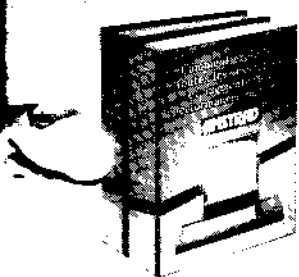


Comment exploiter
toutes les ressources
et augmenter les
performances de votre

AMSTRAD



ÉDITIONS WEKA



Comment exploiter toutes les ressources et augmenter les performances de votre AMSTRAD CPC 464 / 664 / 6128

AM/2 9400

Tables des matières détaillées des compléments déjà parus

N° 5 (158 pages + 2 mylars)

Présentation générale

Dictionnaire technique Français-Anglais, Anglais-Français

Langages du CPC

- Liste alphabétique des primitives du LOGO

Logiciels à "caractère professionnel"

- DBASE II
- Programmation sous DBASE II - Introduction.

Programmes

- Tours de Hanoi
- Jeu des allumettes
- Awari
- Biorythmes
- Gestion de compte bancaire

Fabrication de circuits additionnels pour Amstrad

Construisez vos micro-ordinateurs
Prises et connecteurs

Maintenance

- Le lecteur de cassettes du CPC 464

Notions scientifiques de base

- Electronique logique

N° 6 (182 pages + 2 mylars)

Systèmes d'exploitation des 464 et 6128

- Programmation sous CP/M plus
- Table d'indirection des fonctions BDOS en page zéro (suite)

Logiciels à "caractère professionnel"

- Pocket Wordstar
- Les commandes de Wordstar
- Multiplan
- Applications de Multiplan : budget familial.
- DBASE II
- DBASE II programmation avancée : le travail multi-fichiers, l'intégration de modules binaires, l'option DELIMITED

Périphériques

- Systèmes et méthodes de transfert de fichiers
- Copies et transmissions de cassettes
- L'interface RS-232
- Les modems

Programmes

- Commande PIP en Basic
- Transformation du clavier Qwerty en clavier Azerty sous CP/M plus.

Fabrication de circuits additionnels pour Amstrad

- Commande d'un train électrique à partir de l'Amstrad
- Commande d'un projecteur de diapositives
- Un redirecteur d'appels téléphoniques
- Contrôle d'alarme à transmission téléphonique
- Prises et connecteurs

N° 7 (170 pages + 2 mylars)

Langages du CPC

- Identificateurs standard

Son

- Emission d'un Beep sonore en utilisant la macro du Firmware MC SOUND REGISTER

Logiciels à "caractère professionnel"

- Calcumat : le tableur/graphueur

Périphériques

- L'interface RS-232
- La RS-232 Amstrad
- Connexions et programmes

Programmes

- Checksum, vérificateur de données
- Dump hexadécimal et ASCII

Fabrication de circuits additionnels pour Amstrad

- Un compositeur acoustique de numéros de téléphone
- Technologie des montages électroniques

N° 8 (170 pages)

Langages du CPC

- Les banques ROM en Firmware
- Utilisation du Turbo-Pascal
- Optimisation d'écriture dans un fichier texte
- Définition de routines sonores
- Position du curseur sur l'écran
- Le langage Forth 83-Standard pour Amstrad 464, 664, 6128 et PCW
- Le langage Forth sur les Amstrad
- Le compilateur Forth
- Contrôle de l'affichage
- Edition des programmes écrits en Forth

Logiciels à "caractère professionnel"

- Calcumat par le détail

Périphériques

- Les mémoires de masse
- Le lecteur de disquettes 5 1/4 Vortex en lecteur additionnel

Programmes

- Récupération d'un fichier effacé par la commande IERA

Fabrication de circuits additionnels pour Amstrad

- Commande de moteurs pas à pas

N° 9 (176 pages)

Langages du CPC

- Initiation au langage machine
- Les instructions RESTART des CPC
- Programmation d'un traitement de texte

Graphisme

- Graphicomanies
- Jeux de points

Logiciels à "caractère professionnel"

- Tasword
- Les fiches de référence
- Index alphabétique DBASE II
- Index thématique DBASE II
- L'utilitaire Zip

Périphériques

- Le modem Digitelec DTL 2000

Programmes

- Défilement d'un message alphanumérique sur l'écran
- Driver d'imprimante DMP 2000

