cosinus de 30°. La fonction cosinus est spécifiée si C = #0001.



Adresse	Instruction	Opérande
00000	LD	00000
00001	APR(—)	
		# 0001
		DM 0010
		DM 0110

Donnée d'entrée, x

S: DM 0010			
0	10 ¹	10 ⁰	10 ⁻¹
0	3	0	0

La donnée d'entrée de doit pas dépasser #0900 en BCD.

Donnée de résultat

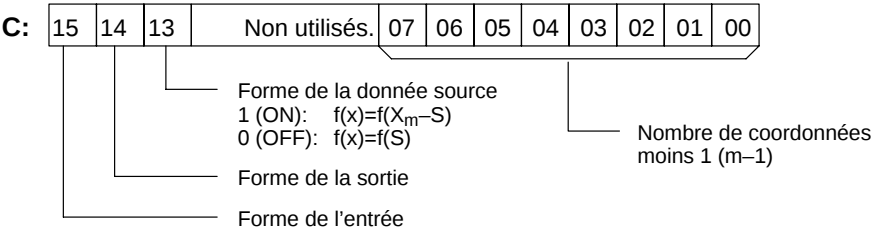
D: DM 0110			
10 ⁻¹	10 ⁻²	10 ⁻³	10 ⁻⁴
8	6	6	0

La donnée de résultat a 4 chiffres significatifs, le 5^e et les suivants sont ignorés. Résultat de cos(0) = 0,9999, et non 1.

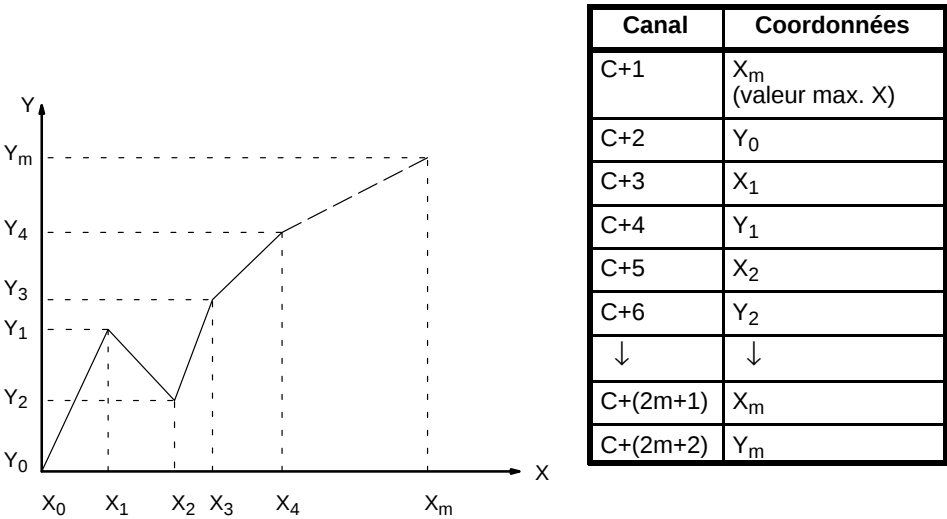
Approximation linéaire

L'approximation linéaire d'APR(—) est spécifiée lorsque C = adresse de mémoire. Le canal C est le 1^{er} du bloc mémoire continu contenant les données d'approximation linéaire.

Le contenu de C spécifie le nombre de segments de ligne de l'approximation, et le forme BCD ou BIN de l'entrée et de la sortie. Les bits 00 à 07 contiennent le nombre de segments de ligne moins 1, m-1, en donnée binaire. Les bits 14 et 15 déterminent respectivement la forme de l'entrée et de la sortie : 0 spécifie la forme BCD et 1 spécifie la forme BIN.

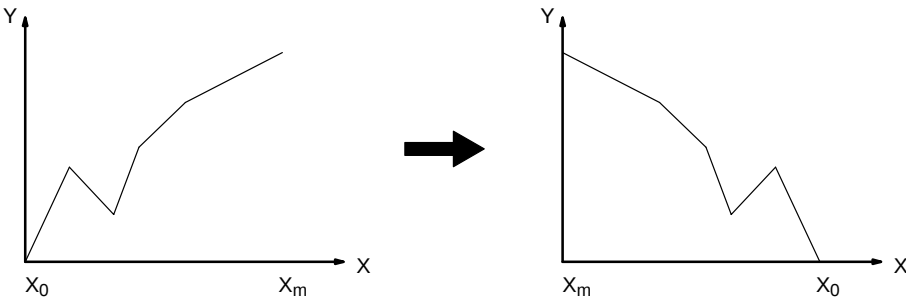


Entrez les coordonnées des points finaux $m+1$ définissant les segments de ligne m comme sur le tableau suivant. Entrez toutes les coordonnées sous forme BIN. Entrez toujours les coordonnées à partir de la valeur X la plus basse (X_1) jusqu'à la valeur (X_m) la plus haute. $X_0 = 0000$ (il n'est pas nécessaire de l'entrer).



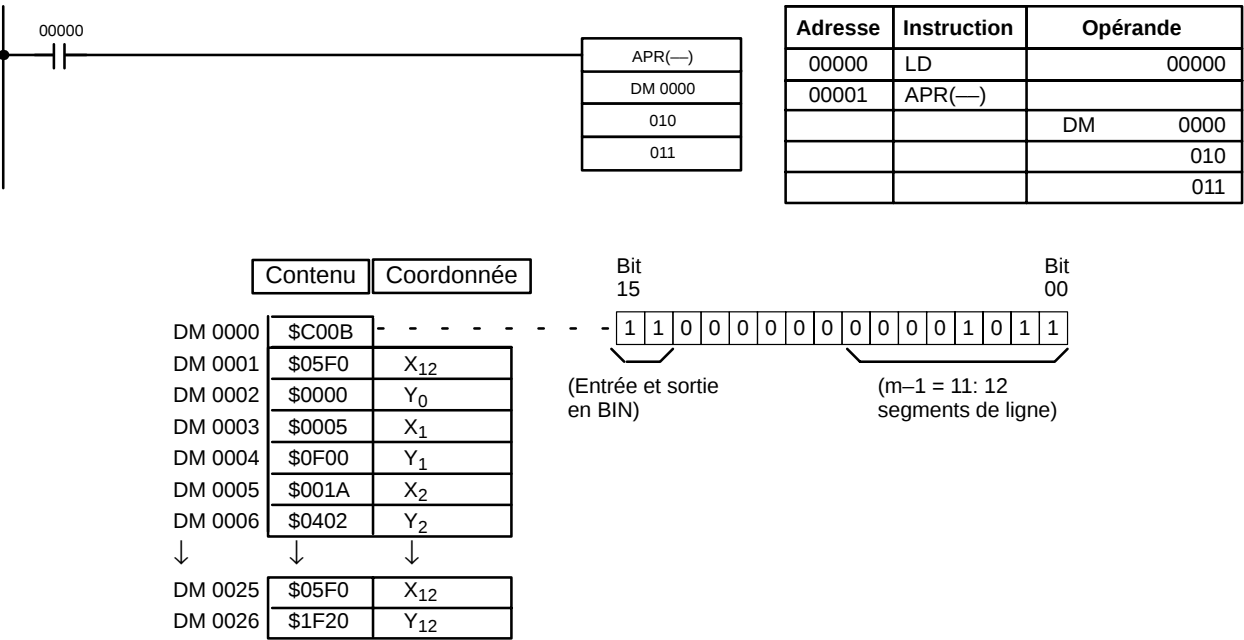
En dehors des valeurs définies, c'est-à-dire au-delà de x^n , la fonction APR(—) n'est pas exécutée. La dernière valeur de sortie est comparée et le bit d'erreur passe à 1.

Si le bit 13 de C est à 1, le graphique doit être réfléchi en miroir de gauche à droite, comme sur le schéma suivant :

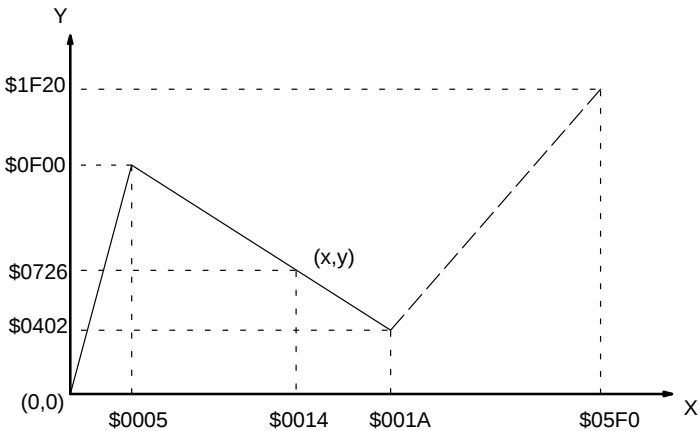


L'exemple suivant montre la construction d'une approximation linéaire avec 12 segments de ligne. Le bloc de données est continu, comme il se doit, de

DM 0000 à DM 0026 (C à C + (2 × 12 + 2)). La donnée d'entrée est prise dans IR 010 et le résultat est sorti dans IR 011.

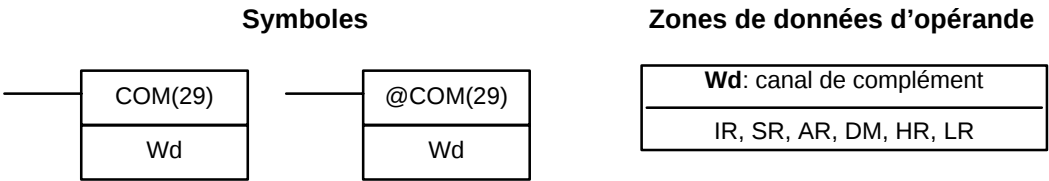


Dans ce cas, le canal de donnée d'entrée, IR 010, contient #0014 et f(0014) = #0726 est sorti dans R, IR 011.



5-23 Instructions logiques

5-23-1 COMPLEMENT – COM(29)



Limites

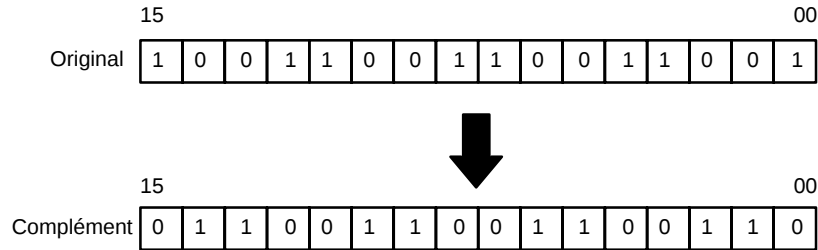
Complémente en une donnée de 16 bits, c'est-à-dire efface tous les bits à ON d'un canal et les met à OFF (ou inversement).
Les DM 6144 à 6655 ne peuvent être utilisés pour Wd.

Description

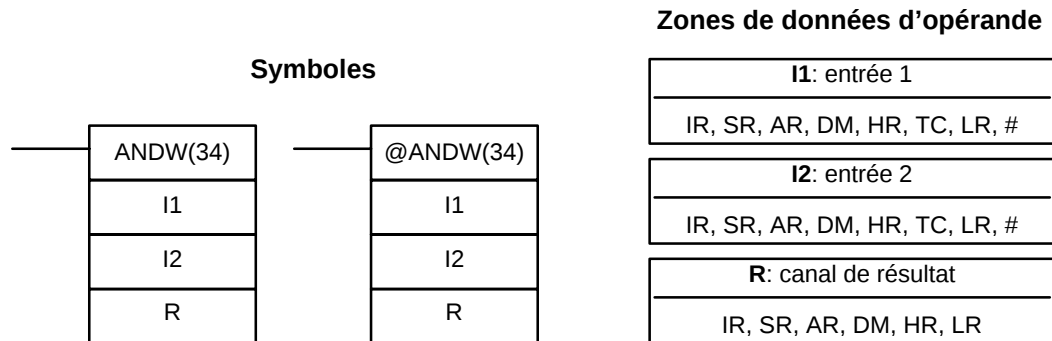
Lorsque la condition d'exécution est à OFF, COM(29) n'est pas exécutée. Lorsque la condition est à ON, COM(29) efface tous les bits à ON et active tous les bits à OFF de Wd.

Précautions

tLe complément de Wd est calculé à chaque tour de scrutation si la forme simple de COM(29) est utilisée. Utilisez la forme sur changement de front (@COM(29)) ou combinez COM(29) avec DIFU(13) ou DIFD(14) pour calculer le complément une seule fois.

Exemple**Drapeaux**

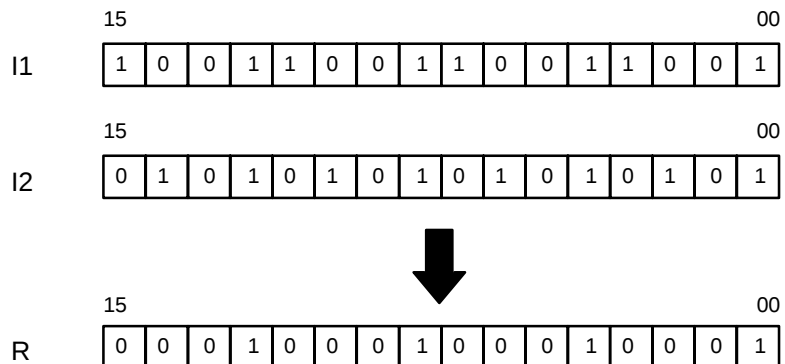
- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- EQ:** ON lorsque le résultat est égal à 0.

5-23-2 LOGICAL AND – MOT ET – ANDW(34)**Limites**

Effectue une opération ET logique entre 2 données à 16 bits.
Les DM 6144 à 6655 ne peuvent être utilisés pour R.

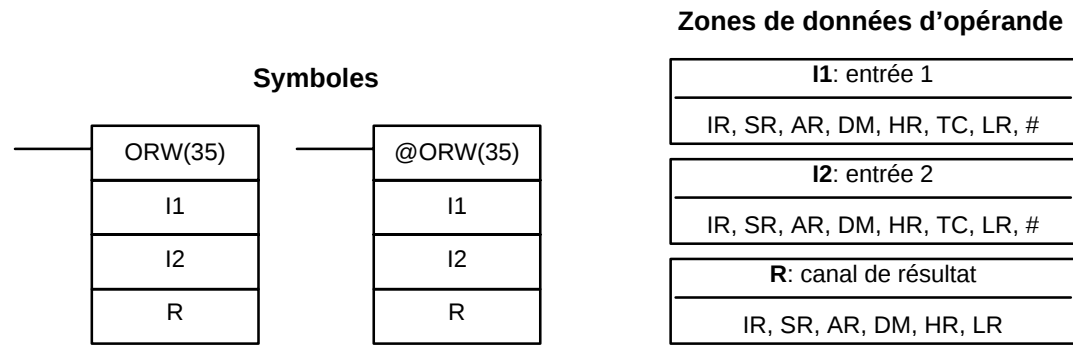
Description

Lorsque la condition d'exécution est à OFF, ANDW(34) n'est pas exécutée.
Lorsque la condition est à ON, ANDW(34) effectue un ET logique entre le contenu de I1 et I2 bit par bit et place le résultat dans R.

Exemple**Drapeaux**

- ER:** Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- EQ:** ON lorsque le résultat est égal à 0.

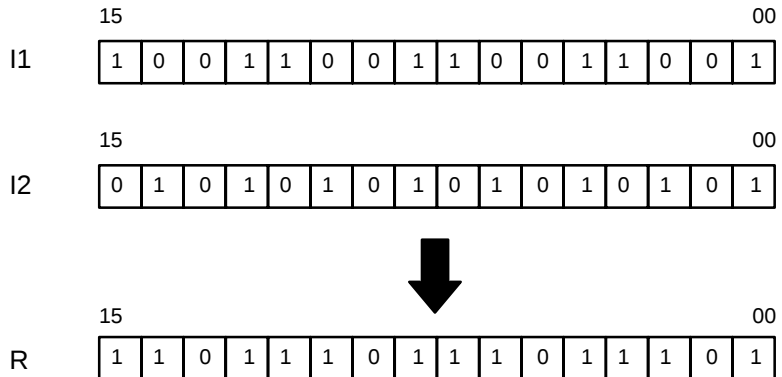
5-23-3 LOGICAL OR – MOT OU – ORW(35)

**Limites**

Effectue une opération OU logique entre deux données à 16 bits.
Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

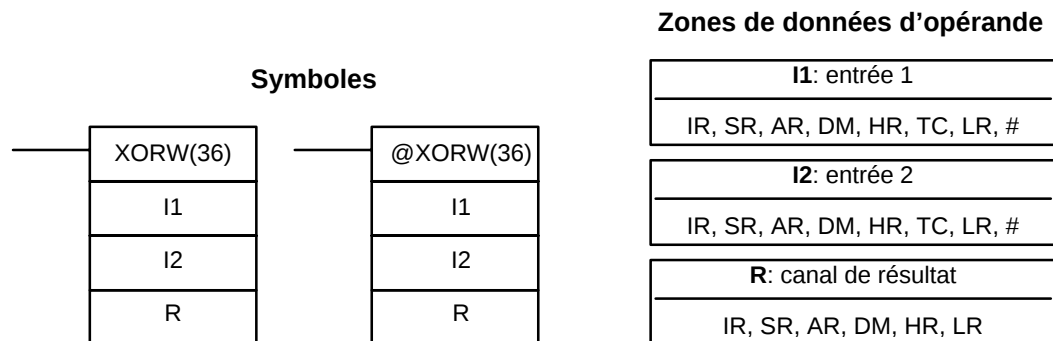
Lorsque la condition d'exécution est à OFF, ORW(35) n'est pas exécutée. Lorsque la condition est à ON, ORW(35) effectue un OU logique entre le contenu de I1 et I2 bit par bit et place le résultat dans R.

Exemple**Drapeaux**

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à 0.

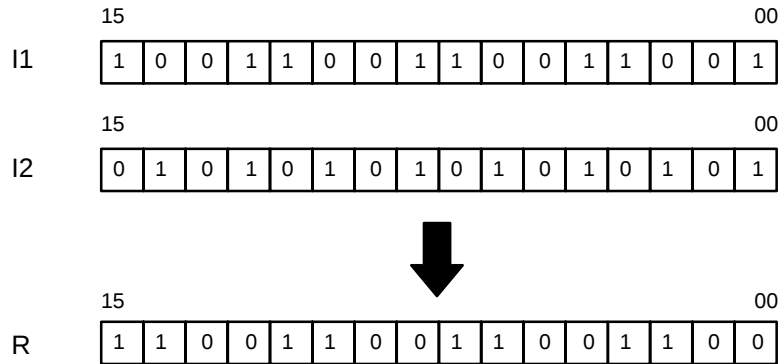
5-23-4 EXCLUSIVE OR– MOT OU EXCLUSIF – XORW(36)

**Limites**

Effectue une opération OU exclusif entre 2 données à 16 bits.
Les DM 6144 à 6655 ne peuvent être utilisés pour R.

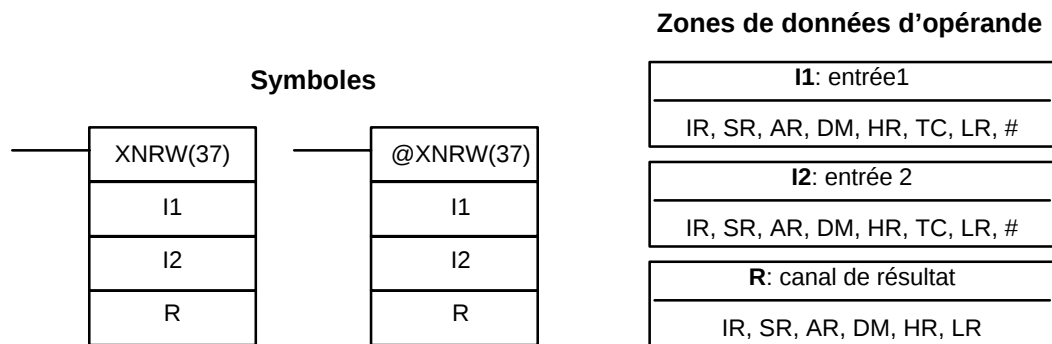
Description

Lorsque la condition d'exécution est à OFF, XORW(36) n'est pas exécutée. Lorsque la condition est à ON, XORW(36) effectue un OR exclusif entre le contenu de I1 et I2 bit par bit et place le résultat dans R.

Exemple**Drapeaux**

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

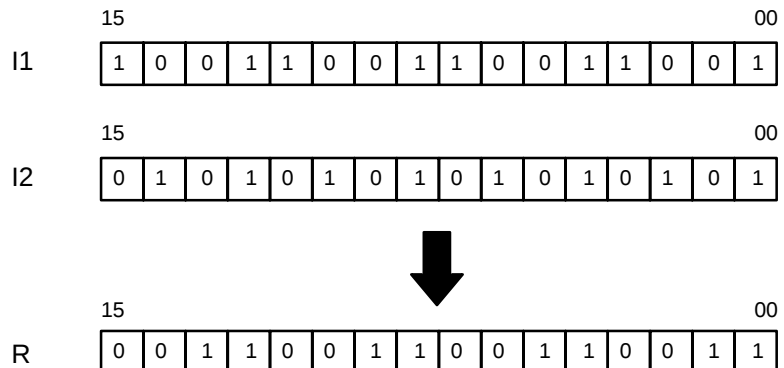
EQ: ON lorsque le résultat est égal à 0.

5-23-5 EXCLUSIVE NOR – MOT OU NON EXCLUSIF – XNRW(37)**Limites**

Effectue une opération OU exclusif complémentée entre 2 données à 16 bits. Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, XNRW(37) n'est pas exécutée. Lorsque la condition est à ON, XNRW(37) effectue un NOR exclusif entre le contenu de I1 et I2 bit par bit et place le résultat dans R.

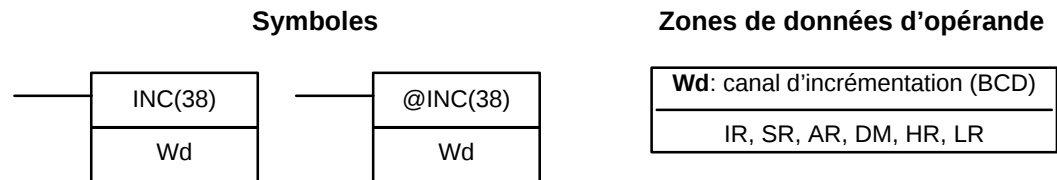
**Drapeaux**

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à 0.

5-24 Instructions d'incrémentation/décrémentation

5-24-1 BCD INCREMENT – INCREMENTATION BCD – INC(38)


Limites

Incrémente de 1 une donnée BCD à 4 chiffres sans la retenue.
 Les DM 6144 à 6655 ne peuvent être utilisés pour Wd.

Description

Lorsque la condition d'exécution est à OFF, INC(38) n'est pas exécutée. Lorsque la condition est à ON, INC(38) incrémente Wd sans affecter Carry (CY).

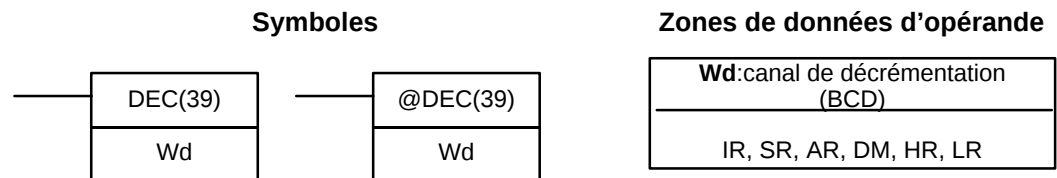
Précautions

Le contenu de Wd est incrémenté à chaque tour de scrutation si la forme simple de INC(38) est utilisée. Utilisez la forme sur changement de front (@INC(38)) ou combinez INC(38) avec DIFU(13) ou DIFD(14) pour incrémenter Wd une seule fois.

Drapeaux

ER: Wd n'est pas en BCD
 Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
EQ: ON lorsque le résultat incrémenté est égal à 0.

5-24-2 BCD DECREMENT – DECREMENTATION BCD – DEC(39)


Limites

Décrémente de 1 une donnée BCD à 4 chiffres sans la retenue.
 Les DM 6144 à 6655 ne peuvent être utilisés pour Wd.

Description

Lorsque la condition d'exécution est à OFF, DEC(39) n'est pas exécutée. Lorsque la condition est à ON, DEC(39) décrémente Wd sans affecter Carry (CY). DEC(39) fonctionne de la même manière que INC(38), mais elle décrémente les valeurs au lieu de les incrémenter.

Précautions

Le contenu de Wd est décrémenté à chaque tour de scrutation si la forme simple de DEC(39) est utilisée. Utilisez la forme sur changement de front (@DEC(39)) ou combinez DEC(39) avec DIFU(13) ou DIFD(14) pour décrémenter Wd une seule fois.

Drapeaux

ER: Wd n'est pas en BCD
 Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
EQ: ON lorsque le résultat incrémenté est égal à 0.

5-25 Instructions de sous-programme

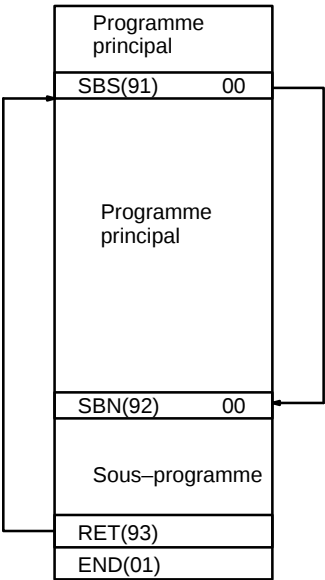
Les sous-programmes permettent de diviser les tâches de contrôle importantes en tâches plus réduites et de ré-utiliser un ensemble d'instructions donné. Lorsque le programme principal appelle un sous-programme, le pointeur est transféré vers ce sous-programme et les instructions du sous-programme sont exécutées. Les instructions d'un sous-programme sont écrites de la même façon que les codes de programmes principaux. Lorsque toutes les instructions d'un sous-programme ont été exécutées, le pointeur retourne au programme principal au point qui suit l'endroit à partir duquel le sous-programme a été appelé (sauf spécification contraire).

5-25-1 SUBROUTINE ENTER – APPEL DE SOUS-PROGRAMME – SBS(91)

Symbole	Zones de données d'opérande		
SBS(91) N	<table><tr><td>N: n° de sous-programme</td></tr><tr><td>000 à 255</td></tr></table>	N: n° de sous-programme	000 à 255
N: n° de sous-programme			
000 à 255			

Limites Appelle le sous-programme spécifié. Lorsqu'une séquence d'exécutions est effectuée plusieurs fois, il est préférable de créer un sous-programme. Le CQM1-CPU11/21-E n'utilise que les numéros de sous-programmes 000 à 127.

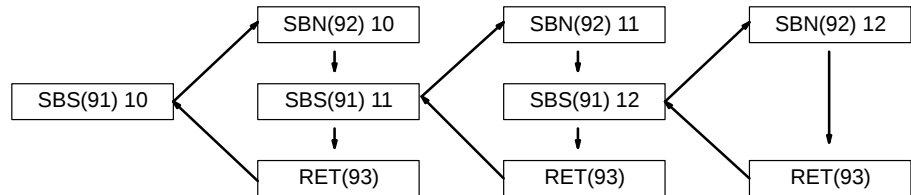
Description On peut exécuter un sous-programme en plaçant SBS(91) dans le programme principal au point où l'on désire placer le sous-programme. Le numéro de sous-programme utilisé dans SBS(91) indique le sous-programme désiré. Lorsque SBS(91) est exécutée (lorsque sa condition d'exécution est à ON), les instructions placées entre SBN(92) portant le même numéro de sous-programme et la première instruction RET(93) qui suit sont exécutées avant que l'exécution ne retourne à l'instruction suivant SBS(91) à l'origine de l'appel.



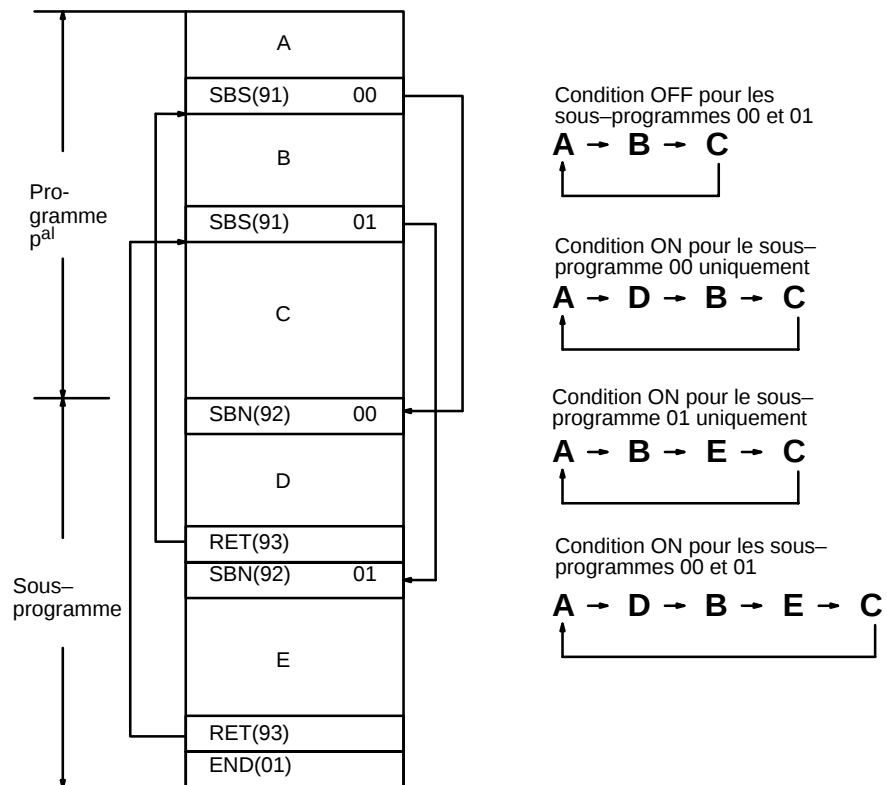
SBS(91) peut être utilisée autant de fois qu'on le désire dans le programme, c'est-à-dire qu'on peut appeler de différents endroits du programme le même sous-programme.

SBS(91) peut également être placée dans un sous-programme pour décaler l'exécution du programme d'un sous-programme à l'autre (imbrication de

sous-programmes). A la fin du deuxième sous-programme (c'est-à-dire au niveau de RET(93)), l'exécution du programme retourne au sous-programme original qui est alors terminé avant de retourner au programme principal. On peut faire des imbrications jusqu'à 16 niveaux différents. Un sous-programme ne peut s'appeler lui-même (par ex., SBS(91) 00 ne peut être programmée dans le sous-programme défini à l'aide de SBN(92) 00). Le schéma suivant illustre deux niveaux d'imbrication :



Exécution du programme pour les conditions d'exécution de deux SBS(91).

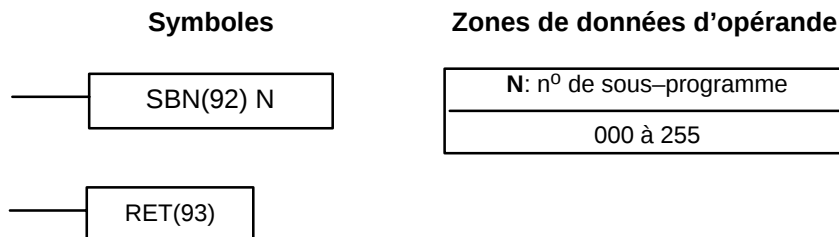


Drapeaux

ER: Pas de sous-programme pour le numéro de sous-programme spécifié.
 Un sous-programme s'appelle.
 Un sous-programme actif a été appelé.

Attention : SBS(91) n'est pas exécutée et le sous-programme n'est pas appelé lorsque ER est à ON.

5-25-2 SUBROUTINE DEFINE and RETURN – DEBUT DE SOUS-PROGRAMME et FIN DE SOUS-PROGRAMME – SBN(92)/RET(93)



Limites	Indique le début et la fin du sous-programme, ainsi que le retour à la première ligne de sous-programme après appel (SBS). Le CQM1-CPU11/21-E n'utilise que les numéros de sous-programmes 000 à 127. Chaque sous-programme ne peut être utilisé qu'une seule fois dans SBN(92).
Description	SBN(92) s'utilise pour marquer le début d'un sous-programme ; RET(93) s'utilise pour en marquer la fin. Chaque sous-programme est identifié à l'aide d'un numéro de sous-programme, N, programmé comme donnée d'opérande de SBN(92). Ce même numéro de sous-programme s'utilise dans toute instruction SBS(91) appelant le sous-programme (cf chapitre 5-25-1 SBS(91)). Aucun numéro de sous-programme n'est nécessaire avec RET(93). Tous les sous-programmes doivent être programmés à la fin du programme principal. Lors de la programmation d'un ou de plusieurs sous-programmes, le programme principal est exécuté jusqu'à la première instruction SBN(92) avant de retourner à l'adresse 00000 pour le tour de scrutation suivant. Les sous-programmes ne sont pas exécutés s'ils ne sont pas appelés par SBS(91). END(01) doit être placée à la fin du dernier sous-programme, c'est-à-dire après la dernière instruction RET(93). Elle n'est pas nécessaire en d'autres endroits du programme.
Précautions	Si SBN(92) est placée à un mauvais endroit du programme, elle en inhibe l'exécution au-delà de ce point : l'exécution retourne au début du programme lorsque SBN(92) est rencontrée. Si DIFU(13) ou DIFU(14) sont placées dans un sous-programme, le bit d'opérande n'est pas placé à OFF jusqu'à la prochaine exécution du sous-programme, c'est-à-dire que le bit d'opérande peut rester à ON plus longtemps qu'un cycle.
Drapeaux	Aucun drapeau n'est directement affecté par ces instructions.

5-26 Instructions spéciales

5-26-1 TRACE MEMORY SAMPLING – CHRONOGRAMME DYNAMIQUE – TRSM(45)

Attention : cette instruction n'est pas disponible avec le CQM1-CPU11-E/21-E.

La création d'un chronogramme peut être utilisée pour faciliter la mise au point des machines en fonctionnement. Le lancement et l'utilisation du chronogramme sont obligatoirement effectués par Sysmate LSS ; aucune recherche de données n'est possible à partir de la console de programmation. Cette recherche est décrite en détail dans le manuel LSS. Ce chapitre montre le symbole de TRSM(45) et donne un programme en exemple :

Symbole



Description

TRSM(45) permet la lecture d'un morceau de programme ou la recherche de pannes aléatoires en fonctionnement dynamique. On peut désigner jusqu'à 12 bits et 3 canaux pour la lecture dynamique (cf manuel LSS).

TRSM(45) n'est pas contrôlé par les conditions d'exécution mais par deux bits de la zone AR : AR 2515 et AR 2514. AR 2515 est le bit de départ d'échantillonnage du chronogramme ; il est placé à ON pour lancer l'échantillonnage servant à la recherche. Le bit de départ de l'échantillonnage ne doit être placé à ON qu'à partir d'un appareil périphérique et non à partir du programme. AR 2514 est le bit de départ de la recherche. Lorsqu'il est sélectionné, la donnée spécifiée est enregistrée dans la mémoire du chronogramme. Le bit de départ de la recherche peut être sélectionné à partir du programme ou d'un appareil de programmation. Un retard positif ou négatif peut être sélectionné pour modifier le point réel de départ de la recherche.

Les données peuvent être enregistrées de 3 façons. TRSM(45) peut être placée à un ou plusieurs endroits du programme pour indiquer où la donnée spécifiée doit être recherchée. Si l'on n'utilise pas TRSM(45), les données spécifiées sont recherchées lorsque END(01) est exécutée. La troisième méthode implique la sélection d'un intervalle de temps à partir de l'appareil périphérique de façon à ce que la donnée spécifiée soit recherchée à un intervalle régulier indépendant du temps de cycle (cf manuel LSS).

TRSM(45) peut être placée n'importe où dans un programme et autant de fois qu'on le désire. Les données du chronogramme peuvent alors être surveillées par une console de programmation, un ordinateur central, etc.

Bits de contrôle d'AR et drapeaux

Les bits et drapeaux de contrôle suivants sont utilisés pendant la recherche de données. Le drapeau de recherche est à ON pendant les opérations de recherche. Le drapeau de fin de recherche est à ON lorsqu'assez de données ont été retrouvées pour remplir la mémoire du chronogramme.

Drapeau	Fonction
AR 2515	Bit de départ d'échantillonnage*
AR 2514	Bit de départ de recherche
AR 2513	Drapeau de recherche
AR 2512	Drapeau de recherche terminée

Rem. : * ne changez pas l'état de AR 2515 à partir du programme.

Précautions

Si TRSM(45) est exécutée, elle ne peut avoir lieu dans un bloc JMP(08) – JME(09) lorsque la condition de saut est à OFF.

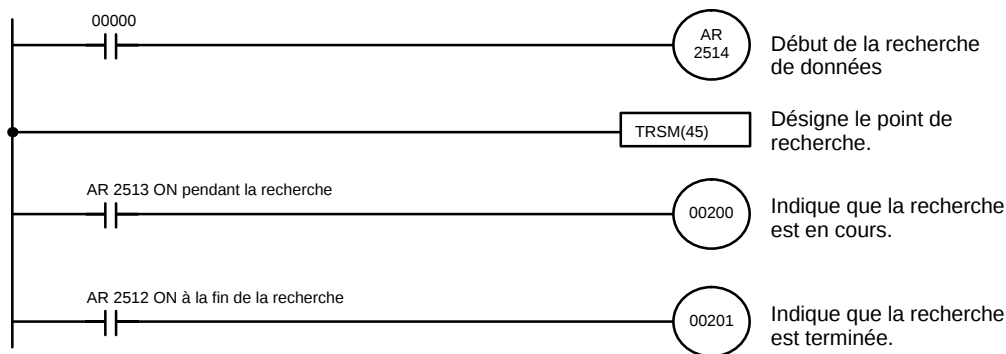
Exemple

L'exemple suivant montre le programme de base et de fonctionnement pour la recherche de données. Forcez le bit de départ d'échantillonnage (AR 2515) à l'état ON pour le départ de l'échantillonnage ; ce bit ne doit pas être placé à ON à partir du programme. La donnée est lue et stockée dans le chronogramme.

Lorsque IR 00000 est à ON, le bit de départ de la recherche (AR 2514) est également placé à ON et l'UC recherche le retard pour l'enregistrer dans le chronogramme. Cela peut signifier que certains des échantillonnages déjà réalisés

seront enregistrés comme mémoire trace (retard négatif) ou que certains échantillonnages seront effectués avant d'être enregistrés (retard positif).

Les données échantillonnées sont écrites dans le chronogramme en sautant au début de la zone mémoire lorsque la fin en a été atteinte et en continuant jusqu'à la marque de départ. Cela peut signifier que les données précédemment enregistrées (celles de l'échantillonnage précédant la marque de départ) sont effacées au profit d'autres données (surtout si le retard est positif). Un retard négatif ne peut être tel que la donnée désirée est exécutée avant le départ de l'échantillonnage.

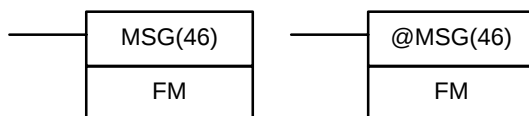


Adresse	Instruction	Opérande
00000	LD	0000
00001	OUT	AR 2514
00002	TRSM(45)	
00003	LD	AR 2513

Adresse	Instruction	Opérande
00004	OUT	00200
00005	LD	AR 2512
00006	OUT	00201

5-26-2 MESSAGE DISPLAY – MESSAGE – MSG(46)

Symboles



Zones de données d'opérande

FM: 1 ^{er} canal de message
IR, SR, AR, DM, HR, LR

Limites

L'instruction MSG lit les données ASCII de huit canaux (16 caractères) et affiche le message sur la console de poche.

Les DM 6649 à 6655 ne peuvent être utilisés pour FM.

Description

Exécutée avec une condition à ON, MSG(46) lit huit canaux de code ASCII étendu de FM à FM+7 et affiche le message sur la console de programmation. Le message affiché peut comporter jusqu'à 16 caractères, c'est-à-dire que chaque code de caractère ASCII demande huit bits (2 chiffres) (cf annexe K pour les codes ASCII).

Si vous n'avez pas besoin des 8 canaux pour le message, ils peuvent être arrêtés à n'importe quel endroit en entrant "OD." Lorsque l'on rencontre OD dans un message, plus aucun canal n'est lu et les canaux qui auraient pu être utilisés pour le message peuvent l'être à d'autres fins.

Message stockés en tampon et priorité

On peut stocker jusqu'à 3 messages dans la mémoire tampon : une fois stockés, ils apparaissent en premier s'ils ont été stockés en dernier. Plus de 3 MSG(46) pouvant être exécutés dans le même tour de scrutation, il existe un ordre de

priorité fondé sur la zone de stockage des messages pour la sélection des messages à stocker en tampon.

Ordre de priorité de l'affichage des messages :

LR > IR > HR > AR > TC > DM

Dans une même zone, ceux ayant l'adresse la plus basse ont priorité.

Pour les messages indirectement adressés (*DM), ceux ayant les adresses DM finales les plus basses ont priorité.

Effacement des messages

Pour effacer un message, exécutez FAL(06) 00 ou effacez-le par la console de programmation ou le logiciel Sysmate LSS.

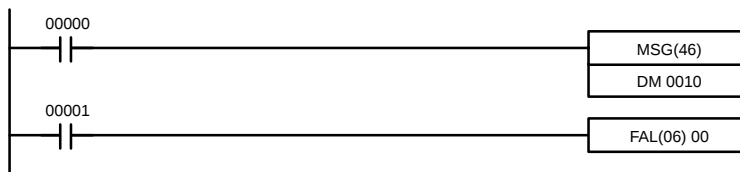
Si les données du message changent pendant l'affichage, l'affichage est également modifié.

Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Exemple

L'exemple suivant montre le message produit par l'instruction et les données lorsque 00000 est à ON. Si 00001 passe à ON, un message est effacé.



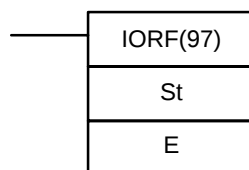
Adresse	Instruction	Opérande
00000	LD	00000
00001	MSG(46)	
		DM 0010
00002	LD	00001
00003	FAL(06)	00

Contenu DM					Equivalent ASCII	
DM 0010	4	1	4	2	A	B
DM 0011	4	3	4	4	C	D
DM 0012	4	5	4	6	E	F
DM 0013	4	7	4	8	G	H
DM 0014	4	9	4	A	I	J
DM 0015	4	B	4	C	K	L
DM 0016	4	D	4	E	M	N
DM 0017	4	F	5	0	O	P

MSG
ABCDEFGHIJKLMN OP

5-26-3 I/O REFRESH – RAFFRAICHISSEMENT DES E/S – IORF(97)

Symbole



Zones de données d'opérande

St: canal de départ
IR 000 to IR 115
E: canal de fin
IR 000 to IR 115

Limites

Raffraîchit les données situées entre les canaux de début et de fin spécifiés. St doit être inférieur ou égal à E.

Description

Pour raffraîchir les E/S, indiquez simplement les premier (St) et dernier (E) canaux d'E/S à raffraîchir. Lorsque la condition d'exécution de IORF(97) est à

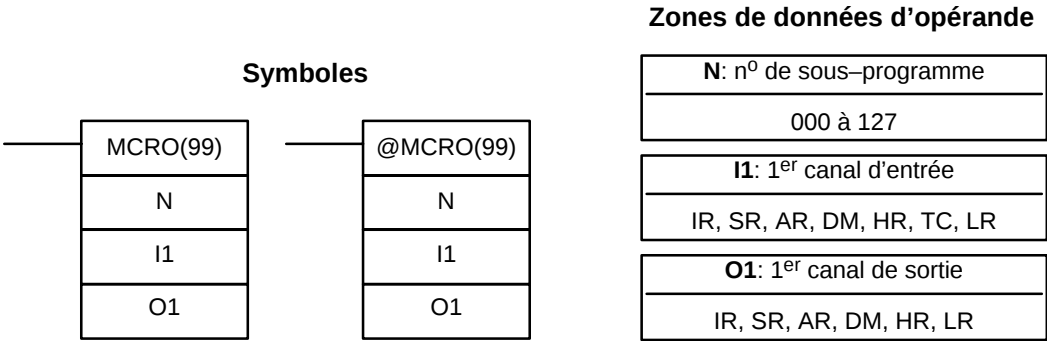
ON, tous les canaux entre St et E sont rafraîchis, en plus du rafraîchissement d'E/S normal effectué pendant le tour de scrutation de l'UC.

Rem. : cette instruction est sans effet sur les canaux qui ne sont pas utilisés pour les E/S.

Drapeaux

Aucun drapeau n'est affecté par cette instruction.

5-26-4 MACRO – MCRO(99)



Limites

Les DM 6144 à 6655 ne peuvent être utilisés pour O1.

Description

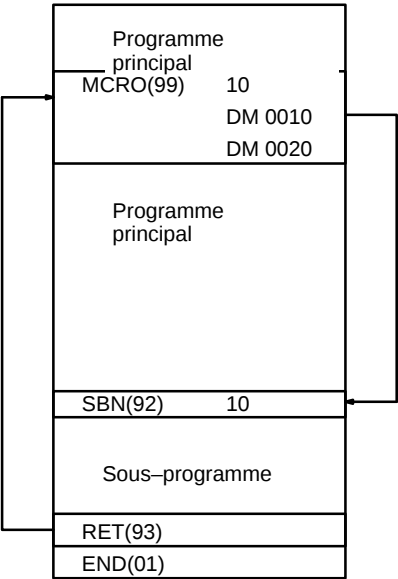
L'instruction MACRO permet de remplacer plusieurs sous-programmes par un seul ayant une structure identique mais des opérandes différentes. Il y a 4 canaux d'entrée, IR 096 à IR099 et 4 canaux de sortie, IR 196 à IR 199, attribués à MCRO(99). Ces 8 canaux sont utilisés dans le sous-programme et prennent leur contenu entre I1 et I1+3 et entre O1 et O1+3 lors de l'exécution du sous-programme.

Lorsque la condition d'exécution est à OFF, MCRO(99) n'est pas exécutée. Lorsqu'elle est à ON, MCRO(99) copie le contenu de I1 à I1+3 dans IR 096 à IR 099, et copie le contenu de O1 à O1+3 dans IR 196 à IR 199, puis appelle et exécute le sous-programme spécifié dans N. Lorsque le sous-programme est terminé, le contenu de IR 196 à IR 199 est retransféré vers O1 à O1+3 avant la fin de MCRO(99).

Exemple

Dans cet exemple, le contenu de DM 0010 à DM 0013 est copié dans IR 096 à IR 099, le contenu de DM 0020 à DM 0023 est copié dans IR 196 à IR 199 et le

sous-programme 10 est appelé et exécuté. Lorsque le sous-programme est terminé, le contenu de IR 196 à IR 199 est recopié dans DM 0020 à DM 0023.



Drapeaux

- ER:** Il n'y a pas de sous-programme attribué au numéro spécifié.
Une opérande a dépassé la limite de la zone de données.
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
Un sous-programme s'est appelé lui-même.
Un sous-programme actif a été appelé.

5-26-5 BIT COUNTER – COMPTEUR DE BITS – BCNT(--)

Symboles

BCNT(—)	@BCNT(—)
N	N
SB	SB
R	R

Zones de données d'opérande

N: Nombre de canaux (BCD)
IR, SR, AR, DM, HR, TC, LR, #
SB: canal source de début
IR, SR, AR, DM, HR, TC, LR
R: canal de destination
IR, SR, AR, DM, HR, TC, LR

Exemple

BCNT(—)
0003
001
DM0010

	15		00	
CH001	0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0			Nombre de "1"
				= 5
CH002	1 0 0 0 0 1 1 1 0 1 1 0 0 1 0 1			+
				= 8
CH003	1 1 0 0 1 0 1 1 1 0 1 0 1 0 0 1			+
				= 9
DM0010	0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0			=
	0 0 2 2			

Limites

Compte le nombre total de bits à ON dans un ou plusieurs canaux spécifiés. Ils peuvent être utilisés comme vérification de somme de données.
N ne peut être égal à 0.

Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description

Lorsque la condition d'exécution est à OFF, BCNT(—) n'est pas exécutée. Lorsque la condition est à ON, BCNT(—) compte le nombre total de bits à ON dans tous les canaux entre SB et SB+(N-1) et place le résultat dans R.

Drapeaux

ER: N n'est pas en BCD, ou N est égal à 0 ; SB et SB+(N-1) n'appartiennent pas à la même zone de données.

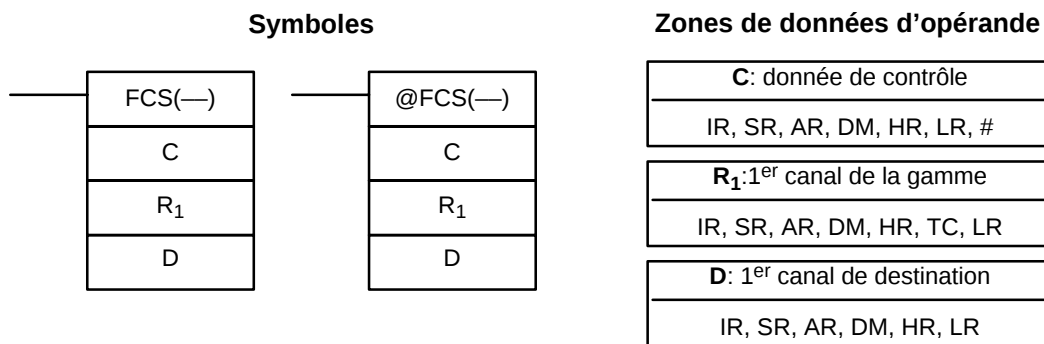
La valeur de comptage résultante dépasse 9999.

N est égal à 0 ou n'est pas en BCD.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

EQ: ON lorsque le résultat est égal à 0.

5-26-6 FRAME CHECKSUM – CALCUL DE CONTROLE DE TRAME – FCS(—)



Limites

Les 3 chiffres d'extrême droite de C doivent être en BCD entre 001 et 999.

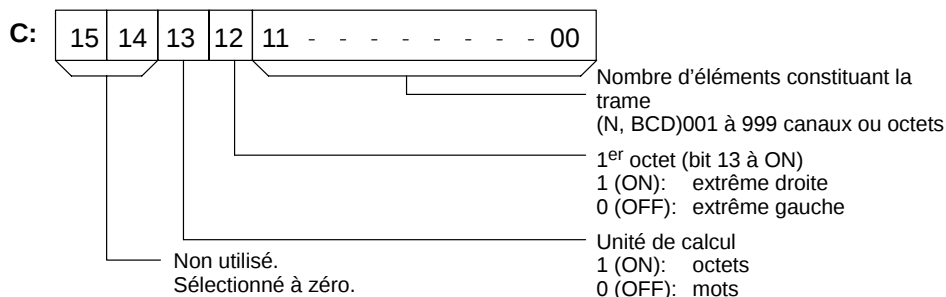
Les DM 6143 à 6655 ne peuvent être utilisés pour D.

Description

FCS(—) peut être utilisée pour vérifier si des erreurs apparaissent lors de transferts de données par les ports de communication.

Lorsque la condition d'exécution est à OFF, FCS(—) n'est pas exécutée. Lorsqu'elle est à ON, FCS(—) calcule la somme de vérification de trame de la gamme spécifiée en effectuant un OR exclusif entre le contenu des canaux R₁ à R₁+N-1 ou entre les octets des canaux R₁ à R₁+N-1. La valeur de la somme de vérification de trame (hexadécimale) est alors convertie en ASCII et sortie sur les canaux de destination (D and D+1).

La fonction des bits de C est illustrée et expliquée ci-dessous :



Nombre d'éléments de la trame

Le nombre d'éléments de la trame (N) est contenu dans les 3 chiffres d'extrême droite de C, qui doit être en BCD entre 001 et 999.

Unités de calcul

Le calcul de la somme de vérification est effectué sur mot si le bit 13 est à OFF et la somme de vérification des octets si le bit 13 est à ON.

Si l'on choisit le calcul sur octet, la trame peut commencer par l'octet d'extrême droite ou d'extrême gauche de R₁. L'octet d'extrême gauche de R₁ n'est pas inclus si le bit 12 est à ON.

	MSB	LSB
R ₁	1	2
R ₁ +1	3	4
R ₁ +2	5	6
R ₁ +3	7	8
⋮	⋮	⋮

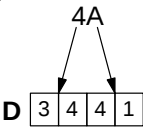
Lorsque le bit 12 est à OFF, les octets sont liés par un OR logique dans cet ordre : 1, 2, 3, 4,

Lorsque le bit 12 est à ON, les octets sont liés par un OR logique dans cet ordre : 2, 3, 4, 5,

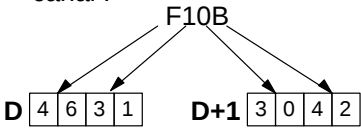
Conversion en ASCII

Le calcul de la somme de vérification sur octet donne une valeur hexadécimale à deux chiffres qui est convertie en son équivalent ASCII à 4 chiffres. Le calcul sur mot donne une valeur hexadécimale à quatre chiffres qui est convertie en son équivalent ASCII à 8 chiffres, comme le montre l'exemple ci-dessous :

Somme de vérification sur octet :



Somme de vérification sur canal :



Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Le nombre d'éléments n'est pas compris entre 001 et 999 en BCD.

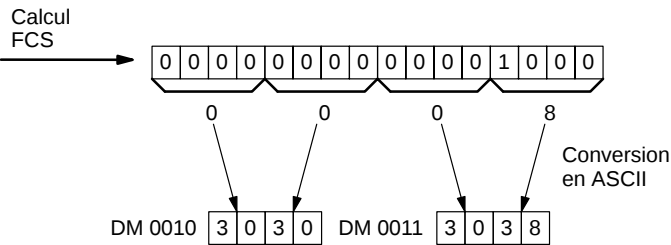
Exemple

Lorsque IR 00000 est à ON dans l'exemple suivant, la somme de vérification (0008) est calculée pour les 8 mots de DM 0000 à DM 0007 et l'équivalent ASCII (30 30 30 38) est écrit dans les DM 0010 et 0011.



Adresse	Instruction	Opérande
00000	LD	00000
00001	@FCS(—)	
		# 0008
		DM 0000
		DM 0010

DM 0000	0001
DM 0001	0002
DM 0002	0003
DM 0003	0004
DM 0004	0005
DM 0005	0006
DM 0006	0007
DM 0007	0008



5-26-7 FAILURE POINT DETECTION – DETECTION D'UN POINT DE DEFAUT – FPD(--)

Symboles	Zones de données d'opérande
FPD(--) C T D	C: donnée de contrôle # T: temps de surveillance (BCD) IR, SR, AR, DM, HR, TC. LR, # D: 1^{er} canal d'enregistrement IR, SR, AR, DM, HR, LR

Limites

D et D+8 doivent appartenir à la même zone de données lorsque le bit 15 de C est à ON.

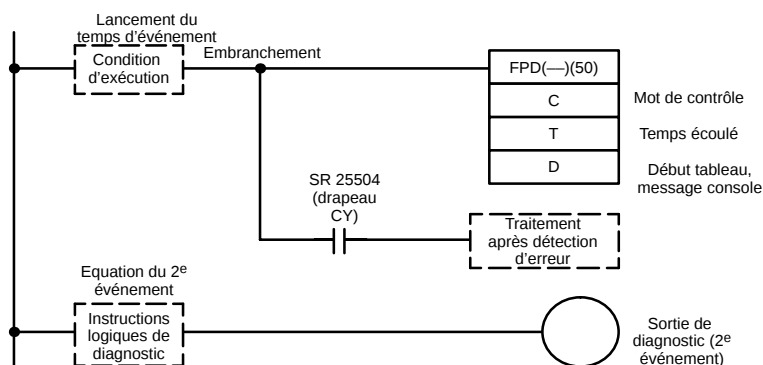
Les DM 6144 à 6655 ne peuvent être utilisés pour T ou D.

C doit être entré comme constante.

Description

FPD(--) peut être utilisée dans le programme autant de fois qu'on le désire, mais chacune des instructions doit avoir une valeur D différente. Cette instruction permet de vérifier si le temps écoulé entre deux événements a été dépassé et de diagnostiquer dans l'équation du second événement l'élément qui manque. Ce diagnostic peut être sorti sur la console de programmation par l'affichage en clair de l'adresse défectueuse d'un message complémentaire de 8 caractères et d'un numéro de FAL(06).

Les parties de programmes marquées de tirets dans le schéma suivant peuvent être écrites selon les besoins d'applications particuliers du programme. La partie de programme lancée par CY est optionnelle et peut utiliser n'importe quelle instruction excepté LD et LD NOT. Les instructions logiques de diagnostic et la condition d'exécution peuvent consister en une combinaison de conditions NF ou NO quelconque.



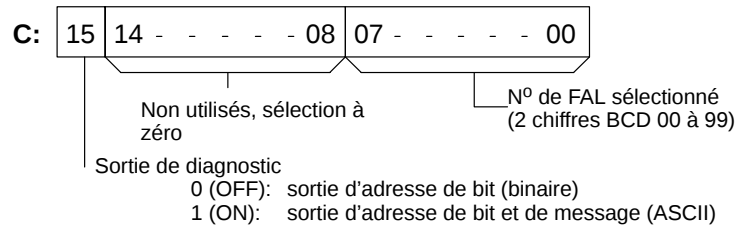
Lorsque la condition d'exécution est à OFF, FPD(--) n'est pas exécutée. Lorsque la condition est à ON, FPD(--) surveille le temps de mise à ON des conditions logiques de diagnostic, et met la sortie de diagnostic à ON. Si le temps dépasse T, voici ce qu'il se passe :

- 1, 2, 3... 1. Une erreur FAL(06) est générée avec le numéro de FAL spécifié dans les deux premiers chiffres de C. Si 00 est spécifié, aucune erreur ne sera générée.
2. Les instructions logiques de diagnostic sont examinées pour y trouver la première condition d'entrée à OFF et l'adresse de ce bit de condition est sortie dans les canaux de destination en commençant à D.

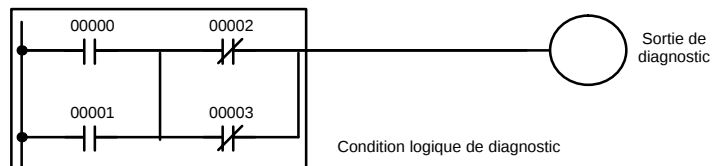
3. Le drapeau CY (SR 25504) est placé à ON. On peut exécuter une partie de programme de traitement d'erreur avec le drapeau CY si on le désire.
4. Si le bit 15 de C est à ON, un message présélectionné contenant jusqu'à 8 caractères ASCII est affiché sur le périphérique avec l'adresse du bit mentionnée en 2.

Donnée de contrôle

Fonction des bits de données de contrôle de C :

**Instructions logiques de diagnostic**

Si le temps qui s'écoule jusqu'à la mise à ON de la sortie logique de diagnostic dépasse T, les instructions logiques de diagnostic examinées pour y trouver la condition d'entrée à OFF. Si plus d'une condition d'entrée se trouve à OFF, la condition d'entrée de la ligne d'instruction la plus haute et la plus proche de la barre de bus de gauche est sélectionnée.



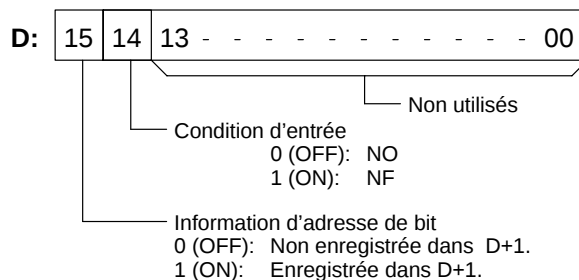
Lorsque IR 00000 à IR 00003 sont à ON, la condition NF IR 00002 est considérée comme la cause de la sortie de diagnostic ne passant pas à ON.

Sortie de diagnostic

Il y a deux façons de sortir l'adresse du bit de la condition OFF détectée dans la condition logique de diagnostic.

- 1, 2, 3...** 1. Sortie de l'adresse de bit (utilisée lorsque le bit 15 de C est à OFF).

Le bit 15 de D indique si l'information de l'adresse de bit est stockée dans D+1. Si c'est le cas, le bit 14 de D indique si la condition d'entrée est NO ou NF.



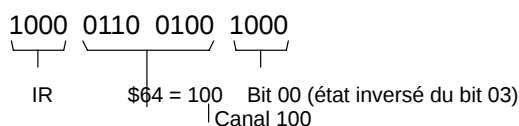
D+1 contient le code d'adresse du bit de la condition d'entrée (cf ci-dessous). Les adresses de canal, les numéros de bit et les numéros de TC sont en binaire :

Zone de donnée	Etat du bit D+1																			
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00				
IR, SR	1	0	0	0	Adresse de canal								Numéro de bit							
HR	1	0	0	1	1	Adresse de canal								Numéro de bit						
LR	1	0	0	1	0	0	Adresse de canal								Numéro de bit					
TC*	1	0	0	1	0	1	*	Numéro de tempo./compteur (TC)												

*Pour la zone TC, le bit 09 de D+1 indique si le numéro est une tempo. ou un compteur. 0 indique une temporisation et 1 un compteur.

- a) L'état du bit d'extrême gauche du numéro de bit (bit 03) est inversé.

Exemple: si D + 1 contient 1000 0110 0100 1000, IR 10000 est indiqué de la façon suivante :



2. Adresse de bit et sortie de message (sélectionnée lorsque le bit 15 de C est à ON).

Le bit 15 de D indique si l'information d'adresse de bit est stockée ou non dans D+1 à D+3. Si c'est le cas, le bit 14 de D indique si la condition d'entrée est normalement ouverte (NO) ou normalement fermée (NF) (cf tableau suivant).

Les canaux D+5 à D+8 contiennent l'information ASCII affichée sur un périphérique avec l'adresse de bit lorsque FPD(—) est exécutée. Les canaux D+5 à D+8 contiennent le message présélectionné par l'utilisateur comme suit :

Canal	Bits 15 à 08	Bits 07 à 00
D+1	20 = espace	1 ^{er} caractère ASCII
D+2	Second caractère ASCII	3 ^e caractère ASCII
D+3	4 ^e caractère ASCII	5 ^e caractère ASCII
D+4	2D = "—"	"0"=NO, "1"=NF
D+5	1 ^{er} caractère ASCII	Second caractère ASCII
D+6	3 ^e caractère ASCII	4 ^e caractère ASCII
D+7	5 ^e caractère ASCII	6 ^e caractère ASCII
D+8	7 ^e caractère ASCII	8 ^e caractère ASCII

Rem. : si l'on n'a pas besoin de 8 caractères dans le message, entrer "0D" après le dernier caractère.

Détermination du temps de surveillance

La procédure suivante peut être utilisée pour sélectionner automatiquement le temps de surveillance T, sous conditions de fonctionnement réelles lors de la spécification d'une opérande de canal pour T. Cette opération ne peut être effectuée si une constante est sélectionnée pour T ; le temps est sélectionné au dixième de seconde.

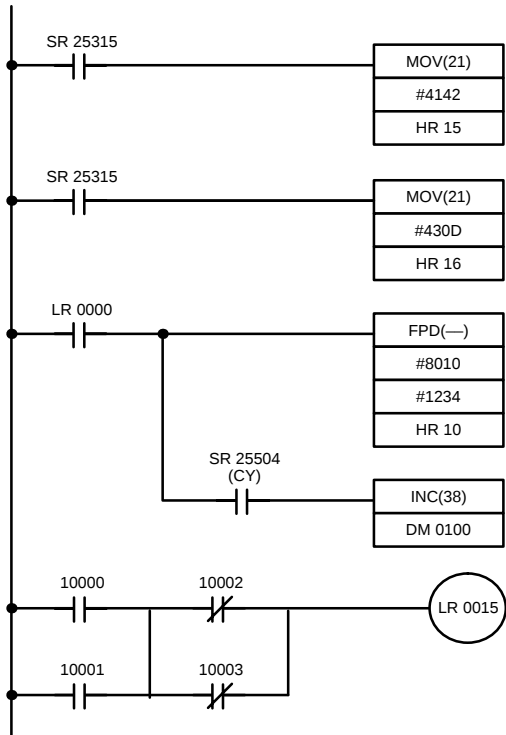
- 1, 2, 3...**
1. Mettez le CQM1 en mode MONITOR.
 2. Connectez un périphérique comme la console de programmation.
 3. Utilisez le périphérique pour placer à ON le bit de contrôle AR 2508.
 4. Exécutez le programme avec AR 2508 à ON. Si le temps de surveillance en cours dans T est dépassé 1,5 fois, le temps de surveillance réel est alors

stocké dans T. Les erreurs FAL(06) ne se produisent pas lorsque AR 2508 est à ON.

5. Placez AR 2508 à OFF lorsqu'une valeur acceptable a été stockée dans T.

Exemple

FPD(—) sélectionnée pour afficher l'adresse de bit et le message ("ABC") lorsqu'un temps de surveillance de 123,4 s est dépassé :



Adresse	Instruction	Opérande
00000	LD	25315
00001	MOV(21)	
		# 4142
		HR 15
00002	LD	25315
00003	MOV(21)	
		# 430D
		HR 16
00004	LD	LR 0000
00005	FPD(—)	
		# 8010
		# 1234
		HR 10
00006	AND	25504
00007	INC(38)	
		DM 0100
00008	LD	10000
00009	OR	10001
00010	LD NOT	10002
00011	OR NOT	10003
00012	AND LD	
00013	OUT	LR 0015

FPD(—) est exécutée et commence la surveillance lorsque LR 0000 passe à ON. Si LR 0015 ne passe pas à ON en 123,4 et si IR 10000 à IR 10003 sont tous à ON, IR 10002 est sélectionné comme cause de l'erreur et une erreur FAL(06) est générée avec un numéro de FAL de 10 ; l'adresse de bit et le message pré-sélectionné ("10002-1ABC") sont affichés sur le périphérique :

HR 10	0000
HR 11	0000
HR 12	0000
HR 13	0000
HR 14	0000
HR 15	4142
HR 16	430D
HR 17	0000
HR 18	0000



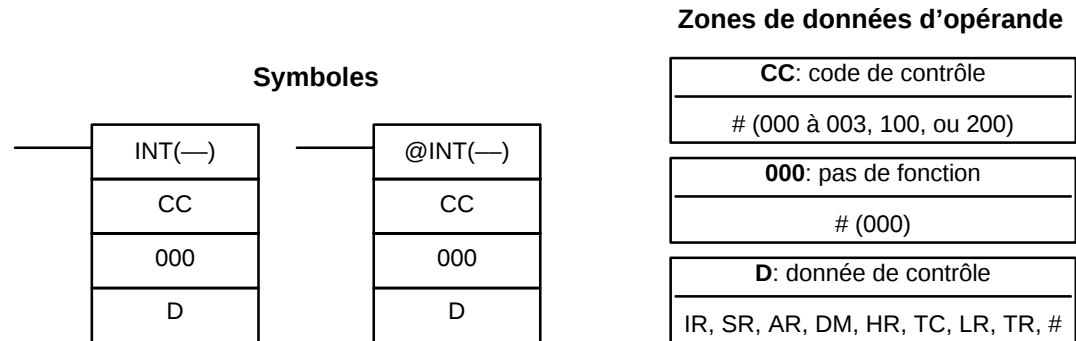
HR 10	C000
HR 11	2031
HR 12	3030
HR 13	3032
HR 14	2D31
HR 15	4142
HR 16	430D
HR 17	0000
HR 18	0000

Indique une information, une condition NF
"1"
"00"
"02"
"-1"
"AB"
"C", et code CR code
2 derniers canaux ignorés
(affichés comme espaces)

Drapeaux

- ER:** T n'est pas en BCD.
C n'est pas une constante ou n'est pas en BCD 00 à 99.
- Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- CY:** ON lorsque le temps entre l'exécution de FPD(—) et l'exécution d'une sortie de diagnostic dépasse T.

5-26-8 INTERRUPT CONTROL – CONTROLE D'INTERRUPTION DES ENTREES – INT(--)



Limites

Les DM 6644 à 6655 ne peuvent être utilisés pour D lorsque CC=002.

Description

Lorsque la condition d'exécution est à OFF, INT(—) n'est pas exécutée. Lorsque la condition est à ON, INT(—) est utilisée pour contrôler les interruptions et effectuer une des 6 fonctions du tableau suivant en fonction de la valeur de CC :

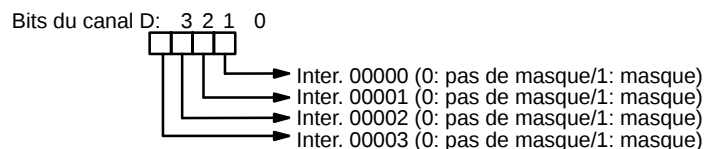
Fonction INT(--)	CC
Interruptions d'entrée masquées ou non masquées	000
Effacement des interruptions d'entrée	001
Lecture de l'état du masque en cours	002
Renouvellement de la SV du compteur	003
Toutes les interruptions sont masquées	100
Masque ôté sur toutes les interruptions	200

Ces 6 fonctions sont décrites dans de plus amples détails ci-dessous.

Interruptions d'entrées masquées ou non (CC=000)

Cette fonction s'utilise pour appliquer et ôter les masques des entrées d'interruption 00000 à 00003. Les entrées masquées sont enregistrées mais ignorées. Lorsqu'une entrée est masquée, son programme d'interruption est lancé aussitôt que le masque est ôté du bit (sauf si il est effacé avant par INT(—) avec CC=001).

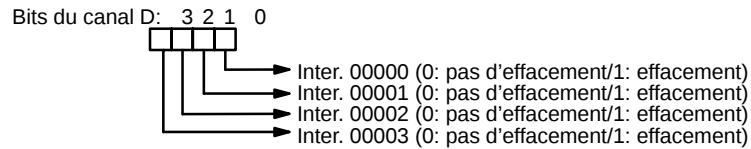
Sélectionnez le bit correspondant dans D en position 0 ou 1 pour appliquer ou ôter un masque d'entrée d'interruption. Les bits 00 à 03 correspondent à 00000 à 00003. Les bits 04 à 15 doivent être à 0.



Effacement des interruptions d'entrée (CC=001)

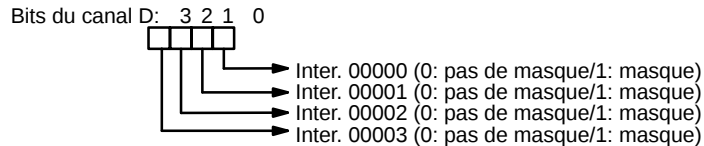
Cette fonction s'utilise pour effacer les entrées d'interruption 00000 à 00003. Les entrées d'interruption étant enregistrées, les interruptions masquées sont mises en service une fois le masque enlevé sauf si il est effacé auparavant.

Sélectionnez le bit correspondant dans D en position 1 pour effacer une entrée d'interruption d'E/S. Les bits 00 à 03 correspondent à 00000 à 00003. Les bits 04 à 15 doivent être à 0.



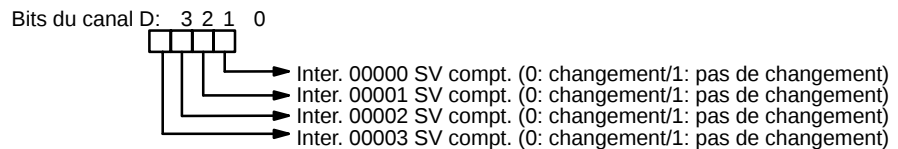
Lecture de l'état du masque en cours (CC=002)

Cette fonction est utilisée pour écrire l'état du masque en cours pour les entrées d'interruption 00000 à 00003 dans le canal D. Le bit correspondant est à ON si l'entrée est masquée (les bits 00 à 03 correspondent à 00000 à 00003.)



Renouvellement de la SV du compteur (CC=003)

Cette fonction est utilisée pour renouveler la SV (valeur sélectionnée) du compteur des entrées d'interruption 00000 à 00003 pour le canal D. Sélectionnez le bit correspondant dans D en position 1 pour renouveler la SV du compteur d'entrée (les bits 00 à 03 correspondent à 00000 à 00003).



Toutes les interruptions sont masquées (CC=100/200)

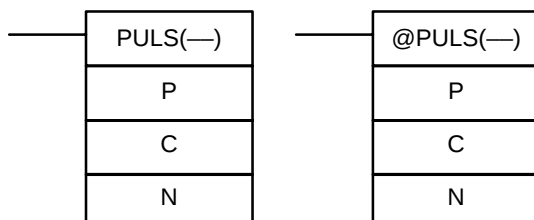
Cette fonction est utilisée pour appliquer ou ôter un masque sur tous les traitements d'interruption. Les entrées masquées sont enregistrées mais ignorées. La donnée de contrôle D n'est pas utilisée pour cette fonction. Sélectionnez D en position #0000.

Drapeaux

ER: SV de compteur incorrecte (CC=003 uniquement)
 Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
 CC=100 ou 200 pendant l'exécution d'un programme d'interruption
 CC=100 alors que toutes les entrées étaient déjà masquées.
 CC=200 alors que tous les masques d'entrée avaient déjà été enlevés.
 CC et/ou D ne sont pas comprises dans les gammes de valeurs spécifiées.

5-26-9 SET PULSES – PARAMETRES D'IMPULSION – PULS(—)

Symboles



Zones de données d'opérande

P: spécification du port
000, 001, ou 002
C: donnée de contrôle
000 à 005
N: nombre d'impulsions
IR, SR, AR, DM, HR, LR

Limites

N et N+1 doivent appartenir à la même zone de données.

Les DM 6143 à 6655 ne peuvent être utilisés pour N.

Description

PULS(—) est utilisée pour sélectionner les paramètres des sorties d'impulsion lancées plus tard dans le programme avec SPED(—) ou ACC(—). Les paramètres qui peuvent être sélectionnés sont le nombre d'impulsions sorties en mode indépendant, le sens des sorties d'impulsions des ports 1 et 2 et le point de décélération pour les sorties d'impulsion contrôlées par le mode 0 de ACC(—).

PULS(—) ayant un temps d'exécution relativement long, le temps de scrutation peut être réduit avec la version @PULS(—), mais seulement en cas de nécessité.

Spécification du port (P)

La spécification du port indique l'endroit de la sortie d'impulsions. Les paramètres sélectionnés dans C et N s'appliquent à l'instruction SPED(—) ou ACC(—) suivante dans laquelle le même port de sortie est spécifié.

P	Point de sortie de l'impulsion
000	Bit de sortie
001	Port 1
002	Port 2

Donnée de contrôle (C)

La donnée de contrôle détermine le sens de sortie de l'impulsion vers les ports 1 et 2 et indique si le nombre d'impulsions et/ou le point de décélération sont spécifiés dans N à N+3. Cette opérande doit être sélectionnée en position 000 lorsque P=000 (CW : sens horaire ; CCW : sens anti-horaire).

C	Sens	Nombre d'impulsions	Point de décélération
000	CW	Sélectionné dans N et N+1	Non sélectionné
001	CCW	Sélectionné dans N et N+1	Non sélectionné
002	CW	Sélectionné dans N et N+1	Sélectionné dans N+2 et N+3
003	CCW	Sélectionné dans N et N+1	Sélectionné dans N+2 et N+3
004	CW	Non sélectionné	Non sélectionné
005	CCW	Non sélectionné	Non sélectionné

La sélection du sens reste valable jusqu'à la fin de l'exécution du programme ou jusqu'à la ré-exécution de PULS(—).

Nombre d'impulsions et point de décélération

Lorsque C=000 à 003, N+1, N contient le nombre à 8 chiffres d'impulsions des sorties d'impulsion en mode indépendant. N+1, N peuvent être compris entre 00000001 et 16777215. La sortie d'impulsions lancée par SPED(—) ou ACC(—) s'arrête automatiquement lorsque ce nombre d'impulsions est atteint.

4 chiffres d'extrême gauche 4 chiffres d'extrême droite Gamme
 Nombre d'impulsions: N+1 N 0000 0001 à 1677 7215

Lorsque C=002 ou 003, N+3, N+2 contient le nombre d'impulsion à 8 chiffres du point de décélération utilisé dans le mode 0 de ACC(—). N+3, N+2 peuvent être compris entre 00000001 et 16777215. La sortie d'impulsions lancée par ACC(—) commence la décélération lorsque ce nombre d'impulsions a été atteint.

4 chiffres d'extrême gauche 4 chiffres d'extrême droite Gamme
 Point de décélération : N+3 N+2 0000 0001 à 1677 7215

Lorsque C=004 ou 005, ni le nombre d'impulsions ni le point de décélération ne sont sélectionnés. Sélectionnez N en position 000 lorsque C=004 ou 005.

Changements de fréquence Le nombre d'impulsions sélectionné pour la sortie est utilisé même si SPED(—) sert à changer la fréquence d'impulsions pendant le fonctionnement.
 Par exemple, si le nombre d'impulsions sélectionné est 2,100 et que la fréquence est modifiée de 1 kHz à 100 Hz, la sortie d'impulsions s'arrête en :
 12 s si la fréquence est modifiée au bout d'1 s en 1 kHz.
 3 s si la fréquence est modifiée au bout d'2 s en 1 kHz.

Drapeaux

ER: Erreur dans les sélections d'instruction.
 Zone de données dépassée.
 Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
 PULS(—) est exécutée dans un sous-programme d'interruption alors qu'une instruction d'E/S d'impulsion ou de compteur rapide est exécutée dans le programme principal.

5-26-10 SPEED OUTPUT – FREQUENCE D'IMPULSION – SPED(—)**Symboles**

SPED(—)	@SPED(—)
P	P
M	M
F	F

Zones de données d'opérande

P: spécification du port
001, 002, ou 010 à 150
M: mode de sortie
000 ou 001
F: fréquence d'impulsion
IR, SR, AR, DM, HR, LR, #

Limites

F doit être en BCD, #0000 à #5000 lorsqu'un port est spécifié, #0000 ou #0002 à #0100 lorsqu'un bit de sortie est spécifié.

Les DM 6144 à 6655 ne peuvent être utilisés pour F.

Description

SPED(—) est utilisée pour sélectionner, modifier ou arrêter une sortie d'impulsion à partir du port spécifié ou du bit de sortie. Lorsque la condition d'exécution est à OFF, SPED(—) n'est pas exécutée. Lorsque la condition est à ON, SPED(—) sélectionne la fréquence d'impulsion F pour le port ou le bit de sortie spécifié par P. M détermine le mode de sortie.

SPED(—) ayant un temps d'exécution relativement long, le temps de scrutation peut être réduit en utilisant la version @SPED(—), mais seulement en cas de nécessité.

Rem. : cf chapitre *AUCUN LIEN* pour de plus amples détails.

Spécification du port (P)

P spécifie le port ou le bit de sortie sur lequel les impulsions doivent être sorties :

P	Point de sortie d'impulsion
001	Port 1
002	Port 2
000 à 150	Bits de sortie IR 10000 à IR 10015. Les 2 premiers chiffres de P spécifient le bit de IR 100 qui est le bit de sortie et le 3 ^e chiffre de P est toujours à 0. Par ex., P=000 spécifie IR 10000, P=010 spécifie IR 10001, ... et P=150 spécifie le bit IR 10015.

Mode de sortie (M)

La valeur de M détermine le mode de sortie. Une valeur de 000 indique le mode indépendant et 001 indique le mode continu.

En mode indépendant, la sortie d'impulsions continue jusqu'à apparition d'un des trois phénomènes suivants :

- 1, 2, 3... 1. Le nombre d'impulsions spécifié par PULS(—) est atteint (exécutez PULS(—) avant SPED(—) en mode indépendant).
2. INI(—) est exécutée avec C=003.
3. SPED(—) est ré-exécutée avec la fréquence de sortie F à 000.