N° 10 (174 pages + 2 mylars)

Graphisme

- Jeux de lignes
- Les espaces inconnus

Logiciels à "caractère professionnel"

- Tasprint

Périphériques

- Réalisation d'un serveur télématique
- La structure matérielle
- La structure logicielle
- Les routines de base

Programmes

- Analyse syntaxique d'une phrase
- Instruction CAT évoluée
- Edition et modification des secteurs d'une disquette
- Traitement de texte
- Mise en œuvre d'utilitaires
- Le traitement de texte Weka
- Fonctions élémentaires

Fabrication de circuits additionnels pour Amstrad

- Un relais de fréquence pour sortie audio
- Amstrad et hi-fi
- Un amplificateur de casque stéréo

N° 11 (174 pages)

Conception matérielle des CPC

- Exploitation du PIO 8255

Langages du CPC

- Les RSX

Logiciels à "caractère professionnel"

- Masterfile III
- Présentation générale
- Utilisation de Masterfile III
- Fonctions avancées et adaptations

Périphériques

- Le programme du serveur
- Multiface II, sauvegarde mémoire

Programmes

- Bataille navale
- Danger piranhas
- CAPS LOCK interactif
- Correcteurs orthographiques
- Correcteur orthographique de base

Fabrication de circuits additionnels pour AMSTRAD

- Le wrapping (connexions enroulées)

N° 12 (168 pages + 2 mylars)

Périphériques

- Le programme du serveur
- OUT et les ports de sortie

Programmes

- Fonction LOCATE-INPUT
- Fonction HELP
- Turbo copie d'écran graphique
- Protection écran (screen saver)
- Filtrage de fichiers ASCII
- Gestion de logiciels
- Premier jeu de fonctions évoluées

Fabrication de circuits additionnels pour AMSTRAD

- Un lecteur-enregistreur de télécartes usagées
- Technologie de pointe
- Pupitre de saisie

N° 13 (172 pages + 1 mylar)

Systèmes d'exploitation des 664 et 6128

- Tables d'indirection des fonctions BDOS en page zéro (suite)

Langages des CPC

- Basic approfondi
- SYMBOL et SYMBOL AFTER
- L'instruction CALL et les RSX en Basic
- Identificateurs standard
- Travail en Assembleur 8080 sous CP/M 2.2 ou CP/M Plus
- Les instructions du 8080

Logiciels à "caractère professionnel"

- Les commandes de Wordstar (suite)

Périphériques

- Création de jeux d'aventures
- Chargeur hexadécimal
- Filtrage de fichiers ASCII

Fabrication de circuits additionnels pour Amstrad

- Une carte à 8 entrées analogiques
- Support de moniteur

N° 14 (182 pages + 1 mylar)

Langages du CPC

- La gestion des variables dans les Amstrad CPC
- Utilisation des vecteurs du système d'exploitation sous Basic

Son

- La synthèse vocale
- Le synthétiseur vocal TECHN-MUSIQUE

Logiciels à "caractère professionnel"

- Le compilateur dBase II DB COMPILER de Wordtech

Programmes

- Exécution de jeux d'aventures
- Transformez votre Amstrad CPC + DMP 2000 en machine à écrire
- Second jeu de fonctions évaluées

Fabrication de circuits additionnels pour AMSTRAD

- Support d'imprimante 80 colonnes avec récepteur de listing
- Amstrad et Vidéo
- Mariez votre unité centrale avec d'autres écrans de visualisation

Maintenance

- Entretien des claviers,
- Entretien des écrans
- Entretien des lecteurs de disquettes
- Entretien des imprimantes

Notions scientifiques de base

- Eléments de mathématiques générales
- Langage des ensembles
- Notions générales de géométrie
- Notions générales de trigonométrie



EDITIONS WEKA

Chère Cliente, Cher Client,

Nous sommes heureux de vous adresser aujourd'hui votre exemplaire de l'ouvrage *"Comment exploiter toutes les ressources et augmenter les performances de votre Amstrad..."*, et nous vous remercions de la confiance que vous nous accordez.

Cet ouvrage, nous en sommes convaincus, vous sera d'une aide précieuse pour tirer le meilleur parti de votre micro-ordinateur et exploiter à fond toutes ses possibilités.