Lors de la sortie d'impulsions en mode indépendant, spécifiez auparavant le nombre d'impulsions en exécutant PULS(—). Lors d'une sortie à partir des ports 1 ou 2, spécifiez également le sens (CW ou CCW).

En mode indépendant, le nombre d'impulsions sorties vers les ports 1 et 2 est contenu dans IR 236 et 237 (port 1) et dans IR 238 et IR 239 (port 2).

	4 chiffres d'extrême gauche	4 chiffres d'extrême droite
PV de la sortie d'impulsions du port 1 :	IR 237	IR 236
PV de la sortie d'impulsions du port 2 :	IR 239	IR 238

En mode continu, les impulsions sont sorties jusqu'à ce que l'instruction INI(—) soit exécutée avec C=003 ou SPED(—) est ré-exécutée avec F=0000. Si le sens (CW ou CCW) n'est pas spécifié lors de la sortie du port 1 ou 2, les impulsions sont produites en sens horaire (CW).

Fréquence d'impulsions (F)

La valeur de F sélectionne la fréquence d'impulsion par unités de 10 Hz, comme ci-dessous. La sélection de F à 0000 arrête la sortie d'impulsions en un point spécifié :

Point de sortie	Valeurs possibles de F
Port 1 ou 2	0000 (arrête la sortie d'impulsions) ou 0001 à 5000 (10 Hz à 50 kHz)
Bits de sortie	0000 (arrête la sortie d'impulsions) ou 0002 à 0100 (20 Hz à 1 kHz)

Précautions

Avec le CQM1-CPU11/21-E, la méthode de rafraîchissement de sortie de DM 6639 (configuration API) doit être sélectionnée en sortie directe avant de commencer la sortie d'impulsions.

La sortie d'impulsions ne peut être utilisée lorsque la tempo. cyclique 0 est en fonctionnement.

Lorsqu'une sortie d'impulsions d'une fréquence de 500 Hz ou plus est produite à partir d'un bit de sortie, sélectionnez le traitement d'interruption pour les numéros de TC de TIMH(15) 000 à 003 en choisissant #0104 dans le DM 6629 de la configuration API.

Une sortie d'impulsion n'est possible que sur un seul bit de sortie.

Drapeaux

ER: SPED(—) est exécutée alors qu'une tempo. cyclique 0 est en fonctionnement.

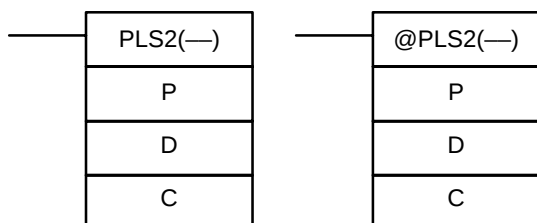
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Erreur dans les sélections d'instructions.

SPED(—) est exécutée dans un sous-programme d'interruption alors qu'une E/S d'impulsion ou qu'une instruction de compteur rapide est exécutée dans le programme principal.

5-26-11 PULSE OUTPUT- FREQUENCE D'IMPULSION 2 – PLS2(—)

Symboles



Zones de données d'opérande

P: Port de communication
001 ou 002
D: spécification de sens
000 ou 001
C: 1 ^{er} canal de contrôle
IR, SR, AR, DM, HR, LR

Limites

Instruction disponible avec le **CQM1-CPU43-E**.

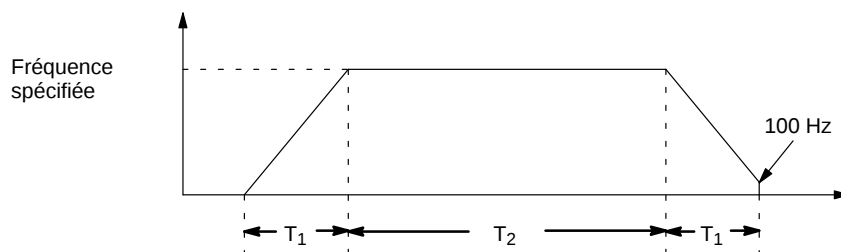
PLS2(—) ne peut être utilisée si la configuration API (DM 6611) est sélectionnée en mode compteur rapide.

P doit être égal à 001 ou 002 et D doit être égal à 000 ou 001.

C à C+3 doivent appartenir à la même zone de données.

Description

PLS2(—) est utilisée pour sortir un nombre spécifié d'impulsions CW ou CCW à partir du port 1 ou 2. La sortie par impulsions accélère jusqu'à la fréquence spécifiée et décélère à la même vitesse (la sortie d'impulsions s'arrête à 100 Hz).



Les équations suivantes montrent comment calculer le temps d'accélération/décélération approximatif et le temps de fonctionnement T₂ (les deux temps sont en secondes) :

$$T_1 = \frac{0.004 \times \text{Fréquence spécifiée}}{\text{Vitesse d'accél. / décél.}}$$

$$T_2 = \frac{\text{Nombre d'impuls.} - (T_1 \times \text{fréq. spécifiée})}{\text{Fréquence spécifiée}}$$

Rem. : 1. Bien que T₁ et T₂ varient légèrement selon les conditions de fonctionnement, le nombre de sorties d'impulsions est précis.

2. PLS2(—) ne fonctionne pas si les impulsions sont déjà sorties à partir du port spécifié. Vérifiez les drapeaux de sortie d'impulsion (AR 0515 pour le port 1 et AR 0615 pour le port 2) avant l'exécution de PLS2(—).

Sélection des opérandes

P spécifie le port d'où les impulsions sont sorties. Les impulsions sont sorties à partir du port 1 quand P=001 et les impulsions sont sorties à partir du port 2 quand P=002.

D spécifie le signal de sortie CW ou CCW. La sortie est CW lorsque D=000 et CCW lorsque D=001.

Le contenu de C détermine la vitesse d'accélération/décélération. Pendant l'accélération/décélération, la fréquence de sortie est augmentée ou réduite selon

les indications contenues dans C toutes les 4,08 ms. C doit être en BCD entre 0001 et 0200 (10 Hz et 2 kHz).

Le contenu de C+1 spécifie la fréquence. C+1 doit être en BCD entre 0001 et 5000 (10 Hz à 50 kHz).

Le contenu à 8 chiffres de C+3, C+2 détermine le nombre d'impulsions sorties. C+3, C+2 doivent être en BCD entre 0000 0001 et 1677 7215.

Drapeaux

ER: Erreur dans la sélection des opérandes.

L'UC n'est pas une CQM1-CPU43-E.

La configuration API n'est pas sélectionnée en sortie d'impulsions.

Fréquence spécifiée, vitesse d'accélération/décélération et nombre d'impulsions incorrects

(nombre d'impulsions < $T_1 \times$ fréquence spécifiée).

PLS2(—) est exécutée dans un sous-programme d'interruption alors qu'une E/S d'impulsion ou une instruction de compteur rapide est exécutée dans le programme principal.

Limite de zone de données dépassée.

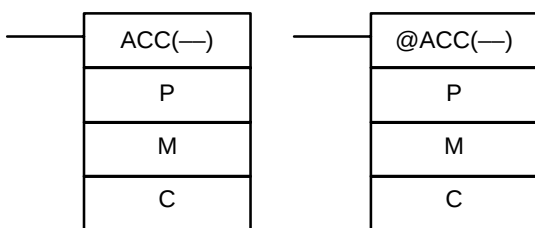
Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

AR 0515: Drapeau de sortie du port 1. ON lorsque les impulsions proviennent du port 1.

AR 0615: Drapeau de sortie du port 2. ON lorsque les impulsions proviennent du port 2.

5-26-12 ACCELERATION CONTROL – CONTROLE D'ACCELERATION – ACC(—)

Symboles



Zones de données d'opérande

P: port de communication
001 ou 002
M: spécification de mode
000 à 003
C: 1 ^{er} canal de contrôle
IR, SR, AR, DM, HR, LR

Limites

Cette instruction est disponible sur le **CQM1-CPU43-E**.

Le mode 0 de ACC(—) ne peut être utilisé si la configuration API (DM 6611) est en mode compteur rapide.

P doit être égal à 001 ou 002 et M doit être compris entre 000 et 003.

C à C+3 doivent appartenir à la même zone de données.

Description

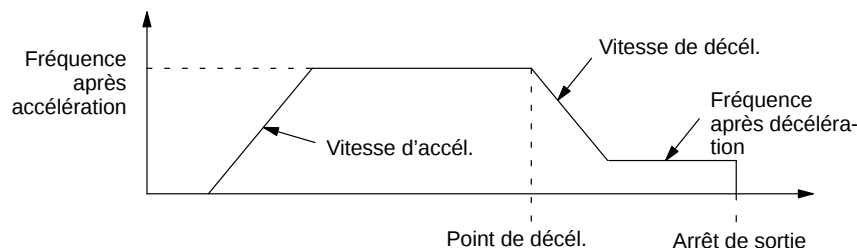
ACC(—) est utilisée avec PULS(—) pour contrôler l'accélération et/ou la décélération des sorties d'impulsions à partir des ports 1 ou 2.

Voici les 4 modes disponibles :

La fonction des canaux de contrôle varie selon les modes mais P spécifie toujours le port de sortie des impulsions et M le mode. Sélectionnez P en position 001 ou 002 pour indiquer le port 1 ou 2. Sélectionnez M en position 000 à 003 pour indiquer les modes 0 à 3.

Mode 0 (M=000)

Le mode 0 s'utilise pour sortir un nombre spécifié d'impulsions CW ou CCW à partir des ports 1 ou 2. La vitesse d'accélération, la fréquence après accélération, le point de décélération, la vitesse de décélération et la fréquence après décélération peuvent être contrôlés.



Sélection d'opérande PULS(---)PULS(—) doit être exécutée avant ACC(—) pour spécifier le sens, le nombre total d'impulsions à sortir et le point de décélération. Voici les fonctions des opérands de PULS(—); reportez-vous au chapitre 5-26-9 pour de plus amples détails :

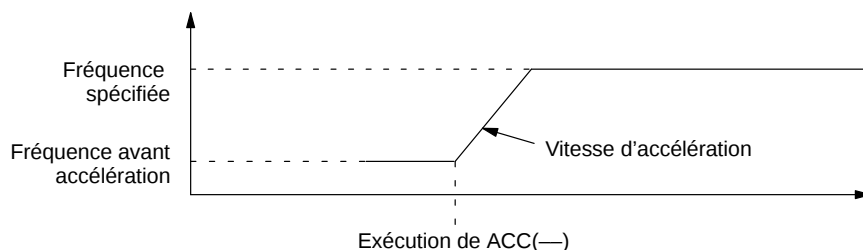
- 1, 2, 3...**
 1. La première opérande de PULS(—) spécifie la port de sortie. Les impulsions sont sorties à partir du port 1 lorsque P=001, et à partir du port 2 lorsque P=002.
 2. La deuxième opérande spécifie le sens. La sortie est en sens horaire (CW) lorsque C=002 et en sens anti-horaire (CCW) lorsque C=003.
 3. La troisième opérande spécifie le premier des 4 canaux de contrôle.
 - a) Le contenu à 8 chiffres de N+1,N (0000 0001 à 1677 7215) détermine le nombre total d'impulsions à sortir.
 - b) Le contenu à 8 chiffres de N+3,N+2 (0000 0001 à 1677 7215) détermine le point de décélération.

Canaux de contrôle ACC(---) Les 4 canaux de contrôle indiquent la vitesse d'accélération, la fréquence après accélération, la vitesse de décélération et la fréquence après décélération.

- 1, 2, 3...**
 1. Le contenu de C détermine la vitesse d'accélération. Pendant l'accélération, la fréquence de sortie est augmentée de la quantité sélectionnée dans C toutes les 4,08 ms. C doit être en BCD de 0001 à 0200 (10 Hz à 2 kHz).
 2. Le contenu de C+1 spécifie la fréquence après accélération. C+1 doit être en BCD de 0000 à 5000 (0 Hz à 50 kHz).
 3. Le contenu de C+2 détermine la vitesse de décélération. Pendant la décélération, la fréquence de sortie est réduite de la quantité sélectionnée dans C+2 toutes les 4,08 ms. C doit être en BCD de 0001 à 0200 (10 Hz à 2 kHz).
 4. Le contenu de C+3 spécifie la fréquence après décélération. C+3 doit être en BCD de 0000 à 5000 (0 Hz à 50 kHz).

Mode 1 (M=001)

Le mode 1 est utilisé pour augmenter la fréquence sortie vers une fréquence spécifiée à une vitesse spécifiée. La sortie des impulsions continue jusqu'à l'arrêt.

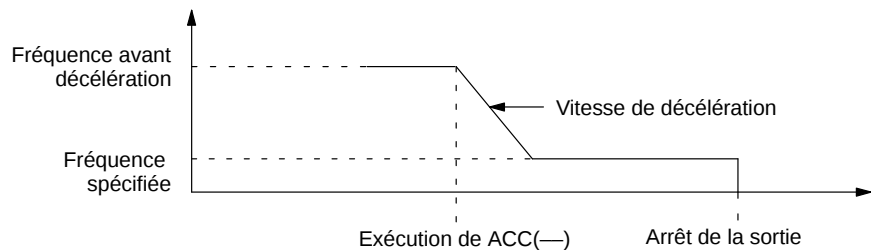


Les deux canaux de contrôle indiquent la vitesse d'accélération et la fréquence spécifiée.

- 1, 2, 3...**
1. Le contenu de C détermine la vitesse d'accélération. Pendant l'accélération, la fréquence de sortie est augmentée de la quantité sélectionnée dans C toutes les 4,08 ms. C doit être en BCD entre 0001 et 0200 (10 Hz à 2 kHz).
 2. Le contenu de C+1 indique la fréquence spécifiée. C+1 doit être en BCD de 0000 à 5000 (0 Hz à 50 kHz).

Mode 2 (M=002)

Le mode 2 est utilisé pour réduire la fréquence sortie vers une fréquence spécifiée à une vitesse spécifiée. La sortie s'arrête lorsque le nombre total d'impulsions spécifié dans PULS(—) a été sorti.

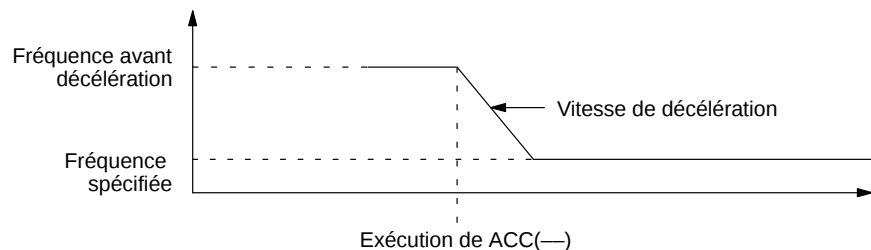


Les 2 canaux de contrôle indiquent la vitesse de décélération et la fréquence spécifiée.

- 1, 2, 3...**
1. Le contenu de C détermine la vitesse de décélération. Pendant la décélération, la fréquence sortie est réduite de la quantité sélectionnée dans C toutes les 4,08 ms. C doit être en BCD entre 0001 et 0200 (10 Hz à 2 kHz).
 2. Le contenu de C+1 indique la fréquence spécifiée. C+1 doit être en BCD entre 0000 et 5000 (0 Hz à 50 kHz).

Mode 3 (M=003)

Le mode 3 est utilisé pour diminuer la fréquence sortie vers une fréquence spécifiée à une vitesse spécifiée. La sortie des impulsions continue jusqu'à l'arrêt.



Les 2 canaux de contrôle indiquent la vitesse d'accélération et la fréquence spécifiée.

- 1, 2, 3...**
1. Le contenu de C détermine la vitesse d'accélération. Pendant l'accélération, la fréquence sortie est augmentée de la quantité sélectionnée dans C toutes les 4,08 ms. C doit être en BCD de 0001 à 0200 (10 Hz à 2 kHz).
 2. Le contenu de C+1 indique la fréquence spécifiée. C+1 doit être en BCD de 0000 à 5000 (0 Hz à 50 kHz).

Drapeaux

ER: Erreur dans les sélections d'opérandes.

L'UC n'est pas une CQM1-CPU43-E.

Configuration API non sélectionnée en sortie par impulsions.

ACC(—) est exécutée avec M=000 et le port de sortie spécifié est utilisé.

ACC(—) est exécutée dans un sous-programme d'interruption tandis qu'une entrée par impulsion ou une instruction de compteur rapide est exécutée dans le programme principal.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

AR 0515: Drapeau de sortie du port 1. ON lorsque les impulsions sont sorties à partir du port 1.

AR 0615: Drapeau de sortie du port 2. ON lorsque les impulsions sont sorties à partir du port 2.

5-26-13 PULSE WITH VARIABLE DUTY RATIO – IMPULSION A RAPPORT CYCLIQUE VARIABLE – PWM(—)

Symboles		Zones de données d'opérande
—	PWM(—)	P: Port de communication
	P	001 ou 002
	F	F: fréquence
	D	000, 001 ou 002
—	@PWM(—)	D: taux de service
	P	IR, SR, AR, DM, HR, TC, LR, #
	F	
	D	

Limites

Cette instruction est disponible avec le **CQM1-CPU43-E**.

PWM(—) ne peut être utilisée si la configuration API (DM 6643 ou DM 6644) est sélectionnée en rapport cyclique d'impulsion.

P doit être égal à 001 ou 002 et F à 000, 001 ou 002.

D doit être en BCD entre 0001 et 0099.

Description

PWM(—) est utilisée pour sortir des impulsions au rapport cyclique variable à partir du port 1 ou 2. La sortie peut être choisie entre 3 fréquences : 5,9 kHz, 1,5 kHz ou 91,6 Hz. La sortie par impulsion continue jusqu'à l'exécution de INI(—) pour l'arrêter.

Pour exécuter PWM(—), le port spécifié doit être sélectionné en rapport cyclique d'impulsion dans la configuration API. Sélectionnez le chiffre d'extrême gauche de DM 6643 en position 1 pour permettre la sortie d'impulsions en rapport cyclique à partir du port 1 et le chiffre d'extrême gauche de DM 6644 en position 1 pour permettre la sortie d'impulsions en rapport cyclique à partir du port 2. Il est impossible de sortir des impulsions normales à partir d'un port sélectionné en sortie d'impulsions en rapport cyclique.

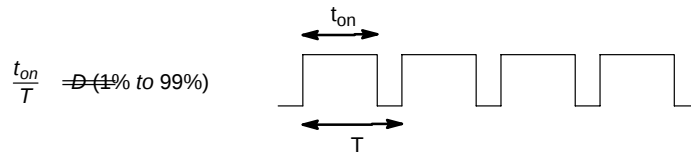
Sélections d'opérande

P spécifie le port de sortie des impulsions. Les impulsions sont sorties à partir du port 1 lorsque P=001 et à partir du port 2 lorsque P=002.

F spécifie la fréquence de la sortie par impulsion comme suit :

F	Fréquence
000	5,9 kHz
001	1,5 kHz
002	91,6 Hz

D spécifie le rapport cyclique d'impulsions, c'est-à-dire le pourcentage de temps pendant lequel la sortie est à ON. D doit être en BCD de 0001 à 0099 (1% à 99%). Le rapport cyclique est de 75 % dans le schéma suivant :

**Drapeaux**

ER: Erreur dans les sélections d'opérandes.
 L'UC n'est pas une CQM1-CPU43-E.
 Configuration API non sélectionnée en rapport cyclique d'impulsions.
 PMW(—) est exécutée dans un sous-programme d'interruption tandis qu'une E/S par impulsion ou une instruction de compteur rapide est exécutée dans le programme principal.
 Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

5-26-14 DATA SEARCH – RECHERCHE DE DONNEE – SRCH(—)**Symboles**

—	SRCH(—)	—	@SRCH(—)
	N		N
	R ₁		R ₁
	C		C

Zones de données d'opérande

N: Nombre de canaux
IR, SR, AR, DM, HR, TC, LR, #
R₁: 1 ^{er} canal de la gamme
IR, SR, AR, DM, HR, TC, LR
C: donnée de comparaison, canal de résultat
IR, SR, AR, DM, HR, LR

Limites

N doit être en BCD entre 0001 et 9999.
 R₁ et R₁+N-1 doivent appartenir à la même zone de données.
 Les DM 6143 à DM 6655 ne peuvent être utilisés pour C.

Description

Lorsque la condition d'exécution est à OFF, SRCH(—) n'est pas exécutée. Lorsque la condition est à ON, SRCH(—) recherche dans la gamme de mémoire R₁ à R₁+N-1 les adresses contenant des données de comparaison de C. Si une ou plusieurs adresses contiennent des données de comparaison, le drapeau EQ (SR 25506) est placé à ON et seule l'adresse la plus basse contenant les données de comparaison est identifiée dans C+1. L'adresse est identifiée différemment pour la zone DM :

- 1, 2, 3...**
1. Pour une adresse de la zone DM, l'adresse de canal est écrite dans C+1. Par ex., si l'adresse la plus basse contenant la donnée de comparaison est DM 0114, alors #0114 est écrit dans C+1.
 2. Pour une adresse d'une autre zone de données, le nombre d'adresses du début de recherche est écrit dans C+1. Par ex., si l'adresse la plus basse contenant les données de comparaison est IR 114 et que le premier canal de la gamme de recherche est IR 014, alors #0100 est écrit dans C+1.

Si aucune des adresses de la gamme ne contient de données de comparaison, le drapeau EQ (SR 25506) est placé à OFF et C+1 reste inchangé.

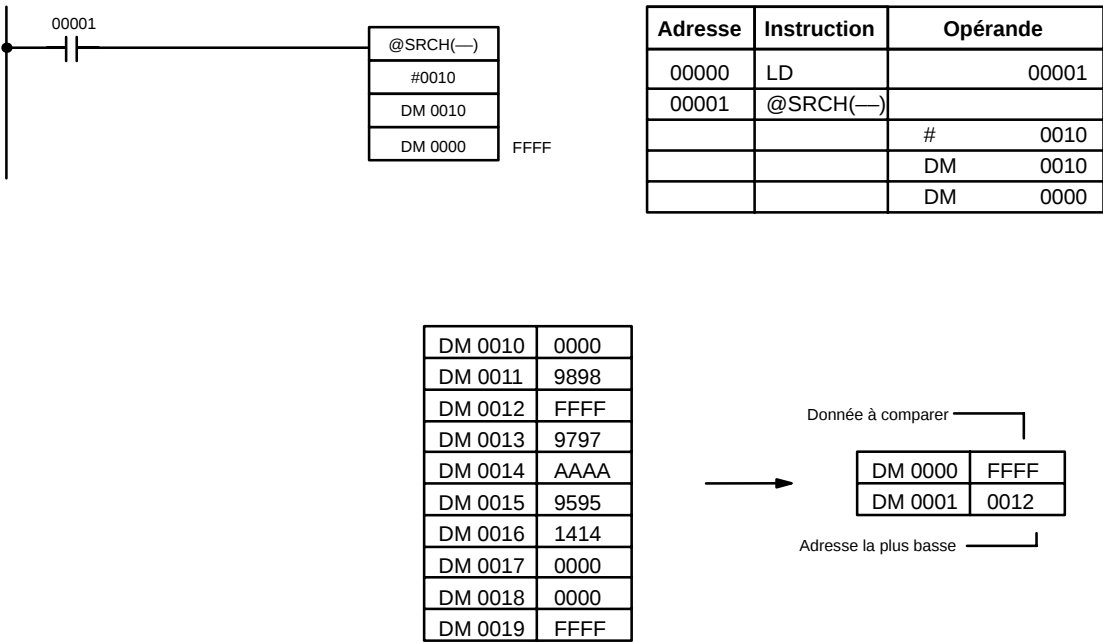
Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
 N n'est pas en BCD entre 0001 et 9999.

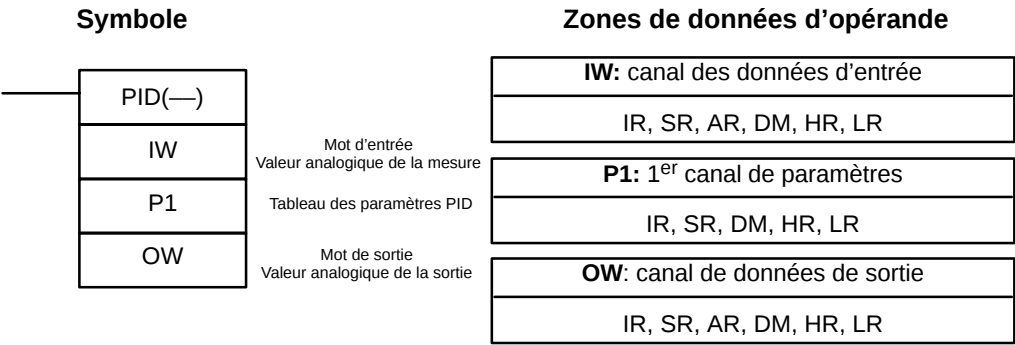
EQ: ON lorsque la donnée de comparaison correspond à une donnée de la gamme de recherche.

Exemple

Dans l'exemple suivant, les 10 mots sélectionnés (#0010) dans DM 0010 à DM 0019 est examinée pour y rechercher les adresses contenant les mêmes données que DM 0000 (#FFFF). DM 0012 contenant les mêmes données, le drapeau EQ (SR 25506) passe à ON et #0012 est inscrit dans le DM 0001.



5-26-15 PID CONTROL – CONTROLE PID – PID(—)



Limites

Cette instruction est disponible avec l'UC **CQM1-CPU4 I-E.**

Les DM 6144 à 6655 ne peuvent être utilisés pour IW, P1 à P1+32 ou OW.
P1 à P1+32 doivent appartenir à la même zone de données.

Description

PID(—) effectue un contrôle PID ayant pour base les paramètres spécifiés dans P1 à P1+6. Les données de IW sont utilisées pour calculer les données de sortie

écrites dans OW. Le tableau suivant montre les fonctions des canaux de paramètres :

Canaux	Bits	Paramètre	Fonction/gamme de sélection
P1	00 à 15	Valeur sélectionnée (SV).	Consigne pour le contrôle PID. Sélection en hexadécimal, le nombre de bits étant sélectionné par le paramètre de plage d'entrée.
P1+1	00 à 15	Largeur de la bande proportionnelle	Largeur de la bande proportionnelle et de la plage d'entrée de 0,1 à 999,9 % : doit être en BCD de 0001 à 9999 (10 = 10 % de la pleine échelle).
P1+2	00 à 15	Temps d'intégrale	Rapport du temps d'intégrale et de la période d'échantillonnage utilisé dans le contrôle intégral ; doit être en BCD de 0001 à 8191, ou égal à 9999 (9999 inhibe le contrôle intégral).
P1+3	00 à 15	Temps de dérivée	Rapport du temps de dérivée et de la période d'échantillonnage ; doit être en BCD de 0001 à 8191.
P1+4	00 à 15	Période d'échantillonnage	Intervalle entre les échantillonnages des données d'entrée de 0,1 à 102,3 s ; doit être en BCD de 0001 à 1023.
P1+5	00 à 03	Spécification du fonctionnement	Sélectionne le fonctionnement normal ou inverse ; 0 spécifie le fonctionnement inverse (chaud) et 1 le fonctionnement normal (froid).
	04 à 15	Coefficient du filtre d'entrée	Détermine le coefficient du filtre d'entrée. Plus le coefficient est bas, plus le filtre est faible. Cette sélection doit être en BCD de 100 à 199, ou égale à 000. 000 sélectionne la valeur par défaut (0,65) et 100 à 199 le coefficient (0,00 à 0,99).
P1+6	00 à 07	Plage de sortie	Détermine la résolution de la plage de sortie. Cette sélection doit être comprise entre 00 et 08, ce qui donne une gamme comprise entre 8 et 16 bits (0 = 8 ; 1 = 9 ; 2 = 10 ; 3 = 11... ; 8 = 16).
	08 à 15	Plage d'entrée	Détermine la résolution de la plage d'entrée ; cette sélection doit être comprise entre 00 et 08, ce qui donne une plage d'entrée comprise entre 8 et 16 bits (0 = 8 ; 1 = 9 ; 2 = 10 ; 3 = 11... ; 8 = 16).
P1+7 à P1+32	00 à 15	Zone de travail	Ne pas utiliser (utilisés par le système).

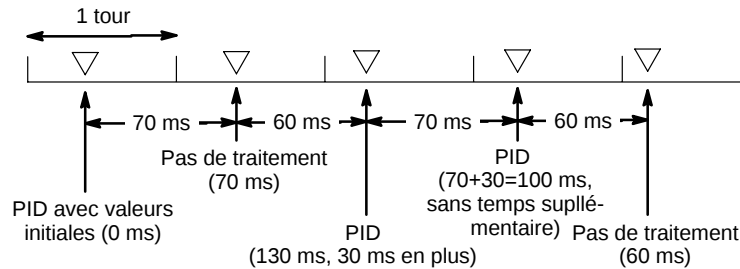
Lorsque la condition d'exécution est à OFF, PID(—) n'est pas exécutée et les données d'instruction sont maintenues. Lorsque la condition est à OFF, les données de sortie voulues peuvent être écrites directement dans OW pour le contrôle manuel.

Lorsque la condition d'exécution passe de OFF à ON, PID(—) lit les paramètres et initialise la zone de données. Il existe une fonction intégrée permettant de changer les données de sortie en continu au départ (les changements brutaux des données de sortie peuvent affecter le fonctionnement du système).

Attention : les changements de paramètres ne sont effectifs que jusqu'à ce que la condition d'exécution de PID(—) passe de OFF à ON.

Lorsque la condition d'exécution est à ON, PID(—) effectue le calcul PID sur la donnée d'entrée lorsque la période d'échantillonnage est écoulée. La période d'échantillonnage est le temps qui doit s'écouler avant que les données ne soient lues pour traitement.

Le schéma suivant montre le lien existant entre la période d'échantillonnage et le traitement PID. Le traitement PID n'est effectué que lorsque la période d'échantillonnage (100 ms dans ce cas) est écoulée.



Drapeaux

- ER:** Erreur dans la sélection des paramètres.
Le temps de scrutation est deux fois plus long que la période d'échantillonnage et PID(—) ne peut donc pas être exécutée précisément. Dans ce cas, PID(—) n'est pas exécutée.
- Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
- CY:** ON lorsque le traitement PID est terminé (OFF lorsque la période d'échantillonnage n'est pas écoulée).

5-27 Instructions de communication

5-27-1 RECEIVE – RECEPTION DE DONNEES – RXD(—)

Symboles		Zones de données d'opérande
RXD(—)	@RXD(—)	D: 1 ^{er} canal de destination
		IR, SR, AR, DM, HR, TC, LR
		C: canal de contrôle
		#
D	D	N: nombre d'octets
		IR, SR, AR, DM, HR, TC, LR, #
C	C	
N	N	

Limites

D et $D+(N \div 2) - 1$ doivent appartenir à la même zone de données.
Les DM 6144 à 6655 ne peuvent être utilisés pour D ni pour N.
N doit être en BCD de #0000 à #0256 (#0000 à #0061 en mode liaison à l'ordinateur).

Description

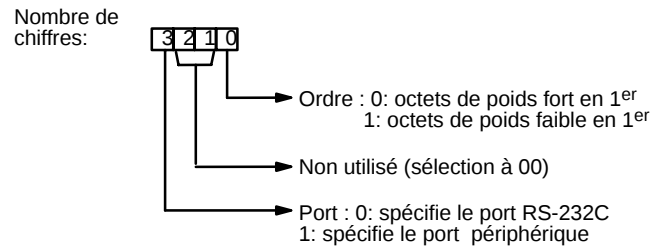
Lorsque la condition d'exécution est à OFF, RXD(—) n'est pas exécutée. Lorsque la condition est à ON, RXD(—) lit le nombre N d'octets de données reçues au port sélectionné par les mots de contrôle et écrit ces données dans les mots D à $D+(N \div 2) - 1$. On peut lire jusqu'à 256 octets de données en une seule fois. Si l'on reçoit moins de N octets, la quantité reçue est lue.

Reportez-vous au chapitre concernant RXD(—) pour de plus amples détails sur l'instruction RXD(—), le protocole de sélection des communications dans la configuration API, etc.

Attention : le CQM1 est incapable de recevoir d'autres données lorsque les 256 octets sont atteints et si l'on n'est pas en lecture RXD(—). Lisez les données dès que possible après le passage à ON du drapeau de réception terminée (AR 0806 pour le port RS-232C, AR 0814 pour le port périphérique).

Canal de contrôle

La valeur du mot de contrôle détermine le port à partir duquel les données sont lues et l'ordre d'écriture des données qui seront mémorisées :



L'ordre d'écriture des données mémorisées dépend de la valeur du digit 0 de C. 8 octets de données 12345678... s'écrivent de la manière suivante :

Digit 0 = 0

	MSB	LSB
D	1	2
D+1	3	4
D+2	5	6
D+3	7	8
⋮	⋮	⋮

Digit 0 = 1

	MSB	LSB
D	2	1
D+1	4	3
D+2	6	5
D+3	8	7
⋮	⋮	⋮

Flags

ER: L'UC n'est pas équipée de port RS-232C.

Pas de connexion d'autre appareil au port sélectionné.

Erreur dans les sélections de communication ou d'opérande de l'API.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Les canaux de destination (D à D+(N÷2)-1) dépassent la zone de données.

AR 08: AR 0806 passe à ON lorsque la donnée est reçue normalement par le port RS-232C. Remettez RXD(—) à 0 après exécution.

AR 0814 passe à ON lorsque la donnée est reçue normalement au port périphérique. Remettez RXD(—) à 0 après exécution.

AR 09: Contient le nombre d'octets reçus par le port RS-232C. Remettez RXD(—) à 0 après exécution.

AR 10: Contient le nombre d'octets reçus par le port périphérique. Remettez RXD(—) à zéro après exécution.

Rem. : les drapeaux de communication et les compteurs peuvent être effacés en spécifiant 0000 pour N ou en utilisant les bits de remise à zéro de port (SR 25208 pour le port périphérique et SR 25209 pour le port RS-232C).

5-27-2 TRANSMIT– ENVOI DE DONNEES – TXD(—)**Symboles**

TXD(—)	@TXD(—)
S	S
C	C
N	N

Zones de données d'opérande

S: 1 ^{er} canal source
IR, SR, AR, DM, HR, TC, LR
C: canal de contrôle
#
N: nombre d'octets
IR, SR, AR, DM, HR, TC, LR, #

Limites

S et $S+(N/2)-1$ doivent appartenir à la même zone de données.

Les DM 6144 à 6655 ne peuvent être utilisés pour S ni N.

N doit être en BCD de #0000 à #0256 (#0000 à #0061 en mode liaison à l'ordinateur).

Description

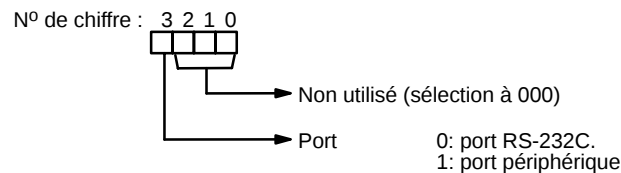
Lorsque la condition d'exécution est à OFF, TXD(—) n'est pas exécutée. Lorsque la condition est à ON, TXD(—) lit N octets de données des canaux S à $S+(N/2)-1$, les convertit en ASCII et sort les données du port sélectionné. TXD(—) fonctionne différemment en mode SYSMAC-WAY et RS-232C.

Reportez-vous au chapitre 1-4 pour de plus amples détails sur l'utilisation de l'instruction TXD(—), la sélection du protocole de communication dans la configuration API, etc.

Rem. : le drapeau AR 0805 est à ON lorsque le CQM1 est capable de transmettre les données par le port RS-232C et AR 0813 est à ON lorsque le CQM1 est capable de transmettre les données par le port périphérique.

Mode liaison SYSMAC-WAY

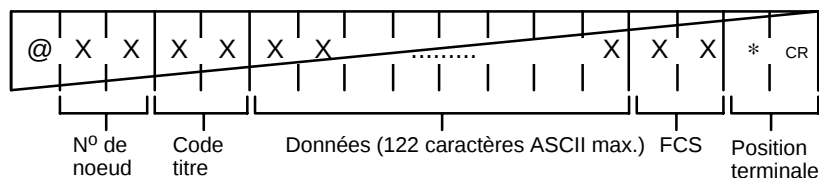
N doit être en BCD de #0000 à #0061 (c'est-à-dire jusqu'à 122 octets ASCII). La valeur du mot de contrôle détermine le port à partir duquel les données sont sorties :



Le nombre d'octets spécifié est lu de S à $S+(N/2)-1$, converti en ASCII, et transmis par le port sélectionné. Les octets des données source ci-dessous sont transmis dans cet ordre : 12345678...

	MSB	LSB
S	1	2
S+1	3	4
S+2	5	6
S+3	7	8
⋮	⋮	⋮
⋮	⋮	⋮

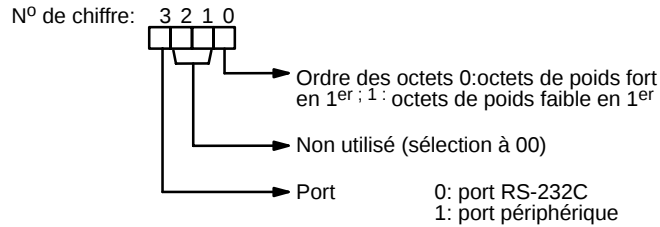
Le schéma suivant montre la trame SYSMAC-WAY (TXD) envoyée par le CQM1. Le CQM1 attache automatiquement les préfixes et suffixes comme le numéro de noeud, le titre et le FCS.

**Mode RS-232C**

N doit être en BCD de #0000 à #00256. La valeur du mot de contrôle détermine le port à partir duquel les données sont sorties et l'ordre dans lequel les données sont écrites en mémoire.

Mot de contrôle

La valeur du mot de contrôle détermine le port à partir duquel les données sont lues et l'ordre dans lequel les données sont écrites en mémoire.



Le nombre d'octets spécifié est lu de S à S+(NP2)-1 et transmis par le port sélectionné.

	MSB	LSB
S	1	2
S+1	3	4
S+2	5	6
S+3	7	8
⋮	⋮	⋮

Lorsque le chiffre 3 de C est égal à 0, les octets des données source ci-dessus sont transmises dans cet ordre : 12345678...

Lorsque le chiffre 3 de C est égal à 1, les octets des données source ci-dessus sont transmises dans cet ordre : 21436587...

Rem. : lorsque les codes de fin et de départ sont spécifiés, la longueur totale des données doit être de 256 octets max., y compris les codes de départ et de fin.

Drapeaux

ER: L'UC n'est équipée d'un port RS-232C.

Un autre appareil n'a pas été relié au port périphérique.

Erreur dans les sélections de communication (configuration API) ou dans les sélections d'opérande.

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Les canaux source (S à S+(N+2)-1) dépassent la zone de données.

AR 08: AR 0805 passe à ON lorsqu'il est possible de transmettre des données par le port RS-232C. AR 0813 passe à ON lorsque la transmission est possible par le port périphérique.

5-28 Instructions d'E/S étendues

5-28-1 7-SEGMENT DISPLAY OUTPUT – SORTIE 7 SEGMENTS – 7SEG(--)

Symboles	Zones de données d'opérande
7SEG(—)	S: 1 ^{er} canal source
S	IR, SR, AR, DM, HR, TC, LR
O	O: canal de sortie
C	IR, SR, AR, HR, LR, TC, DM
	C: donnée de contrôle
	000 à 007

Limites

N'utilisez pas 7SEG(—) plus de deux fois dans le programme.

Description

Lorsque la condition d'exécution est à OFF, 7SEG(—) n'est pas exécutée. Lorsque la condition est à ON, 7SEG(—) lit les données source (4 ou 8 chiffres), les convertit en données d'affichage à 7 segments et sort ces données sur l'affichage à 7 segments connecté à la sortie indiquée par O.