Au fil de ses 1 224 pages, vous pourrez en effet :

- maîtriser les principaux langages et approfondir vos connaissances en programmation ;
- créer des programmes de jeux, de gestion, de création graphique et sonore ;
- utiliser les logiciels tels que Multiplan, D Base II avec l'aisance des professionnels et pour vos besoins personnels ;
- découvrir de nombreux programmes prêts à tourner, utiles ou divertissants ;
- bénéficier des "trucs" et "astuces" que mettent à votre disposition nos experts ;
- accroître les performances de votre micro en l'équipant des montages proposés ;
- diagnostiquer les pannes et réparer votre Amstrad en vous aidant des schémas, circuits, vues éclatées... et nombreux conseils.

Dès maintenant, vous bénéficiez de notre service exclusif : et vous recevrez, à ce titre, dans les prochaines semaines, un complément que vous pourrez examiner chez vous, pendant 15 jours, en bénéficiant de la garantie Weka "satisfait ou remboursé".

Nous partageons votre passion et espérons vivement que cet ouvrage répondra à vos souhaits.

Bien cordialement,

L'Editeur.

Comment exploiter toutes les ressources et augmenter les performances de votre

AMSTRAD

CPC
464 / 664
6128

15^e Complément

Guy Aletti
Jean-Marc Campaner
Jean-Paul Carré
Patrick Gueulle
Michel Martin
Marc Petremann
Christine Saincir
Philippe Tixier

EDITIONS WEKA

Paris - Kissing - Zurich - Milan - Amsterdam - Vienne - Londres - New York

Avertissement

Le chapitre sur Turbo-Pascal développé dans cet ouvrage n'est pas un substitut du manuel d'utilisation livré avec le logiciel. C'est un chapitre complémentaire destiné aux personnes qui veulent se perfectionner à l'utilisation de ce logiciel.

- Turbo-Pascal® est un logiciel conçu et commercialisé par **Borland International Inc.**, qui en détient le copyright à l'échelon mondial.

- « Turbo-Pascal » ou « Turbo C » est une marque enregistrée de Borland International Inc.

Les Editions WEKA rappellent que selon la loi n° 85660 du 3 juillet 1985 tout utilisateur de ce logiciel doit être en possession de l'original. Il a toutefois le droit de faire une copie pour ses besoins propres, cette copie n'étant pas cessible à une tierce personne.

Extrait de la loi du 3 juillet 1985

« Toute reproduction autre que l'établissement d'une copie de sauvegarde par l'utilisateur, ainsi que toute utilisation d'un logiciel non expressément autorisée par l'auteur ou ses ayants droit, est passible des sanctions prévues par la loi. »

Tous droits de traduction, d'adaptation et de reproduction de la version française
par tous procédés réservés pour tous pays

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1^{er} de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal.

15^e complément

© 1989 Editions WEKA, 82, rue Curial, 75935 Paris cedex 19
Tél. : (1) 40 37 01 00 - Télex 210 504

Editions WEKA - Paris - Kissing - Zurich - Milan - Amsterdam - Vienne - Londres - New York

En Suisse : WEKA Verlag, Hermetschloostr. 77, Zurich

Editeur : Brigitte Morvant

Secrétariat d'édition : Brigitte Chevallier

Fabrication : Tina Le Xuan

Composition : Compo-Méca - 64990 Mouguerre

Corlet imprimeur s.a. - 14110 Condé-sur-Noireau

Reproduction interdite - Tous droits réservés

Imprimé en France, 1989

Dépôt légal (ouvrage de base) : 2^e trimestre 1987

ISBN 2-7337-0041-3

New York - Londres - Paris

Partie 1

Présentation générale

1/0

Table des matières générale

Tome 1**1 Présentation générale**

- 1/0** Table des matières générale
- 1/1** Le système CPC : A quoi et à qui va-t-il servir ?
Carte contact-lecteurs « Ecrivez-nous »
- 1/2** Glossaire
- 1/3** Index
- 1/4** Dictionnaire technique Français-Anglais/Anglais-Français