La valeur de C indique le nombre de chiffres des données source et la logique des cartes d'E/S comme sur le tableau suivant :

Données source	Logique d'entrée des données d'affichage	Logique d'entrée du verrouillage d'affichage	C
4 chiffres (S)	Identique à la carte de sortie	Identique à la carte de sortie	0000
		Différente de la carte de sortie	0001
	Différente de la carte de sortie	Identique à la carte de sortie	0002
		Différente de la carte de sortie	0003
8 chiffres (S, S+1)	Identique à la carte de sortie	Identique à la carte de sortie	0004
		Différente de la carte de sortie	0005
	Différente de la carte de sortie	Identique à la carte de sortie	0006
		Différente de la carte de sortie	0007

S'il y a 8 chiffres de données de source, ils sont placés dans S et S+1, les chiffres de poids fort étant placés dans S+1. S'il y a 4 chiffres de données de source, ils sont placés dans S.

7SEG(—) affiche les données à 8 ou 4 chiffres en 12 cycles, puis recommence et affiche à nouveau les données.

Drapeaux

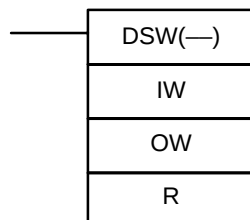
ER: S et S+1 n'appartiennent pas à la même zone de données (en sélection d'affichage 8 chiffres).

Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).

Erreur dans les sélections d'opérande.

SR 25409: ON lorsque 7SEG(—) est en cours d'exécution.

5-28-2 DIGITAL SWITCH INPUT- ENTREE ROUE CODEUSE – DSW(—)

Symboles**Zones de données d'opérande**

IW: canal d'entrée
IR, SR, AR, DM, HR, TC, LR
OW: canal de sortie
IR, SR, AR, DM, HR, TC, LR
R: 1 ^{er} canal de résultat
IR, SR, AR, DM, HR, TC, LR

Limites

Les DM 6144 à 6655 ne peuvent être utilisés pour R.

Description	DSW(—) est utilisée pour lire la valeur sélectionnée sur une roue codeuse connectée aux cartes d'E/S. Lorsque la condition d'exécution est à OFF, DSW(—) n'est pas exécutée. Lorsque la condition est à ON, DSW(—) lit la valeur (4 ou 8 chiffres) sélectionnée sur la roue codeuse à partir de IW et place le résultat dans R. Si la valeur est un nombre à 8 chiffres, elle est placée dans R et R+1, les chiffres de poids fort étant placés dans R+1. Le nombre de chiffres est sélectionné dans le DM 6639 de la configuration API. DSW(—) lit les données à 8 ou 4 chiffres en 12 cycles, puis recommence et continue la lecture des données.
Drapeaux	ER: IW et/ou OW ne sont pas attribués aux bonnes cartes d'E/S. Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée). R et R+1 n'appartiennent pas à la même zone de données (lorsque le CQM1 est sélectionné pour recevoir les données à 8 chiffres). SR 25410: ON lorsque DSW(—) est en cours d'exécution

5-28-3 **HEXADECIMAL KEY INPUT – ENTREE CLAVIER HEXADECIMAL – HKY(—)**

Symboles	Zones de données d'opérande						
HKY(—) IW OW D	<table><tr><td>IW: canal d'entrée</td></tr><tr><td>IR, SR, AR, DM, HR, TC, LR</td></tr><tr><td>OW: canal de sortie du signal de contrôle</td></tr><tr><td>IR, SR, AR, DM, HR, TC, LR</td></tr><tr><td>D: 1^{er} canal d'enregistrement</td></tr><tr><td>IR, SR, AR, DM, HR, TC, LR</td></tr></table>	IW: canal d'entrée	IR, SR, AR, DM, HR, TC, LR	OW: canal de sortie du signal de contrôle	IR, SR, AR, DM, HR, TC, LR	D: 1 ^{er} canal d'enregistrement	IR, SR, AR, DM, HR, TC, LR
IW: canal d'entrée							
IR, SR, AR, DM, HR, TC, LR							
OW: canal de sortie du signal de contrôle							
IR, SR, AR, DM, HR, TC, LR							
D: 1 ^{er} canal d'enregistrement							
IR, SR, AR, DM, HR, TC, LR							

Limites	D et D+2 doivent appartenir à la même zone de données. N'utilisez pas HKY(—) plus de deux fois dans le programme. Les DM 6144 à 6655 ne peuvent être utilisés pour D.
Description	Lorsque la condition d'exécution est à OFF, HKY(—) n'est pas exécutée. Lorsque la condition est à ON, HKY(—) entre les données à partir d'un clavier hexadécimal connecté à l'entrée indiquée par IW. Les données sont entrées de deux façons : 1, 2, 3... 1. Un registre à décalage de 8 chiffres est créé dans D et D+1. Lorsqu'une touche est sollicitée sur le clavier hexadécimal, le chiffre correspondant est décalé vers le chiffre de poids le plus faible de D. Les autres chiffres de D, D+1 sont décalés vers la gauche et le chiffre de poids le plus fort de D+1 est perdu. 2. Les bits de D+2 et le bit 4 de OW indiquent l'entrée de touche. Lorsqu'une des touches du clavier (0 à F) est sollicitée, le bit correspondant dans D+2 (00 à 15) et le bit 4 de OW passent à ON. Rem. : lorsque l'on appuie sur l'une des touches, les autres sont désactivées. HKY(—) entre chaque chiffre en 3 à 12 cycles, puis recommence et continue à effectuer les entrées.
Drapeaux	ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée). D et D+2 n'appartiennent pas à la même zone de données.

SR 25408: ON lorsque HKY(—) est en cours d'exécution.

5-28-4 TEN KEY INPUT – ENTREE CLAVIER 10 TOUCHES – TKY(—)

Symboles	Zones de données d'opérande
— TKY(—)	IW: canal d'entrée IR, SR, AR, DM, HR, TC, LR
IW	D₁: 1 ^{er} canal d'enregistrement IR, SR, AR, DM, HR, TC, LR
D ₁	D₂: canal d'entrée de touche IR, SR, AR, DM, HR, TC, LR
D ₂	

Limites

D₁ et D₁+1 doivent appartenir à la même zone de données.

Les DM 6143 à 6655 ne peuvent être utilisés pour D₁.

Description

Lorsque la condition d'exécution est à OFF, TKY(—) n'est pas exécutée. Lorsque la condition est à ON, TKY(—) entre les données à partir d'un clavier à 10 touches connecté à l'entrée indiquée par IW. Les données sont entrées de deux façons :

- 1, 2, 3...** 1. Un registre à décalage de 8 chiffres est créé dans D₁ et D₁+1. Lorsqu'une touche est sollicitée sur le clavier à 10 touches, le chiffre BCD correspondant est décalé vers le chiffre de poids le plus faible de D₁. Les autres chiffres de D₁, D₁+1 sont décalés vers la gauche et le chiffre de poids le plus fort de D₁+1 est perdu.
2. Les 10 premiers bits de D₂ indiquent l'entrée de touche. Lorsqu'une des touches du clavier (0 à 9) est sollicitée, le bit correspondant dans D₂ (00 à 09) passe à ON.

Rem. : lorsque l'on appuie sur l'une des touches, les autres sont désactivées.

TKY(—) peut être utilisée plusieurs fois dans le programme en changeant le canal d'entrée IW.

Drapeaux

ER: Le canal DM indirectement adressé n'existe pas (le contenu du canal *DM n'est pas en BCD ou la limite de la zone DM a été dépassée).
D₁ et D₁+1 n'appartiennent pas à la même zone de données.

Chapitre 6

Commandes de liaison à l'ordinateur

Ce chapitre expose les méthodes d'utilisation des commandes de liaison de l'ordinateur vers les ports du CQM1.

6-1	Procédure de communication	300
6-2	Formats de commande et de réponse	302
6-2-1	Commandes venant de l'ordinateur	302
6-2-2	Commandes venant de l'API	304
6-3	Commandes de liaison à l'ordinateur	305
6-3-1	IR/SR AREA READ — RR	305
6-3-2	LR AREA READ — RL	305
6-3-3	HR AREA READ — RH	306
6-3-4	PV READ — RC	306
6-3-5	TC STATUS READ — RG	307
6-3-6	DM AREA READ — RD	307
6-3-7	AR AREA READ — RJ	307
6-3-8	IR/SR AREA WRITE — WR	308
6-3-9	LR AREA WRITE — WL	308
6-3-10	HR AREA WRITE — WH	309
6-3-11	PV WRITE — WC	309
6-3-12	TC STATUS WRITE — WG	310
6-3-13	DM AREA WRITE — WD	310
6-3-14	AR AREA WRITE — WJ	311
6-3-15	SV READ 1 — R#	311
6-3-16	SV READ 2 — R\$	312
6-3-17	SV READ 3 — R%	313
6-3-18	SV CHANGE 1 — W#	314
6-3-19	SV CHANGE 2 — W\$	315
6-3-20	SV CHANGE 3 — W%	316
6-3-21	STATUS READ — MS	317
6-3-22	STATUS WRITE — SC	318
6-3-23	ERROR READ — MF	318
6-3-24	FORCED SET — KS	319
6-3-25	FORCED RESET — KR	320
6-3-26	MULTIPLE FORCED SET/RESET — FK	321
6-3-27	FORCED SET/RESET CANCEL — KC	322
6-3-28	PC MODEL READ — MM	323
6-3-29	TEST — TS	323
6-3-30	PROGRAM READ — RP	324
6-3-31	PROGRAM WRITE — WP	324
6-3-32	COMPOUND COMMAND — QQ	324
6-3-33	ABORT — XZ	326
6-3-34	INITIALIZE — **	326
6-3-35	Commande indéfinie — IC	327

Tableau des commandes Les commandes ci-dessous peuvent être utilisées pour la communication de l'ordinateur vers l'API.

Code titre	Mode API			Dénomination
	RUN	MON	PRG	
RR	Valide	Valide	Valide	IR/SR AREA READ
RL	Valide	Valide	Valide	LR AREA READ
RH	Valide	Valide	Valide	HR AREA READ
RC	Valide	Valide	Valide	PV READ
RG	Valide	Valide	Valide	TC STATUS READ
RD	Valide	Valide	Valide	DM AREA READ
RJ	Valide	Valide	Valide	AR AREA READ
WR	Non val.	Valide	Valide	IR/SR AREA WRITE
WL	Non val.	Valide	Valide	LR AREA WRITE
WH	Non val.	Valide	Valide	HR AREA WRITE
WC	Non val.	Valide	Valide	PV WRITE
WG	Non val.	Valide	Valide	TC STATUS WRITE
WD	Non val.	Valide	Valide	DM AREA WRITE
WJ	Non val.	Valide	Valide	AR AREA WRITE
R#	Valide	Valide	Valide	SV READ 1
R\$	Valide	Valide	Valide	SV READ 2
R%	Valide	Valide	Valide	SV READ 3
W#	Non val.	Valide	Valide	SV CHANGE 1
W\$	Non val.	Valide	Valide	SV CHANGE 2
W%	Non val.	Valide	Valide	SV CHANGE 3
MS	Valide	Valide	Valide	STATUS READ
SC	Valide	Valide	Valide	STATUS WRITE
MF	Valide	Valide	Valide	ERROR READ
KS	Non val.	Valide	Valide	FORCED SET
KR	Non val.	Valide	Valide	FORCED RESET
FK	Non val.	Valide	Valide	MULTIPLE FORCED SET/RESET
KC	Non val.	Valide	Valide	FORCED SET/RESET CANCEL
MM	Valide	Valide	Valide	PC MODEL READ
TS	Valide	Valide	Valide	TEST
RP	Valide	Valide	Valide	PROGRAM READ
WP	Non val.	Non val.	Valide	PROGRAM WRITE
QQ	Valide	Valide	Valide	COMPOUND COMMAND
XZ	Valide	Valide	Valide	ABORT (commande uniquement)
**	Valide	Valide	Valide	INITIALIZE (commande uniquement)
IC	---	---	---	Commande indéfinie (réponse uniquement)

6-1 Procédure de communication

La communication est exécutée par un échange de commandes et de réponses entre l'ordinateur et l'API. Avec le CQM1, il existe deux méthodes de communication : la normale (commandes allant de l'ordinateur vers l'API) et une autre, permettant de faire remonter les commandes de l'API vers l'ordinateur.

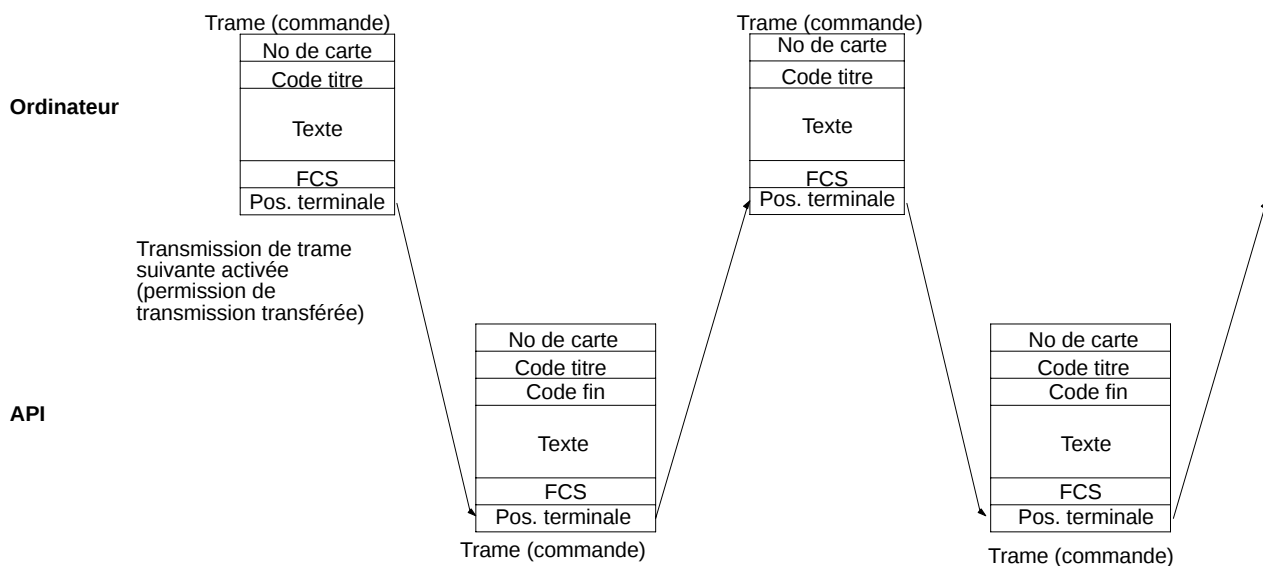
Transmission et réception de trame

Commandes et réponses s'échangent comme sur les schémas ci-dessous. Un bloc de données transféré en une seule transmission porte le nom de "trame". Une seule trame contient au maximum 131 caractères de données.

La permission d'envoyer une trame s'appelle "permission de transmission". La carte ayant cette permission peut effectuer l'envoi à n'importe quel moment. La permission est échangée entre l'ordinateur et l'API à chaque transmission ; elle passe de la carte de transmission à la carte de réception lors de la réception d'une position terminale (code marquant la fin d'une commande ou d'une réponse) ou d'un code de limite (code de séparation des trames).

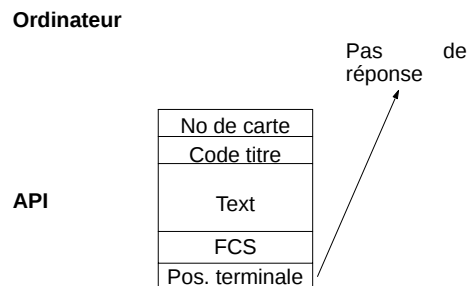
Commandes de l'ordinateur

En communication à l'ordinateur, c'est ce dernier qui a la permission de transmission en premier lieu et qui commence la communication. L'API envoie alors automatiquement une réponse.



Commandes de l'API

Il est également possible d'envoyer des commandes à l'ordinateur en mode communication à l'ordinateur. Dans ce cas, l'API a la permission de transmission et commence la communication.



Lorsque les commandes sont envoyées vers l'ordinateur, les données sont transmises de l'API vers l'ordinateur. Si l'on a besoin d'une réponse à la commande, on peut utiliser une commande de communication à l'ordinateur pour transmettre une réponse de l'ordinateur vers l'API.

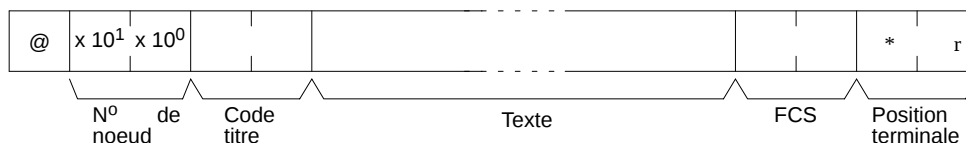
6-2 Formats de commande et de réponse

On traite ici des formats de commande et de réponse échangés dans les communications de liaison à l'ordinateur.

6-2-1 Commandes venant de l'ordinateur

Format de commande

Format à préparer pour la transmission d'une commande venant de l'ordinateur:



@ : un symbole "@" doit être placé au début.

N° de noeud

Identifie l'API qui communique avec l'ordinateur.

Spécifie le numéro de noeud sélectionné pour l'API dans la configuration API (DM 6648, DM 6653).

Code titre

Sélectionnez le code de commande à 2 caractères.

Texte

Sélectionnez les paramètres de commande.

FCS

Sélectionnez le code à 2 caractères du FCS ("Frame Check Sequence") : cf page suivante.

Position terminale

Sélectionnez deux caractères, "*" et le retour chariot (CHR\$(13)) indiquant la fin de la commande.

Format de réponse

La réponse de l'API est envoyée dans le format suivant. Préparez le programme de façon à ce que les données de réponse puissent être interprétées et traitées.



@, n° de noeud, code titre

Contenu identique à celui de la commande.

Code fin

Etat de commande terminée envoyé (par ex., erreur ou pas d'erreur).

Texte

Le texte est envoyé seulement en cas de données du type donnée de lecture.

FCS, position terminale

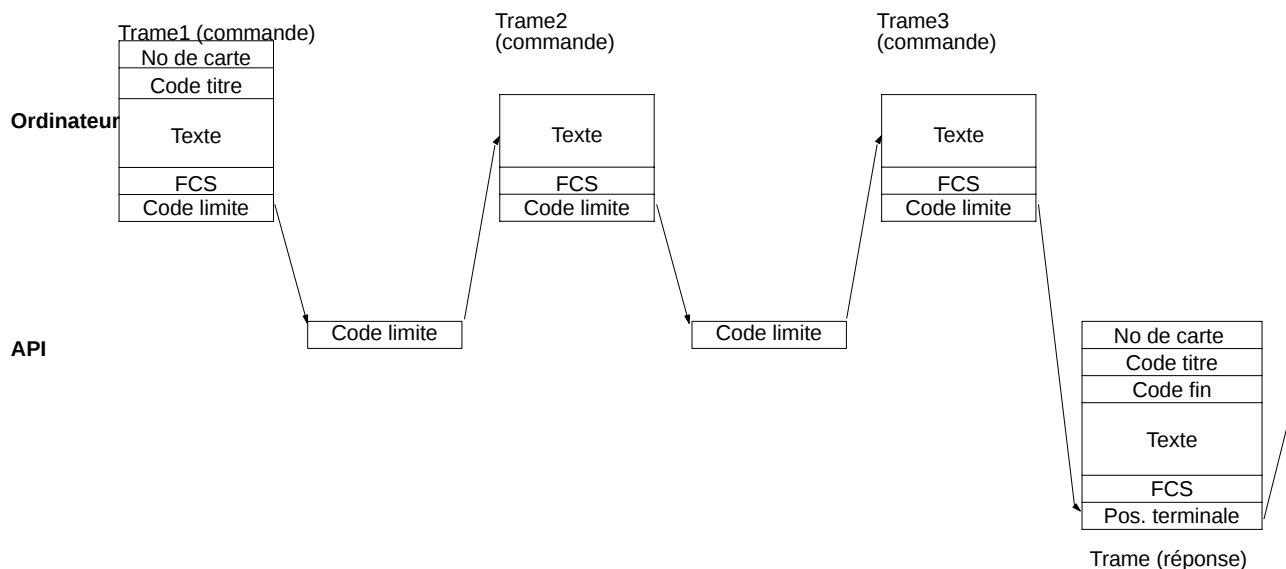
Cf rubrique "Format de commande".

Transmissions longues

Le bloc de données transmissibles en une seule trame le plus important est de 131 caractères. Une commande ou une réponse de 132 caractères minimum doit donc être divisée en plusieurs trames avant transmission. Lorsqu'une transmission est divisée, la fin des trames de début et intermédiaires est marquée par un code de limite au lieu d'une position terminale.

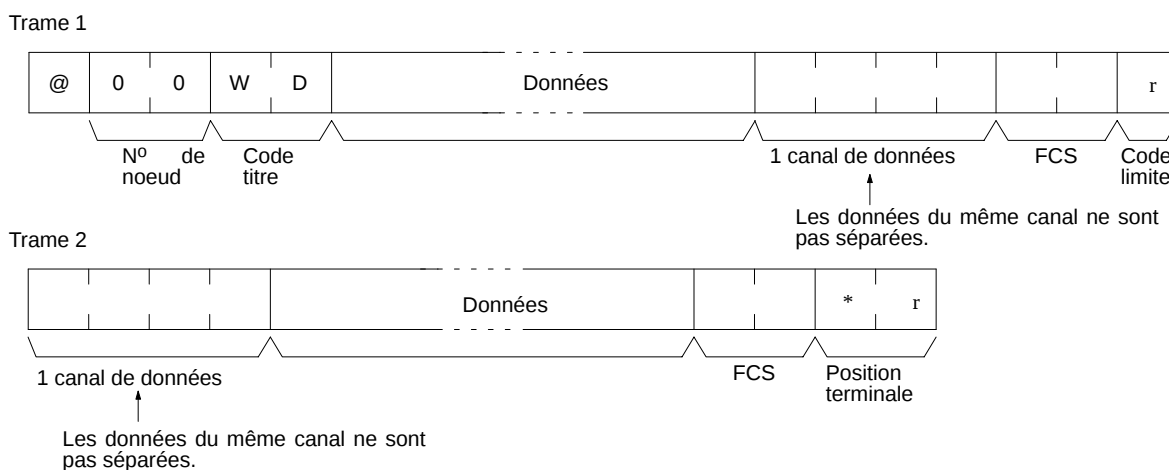
Pendant la transmission des trames, le noeud de réception attend le code de limite. Après transmission de ce dernier, la trame suivante est envoyée. La procédure est répétée jusqu'à ce que la commande ou la réponse entière soient

transmises. Voici un exemple de communication en liaison à l'ordinateur adressée à un API :



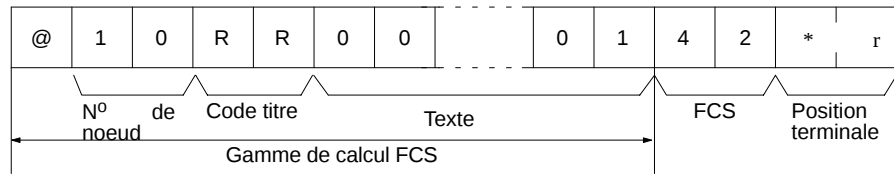
Précautions pour les transmissions longues

Lorsque vous divisez des commandes comme WR, WL, WC ou WD exécutant des opérations d'écriture, veillez à ne pas diviser en trames séparées les données devant être écrites en un seul canal. Comme sur le schéma suivant, veillez à séparer les trames de façon à ce qu'elles coïncident avec les divisions de canaux.



FCS (Frame Check Sequence) Lors de la transmission d'une trame, un FCS est placé juste avant le code de limite ou la position terminale pour vérifier si une éventuelle erreur a été générée. Le FCS est une donnée à 8 bits convertie en deux caractères ASCII. La donnée à 8 bits est le résultat un OR EXCLUSIF effectué sur la donnée de début de trame et jusqu'à la fin du texte de cette trame (juste avant le FCS). En calculant

le FCS à chaque réception de trame et en vérifiant le résultat par rapport au FCS de la trame, on peut vérifier s'il se trouve des erreurs dans la trame.



Code ASCII				
@	40	0100	0000	
		EOR		
1	31	0011	0001	
		EOR		
0	30	0011	0000	
		EOR		
R	52	0101	0010	
		EOR		
1	31	0011	0001	
		EOR		
Résultat du calcul		0100	0010	
		i	i	Converti en hexadécimal.
		4	2	Traité comme ASCII

Exemple de programme FCS Voici un exemple de sous-programme BASIC pour l'exécution d'une vérification FCS sur une trame reçue par un ordinateur :

```

400 *FSCHECK
410 L=LEN(RESPONSE$) ' ..... Donnée transmise et reçue
420 Q=0:FCSCCK$=""
430 A$=RIGHT$(RESPONSE$,1)
440 PRINT RESPONSE$,A$,L
450 IF A$="*" THEN LENG$=LEN(RESPONSE$)-3
      ELSE LENG$=LEN(RESPONSE$)-2
460 FCSP$=MID$(RESPONSE$,LENG$+1,2) ' ... Donnée FCS reçue
470 FOR I=1 TO LENG$ ' ..... Nombre de caractères en FCS
480 Q=ASC(MID$(RESPONSE$,I,1)) XOR Q
490 NEXT I
500 FCSD$=HEX$(Q)
510 IF LEN(FCSD$)=1 THEN FCSD$="0"+FCSD$ 'Résultat FCS
520 IF FCSD$<>FCSP$ THEN FCSCCK$="ERR"
530 PRINT"FCSD$=";FCSD$,"FCSP$=";FCSP$,"FCSCCK$=";FCSCCK$
540 RETURN

```

Rem. : 1. La réception normale de donnée comprend le FCS, le code de limite et la position terminale, etc. Lorsqu'une erreur se produit dans la transmission, le FCS ou d'autres données peuvent manquer : programmez le système en fonction de cette éventualité.

2. Dans cet exemple de programme, le code CR (CHR\$(13)) n'a pas été entré pour RESPONSE\$. Lors de l'entrée du code CR, effectuez les changements dans les lignes 430 et 450.

6-2-2 Commandes venant de l'API

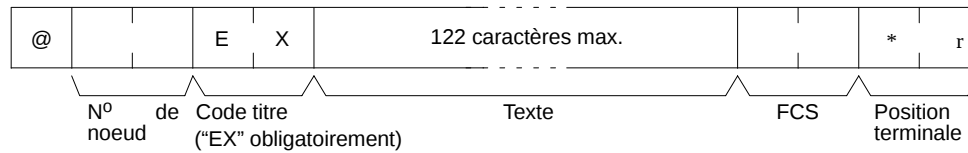
Dans les communications à l'ordinateur, les commandes sont envoyées de l'ordinateur vers l'API mais l'inverse est également possible. En mode de liaison à l'ordinateur, toutes les données peuvent être transmises de l'API vers l'ordinateur. Pour envoyer une commande à l'ordinateur, utilisez l'instruction TRANSMIT (TXD(--)) du programme API en mode de liaison à l'ordinateur.

TXD(—) sort une donnée à partir du port spécifié (RS-232C ou périphérique).

Format de réception

Lorsque TXD(—) est exécutée, les données stockées dans les canaux à partir du premier canal d'envoi sont converties en ASCII et sorties sur l'ordinateur

comme des commandes de liaison à l'ordinateur dans le format suivant. Le symbole "@", le numéro de noeud, le code titre, le FCS et le code de limite sont ajoutés automatiquement lors de la transmission. Au niveau de l'ordinateur, il faut préparer à l'avance un programme d'interprétation et de traitement dans le format suivant :



Un octet de données (2 chiffres hexadécimaux) est converti en deux caractères ASCII pour la transmission ; la quantité de données de la transmission est deux fois supérieure à la quantité de canaux spécifiée pour TXD(—). Le nombre maximum de caractères de transmission est de 122 et le nombre maximum d'octets pouvant être désignés pour TXD(—) est de la moitié, soit 61.

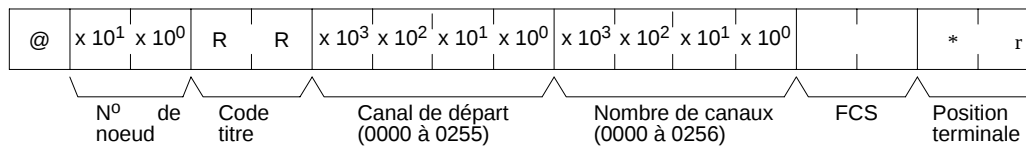
6-3 Commandes de liaison à l'ordinateur

On traite ici des commandes pouvant être envoyées de l'ordinateur à l'API.

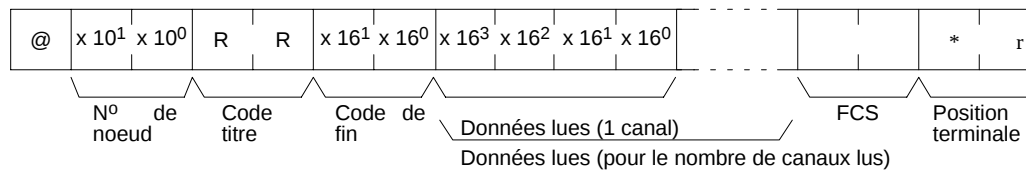
6-3-1 IR/SR AREA READ -- RR

Lecture du contenu du numéro spécifié de canaux IR et SR, en partant du canal spécifié.

Format de commande



Format de réponse



Paramètres

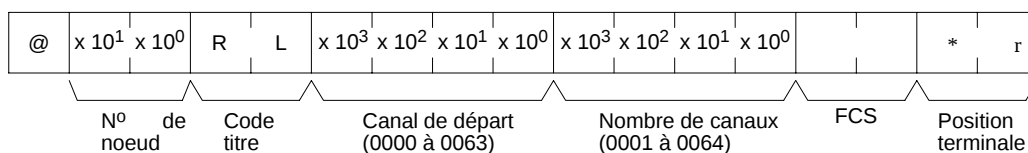
Données lues (Réponse)

Le contenu du numéro de canal spécifié par la commande est envoyé en hexadécimal comme réponse. Les canaux sont renvoyés par ordre, en commençant par le canal de départ spécifié.

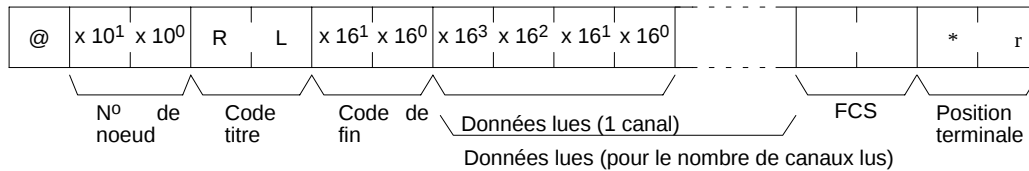
6-3-2 LR AREA READ -- RL

Lecture du contenu du nombre spécifié de canaux LR, en commençant par le canal spécifié.

Format de commande



Format de réponse



Paramètres

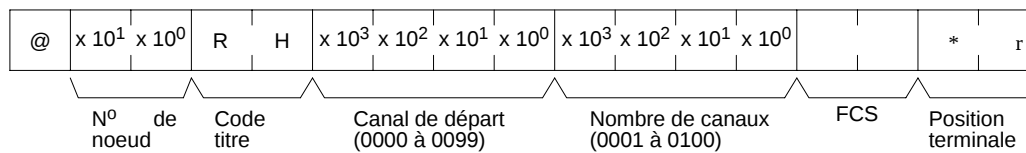
Données lues (Réponse)

Le contenu du numéro de canal spécifié par la commande est envoyé sous forme hexadécimale comme réponse. Les canaux sont renvoyés par ordre, en commençant par le canal de départ spécifié.

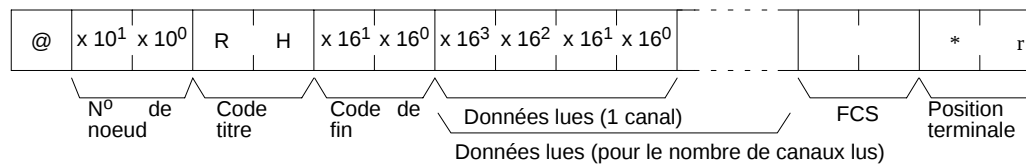
6-3-3 HR AREA READ -- RH

Lecture du contenu des numéros de canaux HR spécifiés, en partant du canal spécifié.

Format de commande



Format de réponse



Paramètres

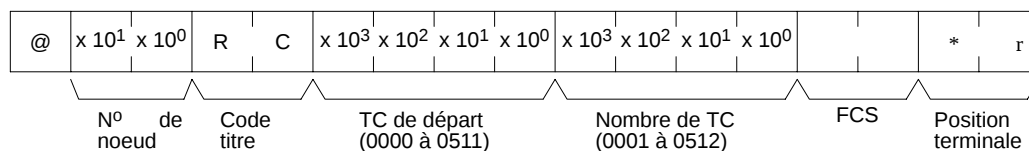
Données lues (Réponse)

Le contenu du numéro de canal spécifié par la commande est envoyé en hexadécimal en réponse. Les canaux sont renvoyés par ordre, en partant du canal de départ spécifié.

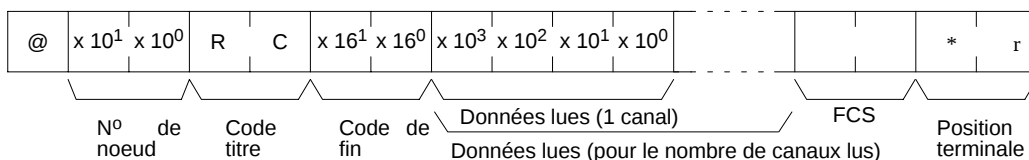
6-3-4 PV READ -- RC

Lecture du contenu de la PV (valeur en cours) des numéros de temporisations et compteurs (TC) spécifiés, en partant du TC spécifié.

Format de commande



Format de réponse



Paramètres

Données lues (Réponse)

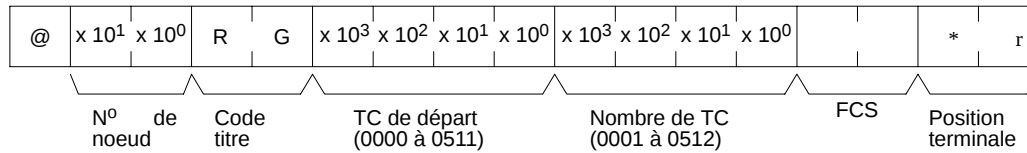
Le numéro des valeurs en cours spécifié par la commande est envoyé en hexa-

décimal comme réponse. Les PV sont renvoyés par ordre, en partant du TC spécifié.

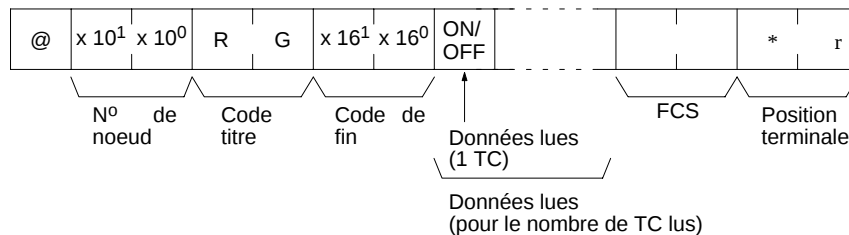
6-3-5 TC STATUS READ -- RG

Lit l'état du drapeau de fin des numéros spécifiés de TC, en partant du TC spécifié.

Format de commande



Format de réponse



Paramètres

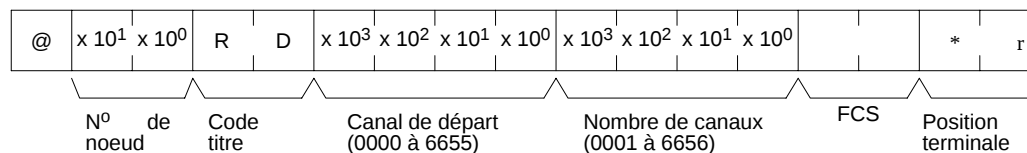
Données lues (Réponse)

L'état du numéro de drapeau de fin spécifié par la commande est renvoyé comme réponse. "1" indique que le drapeau de fin est à ON.

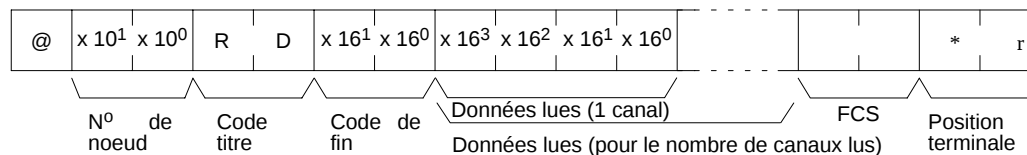
6-3-6 DM AREA READ -- RD

Lit le contenu des numéros de canaux DM spécifiés, en partant du canal spécifié.

Format de commande



Format de réponse



Paramètres

Données lues (Réponse)

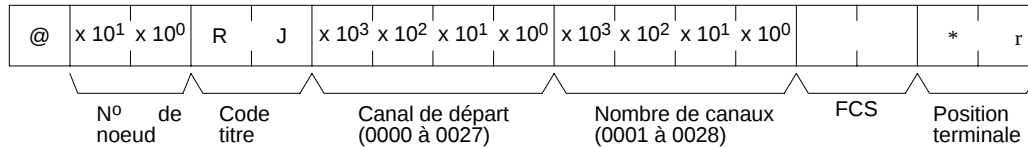
Le contenu des numéros de canaux spécifiés par la commande est envoyé en hexadécimal comme réponse. Les canaux sont renvoyés par ordre, en commençant par le canal de départ spécifié.

Rem. : attention à la configuration de la zone DM, qui varie selon le modèle d'UC.

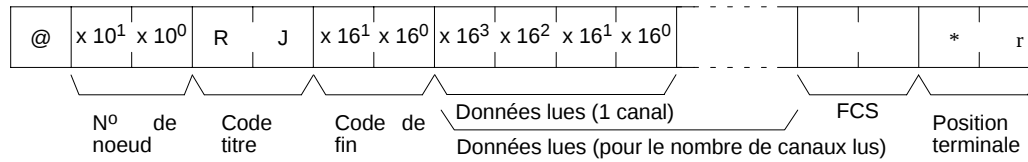
6-3-7 AR AREA READ -- RJ

Lit le contenu des numéros spécifiés de canaux AR, en partant du canal de départ spécifié.

Format de commande



Format de réponse



Paramètres

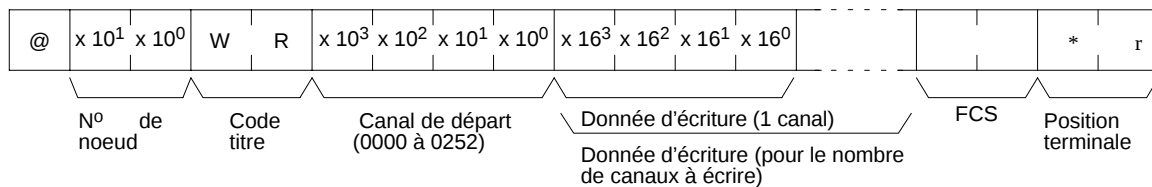
Données lues (Réponse)

Le contenu des numéros de canaux spécifiés par la commande est envoyé en hexadécimal comme réponse. Les canaux sont renvoyés par ordre, en partant du canal de départ spécifié.

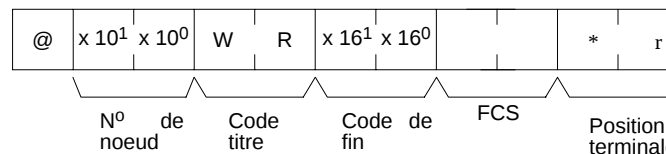
6-3-8 IR/SR AREA WRITE -- WR

Ecrit les données dans les zones IR et SR, en partant du canal spécifié ; l'écriture se fait canal par canal.

Format de commande



Format de réponse



Paramètres

Données d'écriture (Commande)

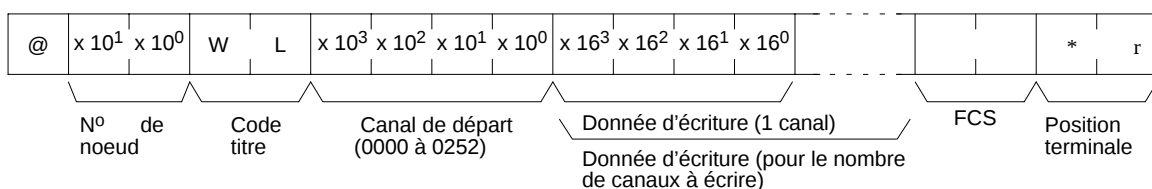
Spécifiez par ordre le contenu des numéros des canaux à écrire dans la zone IR ou SR en hexadécimal, en partant du canal de départ spécifié.

Rem. : si les données spécifiées pour l'écriture dépassent la gamme admissible, une erreur est générée et l'opération d'écriture n'est pas exécutée. Si l'on spécifie par ex. 252 comme canal de départ pour l'écriture et que deux canaux de données sont spécifiés, alors 253 devient le dernier canal d'écriture des données et la commande n'est pas exécutée car SR 253 est au-delà de la gamme admise.

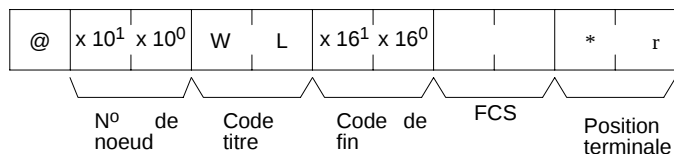
6-3-9 LR AREA WRITE -- WL

Ecrit les données dans la zone LR, en partant du canal spécifié. L'écriture s'effectue canal par canal.

Format de commande



Format de réponse



Paramètres

Données d'écriture (Commande)

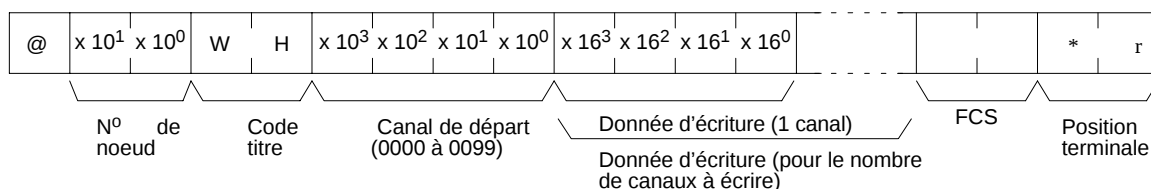
Spécifiez par ordre le contenu des numéros des canaux à écrire dans la zone LR en hexadécimal, en partant du canal de départ spécifié.

Rem. : si les données spécifiées pour l'écriture dépassent la gamme admissible, une erreur est générée et l'opération d'écriture n'est pas exécutée. Si l'on spécifie par ex. 60 comme canal de départ pour l'écriture et que cinq canaux de données sont spécifiés, alors 64 devient le dernier canal d'écriture des données et la commande n'est pas exécutée car TR 64 est au-delà de la gamme admise.

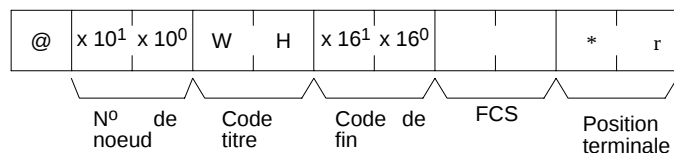
6-3-10 HR AREA WRITE -- WH

Ecrit les données dans la zone HR, en partant du canal spécifié ; l'écriture s'effectue canal par canal.

Format de commande



Format de réponse



Paramètres

Données d'écriture (Commande)

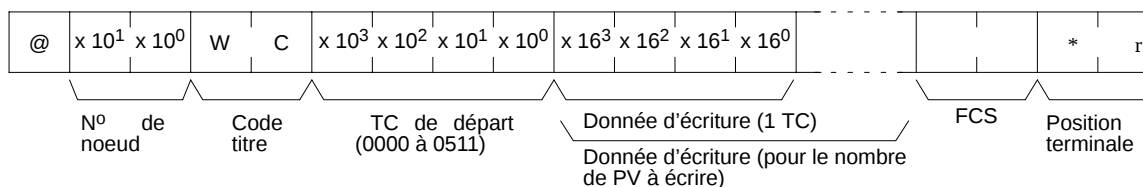
Spécifiez par ordre le contenu des numéros des canaux à écrire dans la zone HR en hexadécimal, en partant du canal de départ spécifié.

Rem. : si les données spécifiées pour l'écriture dépassent la gamme admissible, une erreur est générée et l'opération d'écriture n'est pas exécutée. Si l'on spécifie par ex. 98 comme canal de départ pour l'écriture et que trois canaux de données sont spécifiés, alors 100 devient le dernier canal d'écriture des données et la commande n'est pas exécutée car HR 100 est au-delà de la gamme admise.

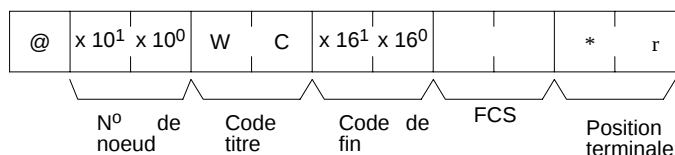
6-3-11 PV WRITE -- WC

Ecrit les valeurs en cours (PV) des TC (tempo./compteurs) en partant du TC spécifié.

Format de commande



Format de réponse



Paramètres

Données d'écriture (Commande)

Spécifiez en nombre décimaux (BCD) les valeurs en cours des TC à écrire, en partant du TC de départ spécifié.

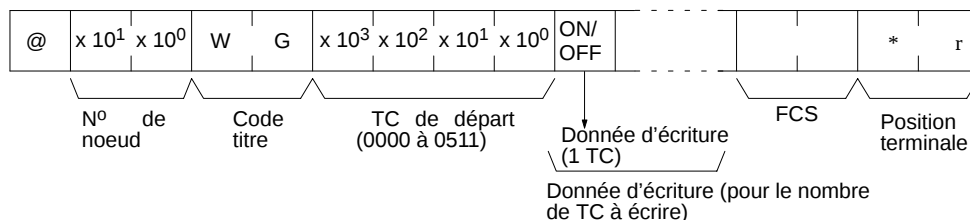
Rem. : 1. Si les données spécifiées pour l'écriture dépassent la gamme admissible, une erreur est générée et l'opération d'écriture n'est pas exécutée. Si l'on spécifie par ex. 510 comme canal de départ pour l'écriture et que trois canaux de données sont spécifiés, alors 512 devient le dernier canal d'écriture des données et la commande n'est pas exécutée car TC 512 est au-delà de la gamme admise.

2. Lorsque cette commande est utilisée pour écrire des données dans la zone PV, les drapeaux de fin des TC écrits sont placés à OFF.

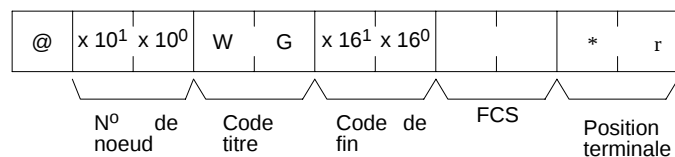
6-3-12 TC STATUS WRITE -- WG

Ecrit l'état des drapeaux de fin des TC de la zone TC, en partant du numéro de TC spécifié ; l'écriture se fait numéro par numéro.

Format de commande



Format de réponse



Paramètres

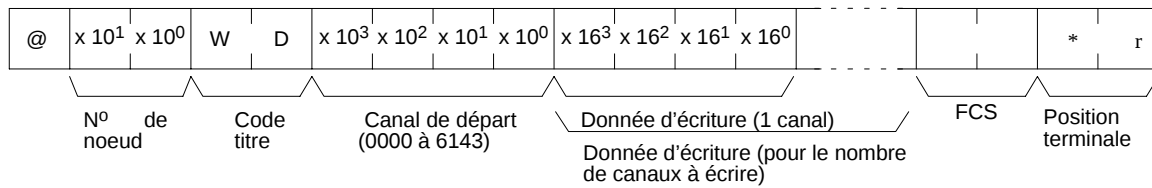
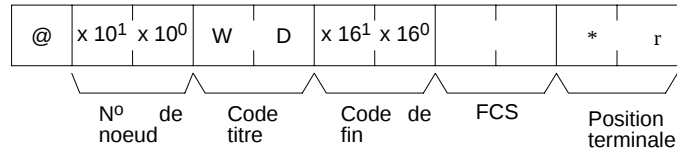
Données d'écriture (Commande)

Spécifiez par ordre l'état des drapeaux de fin pour le nombre de TC à écrire en partant du canal de départ spécifié : ON ("1") ou "OFF" ("0"). Un drapeau de fin à ON indique que le temps ou le comptage est atteint.

Rem. : si les données spécifiées pour l'écriture dépassent la gamme admissible, une erreur est générée et l'opération d'écriture n'est pas exécutée. Si l'on spécifie par ex. 510 comme canal de départ pour l'écriture et que trois canaux de données sont spécifiés, alors 512 devient le dernier canal d'écriture des données et la commande n'est pas exécutée car TC 512 est au-delà de la gamme admise.

6-3-13 DM AREA WRITE -- WD

Ecrit les données dans la zone DM en partant du canal spécifié ; l'écriture s'effectue canal par canal.

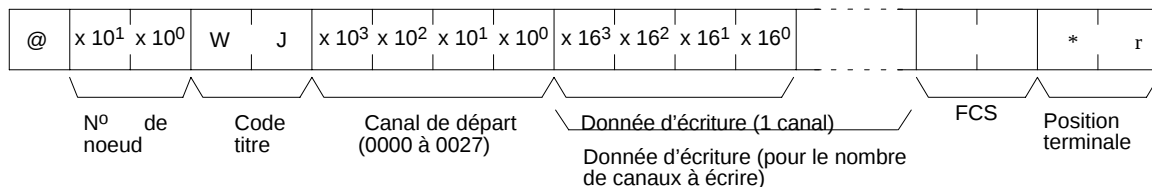
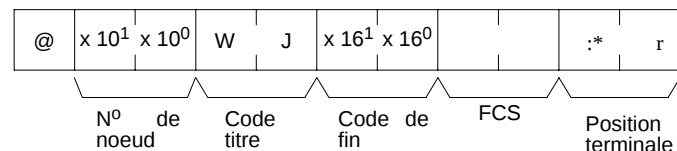
Format de commande**Format de réponse****Paramètres****Données d'écriture (Commande)**

Spécifiez par ordre le contenu des numéros des canaux à écrire dans la zone DM en hexadécimal, en partant du canal de départ spécifié.

- Rem. : 1. Si les données spécifiées pour l'écriture dépassent la gamme admissible, une erreur est générée et l'opération d'écriture n'est pas exécutée. Si l'on spécifie par ex. 6142 comme canal de départ pour l'écriture et que trois canaux de données sont spécifiés, alors 6144 devient le dernier canal d'écriture des données et la commande n'est pas exécutée car DM 6144 est au-delà de la gamme admise.
2. attention à la configuration de la zone DM car elle varie selon l'UC.

6-3-14 AR AREA WRITE -- WJ

Ecrit les données dans la zone AR en partant du canal spécifié ; l'écriture s'effectue canal par canal.

Format de commande**Format de réponse****Paramètres****Données d'écriture (Commande)**

Spécifiez par ordre le contenu des numéros des canaux à écrire dans la zone AR en hexadécimal, en partant du canal de départ spécifié.

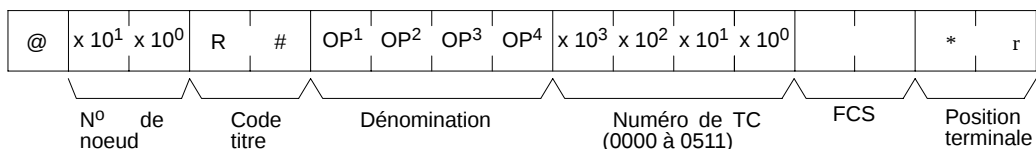
- Rem. : si les données spécifiées pour l'écriture dépassent la gamme admissible, une erreur est générée et l'opération d'écriture n'est pas exécutée. Si l'on spécifie par ex. 26 comme canal de départ pour l'écriture et que deux canaux de données sont spécifiés, alors 28 devient le dernier canal d'écriture des données et la commande n'est pas exécutée car AR 28 est au-delà de la gamme admise.

6-3-15 SV READ 1 -- R#

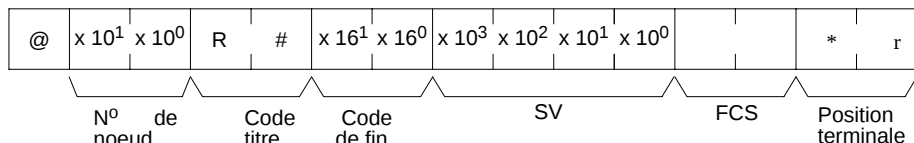
Recherche la première occurrence des instructions TIM, TIMH(15), CNT et CNTR(12) avec le numéro de TC dans le programme utilisateur et lit la PV qui

est normalement sélectionnée comme constante. La SV lue est un nombre décimal à 4 chiffres (BCD). Le programme est examiné depuis le début et la réponse peut prendre environ 10 secondes.

Format de commande



Format de réponse



Paramètres

Dénomination, numéro de TC (Commande)

Spécifiez l'instruction de lecture de SV dans la "Dénomination". Effectuez cette sélection en 4 caractères. Dans "Numéro de TC", spécifiez le TC utilisé pour l'instruction.

Instruction				Classification	Gamme de TC
OP1	OP2	OP3	OP4		
T	I	M	(S)	TIMER	0000 à 0511
T	I	M	H	HIGH-SPEED TIMER	
C	N	T	(S)	COUNTER	
C	N	T	R	REVERSIBLE COUNTER	

(S): espace

SV (Réponse)

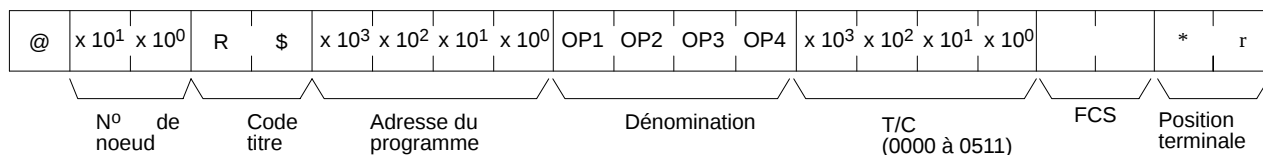
La constante est renvoyée.

- Rem. : 1. L'instruction spécifiée dans "Dénomination" doit comporter 4 caractères. Remplissez les espaces pour obtenir un nombre total de 4 caractères.
2. Si l'on utilise la même instruction plusieurs fois dans un même programme, seule la première est lue.
3. Utilisez cette commande seulement lorsqu'on est certain qu'une constante a été sélectionnée pour la SV.

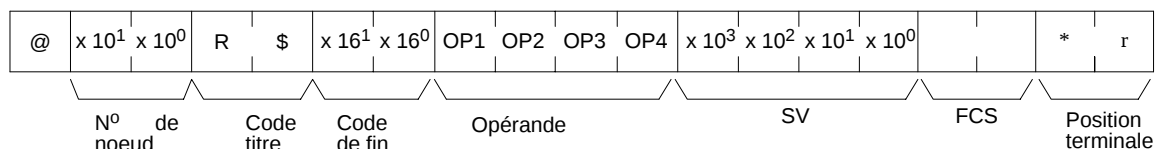
6-3-16 SV READ 2 -- R\$

Lit la constante SV ou l'adresse du canal ou la SV est stockée. La SV lue est un nombre décimal à 4 chiffres (BCD) écrit comme deuxième opérande de l'instruction TIM, TIMH(15), CNT ou CNTR(12) dans l'adresse de programme spécifiée dans le programme utilisateur ; cela ne peut être fait qu'avec un programme inférieur à 10K.

Format de commande



Format de réponse



Paramètres

Dénomination, numéro de TC (Commande)

Spécifiez l'instruction de lecture de SV dans la "Dénomination". Effectuez cette sélection en 4 caractères. Dans "Numéro de TC", spécifiez le TC utilisé pour l'instruction.

Instruction				Classification	Gamme de TC
OP1	OP2	OP3	OP4		
T	I	M	(S)	TIMER	0000 à 0511
T	I	M	H	HIGH-SPEED TIMER	
C	N	T	(S)	COUNTER	
C	N	T	R	REVERSIBLE COUNTER	

(S): espace

Opérande, SV (Réponse)

L'indication de classification de la SV est envoyé dans "Opérande," et l'adresse de canal où est stocké la SV ou la constante de la SV sont renvoyées dans "SV."

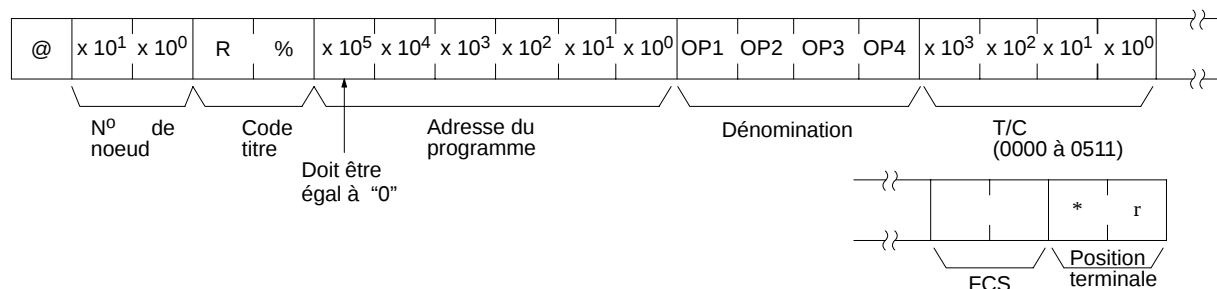
Opérande				Classification	Constante ou adresse de canal
OP1	OP2	OP3	OP4		
C	I	O	(S)	IR ou SR	0000 à 0255
L	R	(S)	(S)	LR	0000 à 0063
H	R	(S)	(S)	HR	0000 à 0099
A	R	(S)	(S)	AR	0000 à 0027
D	M	(S)	(S)	DM	0000 à 6655
D	M	*	(S)	DM (indirect)	0000 à 6655
C	O	N	(S)	Constante	0000 à 9999

Rem. : l'instruction spécifiée doit comporter 4 caractères. Remplissez les blancs avec des espaces pour obtenir un nombre total de 4 caractères.

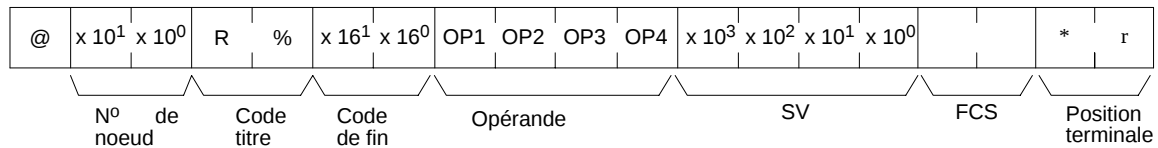
6-3-17 SV READ 3 -- R%

Lit la constante SV ou l'adresse du canal ou la SV est stockée. La SV lue est un nombre décimal à 4 chiffres (BCD) écrit comme deuxième opérande de l'instruction TIM, TIMH(15), CNT ou CNTR(12) dans l'adresse de programme spécifiée dans le programme utilisateur ; cela ne peut être fait qu'avec un programme inférieur à 10K.

Format de commande



Format de réponse



Paramètres

Dénomination, numéro de TC (Commande)

Spécifiez l'instruction de lecture de SV dans la "Dénomination". Effectuez cette sélection en 4 caractères. Dans "Numéro de TC", spécifiez le TC utilisé pour l'instruction.

Instruction				Classification	Gamme de TC
OP1	OP2	OP3	OP4		
T	I	M	(S)	TIMER	0000 to 0511
T	I	M	H	HIGH-SPEED TIMER	
C	N	T	(S)	COUNTER	
C	N	T	R	REVERSIBLE COUNTER	

(S): espace

Opérande, SV (Réponse)

L'indication de classification de la SV est envoyé dans "Opérande", et l'adresse de canal où est stocké la SV ou la constante de la SV sont renvoyées dans "SV."

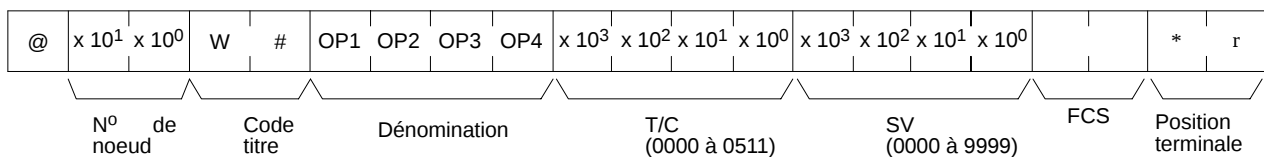
Opérande				Classification	Constante ou adresse de canal
OP1	OP2	OP3	OP4		
C	I	O	(S)	IR ou SR	0000 à 0255
L	R	(S)	(S)	LR	0000 à 0063
H	R	(S)	(S)	HR	0000 à 0099
A	R	(S)	(S)	AR	0000 à 0027
D	M	(S)	(S)	DM	0000 à 6655
D	M	*	(S)	DM (indirect)	0000 à 6655
C	O	N	(S)	Constante	0000 à 9999

Rem. : l'instruction spécifiée doit comporter 4 caractères. Remplissez les blancs avec des espaces pour obtenir un nombre total de 4 caractères.

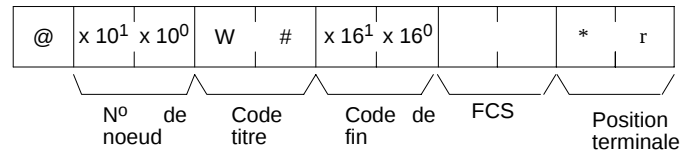
6-3-18 SV CHANGE 1 -- W#

Recherche la première occurrence des instructions TIM, TIMH(15), CNT et CNTR(12) dans le programme utilisateur et change la SV en une nouvelle constante spécifiée dans le deuxième canal de l'instruction. Le programme est examiné depuis le début et la réponse peut prendre environ 10 secondes.

Format de commande



Format de réponse



Paramètres

Dénomination, numéro de TC (Commande)

Spécifiez l'instruction de changement de SV dans la "Dénomination". Effectuez cette sélection en 4 caractères. Dans "Numéro de TC", spécifiez le TC utilisé pour l'instruction.

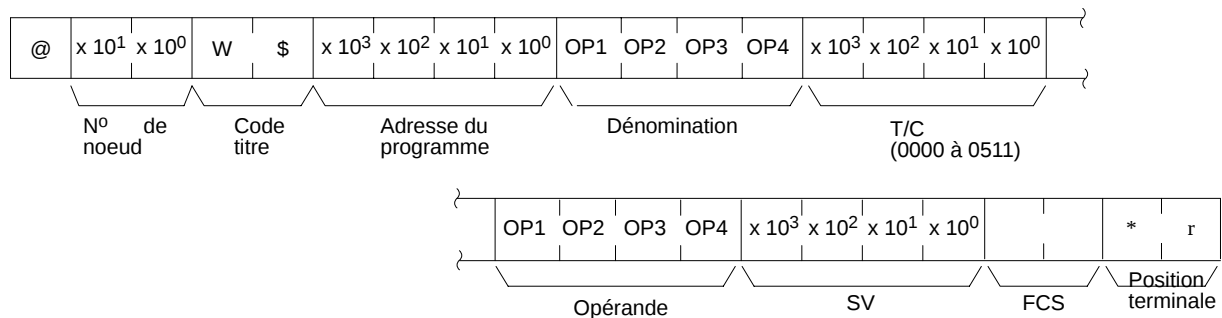
Instruction				Classification	Gamme de TC
OP1	OP2	OP3	OP4		
T	I	M	(S)	TIMER	0000 à 0511
T	I	M	H	HIGH-SPEED TIMER	
C	N	T	(S)	COUNTER	
C	N	T	R	REVERSIBLE COUNTER	

(S): espace

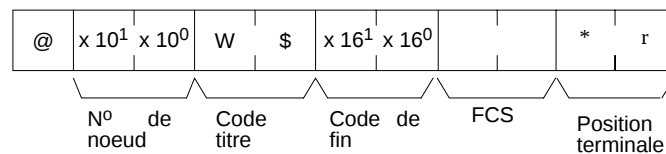
6-3-19 SV CHANGE 2 -- W\$

Change le contenu du deuxième canal de l'instruction TIM, TIMH(15), CNT ou CNTR(12) dans l'adresse de programme spécifiée dans le programme utilisateur ; cela ne peut être fait qu'avec un programme inférieur à 10K.

Format de commande



Format de réponse



Paramètres

Dénomination, numéro de TC (Commande)

Spécifiez l'instruction de changement de SV dans la "Dénomination". Effectuez cette sélection en 4 caractères. Dans "Numéro de TC", spécifiez le TC utilisé pour l'instruction.

Instruction				Classification	Gamme de TC
OP1	OP2	OP3	OP4		
T	I	M	(S)	TIMER	0000 à 0511
T	I	M	H	HIGH-SPEED TIMER	
C	N	T	(S)	COUNTER	
C	N	T	R	REVERSIBLE COUNTER	

(S): espace

Opérande, SV (Réponse)

L'indication de classification de la SV est spécifiée dans "Opérande", (4 caractères) et l'adresse de canal où est stocké la SV ou la constante de la SV sont spécifiées dans "SV."

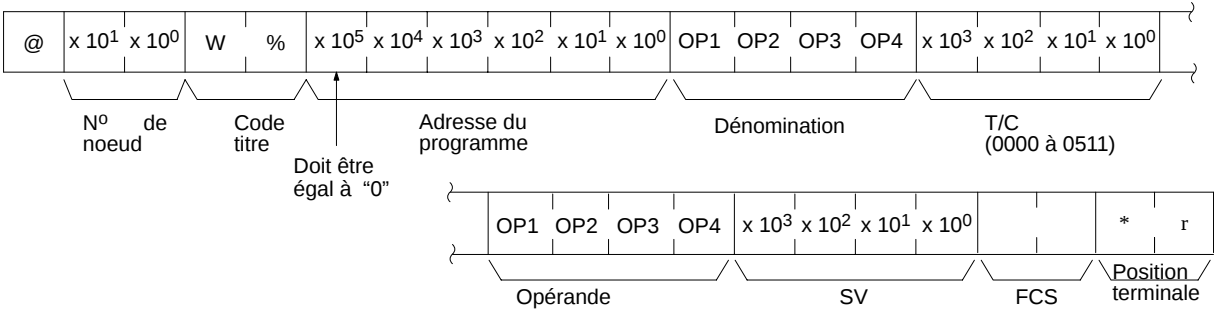
Opérande				Classification	Constante ou adresse de canal
OP1	OP2	OP3	OP4		
C	I	O	(S)	IR ou SR	0000 à 0252
L	R	(S)	(S)	LR	0000 à 0063
H	R	(S)	(S)	HR	0000 à 0099
A	R	(S)	(S)	AR	0000 à 0027
D	M	(S)	(S)	DM	0000 à 6655
D	M	*	(S)	DM (indirect)	0000 à 6655
C	O	N	(S)	Constante	0000 à 9999

(S): espace

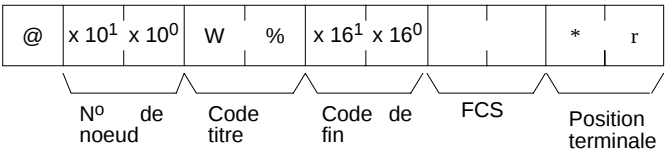
6-3-20 SV CHANGE 3 -- W%

Change le contenu du deuxième canal de l'instruction TIM, TIMH(15), CNT ou CNTR(12) dans l'adresse de programme spécifiée dans le programme utilisateur ; avec cette commande, l'adresse de programme peut être spécifiée pour un programme supérieur à 10K.

Format de commande



Format de réponse



Paramètres

Dénomination, numéro de TC (Commande)

Spécifiez l'instruction de changement de SV dans la "Dénomination". Effectuez cette sélection en 4 caractères. Dans "Numéro de TC", spécifiez le TC utilisé pour l'instruction.

Instruction				Classification	Gamme de TC
OP1	OP2	OP3	OP4		
T	I	M	(S)	TIMER	0000 à 0511
T	I	M	H	HIGH-SPEED TIMER	
C	N	T	(S)	COUNTER	
C	N	T	R	REVERSIBLE COUNTER	

(S): espace

Opérande, SV (Réponse)

L'indication de classification de la SV est spécifiée dans "Opérande" (4 caractères) et l'adresse de canal où est stocké la SV ou la constante de la SV sont stockées dans "SV."

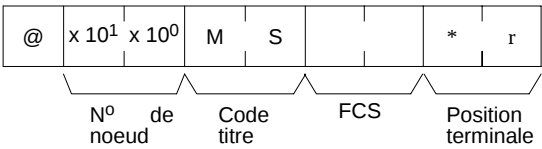
Opérande				Classification	Constante ou adresse de canal
OP1	OP2	OP3	OP4		
C	I	O	(S)	IR ou SR	0000 à 0252
L	R	(S)	(S)	LR	0000 à 0063
H	R	(S)	(S)	HR	0000 à 0099
A	R	(S)	(S)	AR	0000 à 0027
D	M	(S)	(S)	DM	0000 à 6655
D	M	*	(S)	DM (indirect)	0000 à 6655
C	O	N	(S)	Constante	0000 à 9999

(S): espace

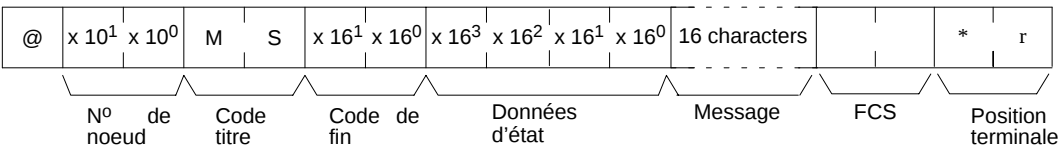
6-3-21 STATUS READ -- MS

Lit les conditions de fonctionnement de l'API.

Format de commande



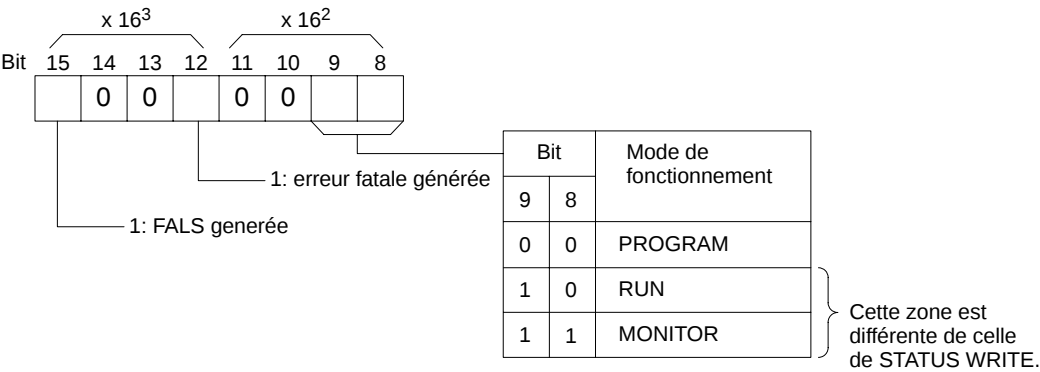
Format de réponse

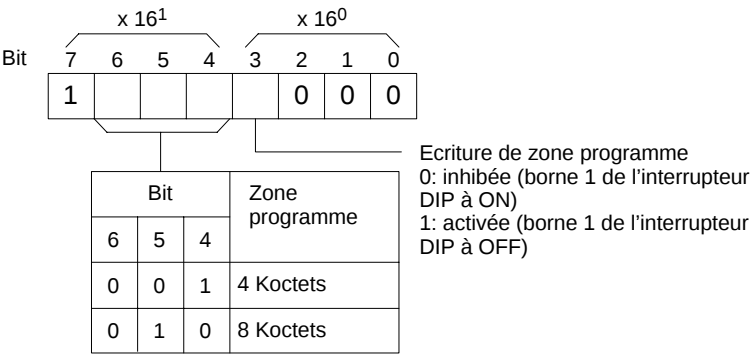


Paramètres

Données d'état, Message (Réponse)

Les "données d'état" sont à 4 chiffres (2 octets) hexadécimaux. L'octet d'extrême gauche indique le mode de fonctionnement de l'UC et l'octet d'extrême droite indique la taille de la zone programme.



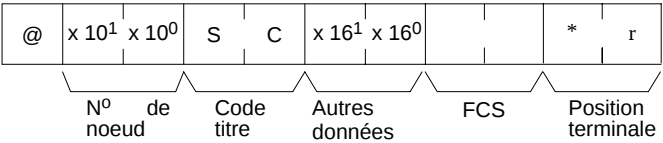


“Message” indique le numéro de FAL/FALS généré au point d'exécution de la commande. En cas d'absence de message, cette indication est omise.

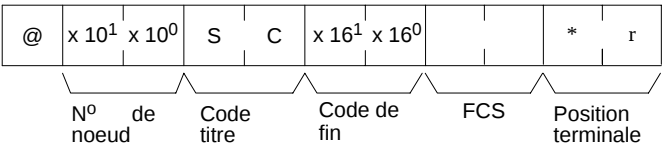
6-3-22 STATUS WRITE -- SC

Change le mode de fonctionnement de l'API.

Format de commande



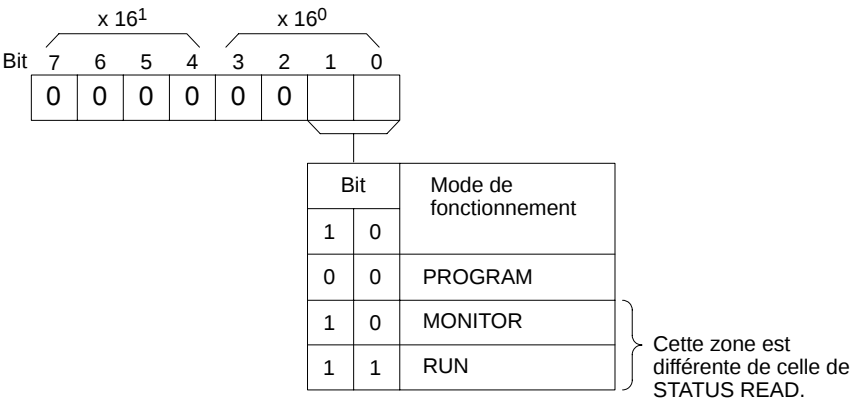
Format de réponse



Paramètres

Autres données (Commande)

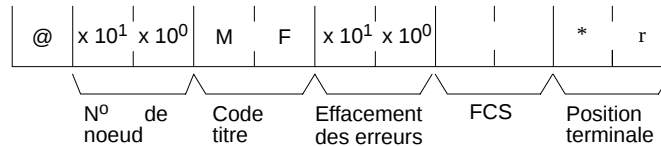
Les “autres données” comprennent 2 chiffres (1 octet) hexadécimaux. Avec les deux bits d'extrême gauche, spécifiez le mode de fonctionnement de l'API. Sélectionnez tous les bits restant à “0.”



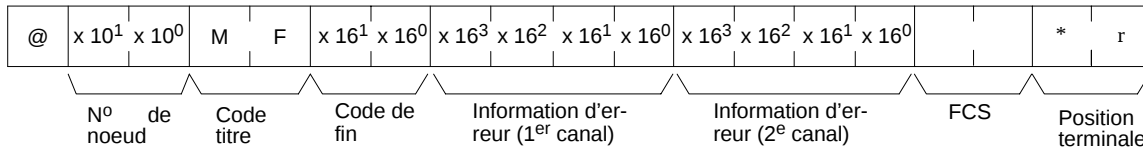
6-3-23 ERROR READ -- MF

Lit et efface les erreurs de l'API ; vérifie également que les erreurs précédentes ont été effacées.

Format de commande



Format de réponse



Paramètres

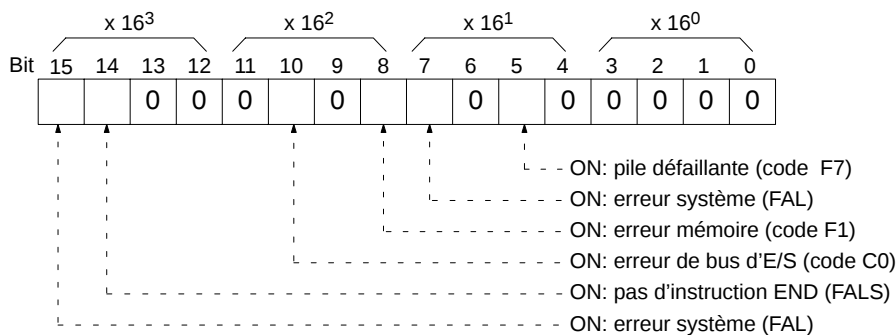
Effacement d'erreur (Commande)

Spécifiez 01 pour effacer les erreurs et 00 pour ne pas les effacer (BCD). Les erreurs fatales ne peuvent être effacées que lorsque l'API est en mode PROGRAM.

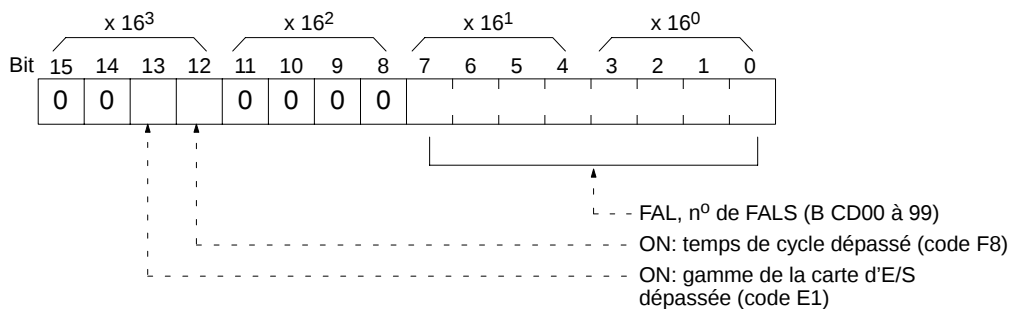
Information d'erreur (Réponse)

L'information d'erreur est en deux canaux.

1^{er} canal



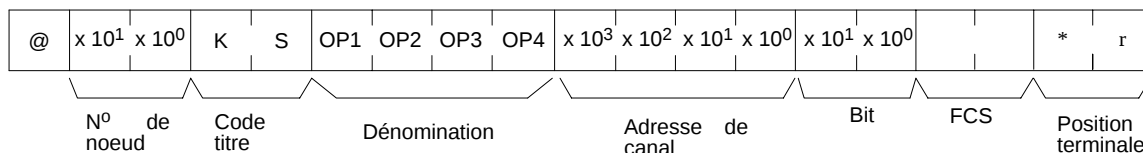
2^e canal



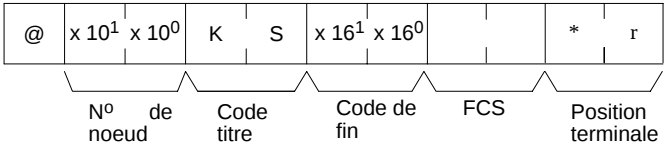
6-3-24 FORCED SET -- KS

Force à l'activation les bits des zones IR, SR, LR, HR, AR ou TC. Une fois un bit activé ou désactivé, cet état est retenu jusqu'à la transmission de FORCED SET/RESET CANCEL (KC).

Format de commande



Format de réponse



Paramètres

Dénomination, adresse de canal, bit (commande)

Dans "Dénomination", spécifiez la zone (IR, SR, LR, HR, AR ou TC) qui doit être forcée à l'activation (4 caractères). Dans "Adresse de canal", spécifiez l'adresse du canal et dans "Bit" le numéro du bit qui doit être forcé à l'activation.

Dénomination				Classification	Gamme de l'adresse de canal	Bit
OP1	OP2	OP3	OP4			
C	I	O	(S)	IR ou SR	0000 à 0252	00 à 15 (décimal)
L	R	(S)	(S)	LR	0000 à 0063	
H	R	(S)	(S)	HR	0000 à 0099	
A	R	(S)	(S)	AR	0000 à 0027	
T	I	M	(S)	Drapeau de fin (tempo.)	0000 à 0511	Constamment à 00
T	I	M	H	Drapeau de fin (tempo. rapide)		
C	N	T	(S)	Drapeau de fin (compteur)		
C	N	T	R	Drapeau de fin (compteur réversible)		

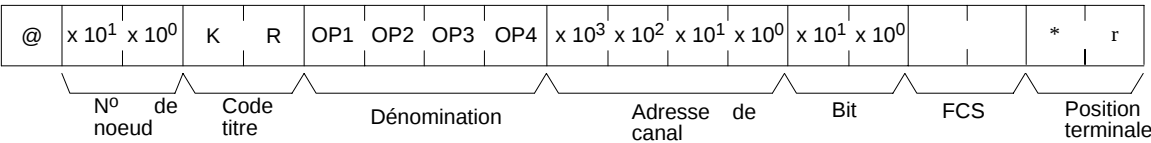
(S): espace

Rem. : la zone spécifiée dans "Dénomination" doit comporter 4 caractères. Remplissez les blancs avec des espaces pour obtenir un nombre total de 4 caractères.

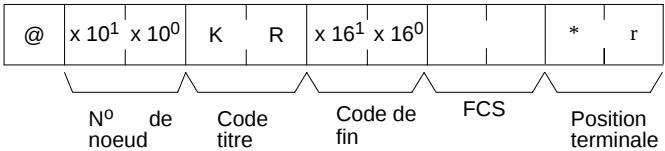
6-3-25 FORCED RESET -- KR

Force un bit à la désactivation dans une zone IR, SR, LR, HR, AR ou TC. Une fois un bit forcé à l'activation ou à la désactivation, cet état est retenu jusqu'à la transmission de FORCED SET/RESET CANCEL (KC).

Format de commande



Format de réponse



Paramètres

Dénomination, adresse de canal, bit (commande)

Dans "Dénomination", spécifiez la zone (IR, SR, LR, HR, AR ou TC) qui doit être forcée à la désactivation (4 caractères). Dans "Adresse de canal", spécifiez l'adresse du canal et dans "Bit" le numéro du bit qui doit être forcé à la désactivation.

Dénomination				Classification	Gamme de sélection de l'adresse de canal	Bit
OP1	OP2	OP3	OP4			
C	I	O	(S)	IR ou SR	0000 à 0252	00 à 15 (décimal)
L	R	(S)	(S)	LR	0000 à 0063	
H	R	(S)	(S)	HR	0000 à 0099	
A	R	(S)	(S)	AR	0000 à 0027	
T	I	M	(S)	Drapeau de fin (tempo.)	0000 à 0511	Constamment à 00
T	I	M	H	Drapeau de fin (tempo. rapide)		
C	N	T	(S)	Drapeau de fin (compteur)		
C	N	T	R	Drapeau de fin (compteur réversible)		

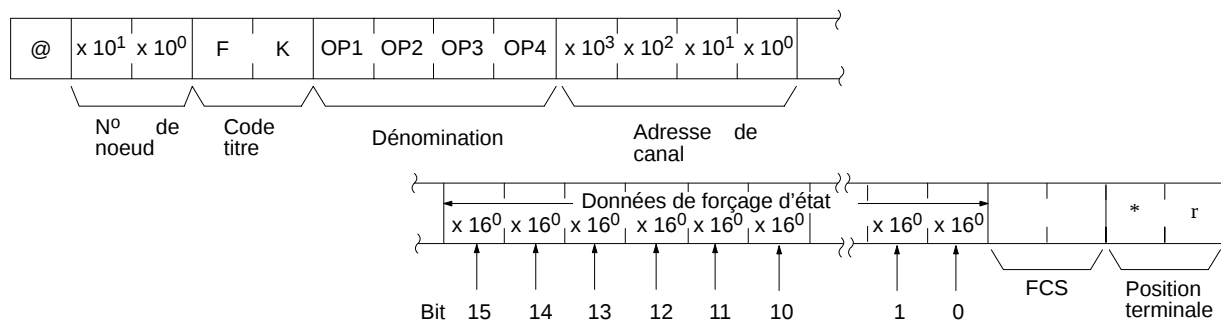
(S): espace

Rem. : la zone spécifiée dans "Dénomination" doit comporter 4 caractères. Remplissez les blancs avec des espaces pour obtenir un nombre total de 4 caractères.

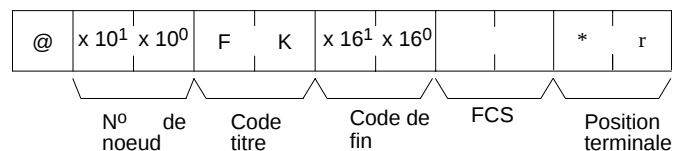
6-3-26 MULTIPLE FORCED SET/RESET -- FK

Force à l'activation ou à la désactivation ou efface l'état des bits d'un canal de la zone IR, SR, LR, HR, AR ou TC.

Format de commande



Format de réponse



Paramètres

Dénomination, adresse de canal (commande)

Dans "Dénomination", spécifiez la zone (IR, SR, LR, HR, AR ou TC) qui doit être forcée à l'activation ou à la désactivation (4 caractères). Dans "Adresse de canal", spécifiez l'adresse du canal qui doit être forcé à l'activation ou à la désactivation.

Dénomination				Classification	Gamme de l'adresse de canal	Bit
OP1	OP2	OP3	OP4			
C	I	O	(S)	IR or SR	0000 à 0252	00 à 15 (décimal)
L	R	(S)	(S)	LR	0000 à 0063	
H	R	(S)	(S)	HR	0000 à 0099	
A	R	(S)	(S)	AR	0000 à 0027	
T	I	M	(S)	Drapeau de fin (tempo.)	0000 à 0511	Constamment à 00
T	I	M	H	Drapeau de fin (tempo. rapide)		
C	N	T	(S)	Drapeau de fin (compteur)		
C	N	T	R	Drapeau de fin (compteur réversible)		

(S): espace

Données de forçage à l'activation, à la désactivation, à l'effacement (Commande)

On utilise un chiffre hexadécimal séparé pour spécifier le processus désiré pour chaque bit dans le canal spécifié (bits 00 à 5). Les bits simplement activés ou désactivés peuvent changer d'état à l'exécution suivante du programme mais les bits dont l'activation ou la désactivation est forcée maintiennent leur état jusqu'à effacement. Si la spécification de "Dénomination" est une tempo. ou un compteur, l'état du drapeau de fin peut être forcé à l'activation ou à la désactivation à l'aide du bit 15, et les autres bits seront ignorés. Seuls les forçages à l'activation ou à la désactivation sont possibles pour les tempo. ou compteurs.

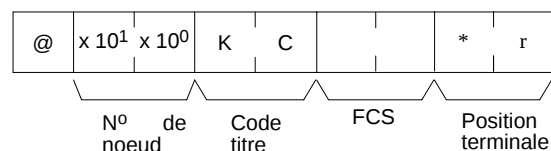
Bits 00 à 15	Sélection
00	Etat du bit inchangé
02	Désactivation
03	Activation
04	Désactivation forcée
05	Activation forcée
08	Effacement de l'état d'activation ou de désactivation forcée

Rem. : la zone spécifiée dans "Dénomination" doit comporter 4 caractères. Remplissez les blancs avec des espaces pour obtenir un nombre total de 4 caractères.

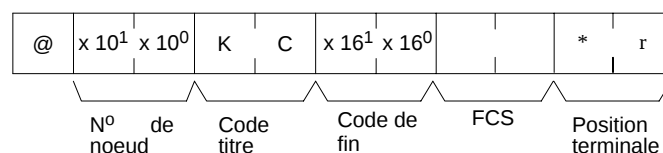
6-3-27 FORCED SET/RESET CANCEL -- KC

Efface tous les bits d'activation ou de désactivation forcée (y compris ceux activés par FORCED SET, FORCED RESET et MULTIPLE FORCED SET/RESET). Si des bits multiples ont été sélectionnés, l'état forcé est effacé pour tous. Il est impossible d'effacer des bits un par un avec KC.

Format de commande



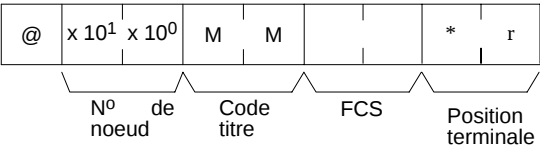
Format de réponse



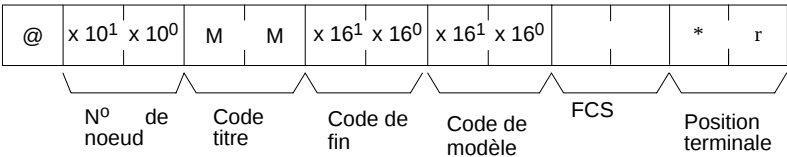
6-3-28 PC MODEL READ -- MM

Lit le modèle de l'API.

Format de commande



Format de réponse



Paramètres

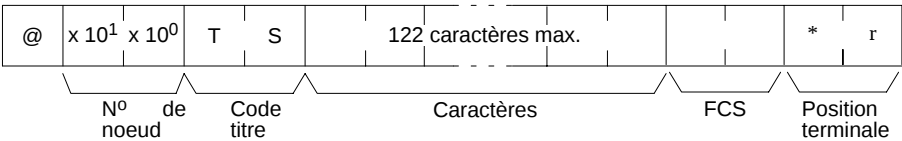
Code du modèle
"Code du modèle" indique le modèle de l'API en 2 chiffres hexadécimaux.

Code du modèle	Modèle
01	C250
02	C500
03	C120
0E	C2000
10	C1000H
11	C2000H/CQM1
12	C20H/C28H/C40H/C200H
20	CV500
21	CV1000
22	CV2000
40	CVM1-CPU01-E
41	CVM1-CPU11-E

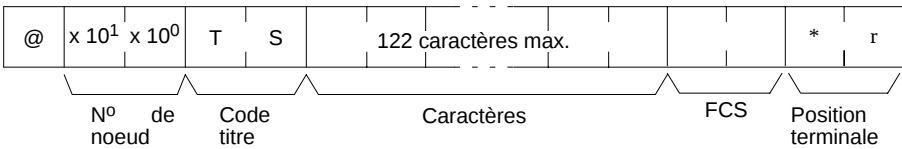
6-3-29 TEST-- TS

Retourne en l'état un bloc de données transmis par l'ordinateur.

Format de commande



Format de réponse



Paramètres

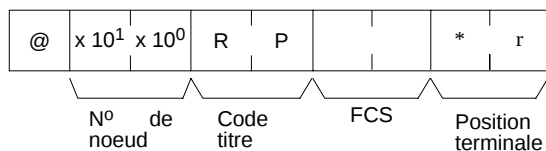
Caractères (Commande, Réponse)
Pour la commande, cette sélection spécifie tout caractère autre que le retour

chariot (CHR\$(13)). Pour la réponse, un caractère identique à celui spécifié par la commande est renvoyé en l'état si l'essai est concluant.

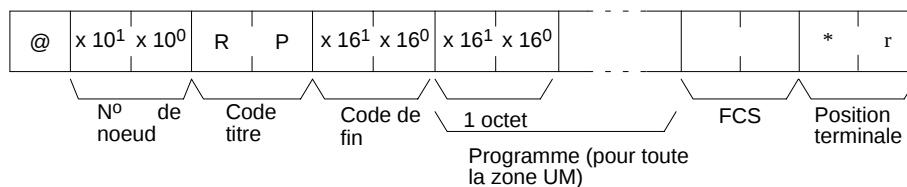
6-3-30 PROGRAM READ -- RP

Lit le contenu du programme utilisateur de l'API en langage machine (code objet). Le contenu est lu en un bloc, du début à la fin.

Format de commande



Format de réponse



Paramètres

Programme (Réponse)

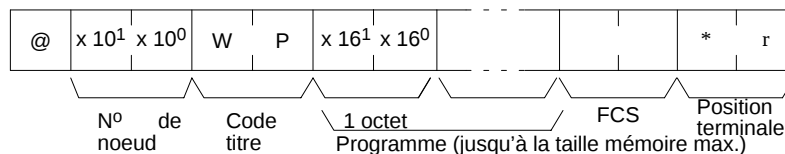
Le programme est lu à partir de toute la zone programme.

Rem. : pour arrêter le fonctionnement de l'appareil lorsqu'il est en cours, exécutez la commande ABORT (XZ).

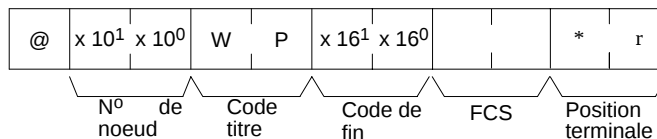
6-3-31 PROGRAM WRITE -- WP

Inscrit dans la zone du programme utilisateur API le programme en langage machine (code objet) transmis à partir de l'ordinateur. Le contenu est écrit en bloc à partir du début.

Format de commande



Format de réponse



Paramètres

Programme (Commande)

Données de programme jusqu'à la taille mémoire maximum.

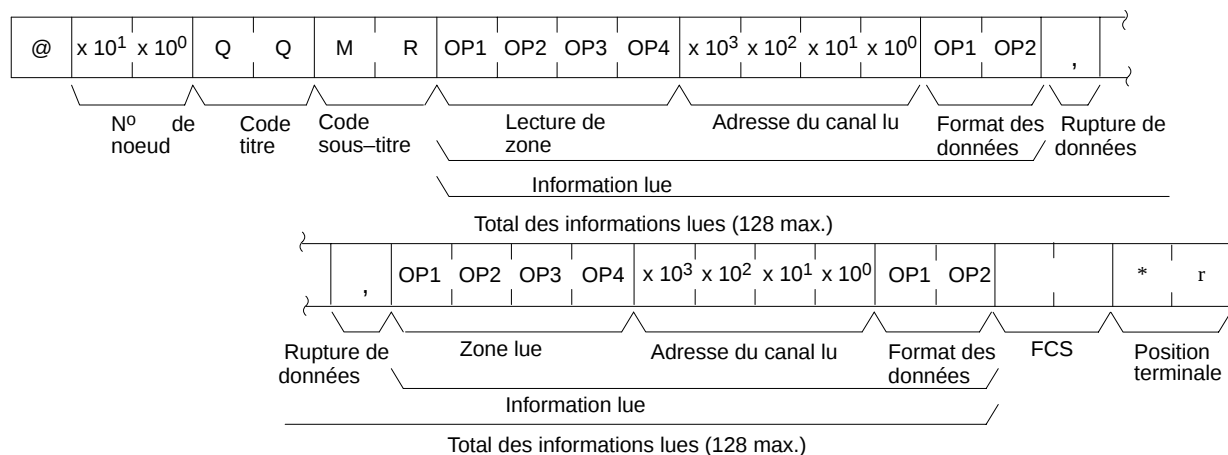
6-3-32 COMPOUND COMMAND -- QQ

Enregistre dans l'API tous les bits, canaux et TC qui doivent être lus et en lit l'état en lot.

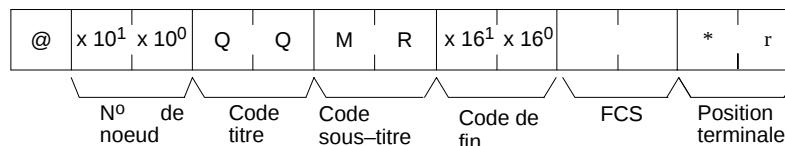
Enregistrement des informations lues

Enregistrement de tous les bits, canaux et TC qui doivent être lus.

Format de commande



Format de réponse



Paramètres

Lecture de zone (Commande)

Spécifiez en 4 caractères le code de la zone à lire. Les codes qui peuvent être spécifiés figurent dans le tableau ci-dessous.

Adresse de canal à lire, format des données (Commande)

Selon la zone et le type de données à lire, l'information lue correspond à celles du tableau suivant. La "Donnée à lire" est spécifiée en 4 caractères BCD et le format des données en 2 chiffres BCD.

Classification de zone	Donnée à lire	Zone à lire	Canal à lire	Data format
IR ou SR	Bit	C I O (S)	0000 à 0255	00 à 15 (decimal)
	Canal			"CH"
LR	Bit	L R (S) (S)	0000 à 0063	00 à 15 (decimal)
	Canal			"CH"
HR	Bit	H R (S) (S)	0000 à 0099	00 à 15 (decimal)
	Canal			"CH"
AR	Bit	A R (S) (S)	0000 à 0027	00 à 15 (decimal)
	Bit			"CH"
Temo.	Drapeau de fin	T I M (S)	0000 à 0511	2 caractères différents de CH
	PV			"CH"
Compteur rapide	Drapeau de fin	T I M H	0000 à 0511	2 caractères différents de CH
	PV			"CH"
Compteur	Drapeau de fin	C N T (S)	0000 à 0511	2 caractères différents de CH
	PV			"CH"
Compteur réversible	Drapeau de fin	C N T R	0000 à 0511	2 caractères différents de CH
	PV			"CH"
DM	Canal	D M (S) (S)	0000 à 6655	2 caractères

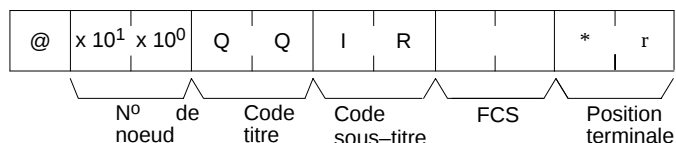
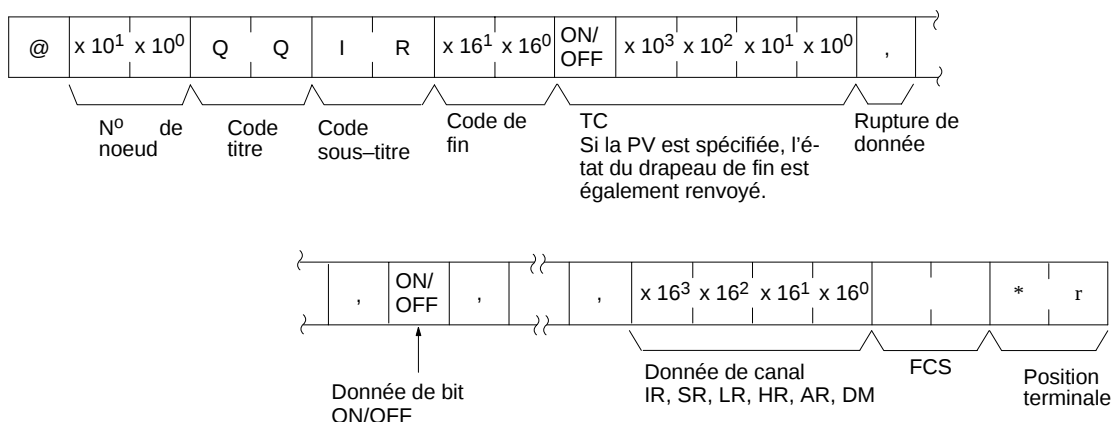
(S): espace

Rupture de données (Commande)

L'information à lire est spécifiée élément par élément avec des codes de rupture pour les séparer (,). Le nombre maximum d'éléments pouvant être spécifiés est de 128 (lorsque la PV d'un TC est spécifiée, toutefois, l'état du drapeau de fin est également retourné et on doit alors compter 2 éléments).

Lecture par lot

L'état des bits, canaux et TC est lu en lot selon l'information à lire enregistrée par QQ.

Format de commande**Format de réponse****Paramètres****Donnée lue (Réponse)**

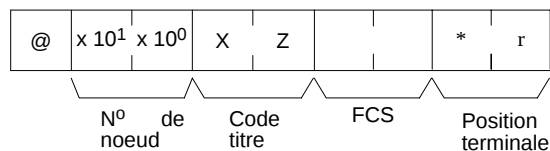
La donnée lue est renvoyée selon le format de la donnée et l'ordre dans lequel l'information à lire a été enregistrée avec QQ. Si le drapeau de fin a été spécifié, la donnée de bit ON ou OFF est renvoyée. Si le canal a été spécifié, alors les données de canal sont renvoyées. Si "PV" a été spécifiée pour les TC, alors la PV est renvoyée à la suite du drapeau de fin.

Rupture de donnée (Réponse)

Le code de rupture (,) est renvoyé entre les parties de données qui sont lues.

6-3-33 ABORT -- XZ

Arrête le fonctionnement de l'ordinateur en cours et permet la réception de la donnée suivante. La commande ABORT ne reçoit pas de réponse.

Format de commande**6-3-34 INITIALIZE -- ****

Initialise la procédure du contrôle de transmission de tous les API connectés à l'ordinateur. La commande INITIALIZE n'utilise pas de numéros de noeud ou de FCS et ne reçoit pas de réponse.

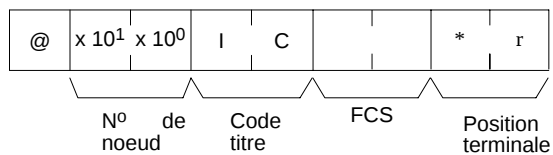
Format de commande



6-3-35 Commande indéfinie -- IC

Cette réponse est renvoyée si le code titre d'une commande ne peut être décodé. Vérifiez le code titre.

Format de commande



Chapitre 7

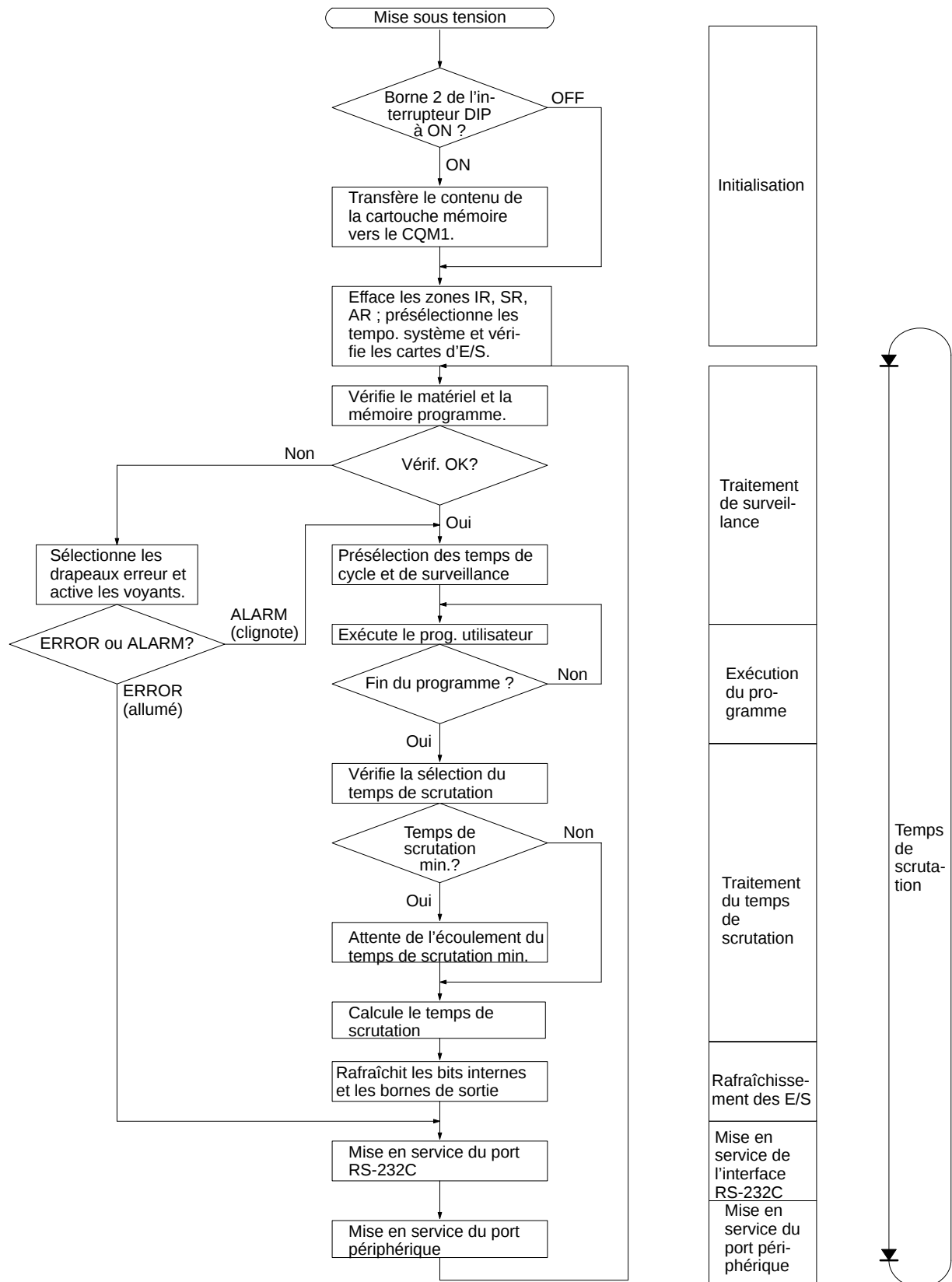
Fonctionnement et temps de traitement du CQM1

Ce chapitre expose le traitement interne du CQM1 et les temps nécessaires aux exécutions ; il vous servira à évaluer les temps précis de traitement de l'API.

7-1	Fonctionnement du CQM1	330
7-2	Temps de scrutation et temps de réponse des E/S	331
7-2-1	Temps de scrutation	331
7-2-2	Temps d'exécution des instructions	333
7-2-3	Temps de réponse des E/S	345
7-2-4	Temps de réponse des E/S en liaison point par point	346
7-2-5	Temps de traitement des interruptions	348

7-1 Fonctionnement du CQM1

Fonctionnement interne du CQM1 :



Modes de rafraîchissement des E/S

Le fonctionnement du rafraîchissement du CQM1 se divise en deux modes. La première méthode, le rafraîchissement des entrées, implique la lecture de l'état ON/OFF des points d'entrée sur les bits d'entrée. La deuxième, le rafraîchissement des sorties, implique la lecture de l'état ON/OFF après l'exécution du programme sur les points de sortie.

E/S	Mode de rafraîchissement des E/S	Fonction
Entrée	Rafraîchissement cyclique	Exécuté à un temps sélectionné une fois par scrutation.
	Rafraîchissement des entrées d'interruption	Effectué avant l'exécution du sous-programme de traitement des interruptions à chaque interruption d'entrée, interruption de tempo. cyclique ou interruption de compteur rapide (rafraîchissement cyclique également effectué).
Sortie	Rafraîchissement cyclique	Exécuté à un temps sélectionné une fois par scrutation.
	Rafraîchissement direct	Lorsqu'il existe une sortie à partir du programme utilisateur, ce point de sortie est immédiatement rafraîchi (rafraîchissement cyclique également effectué).

Etat initial du rafraîchissement des E/S du CQM1 :

Entrée : rafraîchissement cyclique uniquement

Sortie : rafraîchissement cyclique uniquement

Le rafraîchissement cyclique doit être effectué pour les entrées et pour les sorties. Si le rafraîchissement des entrées doit être exécuté au moment où se trouvent des interruptions, sélectionnez alors la gamme de rafraîchissement des entrées dans la configuration API (DM 6630 à DM 6638). L'arrêt du rafraîchissement direct peut être sélectionné dans le DM 6639 de la configuration API.

En plus des 2 méthodes décrites ci-dessus, il est possible d'exécuter des rafraîchissements d'E/S dans le programme par IORF(97).

7-2 Temps de scrutation et temps de réponse des E/S

7-2-1 Temps de scrutation

Processus	Description	Temps de traitement
Surveillance	Sélection de la tempo. chien de garde, de la vérification du bus d'E/S, de la vérification UM, de l'horloge de rafraîchissement, des bits de rafraîchissement attribués à de nouvelles fonctions, etc.	0,8 ms (0,9 ms lorsqu'une cartouche mémoire équipée d'une horloge est montée.) Ajoutez 0,1 ms pour les UC CQM1-CPU4 [-E.]
Exécution du programme	Exécution du programme utilisateur	Temps total pour l'exécution des instructions (varie selon le contenu du programme utilisateur).
Calcul du temps de cycle	Attente jusqu'au moment sélectionné lorsque le temps de cycle min. est sélectionné dans le DM 6619 de la configuration API.	Presque instantané, sauf pour le traitement de l'attente.
Rafraîchissement des E/S	Les informations d'entrée des cartes d'entrée sont lues sur les bits d'entrée. Les informations de sortie (résultat de l'exécution des programmes) sont écrites sur les bits de sortie.	Nombre de mots d'entrée x 0,01 ms
Mise en service du port RS-232C	Appareils connectés au port RS-232C mis en service.	5% max. du temps de cycle (cf Rem.)
Mise en service du port périphérique	Appareils connectés au port périphérique mis en service.	5% max. du temps de cycle (cf Rem.)

Rem. : les pourcentages peuvent être modifiés dans la configuration API (DM 6616, DM 6617).

Effets du temps de scrutation sur le fonctionnement du CQM1

Temps de scrutation	Conditions de fonctionnement
10 ms min.	TIMH(15) peut ne pas être très précise lorsque l'on utilise les TC 016 à 511 (fonctionnement normal en revanche pour les TC 000 à 015) (cf Rem. 1).
20 ms min.	La programmation utilisant le bit d'horloge de 0,02 seconde peut être imprécise (SR 25401).
100 ms min.	La programmation utilisant le bit d'horloge de 0,1 seconde (SR 25500) peut être imprécise. Une erreur CYCLE TIME OVER est générée (SR 25309 passe à ON) (cf Rem. 2).
120 ms min.	La SV du temps de surveillance FALS 9F est dépassée. Une erreur système (FALS 9F) est générée et le fonctionnement est arrêté (cf Rem. 3).
200 ms min.	La programmation utilisant le bit d'horloge de 0,2 seconde (SR 25501) peut être imprécise.

Rem. : Le nombre de tempo. subissant une interruption peut être sélectionné dans le DM 6629 de la configuration API. La sélection par défaut va des TC 00 aux TC 015.

La configuration API (DM 6655) peut être utilisée pour désactiver la détection de l'erreur CYCLE TIME OVER.

Le temps de surveillance du cycle FALS 9F peut être modifié par la configuration API (DM 6618).

Exemple de temps de scrutation

Dans cet exemple, le temps de cycle est calculé pour un CQM1 ayant 80 E/S. Les E/S sont configurées comme suit :

Entrées c.c. : 48 points (3 canaux)

Sorties de bit : 32 points (2 canaux)

Autres conditions de fonctionnement :

Programme utilisateur : 2000 instructions (configurées en instructions LD et OUT)

Horloge : aucune

Port RS-232C : non utilisé

Temps de scrutation : variable (pas de valeur minimum sélectionnée)

Rem. : le temps de traitement moyen pour une seule instruction du programme utilisateur est de 0,625 μ s.

Temps de scrutation :

Traitement	Méthode de calcul	Temps avec périphérique	Temps sans périphérique
Surveillance	Fixe	0,8 ms	0,8 ms
Exécution du programme	$0,625 * 2000$ (μ s)	1,25 ms	1,25 ms
Calcul du temps de scrutation	Négligeable	0 ms	0 ms
Rafraîchissement des E/S	$0,01 * 3 + 0,005 * 2$ (μ s)	0,04 ms	0,04 ms
Mise en service du port RS-232C	Non utilisé	0 ms	0 ms
Mise en service du port périphérique	Temps minimum	0,34 ms	0 ms
Temps de scrutation	(1) + (2) + (3) + (4) + (5) + (6)	2,43 ms	2,09 ms

Rem. : Le temps de scrutation peut être lu automatiquement à partir de l'API par un périphérique.

Les temps de scrutation maximum et en cours sont stockés dans les AR 26 et 27.

Le temps de scrutation peut varier avec les conditions de fonctionnement réelles et peut différer de la valeur calculée.

Les temps de service des ports RS-232C et périphériques sont de 0,34 ms minimum, 87 ms maximum.

7-2-2 Temps d'exécution des instructions

Voici la liste des temps d'exécution des instructions du CQM1. Les temps d'exécution maximum et minimum et leurs conditions sont données lorsque cela est jugé nécessaire. La mention "Mot" dans la colonne des conditions implique le contenu de tout canal, sauf des DM indirectement adressés. Pour ceux-ci, la mention "*DM" est portée, car ils provoquent des temps d'exécution plus longs.

Les temps d'exécution de la plupart des instructions dépendent de leur exécution à l'aide d'une condition ON ou OFF. Les instructions de schéma à relais OUT et OUT NOT sont des exceptions : elles demandent le même temps d'exécution, que la condition soit à ON ou à OFF. Le temps d'exécution OFF d'une instruction peut également varier selon les circonstances ; par exemple si elle se trouve dans un embranchement et que la condition de IL est à OFF, si elle est située entre JMP(04) et JME(05) et que la condition de JMP(04) est à OFF, ou si elle est remise à zéro par une condition OFF. "RSET", "IL" et "JMP" sont utilisées pour indiquer ces trois temps.

Instructions de base

Code	Mnémonique	Temps d'exécution ON (s)	Conditions (min. en haut ; max. en bas)	Temps d'exéc. OFF (s)		
				RSET	IL	JMP
---	LD	0,5	Toutes conditions			
---	LD NOT					
---	AND					
---	AND NOT					
---	OR					
---	OR NOT					
---	AND LD					
---	OR LD					
---	OUT	0,75	Sans sortie directe ou pour les opérandes différentes de IR 10000 à IR 11515 avec les sorties directes.			
---	OUT NOT					
---	SET	1,25	Sorties directes avec IR 10000 à IR 11515			
---	RSET					
---	TIM	1,5	Constante pour la SV	1,5	1,5	1,5
			*DM pour la SV	54,1	1,5	1,5
---	CNT	1,5	Constante pour la SV	1,5	1,5	1,5
			*DM pour la SV	51,6	1,5	1,5

Instructions spéciales

Code	Mnémonique	Temps d'exéc. ON (s)	Conditions (min. en haut ; max. en bas)	Temps d'exéc. OFF (s)
00	NOP	0,5	Toutes conditions	0,0
01	END	29		0,0
02	IL	12,3		10,9
03	ILC	11,3		11,3
04	JMP	18,3		11,9
05	JME	11,0		11,0
06	FAL	56,8		1,5
07	FALS	4,0		1,5
08	STEP	58,2		1,5
09	SNXT	25,0		

Code	Mnémoni- que	Temps d'exéc. ON (µs)	Conditions (min. en haut ; max. en bas)	Temps d'exéc. OFF (µs)		
10	SFT			Déca- lage	IL	JMP
		44,2	Avec registre à décalage d'un canal	43,2	15,0	15,0
		77,7	Avec registre à décalage de 10 canaux	68,5	15,0	15,0
		415,2	Avec registre à décalage de 100 canaux	322,0	15,0	15,0
11	KEEP	0,75	Sans sortie directe ou pour les opérandes dif- férentes de IR 10000 à IR 11515 avec les sor- ties directes.			
		1,25	Sorties directes avec IR 10000 à 11515			
12	CNTR			Déca- lage	IL	JMP
		53,0	Constante pour SV	33,1	20,7	20,7
		79,6	*DM pour SV			
13	DIFU	21,5	Toutes	RAZ	IL	JMP
				21,0	20,8	17,8
14	DIFD	20,8	Toutes	RAZ	IL	JMP
				20,8	20,6	17,6
15	TIMH			Déca- lage	IL	JMP
		36,5	Constante pour SV	54,7	53,0	27,7
		36,5	*DM pour SV	81,0	79,6	27,7
16	WSFT	44,7	Avec registre à décalage d'un canal	2,0		
		77,0	Avec registre à décalage de 10 canaux			
		2,25 ms	Avec registre à décalage de 1 024 canaux utilisant des *DM			
		13,05 ms	Avec registre à décalage de 6 144 utilisant des *DM			
20	CMP	26,7	En cas de comparaison d'une constante et d'un mot	2,0		
		29,5	En cas de comparaison de 2 mots			
		77,3	En cas de comparaison de 2 *DM			
21	MOV	23,5	En cas de transfert d'une constante vers un mot	2,0		
		26,3	En cas de déplacement d'un mot vers un autre mot			
		72,7	En cas de transfert d'un *DM vers un autre *DM			
22	MVN	23,7	En cas de transfert d'une constante vers un mot	2,0		
		26,5	En cas de déplacement d'un mot vers un autre mot			
		72,6	En cas de transfert d'un *DM vers un autre *DM			
23	BIN	50,4	En cas de conversion d'un mot à un autre mot	2,0		
		96,0	En cas de conversion d'un *DM vers un autre *DM			
24	BCD	47,7	En cas de conversion d'un mot à un autre mot	2,0		
		93,3	En cas de conversion d'un *DM à un autre *DM			
25	ASL	24,0	En cas de décalage de mot	1,5		
		45,8	En cas de décalage de *DM			

Code	Mnémoni- que	Temps d'exéc. ON (μ s)	Conditions (min. en haut ; max. en bas)	Temps d'exéc. OFF (μ s)
26	ASR	24,0	En cas de décalage de mot	1,5
		45,8	En cas de décalage de *DM	
27	ROL	24,7	En cas de rotation de mot	1,5
		46,6	En cas de rotation de *DM	
28	ROR	24,7	En cas de rotation de mot	1,5
		46,6	En cas de rotation de *DM	
29	COM	25,9	En cas d'inversion d'un mot	1,5
		48,3	En cas d'inversion d'un *DM	
30	ADD	49,9	Constante + mot A mot	2,5
		53,1	Mot + mot A mot	
		122,1	*DM + *DM A *DM	
31	SUB	49,9	Constante – mot A mot	2,5
		53,1	Mot – mot A mot	
		122,1	*DM – *DM A *DM	
32	MUL	73,7	Constante * mot A mot	2,5
		77,0	Mot * mot A mot	
		144,5	*DM * *DM A *DM	
33	DIV	72,2	Mot ÷ constant A mot	2,5
		75,4	Mot ÷ mot A mot	
		143,0	*DM ÷ *DM A *DM	
34	ANDW	41,9	Constante $\&$ mot A mot	2,5
		45,1	Mot $\&$ mot A mot	
		114,1	*DM $\&$ *DM A *DM	
35	ORW	41,9	Constante V mot A mot	2,5
		45,1	Mot V mot A mot	
		114,1	*DM V *DM A *DM	
36	XORW	41,9	Constante $\vee$ mot A mot	2,5
		45,2	Mot $\vee$ mot A mot	
		114,1	*DM $\vee$ *DM A *DM	
37	XNRW	41,9	Constante ∇ mot A mot	2,5
		45,1	Mot ∇ mot A mot	
		114,1	*DM ∇ *DM A *DM	
38	INC	27,8	En cas de décrémentation d'un mot	1,5
		50,1	En cas de décrémentation d'un *DM	
39	DEC	28,4	En cas de décrémentation d'un mot	1,5
		50,8	En cas de décrémentation d'un *DM	
40	STC	12,0	Toutes	1,0
41	CLC	12,0		1,0
45	TRSM	28,8		1,0
46	MSG	24,6	En cas de message dans les mots	1,5
		48,4	En cas de messages dans les *DM	
50	ADB	53,4	Constante + mot A mot	2,5
		56,6	Mot + mot A mot	
		125,6	*DM + *DM A *DM	

Code	Mnémoni- que	Temps d'exéc. ON (µs)	Conditions (min. en haut ; max. en bas)	Temps d'exéc. OFF (µs)
51	SBB	53,4	Constante – mot A mot	2,5
		56,6	Mot – mot A mot	
		125,6	*DM – *DM A *DM	
52	MLB	45,7	Constante * mot A mot	2,5
		48,9	Mot * mot A mot	
		116,4	*DM * *DM A *DM	
53	DVB	46,7	Mot ÷ constante A mot	2,5
		49,9	Mot ÷ mot A mot	
		117,4	*DM ÷ *DM A *DM	
54	ADDL	59,3	Mot + mot A mot	2,5
		128,9	*DM + *DM A *DM	
55	SUBL	59,3	Mot – mot A mot	2,5
		128,9	*DM – *DM A *DM	
56	MULL	204,5	Mot * mot A mot	2,5
		271,2	*DM * *DM A *DM	
57	DIVL	205,9	Mot ÷ mot A mot	2,5
		272,6	*DM ÷ *DM A *DM	
58	BINL	76,0	Mot A mot	2,0
		120,6	*DM A *DM	
59	BCDL	60,9	Mot A mot	2,0
		105,6	*DM A *DM	
70	XFER	72,9	Lors du transfert d'une constante vers 1 mot	2,5
		76,1	Lors du transfert d'un canal vers un mot	
		2,90 ms	Lors du transfert de 1 024 mots utilisant des *DM	
		16,66 ms	Lors du transfert de 6 144 mots utilisant des *DM	
71	BSET	45,6	Lors de la sélection d'une constante sur 1 mot	2,5
		77,9	Lors de la sélection d'une constante de mot sur 10 mots	
		1,93 ms	Lors de la sélection de *DM sur 1 024 mots	
		10,95 ms	Lors de la sélection de *DM sur 6 144 mots	
72	ROOT	63,9	Calcul de mot A mot	2,0
		110,8	Calcul de *DM A *DM	
73	XCHG	40,9	Mot A mot	2,0
		85,5	*DM A *DM	
74	SLD	41,1	Décalage de 1 mot	2,0
		101,9	Décalage de 10 mots	
		5,49 ms	Décalage de 1024 mots utilisant des *DM	
		32,58 ms	Décalage de 6144 mots utilisant des *DM	
75	SRD	41,1	Décalage de 1 mot	2,0
		101,9	Décalage de 10 mots	
		5,49 ms	Décalage de 1 024 mots utilisant des *DM	
		32,57 ms	Décalage de 6 144 mots utilisant des *DM	
76	MLPX	59,1	En cas de décodage de mot à mot	2,5
		136,4	En cas de décodage de *DM à *DM	

Code	Mnémoni- que	Temps d'exéc. ON (μ s)	Conditions (min. en haut ; max. en bas)	Temps d'exéc. OFF (μ s)
77	DMPX	45,1	En cas de codage de mot à mot	2,5
		120,6	En cas de codage de *DM à *DM	
78	SDEC	60,6	En cas de décodage de mot à mot	2,5
		138,5	En cas de décodage de *DM à *DM	
80	DIST	66,0	Lors de la sélection d'une constante sur un mot + un mot	2,5
		69,3	Lors de la sélection d'un mot sur un mot + un mot	
		144,3	Lors de la sélection d'un *DM sur un *DM + *DM	
		101,0	Lors de la sélection d'une constante sur un empilement	
		104,3	Lors de la sélection d'un mot sur un empilement	
		177,8	Lors de la sélection d'un *DM sur un empilement via un *DM	
81	COLL	65,1	Lors de la sélection d'une constante + un mot sur un mot	2,5
		68,3	Lors de la sélection d'un mot + un mot sur un mot	
		140,1	Lors de la sélection d'un *DM + *DM sur un *DM	
		61,1	Lors de la sélection d'un mot + une constante sur un empilement FIFO	
		64,3	Lors de la sélection d'un mot + un mot sur un empilement FIFO	
		137,6	Lors de la sélection d'un *DM + *DM sur un empilement FIFO via un *DM	
		60,3	Lors de la sélection d'un mot + une constante sur un empilement LIFO	
		63,6	Lors de la sélection d'un mot + un mot sur un empilement LIFO	
		136,8	Lors de la sélection d'un *DM + un *DM sur un empilement LIFO via un *DM	
82	MOVB	46,4	En cas de déplacement d'une constante vers un mot	2,5
		54,9	En cas de déplacement d'un mot vers un mot	
		125,2	En cas de déplacement d'un *DM vers 1 *DM	
83	MOVD	40,7	En cas de déplacement d'une constante vers un mot	2,5
		49,2	En cas de déplacement d'un mot vers un mot	
		119,4	En cas de déplacement d'un *DM vers 1 *DM	
84	SFTR	57,4	Décalage d'un mot	2,5
		98,4	Décalage de 10 mots	
		2,26 ms	Décalage de 1 024 mots utilisant des *DM	
		12,90 ms	Décalage de 6 144 mots utilisant des *DM	
85	TCMP	95,8	Comparaison d'une constante et d'un tableau de sélection des mots	2,5
		98,8	Comparaison d'un mot et d'un tableau de sélection des mots	
		169,0	Comparaison d'un *DM et d'un tableau de sélection des *DM	

Code	Mnémoni- que	Temps d'exéc. ON (µs)	Conditions (min. en haut ; max. en bas)	Temps d'exéc. OFF (µs)
86	ASC	62,5	Mot A mot	2,5
		144,3	*DM A *DM	
91	SBS	41,4	Toutes	1,5
92	SBN	---		---
93	RET	39,0		1,5
97	IORF	37,7	Rafraîchissement de IR 000	2,0
		41,7	Rafraîchissement d'un canal d'entrée	
		46,9	Rafraîchissement d'un canal de sortie	
		112,4	Rafraîchissement de 8 canaux d'E/S	
99	MCRO	140,2	Avec des opérandes d'E/S sélectionnées par canaux	2,5
		188,1	Avec des opérandes d'E/S sélectionnées par *DM	

Instructions étendues

Code	Mnémoni- que	Temps d'exéc. ON (µs)	Conditions	Temps d'exéc. OFF (µs)
17	ASFT	62,7	Décalage d'un canal	2,5
		96,7	Décalage de 10 canaux	
		2,45 ms	Décalage de 1 024 canaux via un *DM	
		16,33 ms	Décalage de 6 144 canaux via un *DM	
18	TKY	81,1	Mot vers mot	2,5
		131,8	*DM vers *DM	
19	MCMP	123,9	Comparaison de mots	2,5
		195,3	Comparaison de *DM	
47	RXD	123,1	Entrée d'un 1 octet via un mot	2,5
		847,3	Entrée de 256 octets via un *DM	
48	TXD	105,1	Sortie de 1 octet via un mot (RS-232C)	2,5
		832,3	Sortie de 256 octets via *DM (RS-232C)	
		86,3	Sortie de 1 octet via un mot (liaison à l'ordinateur)	
		141,9	Sortie de 256 octets via *DM (liaison à l'ordinateur)	
60	CMPL	50,9	Comparaison de mots	2,5
		101,0	Comparaison de *DM	
61	INI	Compteur rapide 0 ou sortie d'impulsion à partir d'un bit de sortie :		2,5
		90,6	Début de comparaison via un mot	
		114,4	Début de comparaison via un *DM	
		72,1	Arrêt de comparaison via un mot	
		83,0	Arrêt de comparaison via un *DM	
		163,6	Changement de PV via un mot	
		182,2	Changement de PV via un *DM	
		56,4	Arrêt de sortie d'impulsion via un mot	
		80,2	Arrêt de sortie d'impulsion via un *DM	

Code	Mnémoni- que	Temps d'exéc. ON ()	Conditions	Temps d'exéc. OFF ()
		Compteurs rapides 1 et 2 ou sortie d'impulsion à partir des ports 1 et 2 :		
		296,8	Début de comparaison via un mot	
		324,3	Début de comparaison via un *DM	
		207,3	Arrêt de comparaison via un mot	
		232,8	Arrêt de comparaison via un *DM	
		468,3	Changement de PV via un mot	
		487,8	Changement de PV via un *DM	
		248,8	Arrêt de sortie d'impulsion via un mot	
		269,8	Arrêt de sortie d'impulsion via un *DM	
		Compteurs rapides absolus 1 et 2 :		
		296,3	Début de comparaison via un mot	
		316,8	Début de comparaison via un *DM	
		202,3	Arrêt de comparaison via un mot	
		226,3	Arrêt de comparaison via un *DM	
62	PRV	Compteur rapide 0 ou sortie d'impulsion à partir d'un bit de sortie :		2,5
		91,5	Désignation de sortie via un mot	
		117,4	Désignation de sortie via un *DM	
		Compteurs rapides 1 et 2 ou sortie d'impulsion à partir des ports 1 et 2 :		
		229,3	Désignation de sortie via un mot (lecture d'état)	
		249,3	Désignation de sortie via un *DM (lecture d'état)	
		229,8	Désignation de sortie via un mot (lecture des résultats de comparaison de gammes)	
		256,3	Désignation de sortie via un *DM (lecture des résultats de comparaison de gammes)	
		Compteurs rapides absolus 1 et 2 :		
		226,3	Désignation de sortie via un mot (lecture d'état)	
		253,3	Désignation de sortie via un *DM (lecture d'état)	
		227,8	Désignation de sortie via un mot (lecture des résultats de comparaison de gammes)	
		253,3	Désignation de sortie via un *DM (lecture des résultats de comparaison de gammes)	

Code	Mnémoni- que	Temps d'exéc. ON ()	Conditions	Temps d'exéc. OFF ()
63	CTBL	Compteur rapide 0 ou sortie d'impulsion à partir d'un bit de sortie :		2,5
		210,3	Tableau spécifié avec 1 spécification dans les mots et départ	
		233,8	Tableau spécifié avec 1 spécification dans les *DM et départ	
		1,31 ms	Tableau spécifié avec 16 spécifications dans les mots et départ	
		1,33 ms	Tableau spécifié avec 16 spécifications dans les *DM et départ	
		1,25 ms	Tableau de gammes en mots et départ	
		1,27 ms	Tableau de gammes en *DM et départ	
		170,8	Tableau spécifié avec 1 spécification dans les mots	
		194,3	Tableau spécifié avec 1 spécification dans les *DM	
		1,27 ms	Tableau spécifié avec 16 spécifications dans les mots	
		1,30 ms	Tableau spécifié avec 16 spécifications dans les *DM	
		1,09 ms	Tableau spécifié en mots	
		1,11 ms	Tableau spécifié en *DM	
		Compteurs rapides 1 et 2 ou sortie d'impulsion à partir des ports 1 et 2 :		
		692,8	Tableau spécifié avec 1 spécification dans les mots et départ	
		721,8	Tableau spécifié avec 1 spécification dans les *DM et départ	
		2,79 ms	Tableau spécifié avec 16 spécifications dans les mots et départ	
		2,81 ms	Tableau spécifié avec 16 spécifications dans les *DM et départ	
		2,26 ms	Tableau spécifié en mots et départ	
		2,27 ms	Tableau spécifié en *DM et départ	
		488,8	Tableau spécifié avec 1 spécification dans les mots	
		517,8	Tableau spécifié avec 1 spécification dans les *DM	
		2,57 ms	Tableau spécifié avec 16 spécifications dans les mots	
		2,61 ms	Tableau spécifié avec 16 spécifications dans les *DM	
		2,19 ms	Tableau de gammes en mots	
		2,21 ms	Tableau de gammes en *DM	
		Compteurs rapides absolus 1 et 2 :		
		600,8	Tableau spécifié avec 1 spécification dans les mots et départ	
		624,8	Tableau spécifié avec 1 spécification dans les *DM et départ	
		2,33 ms	Tableau spécifié avec 16 spécifications dans les mots et départ	
		2,34 ms	Tableau spécifié avec 16 spécifications dans les *DM et départ	
		1,47 ms	Tableau de gammes en mots et départ	
		1,50 ms	Tableau de gammes en *DM et départ	
		460,8	Tableau spécifié avec 1 spécification dans les mots	
		484,8	Tableau spécifié avec 1 spécification dans les *DM	
		2,02 ms	Tableau spécifié avec 16 spécifications dans les mots	
		2,04 ms	Tableau spécifié avec 16 spécifications dans les *DM	
		1,45 ms	Tableau de gammes en mots	
		1,47 ms	Tableau de gammes en *DM	

Code	Mnémoni- que	Temps d'exéc. ON ()	Conditions	Temps d'exéc. OFF ()
64	SPED	Sortie d'impulsion à partir d'un bit de sortie :		2,5
		118,4	Fréquence spécifiée par une constante	
		123,2	Fréquence spécifiée par un mot	
		146,8	Fréquence spécifiée par un *DM	
		Sortie d'impulsion à partir des ports 1 et 2 :		
		302,3	Fréquence spécifiée par une constante	
		310,3	Fréquence spécifiée par un mot	
		320,3	Fréquence spécifiée par un *DM	
65	PULS	Sortie d'impulsion à partir d'un bit de sortie :		2,5
		109,0	Nombre d'impulsions spécifié par un mot	
		137,8	Nombre d'impulsions spécifié par un *DM	
		Sortie d'impulsion à partir des ports 1 et 2 :		
		337,3	Nombre d'impulsions spécifié par un mot	
		360,3	Nombre d'impulsions spécifié par un *DM	
66	SCL	105,8	Désignation de mot	2,5
		180,5	Désignation de *DM	
67	BCNT	88,4	Comptage d'un mot	2,5
		49,32 ms	Comptage de 6,656 mots via *DM	
68	BCMP	140,0	Comparaison de constantes, résultats sur mot	2,5
		143,0	Comparaison de mots, résultats sur mot	
		194,7	Comparaison de *DM, résultats sur *DM	
69	STIM	36,8	Départ d'interruption impulsionnelle sélectionnée par mot	2,5
		73,8	Départ d'interruption impulsionnelle sélectionnée par *DM	
		37,3	Départ d'interruption programmée sélectionnée par mot	
		74,3	Départ d'interruption programmée sélectionnée par *DM	
		66,4	Lecture de tempo. sélectionnée par mot	
		113,6	Lecture de tempo. sélectionnée par *DM	
		35,3	Arrêt de tempo. sélectionné par mot	
		35,6	Arrêt de tempo. sélectionné par *DM	
87	DSW	70,3	Sortie CS de 4 chiffres sélectionnée par mot	2,5
		70,3	Sortie RD de 4 chiffres sélectionnée par mot	
		89,1	Entrée de donnée à 4 chiffres sélectionnée par mot	
		93,1	Sortie CS de 4 chiffres sélectionnée par *DM	
		93,1	Sortie RD de 4 chiffres sélectionnée par *DM	
		110,3	Entrée de donnée à 4 chiffres sélectionnée par *DM	
		74,7	Sortie CS de 8 chiffres sélectionnée par mot	
		75,1	Sortie RD de 8 chiffres sélectionnée par mot	
		105,5	Entrée de données à 8 chiffres sélectionnée par mot	
		103,5	Sortie CS de 8 chiffres sélectionnée par *DM	
		103,9	Sortie RD de 8 chiffres sélectionnée par *DM	
		131,5	Entrée de donnée à 8 chiffres sélectionnée par *DM	
88	7SEG	78,7	4 chiffres, désignation de mot	2,5
		102,6	4 chiffres, désignation de *DM	
		92,1	8 chiffres, désignation de mot	
		117,2	8 chiffres, désignation de *DM	

Code	Mnémoni- que	Temps d'exéc. ON ()	Conditions	Temps d'exéc. OFF ()
89	INT	53,0	Sélection de masque via un mot	2,5
		80,8	Sélection de masque via un *DM	
		49,9	Effacement d'interruption via un mot	
		73,2	Effacement d'interruption via un *DM	
		50,7	Lecture d'état de masque via un mot	
		71,9	Lecture d'état de masque via un *DM	
		64,8	Changement de la SV du compteur via un mot	
		88,1	Changement de la SV du compteur via un *DM	
		27,5	Toutes les interruptions sont masquées via un mot	
		27,5	Toutes les interruptions sont masquées via un *DM	
		28,5	Effacement de toutes les interruptions via un mot	
		28,5	Effacement de toutes les interruptions via un *DM	
	HKY	71,5	Sortie de mot vers mot	2,5
		100,3	Sortie de *DM vers *DM	
		81,5	Entrée de mot vers mot	
		109,5	Entrée de *DM vers *DM	
	FPD	171,6	Désignation de mot, aucun message, exécution	2,5
		279,5	Désignation de *DM, aucun message, exécution	
		204,9	Désignation de mot, aucun message, initialisation	
		312,0	Désignation de *DM, aucun message, initialisation	
	SRCH	62,4	Recherche de mot, résultats sur mot	2,5
		2,64 ms	Recherche de 1 024 mots via *DM, résultats sur *DM	
		15,11 ms	Recherche de 6 144 mots via *DM, résultats sur *DM	
	MAX	56,1	Recherche de mot, résultats sur mot	2,5
		2,56 ms	Recherche de 999 mots via *DM, résultats sur *DM	
	MIN	56,1	Recherche de mot, résultats sur mot	2,5
		2,56 ms	Recherche de 999 mots via *DM, résultats sur *DM	
	APR	57,4	Calcul du sinus	2,5
		460,4	Approximation linéaire avec tableau de 256 éléments via désignation de *DM	
	LINE	93,4	Mot vers mot	2,5
		166,5	*DM vers *DM	
	COLM	115,1	Mot vers mot	2,5
		183,1	*DM vers *DM	
	SEC	92,9	Mot vers mot	2,5
		146,2	*DM vers *DM	
	HMS	94,9	Mot vers mot	2,5
		148,7	*DM vers *DM	
	SUM	72,9	Addition d'un mot, résultats sur mot	2,5
		6,86 ms	Addition de 999 mots via un *DM, résultats sur *DM	
	FCS	73,6	Calcul d'un mot, résultats sur mot	2,5
		2,33 ms	Calcul de 999 mots via un *DM, résultats sur *DM	
	HEX	82,3	Mot vers mot	2,5
		154,3	*DM vers *DM	
	AVG	73,7	Moyenne d'un cycle pour mot	2,5
		282,5	Moyenne de 64 cycles via un *DM	

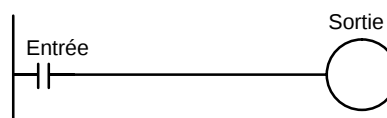
Code	Mnémoni- que	Temps d'exéc. ON ()	Conditions	Temps d'exéc. OFF ()
	PWM	266,8	Rapport cyclique spécifié par une constante	2,5
		272,8	Rapport cyclique spécifié par un mot	
		293,8	Rapport cyclique spécifié par un *DM	
	PID	2,11 ms	Mot vers mot (exécution initiale)	2,5
		2,30 ms	*DM vers *DM (exécution initiale)	
		607,7	Mot vers mot (en cas d'échantillonnage)	
		893,7	*DM vers *DM (en cas d'échantillonnage)	
	ADBL	75,4	Mot + mot → mot	2,5
		152,0	*DM + *DM → *DM	
	SBBL	75,4	Mot – mot → mot	2,5
		152,0	*DM – *DM → *DM	
	MBS	57,9	Constante × mot → mot	2,5
		61,1	Mot × mot → mot	
		135,0	*DM × *DM → *DM	
	DBS	28,3	Constante ÷ mot → mot	2,5
		62,4	Mot ÷ mot → mot	
		136,4	*DM ÷ *DM → *DM	
	MBSL	94,0	Mot × mot → mot	2,5
		167,6	*DM × *DM → *DM	
	DBSL	86,3	Mot ÷ mot → mot	2,5
		160,4	*DM ÷ *DM → *DM	
	CPS	31,0	Comparaison d'une constante et d'un mot	2,5
		33,7	Comparaison de mots	
		82,4	Comparaison de *DM	
	CPSL	51,3	Comparaison de mots	2,5
		102,6	Comparaison de *DM	
	NEG	41,3	Conversion constante → mot	2,5
		44,5	Conversion mot → mot	
		92,7	Conversion *DM → *DM	
	NEGL	51,1	Conversion constant → mots	2,5
		103,2	Conversion *DM → *DM	
	ZCP	38,2	Comparaison constante à gamme de mots	2,5
		44,7	Comparaison mot à gamme de mots	
		114,6	Comparaison *DM à gamme de *DM	
	CPSL	77,7	Comparaison mots à gamme de mots	2,5
		151,4	Comparaison *DM à gamme de *DM	
	XFRB	35,3	Transfert d'1 bit entre mots ayant une constante comme donnée de contrôle	2,5
		56,8	Transfert d'1 bit entre mots ayant une constante comme donnée de contrôle	
		298,3	Transfert de 255 bits entre *DM ayant un *DM comme mot de contrôle	
	PLS2	821,7	Mots pour contrôle de mots	2,5
		849,0	*DM pour contrôle de mots	

Code	Mnémoni- que	Temps d'exéc. ON ()	Conditions	Temps d'exéc. OFF ()
	ACC	547,3	Mode 0: Mots pour contrôle de mots	2,5
		577,0	Mode 0: *DM pour contrôle de mots	
		392,8	Mode 1: Mots pour contrôle de mots	
		424,0	Mode 1: *DM pour contrôle de mots	
		404,8	Mode 2: Mots pour contrôle de mots	
		430,3	Mode 2: *DM pour contrôle de mots	
		259,5	Mode 3: Mots pour contrôle de mots	
		418,3	Mode 3: *DM pour contrôle de mots	
	SCL2	105,0	Conversion de mot à mot, mots pour les mots de paramètres	2,5
		179,8	Conversion *DM à *DM, *DM pour les mots de paramètres	
	SCL3	112,0	Conversion de mot à mot, mots pour les mots de paramètres	2,5
		186,8	Conversion *DM à *DM, *DM pour les mots de paramètres	

7-2-3 Temps de réponse des E/S

Le temps de réponse est le temps nécessaire à l'API après réception d'un signal d'entrée (c'est-à-dire après le passage à ON d'un bit d'entrée) pour vérifier et traiter les informations et pour produire un signal de contrôle (c'est-à-dire pour sortir le résultat de traitement d'un bit de sortie). Le temps de réponse varie en fonction du traitement des conditions et du temps nécessaire à l'exécution de ces conditions.

Les temps de réponse minimum et maximum des E/S sont donnés ci-après, avec le programme suivant comme exemple :



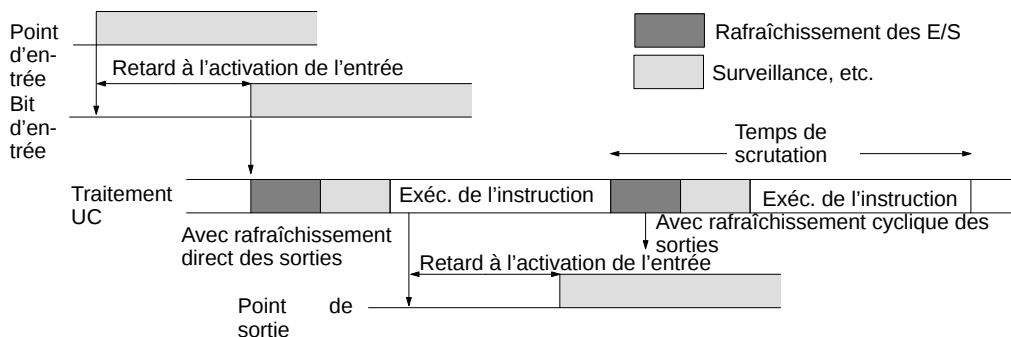
Exemples de conditions pour le calcul des temps de réponse des E/S :

Retard à l'activation des entrées :	8 ms
Temps de surveillance :	1 ms
Temps d'exécution des instructions :	14 ms
Retard à l'activation des sorties relais :	10 ms
Position de l'instruction de sortie :	début du programme
Ports de communication :	non utilisés

Rem. : le retard à l'activation des entrées pour les cartes d'entrée c.c. peut être sélectionné dans la configuration de l'API.

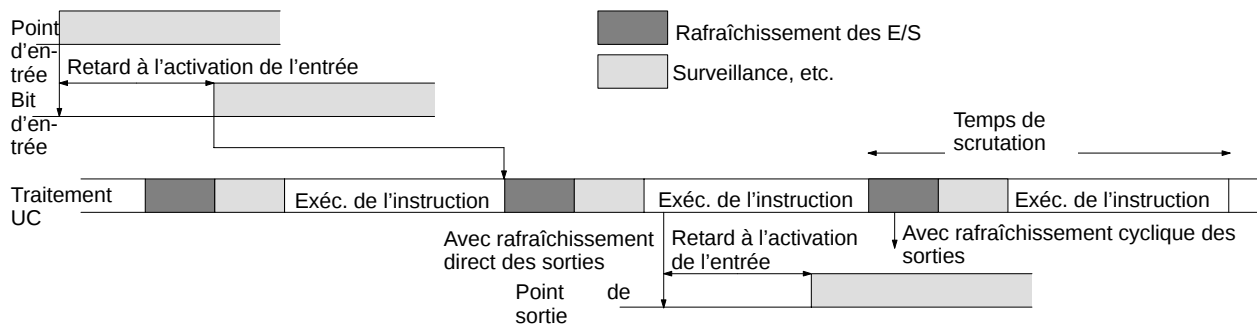
Temps de réponse minimum des E/S

Le CQM1 répond au plus vite lorsqu'il reçoit un signal d'entrée juste avant la phase de rafraîchissement des entrées, comme ci-dessous :



Avec le rafraîchissement cyclique des sorties :
Temps de réponse minimum des E/S = 8 + 15 + 10 = 33 ms
Avec le rafraîchissement direct des sorties :
Temps de réponse minimum des E/S = 8 + 1 + 10 = 19 ms

Temps de réponse maximum des E/S Le CQM1 produit un temps de réponse maximum lorsqu'il reçoit le signal d'entrée juste après la phase de rafraîchissement des entrées, comme ci-dessous. Dans ce cas, un retard d'environ un tour de scrutation est à prévoir.



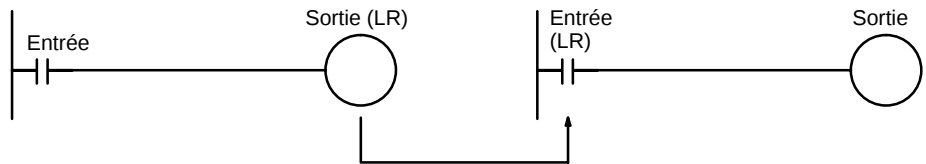
Avec le rafraîchissement cyclique des sorties :
Temps de réponse minimum des E/S = 8 + 15 * 2 + 10 = 48 ms
Avec le rafraîchissement direct des sorties :
Temps de réponse minimum des E/S = 8 + 15 + 10 = 33 ms

7-2-4 Temps de réponse des E/S en liaison point par point

Lorsque deux CQM1 sont reliés point par point, le temps de réponse des E/S est le temps nécessaire à une entrée exécutée sur un CQM1 pour être sortie sur l'autre CQM1 par le moyen d'une communication point par point. Les communications point par point sont exécutées réciproquement entre le maître et l'esclave. Les temps de transmission respectifs sont donnés ci-dessous, selon le nombre de canaux LR utilisés :

Nombre de canaux utilisés	Temps de transmission
64 canaux (LR 00 à LR 63)	39 ms
32 canaux (LR 00 à LR 31)	20 ms
16 canaux (LR 00 à LR 15)	10 ms

Les temps de réponse minimum et maximum des E/S sont donnés ci-dessous, avec les instructions suivantes comme exemples pour le maître et l'esclave. Ici, les communications proviennent du maître et vont vers l'esclave :



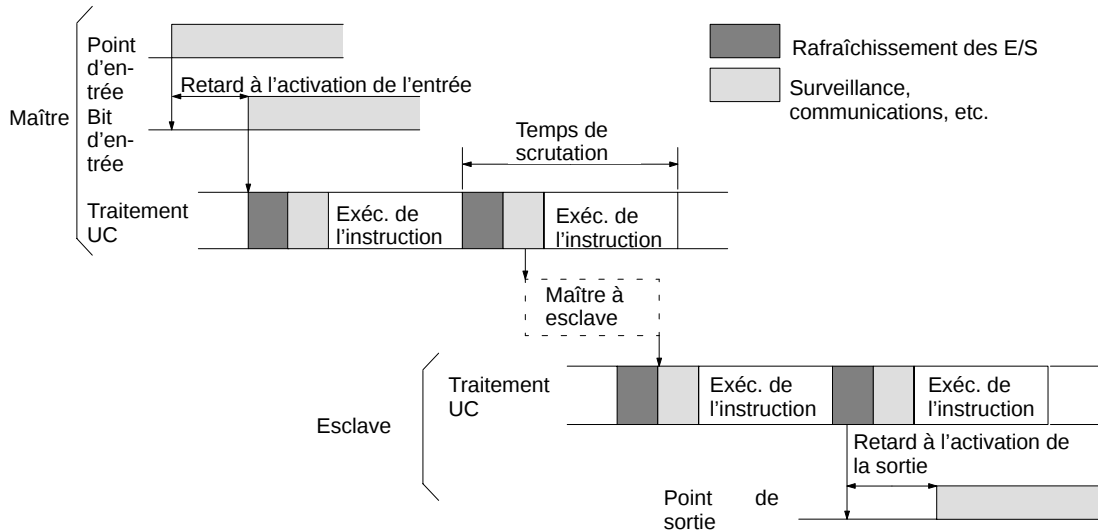
Exemples de conditions pour le calcul des temps de réponse des E/S :

Retard à l'activation des entrées :	8 ms
Temps de scrutation maître :	10 ms
Temps de scrutation esclave :	14 ms
Retard à l'activation des sorties :	10 ms
Sortie directe :	non utilisée
Nombre de canaux LR :	64

Rem. : le retard à l'activation des entrées pour les cartes d'entrée c.c. peut être sélectionné dans la configuration de l'API.

Temps de réponse minimum des E/S Le CQM1 répond au plus vite dans les conditions suivantes :

- 1, 2, 3... 1. Le CQM1 reçoit un signal d'entrée juste avant la phase rafraîchissement des entrées.
2. La transmission maître/esclave commence immédiatement.
3. L'esclave exécute les mises en service des communications juste après la fin des communications.



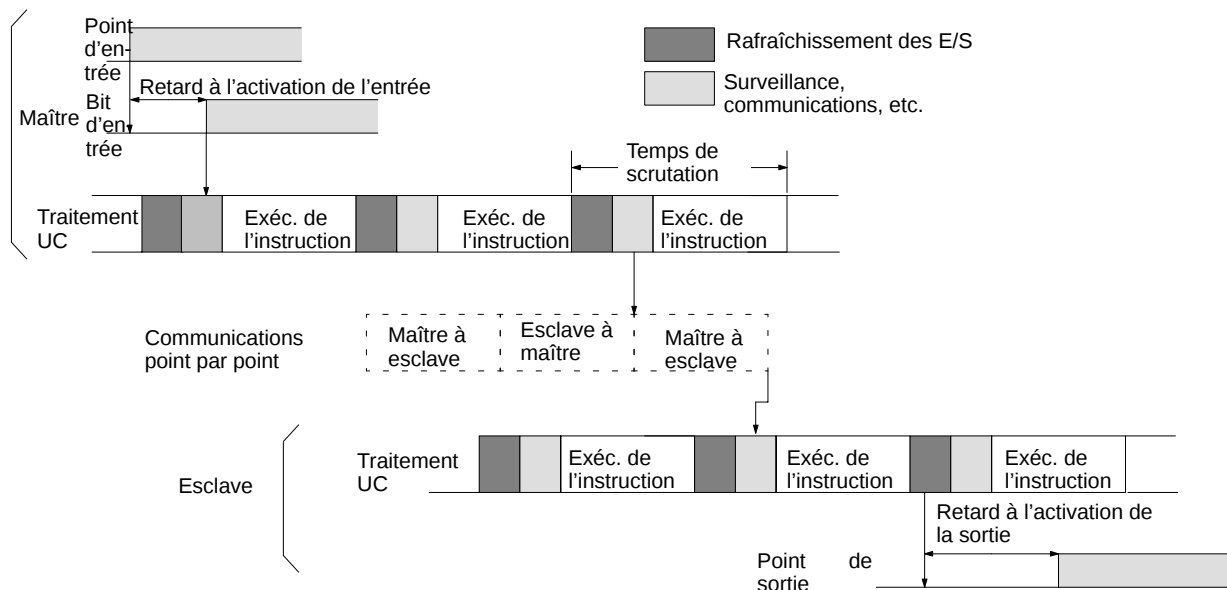
Temps de réponse minimum des E/S :

Retard à l'activation de l'entrée :	8 ms
Temps de scrutation maître :	10 ms
Temps de transmission :	39 ms
Temps de scrutation esclave :	15 ms
<u>+ Retard à l'activation de la sortie :</u>	<u>10 ms</u>
Temps de réponse minimum des E/S :	82 ms

Temps de réponse maximum des E/S Le CQM1 produit un temps de réponse maximum dans les conditions suivantes :

- 1, 2, 3... 1. Le CQM1 reçoit un signal d'entrée juste après la phase rafraîchissement des entrées.
2. La transmission maître/esclave ne commence pas au moment où elle le devrait.

3. Les communications se terminent juste après la mise en service des communications par l'esclave.



Temps de réponse maximum des E/S :

Retard à l'activation de l'entrée :	8 ms
Temps de scrutation maître :	10 ms * 2
Temps de transmission :	39 ms * 3
Temps de cycle esclave :	15 ms * 2
+ Retard à l'activation de la sortie :	10 ms
Temps de réponse maximum des E/S : 185 ms	

7-2-5 Temps de traitement des interruptions

Ce paragraphe traite des temps de traitement des interruptions depuis le départ de l'interruption jusqu'à l'appel du sous-programme et depuis la fin du sous-programme jusqu'au retour à la position originale. Les explications suivantes valent pour les interruptions d'entrée, les interruptions cycliques et les interruptions de compteur rapide.

Temps de traitement

Voici les temps de traitement à partir de la création d'un signal d'interruption jusqu'à l'appel du sous-programme et depuis la fin du sous-programme d'interruption jusqu'au retour à la position originale.

	Contenu	Temps
Retard à l'activation de l'entrée d'interruption	Temps de retard à partir du passage à ON du bit d'entrée d'interruption jusqu'à l'exécution de l'interruption ; ce temps est indépendant des autres interruptions.	50 μ s
i (condition d'interruption réalisée) (cf Rem.)		
Attente jusqu'à la fin du traitement de l'interruption par masque.	Temps d'attente des interruptions jusqu'à la fin du traitement en cas de traitement par masque ; cf paragraphe "Masques".	Cf paragraphe "Masques"
i		
Traitement du passage à l'interruption	Temps nécessaire au passage du fonctionnement normal à une interruption.	40 μ s
i		
Rafraîchissement des entrées au moment de l'interruption	Temps nécessaire au rafraîchissement des entrées lorsqu'il est programmé pour être exécuté au moment de l'appel d'un sous-programme d'interruption (sélection dans la configuration API, DM 6630 à DM 6638.)	10 μ s par mot
i (sous-programme d'interruption exécuté)		
Retour	Temps écoulé entre l'exécution de RET(93) et le retour au processus interrompu.	40 μ s

Rem. : 1. Lorsque le compteur rapide 0 est utilisé avec une table de comparaison par gammes, le temps de traitement de l'interruption peut être affecté par le temps de scrutation.

2. Lorsque les compteurs rapides 1 et 2 sont utilisés avec des tables de comparaison par gammes (avec les UC CQM1-43/44-E), le temps de traitement des interruptions peut être retardé de 1 ms max.

Masques

Les interruptions sont masquées pendant le fonctionnement normal du CQM1 de la façon suivante : jusqu'à la fin du traitement, toutes les interruptions restent masquées durant l'intervalle de temps spécifié.

Tempo. rapides : le temps nécessaire est indiqué ci-dessous, selon (a) : le nombre de tempo. utilisées avec TIMH(15) et (b) : le nombre de tempo. rapides actives à ce moment (le nombre de tempo. rapides est sélectionné dans la configuration API, dans le DM 6629 ; sélection par défaut : 16).
 $0 \leq \text{temps d'attente} \leq 50 + 3 \times (a + b) \mu\text{s}$
 50 μ s max. peuvent être nécessaires, même sans tempo. rapides.

Création et effacement des erreurs fatales : en cas de création d'erreur fatale et d'enregistrement de son contenu dans le CQM1 ou en cas d'effacement d'erreur, les interruptions sont masquées pendant 100 μ s max. jusqu'à la fin du processus.

Modifications "On-line" : les interruptions sont masquées pour 1 s maximum lorsque des modifications sont exécutées en "On-line" pendant le fonctionnement de l'appareil.

La sortie d'impulsion par SPED(64) peut également être affectée par les interruptions, entraînant ainsi des variations dans les temps de sortie.

Exemple de calcul

Cet exemple montre le temps de réponse des interruptions (c'est-à-dire le temps qui s'écoule entre le passage à ON de l'entrée d'interruption et le début du sous-programme d'interruption) lorsque les interruptions sont exécutées dans les conditions suivantes :

Nombre de tempo. rapides : 0 (aucune tempo. rapide lancée)

Modifications "On-line" : non utilisées

Rafraîchissement des entrées au moment de l'interruption : non

Temps de réponse minimum

Retard à l'activation de l'entrée interruptive : 50 μ s

Temps d'attente du masque d'interruption : 0 μ s

+ Passage à l'interruption : 40 μ s

Temps de réponse minimum : 90 μ s

Temps de réponse maximum

Retard à l'activation de l'entrée interruptive : 50 μ s

Temps d'attente du masque d'interruption : 50 μ s

+ Passage à l'interruption : 40 μ s

Temps de réponse maximum : 140 μ s

En plus du temps de réponse ci-dessus, le temps nécessaire à l'exécution du sous-programme d'interruption et un temps de retour de 40 μ s doivent également être pris en compte pour revenir au programme qui a été interrompu.

Veillez à laisser suffisamment de temps pour les interruptions lorsque vous en utilisez dans le programme.

Les sorties provenant des sous-programmes peuvent être produites immédiatement si l'on utilise des sorties directes ; ces dernières s'utilisent pour le programme principal et les sous-programmes d'interruption et ne peuvent être sélectionnées séparément.

Chapitre 8

En cas de problème

Ce chapitre décrit comment procéder à un diagnostic corriger les erreurs de matériel et de logiciel pouvant se produire.

8-1	Introduction	352
8-2	Erreurs de fonctionnement de la console de programmation	352
8-3	Erreurs de programmation	353
8-4	Erreurs définies par l'utilisateur	354
8-5	Erreurs de fonctionnement	355
8-5-1	Erreurs non fatales	355
8-5-2	Erreurs fatales	356
8-6	Stockage des erreurs	357
8-7	Erreurs de liaison à l'ordinateur central	357
8-8	Diagrammes de résolution des problèmes	359

8-1 Introduction

Les erreurs API se divisent à peu près en quatre catégories :

- 1, 2, 3... 1. Erreurs d'entrée de programme
Erreurs produites à l'entrée du programme ou lors de la préparation de l'API.
2. Erreurs de programmation
Erreurs produites pendant la vérification du programme avec la fonction de vérification de programme.
3. Erreurs définies par l'utilisateur
3 instructions peuvent être définies par l'utilisateur pour ses propres erreurs ou messages. Les instructions sont exécutées lorsqu'une condition spéciale (définie par l'utilisateur) apparaît pendant le fonctionnement.
4. Erreurs de fonctionnement
Ces erreurs se produisent après le début de l'exécution du programme :
 - a) Erreurs de fonctionnement non fatales
Le fonctionnement de l'API et l'exécution du programme continuent après l'apparition d'une ou plusieurs de ces erreurs.
 - b) Erreurs de fonctionnement fatales
Le fonctionnement de l'API et l'exécution du programme s'arrêtent et toutes les sorties de l'API passent à OFF lorsqu'une ou plusieurs erreurs de ce type se produisent.

Les voyant de l'API indiquent une erreur ; un message ou code d'erreur est affiché sur la console de programmation ou éventuellement sur l'ordinateur central. Le code d'erreur se trouve également dans les SR 25300 à SR 25307.

Pour les erreurs les plus récentes, le type et l'heure de l'erreur sont enregistrés dans la zone de stockage des erreurs de l'API (DM 6570 à DM 6599).

Les zones SR et AR contiennent des drapeaux et autres informations pouvant être utilisées en cas de problème. Reportez-vous à l'annexe D où figure une liste de ces informations.

Rem. : en plus des erreurs ci-dessus, des erreurs de communication peuvent apparaître en cas de liaison à un ordinateur central.

8-2 Erreurs de fonctionnement de la console de programmation

Les messages d'erreur suivants peuvent apparaître sur la console de programmation ; il faut alors corriger l'erreur comme suit. Les astérisques sont remplacées par des données numériques (une adresse, le plus souvent) lors de l'affichage. Reportez-vous au manuel LSS pour les erreurs pouvant se produire avec le logiciel LSS.

Message	Signification et réponse appropriée
REPL ROM	Essai d'écriture en mémoire protégée. Placez le bouton d'écriture protégée sur OFF (broche 1 de l'interrupteur DIP de l'UC).
PROG OVER	L'instruction de la dernière adresse en mémoire n'est pas NOP(00). Effacez toutes les instructions superflues en fin de programme.
ADDR OVER	Adresse supérieure à la plus importante des adresses mémoire sélectionnée en mémoire programme. Entrez une adresse inférieure.
SETDATA ERR	FALS 00 entré ; "00" ne peut être entré. Entrez à nouveau la donnée.
I/O NO. ERR	Adresse de zone de données désignée dépassant la limite de la zone de données (adresse trop importante par ex.) ; ré-entrez la donnée après vérification des conditions de l'instruction.

8-3 Erreurs de programmation

Les erreurs de syntaxe du programme sont détectées lorsque le programme est vérifié par la fonction de vérification de programme.

3 niveaux de vérification sont possibles. Le niveau désiré doit être désigné pour indiquer le type d'erreur à détecter. Voici les types d'erreurs, les affichages et les explications des erreurs de syntaxe. Le niveau 0 permet de rechercher les erreurs de type A, B et C, le niveau 1 recherche les erreurs de type A et B, le niveau 2 les erreurs de type A uniquement.

Erreurs de niveau A

Message	Signification et réponse appropriée
?????	Le programme a été endommagé par la création d'un code de fonction qui n'existe pas. Entrez à nouveau le programme.
CIRCUIT ERR	Le nombre de blocs logiques ne correspond pas au nombre d'instructions à blocs logiques : LD ou LD NOT commence un bloc logique dont la condition d'exécution n'a pas été utilisée par une autre instruction, ou une instruction à bloc logique n'a pas le nombre de blocs nécessaire. Vérifiez le programme.
OPERAND ERR	Une des constantes entrées pour l'instruction n'est pas comprise dans la gamme définie. Choisissez une constante dans la gamme.
NO END INSTR	Absence de END(01) dans le programme ; inscrivez END(01) dans l'adresse finale du programme.
LOCN ERR	Une instruction du programme ne se trouve pas à la bonne place. Corrigez le programme d'après les règles convenues.
JME UNDEFD	Une instruction JME(04) manque avec une instruction JMP(05). Corrigez le numéro de saut ou insérez une instruction JME(04).
DUPL	Un numéro de saut identique ou un numéro de sous-programme a été utilisé deux fois. Corrigez le programme de façon à n'utiliser le même numéro qu'une seule fois.
SBN UNDEFD	L'instruction SBS(91) a été programmée pour un numéro de sous-programme inexistant. Corrigez le numéro de sous-programme ou programmez le sous-programme adéquat.
STEP ERR	STEP(08) avec un numéro de partie et STEP(08) sans numéro de partie ne sont pas utilisés correctement. Vérifiez les conditions de programmation de STEP(08) et corrigez le programme.

Erreurs de niveau B

Message	Signification et réponse appropriée
IL-ILC ERR	IL(02) et ILC(03) ne sont pas utilisées en paires. Corrigez le programme de façon à ce qu'à chaque IL(02) corresponde une seule ILC(03). Avec plusieurs IL(02) associées à une seule ILC(03), le programme est tout de même exécuté. Vérifiez que le programme est écrit comme vous le souhaitez avant de poursuivre.
JMP-JME ERR	JMP(04) 00 et JME(05) 00 ne sont pas utilisées en paires. Avec plusieurs JMP(04) 00 associées à une seule JME(05) 00, le programme est tout de même exécuté. Vérifiez que le programme est écrit comme vous le souhaitez avant de poursuivre.
SBN-RET ERR	Si l'adresse affichée est celle de SBN(92), deux sous-programmes différents ont été définis avec le même numéro de sous-programme. Changez l'un des numéros ou supprimez un sous-programme. Si l'adresse affichée est celle de RET(93), RET(93) n'a pas été utilisés correctement. Vérifiez les règles de RET(93) et corrigez le programme.

Erreurs de niveau C

Message	Signification et réponse appropriée
COIL DUPL	Le même bit est contrôlé (placé à ON ou OFF) par plusieurs instructions (OUT, OUT NOT, DIFU(13), DIFD(14), KEEP(11), SFT(10)). Bien que cela soit permis pour certaines instructions, vérifiez les règles de l'instruction en cause pour savoir si le programme est correct ou ré-écrivez-le pour que chaque bit ne soit contrôlé que par une seule instruction.
JMP UNDEFD	JME(05) n'est pas utilisé avec JMP(04) avec le même numéro de saut. Ajoutez un JMP(04) portant le même numéro ou supprimez le JME(05) non utilisé.
SBS UNDEFD	Un sous-programme n'est pas appelé par SBS(91). Programmez un appel de sous-programme à l'endroit adéquat ou supprimez le sous-programme si cela est nécessaire.

Attention : Les instructions étendues (codes de fonction 17, 18, 19, 47, 48, 60 à 69, 87, 88, et 89) ne sont pas soumises aux vérifications de programme, ni les DM 1024 à 6143 pour les API qui n'en sont pas pourvus en zone DM (par ex. CQM1-CPU11-E et CQM1-CPU21-E). Les données ne sont pas écrites, même si ces zones sont spécifiées et les données lues sur ces zones sont toujours égales à "0000."

8-4 Erreurs définies par l'utilisateur

Voici les 3 instructions que l'utilisateur peut employer pour définir ses propres erreurs ou messages ; elles envoient des messages à la console de programmation connectée à l'API et causent des erreurs fatales ou non fatales.

MESSAGE – MSG(46)

MSG(46) affiche un message sur la console de programmation ; il peut avoir jusqu'à 16 caractères en longueur et s'affiche lorsque la condition d'exécution de l'instruction est à ON.

FAILURE ALARM – FAL(06)

FAL(06) est une instruction entraînant une erreur non fatale. Voici ce qui se produit lorsqu'une instruction FAL(06) est exécutée :

1, 2, 3...

1. Le voyant ERR/ALM de l'UC clignote. L'API continue à fonctionner.
2. Le numéro de FAL en BCD à 2 chiffres (01 à 99) est écrit dans les SR 25300 à 25307.
3. Le numéro de FAL et l'heure de l'occurrence sont enregistrés dans la zone de stockage d'erreurs de l'API si l'on utilise une cartouche mémoire ayant une horloge (RTC).

Les numéros de FAL peuvent être sélectionnés arbitrairement pour indiquer des conditions particulières. Le même numéro ne peut être utilisé en FAL et FALS. Pour effacer une erreur FAL, corrigez la cause de l'erreur, exécutez FAL 00 et effacez l'erreur utilisant la console de programmation.

SEVERE FAILURE ALARM – FALS(07)

FALS(07) est une instruction entraînant une erreur fatale. Voici ce qui se produit lorsqu'une instruction FALS(07) est exécutée :

1, 2, 3...

1. L'exécution du programme est arrêtée et les sorties sont placées à OFF.
2. Le voyant ERR/ALM de l'UC est allumée.
3. Le numéro de FALS BCD à 2 chiffres (01 à 99) est écrit dans SR 25300 à 25307.
4. Le numéro de FALS et l'heure de l'occurrence sont enregistrés dans la zone de stockage d'erreurs de l'API si l'on utilise une cartouche mémoire ayant une horloge (RTC).

Les numéros de FALS peuvent être sélectionnés arbitrairement pour indiquer des conditions particulières. Le même numéro ne peut être utilisé en FAL et FALS.

Pour effacer une erreur FALS, placez l'API en mode PROGRAM, corrigez la cause de l'erreur et effacez l'erreur utilisant la console de programmation.

FAILURE POINT DETECT – FPD(—)

Les messages d'erreur fatales et non fatales peuvent également être générés avec FPD(—).

8-5 Erreurs de fonctionnement

Il y a deux genres d'erreurs de programmation : les erreurs fatales et les erreurs non fatales. L'API continue de fonctionner après une erreur non fatale mais il s'arrête en cas d'erreur fatale.

Attention : cherchez la cause de toutes les erreurs, fatales ou non ; corrigez-les et remettez l'API en marche. Reportez-vous au manuel LSS pour ce qui concerne les erreurs avec le logiciel LSS.

8-5-1 Erreurs non fatales

L'API continue de fonctionner après une ou plusieurs de ces erreurs mais elles doivent être corrigées et effacées au plus tôt.

Avec l'apparition de l'une de ces erreurs, les voyants POWER et RUN restent allumés et le voyant ERR/ALM clignote :

Message	N° FAL	Signification et réponse appropriée
SYS FAIL FAL**	01 à 99	Une instruction FAL(06) a été exécutée dans le programme. Vérifiez le numéro de FAL pour connaître les conditions qui en causent l'exécution, corrigez et effacez l'erreur.
	9D	Erreur pendant la transmission de données entre UC et cartouche mémoire. Vérifiez l'état des drapeaux AR 1412 à 1415 et corrigez comme suit : AR 1412 ON: passez en mode PROGRAM, effacez l'erreur et retransférez. AR 1413 ON: la destination du transfert est protégée contre l'écriture. La destination est l'API : mettez-le hors tension, assurez-vous que la broche 1 de l'interrupteur DIP de l'UC est à OFF, effacez l'erreur et retransférez. La destination est une cartouche EEPROM : vérifiez que l'alimentation est branchée, effacez l'erreur et retransférez. La destination est une cartouche EPROM : prenez une cartouche où l'écriture est possible. AR 1414 ON: capacité de la destination insuffisante. Vérifiez la taille du programme source dans AR 15 et envisagez l'utilisation d'une autre UC ou d'une autre cartouche. AR 1415 ON: programme absent dans la cartouche mémoire ou présence d'erreurs dans le programme. Vérifiez la cartouche mémoire.
	9C	Erreur dans la fonction d'E/S par impulsions ou d'interface du codeur de type absolu. Vérifiez le contenu des AR 0408 à 0415 (2 chiffres BCD) et corrigez comme suit (codes ne s'appliquant qu'aux CQM1-CPU43-E et CQM1-CPU44-E). 01, 02: erreur de matériel. Débranchez et rebranchez l'alimentation. Si l'erreur persiste, remplacez l'UC. 03: sélections de la configuration API (DM 6611, DM 6612, DM 6643, DM 6644) incorrectes ; corrigez-les. 04: CQM1 arrêté pendant la sortie par impulsions. Vérifiez l'état de la carte qui reçoit les sorties par impulsion ; elle peut avoir été affectée.
SYS FAIL FAL**	9B	Erreur détectée dans la configuration AP. Vérifiez les drapeaux AR 2400 à 2402 et corrigez comme suit : AR 2400 ON: sélections incorrectes dans la configuration API (DM 6600 à 6614) à la mise sous tension ; corrigez-les en mode PROGRAM et rebranchez l'alimentation. AR 2401 ON: sélections incorrectes dans la configuration API (DM 6615 à 6644) à la mise en mode RUN ; corrigez-les en mode PROGRAM et repassez en mode RUN. AR 2402 ON: sélections incorrectes dans la configuration API (DM 6645 à 6655) durant le fonctionnement ; corrigez-les et effacez l'erreur.

Message	N° FAL	Signification et réponse appropriée
SCAN TIME OVER	F8	La tempo. chien de garde a dépassé 100 ms (SR 25309 passe à ON). Le temps de scrutation du programme est donc plus long que ce qui est habituellement conseillé ; réduisez-le si possible.
BATT LOW	F7	Pile de sauvegarde absente ou chute de tension (SR 25308 passe à ON). Remplacez la pile si besoin est ; vérifiez la configuration API (DM 6655) pour vous assurer que les défaillances de la pile sont détectées.

Erreurs de communication

S'il apparaît une erreur dans les communications par port périphérique ou RS-232C, le voyant correspondant (COM1 ou COM2) arrête de clignoter. Vérifiez les câbles de connexion et le programme de l'API et de l'ordinateur central.

Remettez les ports de communication à zéro avec les bits de RAZ de port, SR 25208 et SR 25209.

Inhibition de sortie

Lorsque le voyant OUT INH est allumé, le bit d'inhibition de sortie (SR 25215) est à ON et toutes les sorties de l'UC passent à OFF. Si cet état général n'est pas nécessaire, placez le bit SR 25215 à OFF.

8-5-2 Erreurs fatales

L'API s'arrête, ainsi que l'exécution du programme. Toutes les sorties de l'API passent à OFF lorsque l'une de ces erreurs se produit.

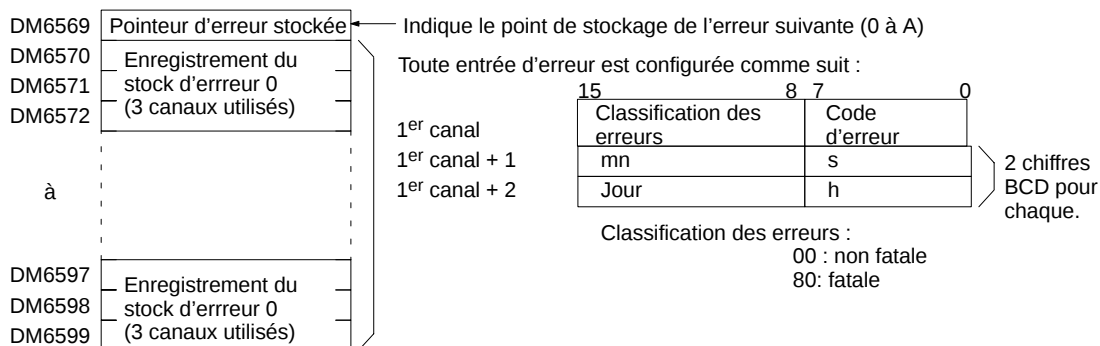
Tous les voyants de l'UC passent à OFF pour l'erreur de coupure de courant. Pour les autres erreurs de fonctionnement fatales, les voyants POWER et ERR/ALM sont allumés. Le voyant RUN est à OFF.

Message	N°FALS	Signification et réponse appropriée
Coupure de courant (aucun message)	Aucun	L'alimentation a été interrompue pendant au moins 10 ms. Vérifiez la tension d'alimentation et les lignes de puissance. Essayez de rebrancher l'alimentation.
MEMORY ERR	F1	AR 1611 ON: erreur de somme de vérification dans la configuration API (DM 6600 à 6655). Initialisez toute la configuration API et ré-effectuez les entrées.
		AR 1612 ON: erreur de somme de vérification dans le programme, indiquant une instruction incorrecte ; vérifiez le programme et corrigez toutes les erreurs.
		AR 1613 ON: erreur de somme de vérification dans les données d'instructions étendues. Initialisez toutes les sélections des instructions étendues et ré-effectuez les entrées.
		AR 1614 ON: cartouche mémoire installée ou démontée avec l'alimentation branchée. Coupez l'alimentation, installez la cartouche et rebranchez l'alimentation.
		AR 1615 ON: contenu de la cartouche mémoire illisible au départ. Vérifiez les drapeaux AR 1412 à 1415 pour trouver la cause du problème, corrigez-le et rebranchez l'alimentation.
NO END INST	F0	Absence de END(01) dans le programme ; inscrivez END(01) dans l'adresse finale du programme.
I/O BUS ERR	C0	Erreur pendant le transfert de données entre l'UC et une carte d'E/S. Trouvez l'endroit qui pose problème à l'aide des drapeaux AR 2408 à 2415, coupez l'alimentation, vérifiez que les cartes d'E/S et capots soient bien fixés et rebranchez l'alimentation.
I/O UNIT OVER	E1	Le nombre de canaux d'E/S sur la carte d'E/S installée dépasse le maximum. Débranchez l'alimentation, ré-organisez le système pour réduire le nombre de canaux d'E/S et rebranchez l'alimentation.
SYS FAIL FALS** (cf Rem.)	01 à 99	Une instruction FALS(07) a été exécutée dans le programme ; vérifiez le numéro de FALS pour connaître les conditions qui causent l'erreur, corrigez-la et effacez-la.
	9F	Le temps de scrutation dépasse le temps de surveillance de la scrutation FALS 9F (DM 6618). Vérifiez le temps de scrutation et effectuez les réductions de temps éventuellement nécessaires.

8-6 Stockage des erreurs

La fonction de stockage des erreurs enregistre le code d'erreur de toute erreur fatale ou non fatale du CQM1, ainsi que la date et l'heure de son occurrence.

Zone de stockage des erreurs Les erreurs sont stockées dans les DM 6569 à 6599, comme suit :

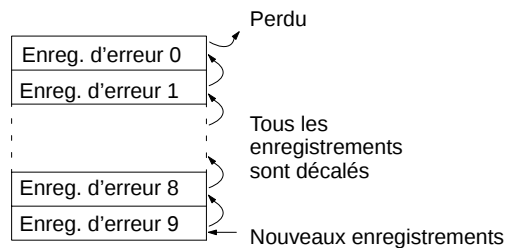


Codes d'erreur : cf supra

Modes de stockage des erreurs

Le mode de stockage des erreurs est sélectionné dans la configuration de l'API (DM 6655). Choisissez le mode qui vous convient :

- 1, 2, 3...**
1. On peut stocker seulement les 10 erreurs les plus récentes et non les plus anciennes par le décalage suivant. L'erreur la plus ancienne (enregistrement 0) est perdue à chaque nouvel enregistrement.



2. On peut ne stocker que les 10 premières erreurs et ignorer les erreurs suivantes.
3. On peut désactiver le stockage pour ne stocker aucune erreur.

La sélection par défaut est la première méthode. Reportez-vous aux sélections du stockage des erreurs au début de ce manuel pour de plus amples détails sur la configuration API.

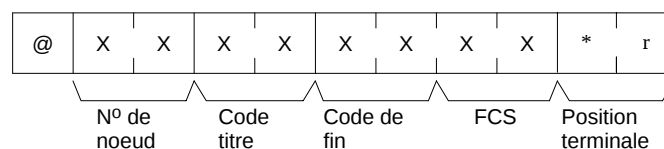
En cas de montage d'une cartouche mémoire sans horloge, des zéros figurent à la place de la date et de l'heure.

Les enregistrements d'erreur sont stockés même si la broche 1 de l'interrupteur DIP du CQM1 DIP est à ON pour protéger les DM 6144 à 6655.

Effacement du stockage des erreurs Pour effacer toutes les erreurs, placez SR 25214 à ON à partir d'un périphérique (après cela, SR 25214 repasse automatiquement à OFF).

8-7 Erreurs de liaison à l'ordinateur

Ces codes d'erreur sont reçus comme codes de réponse (codes de fin) en cas d'impossibilité de traitement d'une commande de l'ordinateur reçue par le CQM1. Voici le format du code d'erreur :



Le code titre varie selon la commande et peut contenir un sous-code (pour les commandes composites).

Code fin	Contenu	Cause probable	Remède
00	Fin normale	---	---
01	Exécution impossible en mode RUN	La commande envoyée ne peut être exécutée quand l'API est en mode RUN.	Vérifiez la correspondance entre la commande et le mode API.
02	Exécution impossible en mode MONITOR	La commande envoyée ne peut être exécutée quand l'API est en mode MONITOR.	
0B	Exécution impossible en mode PROGRAM	La commande envoyée ne peut être exécutée quand l'API est en mode PROGRAM.	Ce code n'est pas utilisé.
13	Erreur FCS	FCS faux. Erreur de calcul FCS ou influence de parasites.	Vérifiez le mode de calcul du FCS. S'il n'y a pas influence de parasites, retransférez la commande.
14	Erreur format	Mauvais format de commande	Vérifiez le format et retransférez la commande.
15	Erreur dans le numéro d'entrée	Mauvaises zones de lecture et d'écriture	Corrigez les zones et retransférez la commande.
16	Commande inexistante	Commande spécifiée inexistante	Vérifiez le code de commande.
18	Erreur de longueur de trame	Longueur maximum de trame dépassée	Séparez la commande en plusieurs trames.
19	Exécution impossible	Éléments à lire non enregistrés pour une commande composite (QQ)	Exécutez QQ pour enregistrer les éléments à lire avant d'essayer la lecture par lots.
23	Mémoire utilisateur protégée en écriture	Broche 1 de l'interrupteur DIP du CQM1 à ON	Placez à OFF la broche 1.
A3	Arrêt suite à une erreur FCS dans les données de transmission	Erreur générée alors qu'une commande sur plusieurs trames était en cours d'exécution. Rem. : les données existant jusqu'à ce point ont déjà été écrites sur la zone adéquate de l'UC.	Vérifiez les données de commande et ré-essayez de transférer.
A4	Arrêt suite à une erreur format dans les données de transmission		
A5	Arrêt suite à une erreur de numéro d'entrée dans les données de transmission		
A8	Arrêt suite à une erreur de longueur de trame dans les données de transmission		
Autres	---	Influence de parasites	Retransférez la commande.

Coupures de courant

Voici les réponses possibles du CQM1 en cas de coupure de courant, même momentanée. Si l'une de ces réponses est reçue pendant ou après une coupure de courant, répétez la commande.

Réponse de commande non définie

@00IC4A*r

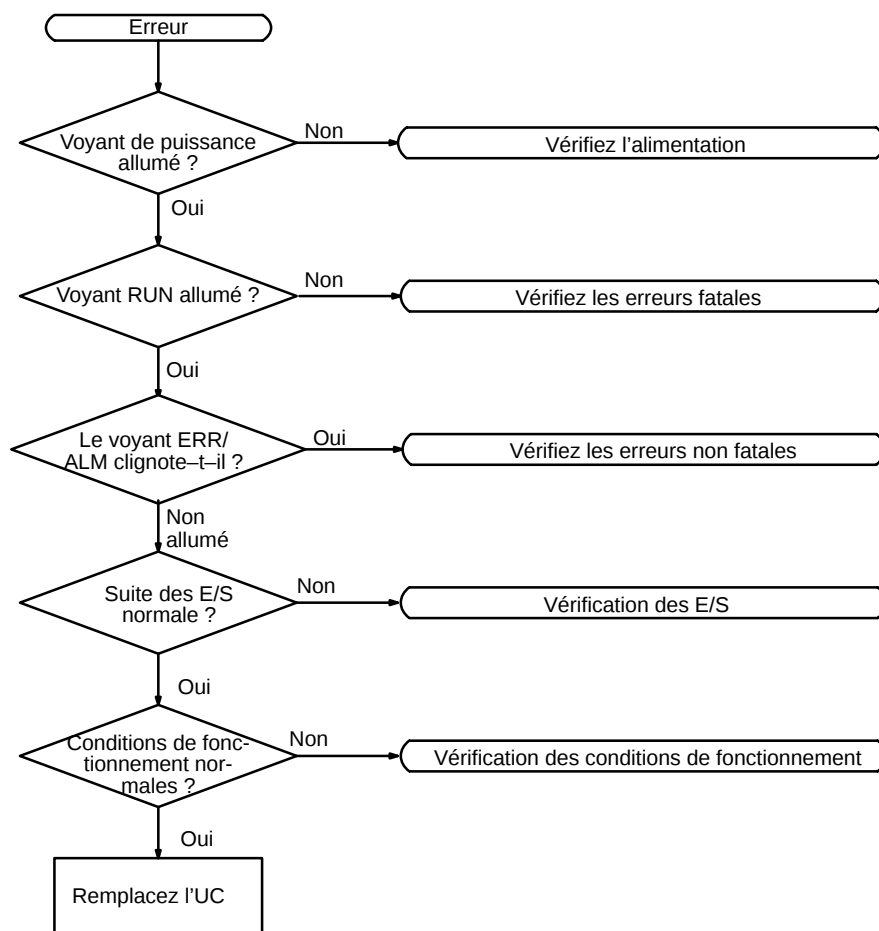
Aucune réponse

Si aucune réponse n'est reçue, arrêtez la dernière commande et répétez l'envoi.

8-8 Diagrammes de résolution des problèmes

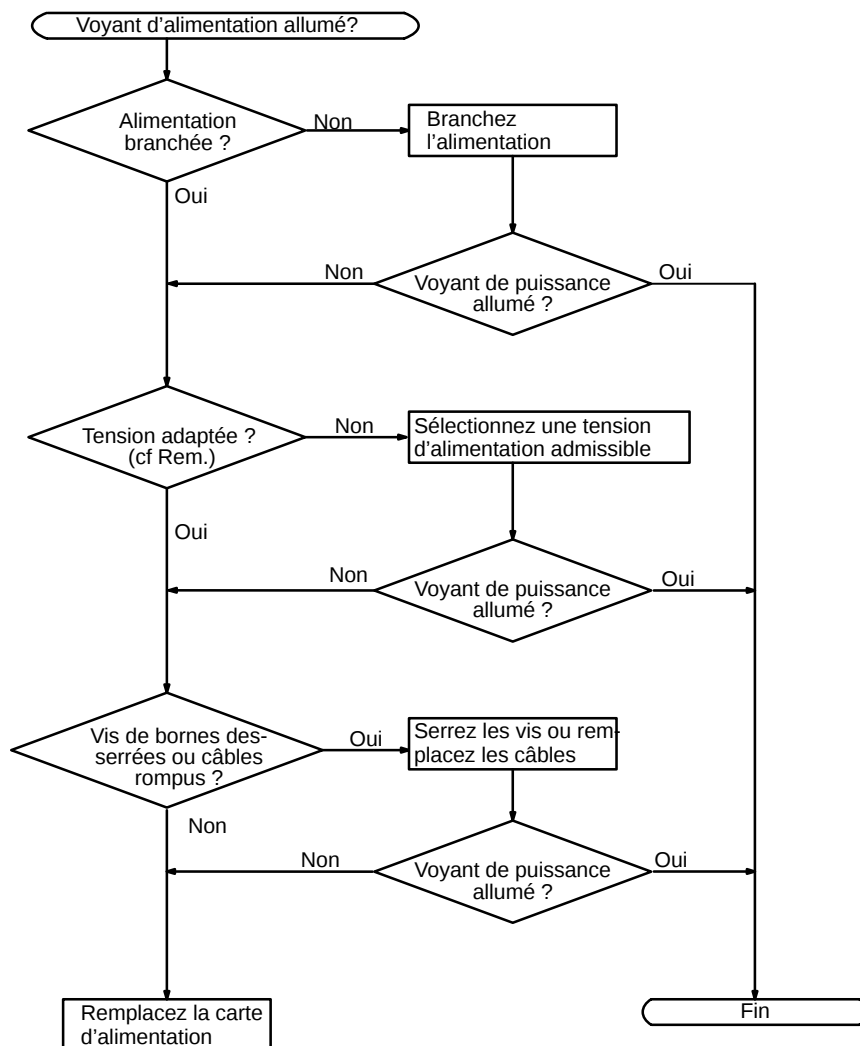
Utilisez les diagrammes suivants pour résoudre les éventuels problèmes de fonctionnement.

Vérification principale



Rem. : coupez toujours l'alimentation de l'API avant de remplacer les cartes, l'UC, les piles, les câbles ou les tableaux.

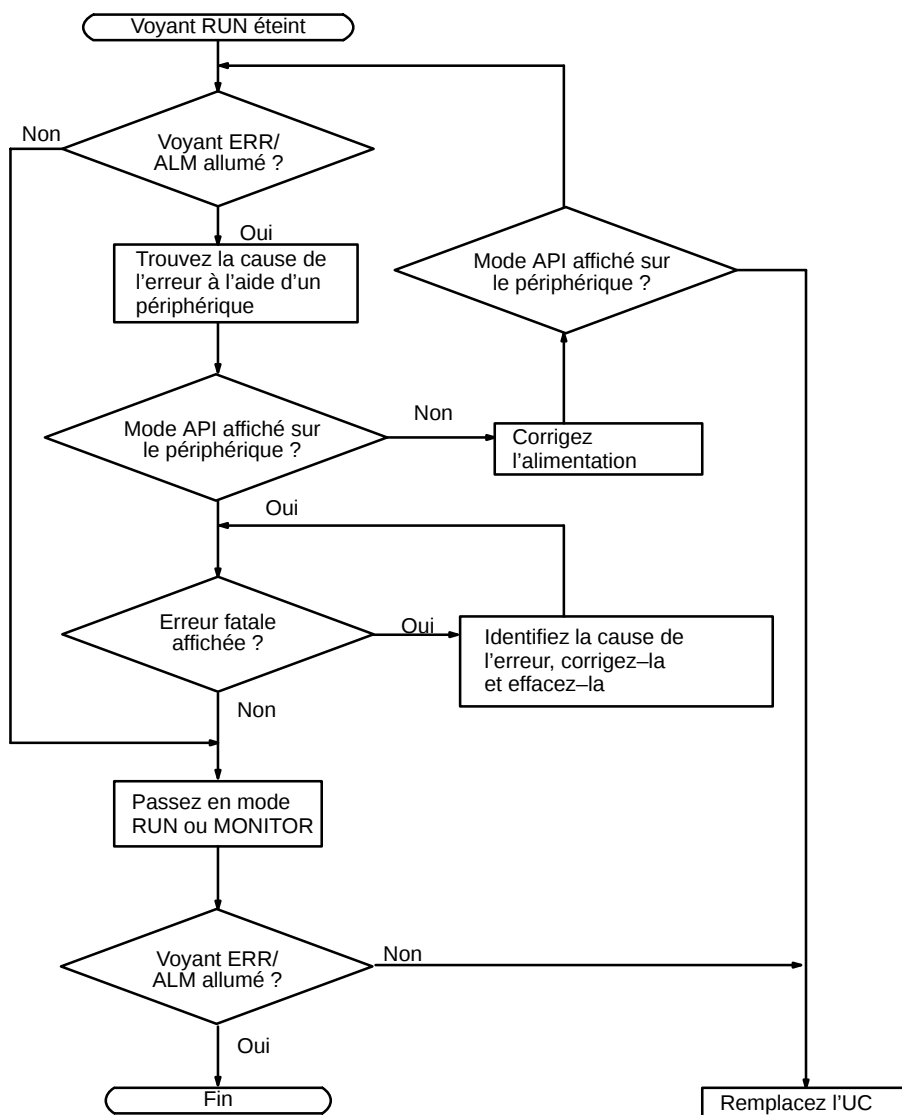
Vérification de l'alimentation



Rem. : gamme de tension admissible pour le CQM1 : 85 à 264 Vc.a.

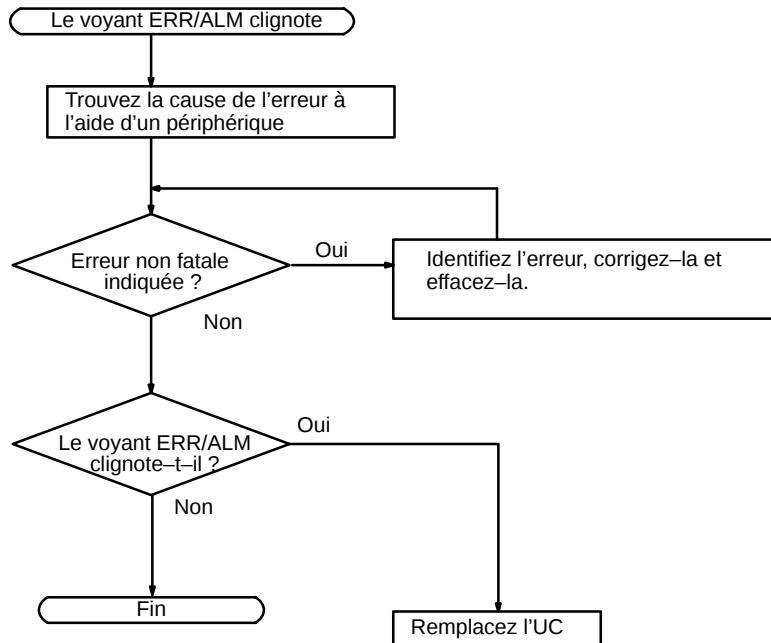
Vérification des erreurs fatales

Résolution des problèmes causés par des erreurs fatales pendant que le voyant de puissance est allumé :



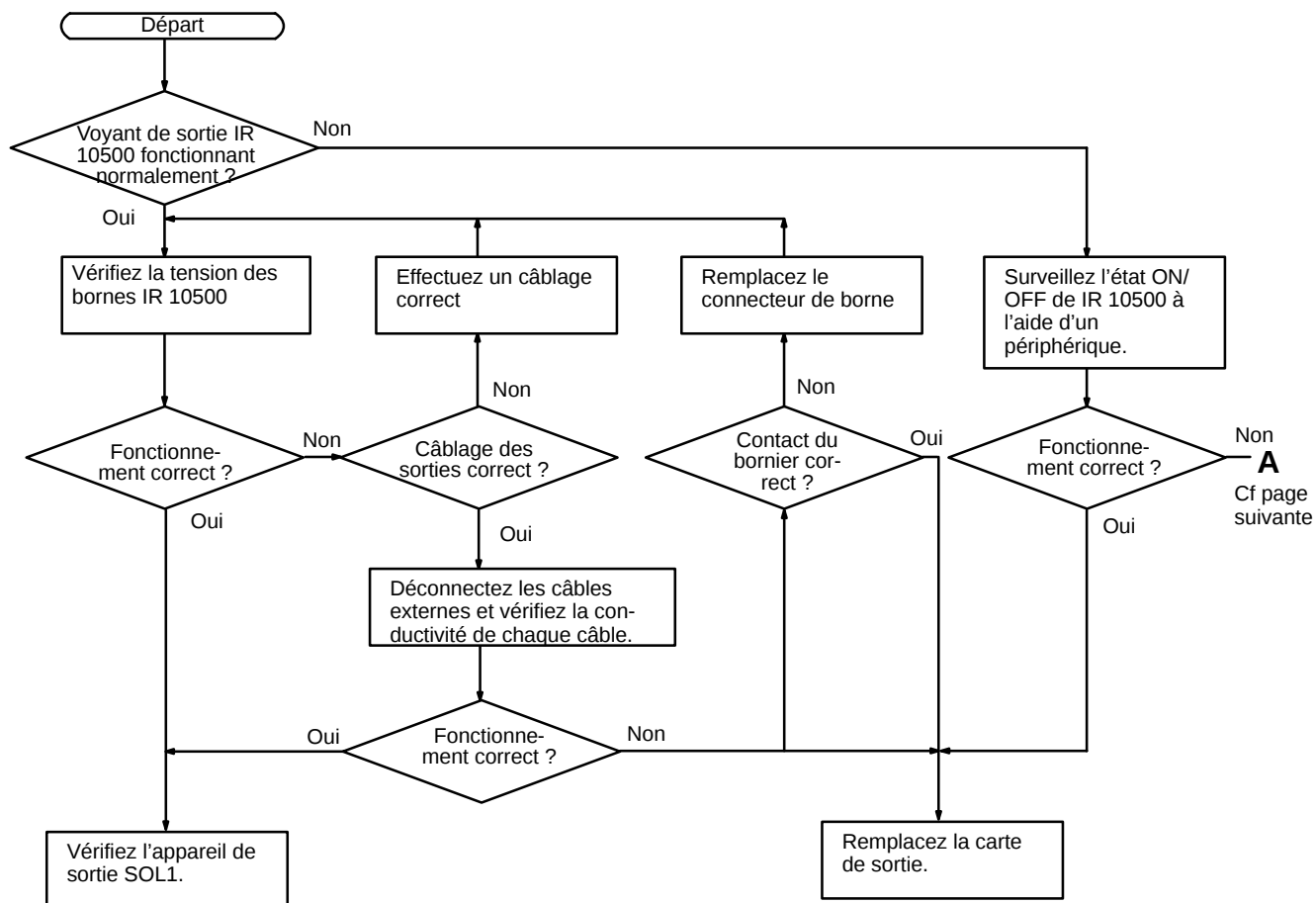
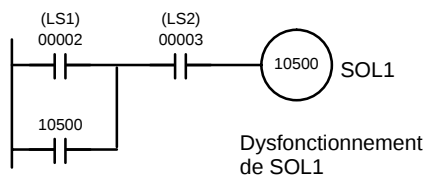
Vérification des erreurs non fatales

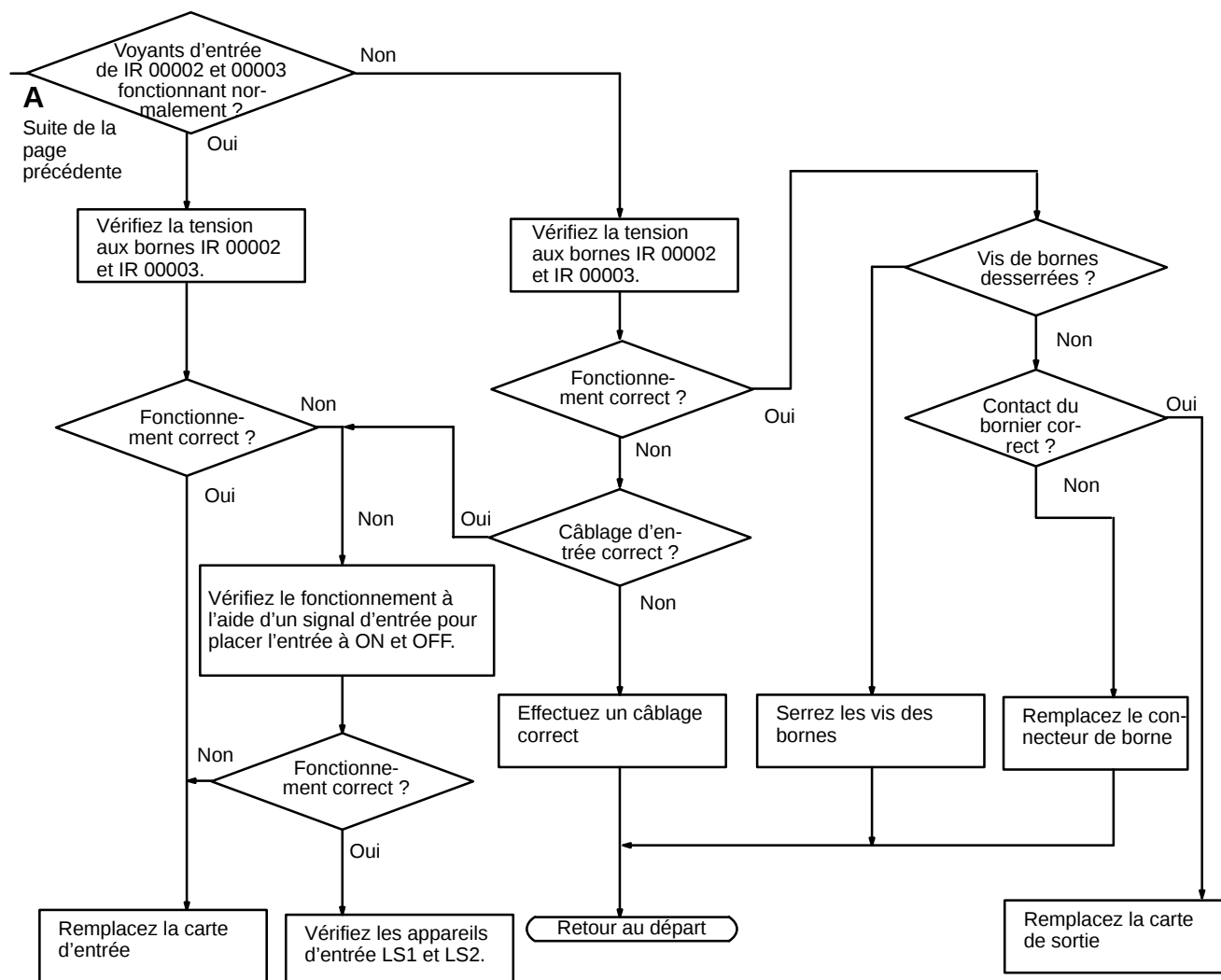
Bien que l'API continue de fonctionner en cas d'erreur non fatale, ces erreurs doivent être corrigées. Il peut être nécessaire d'arrêter l'API pour cela.



Vérification des E/S

Le diagramme suivant repose sur la partie de programme ci-dessous :





Vérification des conditions de fonctionnement