2 Conception matérielle des CPC

- 2/0** Table des matières
- 2/1** Architecture interne
- 2/2** Schémas des cartes mères des Amstrad CPC 464, 664 et 6128
 - 2/2.1 CPC 464
 - 2/2.2 CPC 664
 - 2/2.3 CPC 6128
 - 2/2.4 Différences de conception
- 2/3** Les circuits intégrés spécialisés
 - 2/3.1 Le microprocesseur Z80
 - 2/3.2 Le contrôleur d'écran : CRTC et VGA
 - 2/3.2.1 CRT Controller (ou CRTC)
 - 2/3.3 Le circuit sonore AY3-8912
 - 2/3.4 L'interface parallèle PIO 8255A
 - 2/3.4.1 Exploitation du PIO 8255
 - 2/3.5 Le contrôleur de disque μ PD 765AC

2/4 Les mémoires

- 2/4.1 La mémoire morte (ROM)
- 2/4.2 La mémoire vive (RAM)

2/5 Les circuits intégrés standards

- 2/5.1 Les circuits intégrés linéaires
- 2/5.2 Les circuits intégrés digitaux

2/6 L'horloge interne**3 Systèmes d'exploitation des 664 et 6128****3/0 Table des matières****3/1 Introduction au DOS****3/2 AMSDOS : Définitions, rappels et utilisations**

- 3/2.1 Liste alphabétique des mots clés

3/3 CP/M 2.2 : Définitions, rappels et utilisations

- 3/3.1 Liste alphabétique des mots clés

3/4 CP/M + : Définitions, rappels et utilisations

- 3/4.1 Liste alphabétique des mots clés
- 3/4.2 Programmation sous CP/M plus
 - 3/4.2.1 Organisation mémoire du 6128
 - 3/4.2.1.1 FCB (File Control Bloc)
 - 3/4.2.1.2 SCB (System Control Bloc)
 - 3/4.2.2 Table d'indirection des fonctions BDOS en page zéro

4 Langages du CPC**4/0 Table des matières****4/1 Locomotive BASIC : Définitions et rappels de base**

- 4/1.1 Pourquoi utiliser le BASIC et dans quels domaines
- 4/1.2 Version 1.0 sur CPC 464 : Mots clés et leur utilisation
- 4/1.3 Version 1.1 sur CPC 664 et CPC 6128 : Extensions par rapport à la version 1.0
- 4/1.4 Rappel des ordres BASIC et de leur fonction
- 4/1.5 Cours de programmation
- 4/1.6 Basic approfondi
 - 4/1.6.1 SYMBOL et SYMBOL AFTER
 - 4/1.6.2 L'instruction CALL et les RSX en Basic
 - 4/1.6.3 La gestion des variables dans les Amstrad CPC
 - 4/1.6.4 Utilisation des vecteurs du système d'exploitation sous Basic
 - 4/1.6.5 Formater une disquette sous Basic
 - 4/1.6.6 Accélérez vos programmes Basic - Les tokens

4/2 Assembleur Z80 : Définitions et rappels de base

- 4/2.1 Pourquoi utiliser l'assembleur et dans quels domaines ?
- 4/2.2 Les modes d'adressage
- 4/2.3 Les mots clés de l'assembleur Z80 et leur utilisation
- 4/2.4 Liste alphabétique des codes opératoires de l'assembleur Z80
- 4/2.5 Cours de programmation
 - 4/2.5.1 Initiation au langage machine
- 4/2.6 Assembleurs existants
 - 4/2.6.1 DEVPAC
- 4/2.7 Les banques ROM ou FIRMWARE
- 4/2.8 Les instructions RESTART des CPC
- 4/2.9 Les RSX
- 4/2.10 Accès aux vecteurs mathématiques en Assembleur et utilisation
- 4/2.11 Les interruptions sur Amstrad

Tome 2**4/3 LOGO : Définitions et rappels de base**

- 4/3.1 Pourquoi utiliser le LOGO et dans quels domaines ?
- 4/3.2 Les mots clés du LOGO et leur utilisation
- 4/3.3 Liste alphabétique des primitives du LOGO
- 4/3.4 Franciser le Dr. LOGO de l'Amstrad
- 4/3.5 Programmes d'application
 - 4/3.5.1 Dump mémoire en LOGO 2 et LOGO 3
 - 4/3.5.2 Devine le nombre en LOGO 2 et LOGO 3
 - 4/3.5.3 Chiffres romains en LOGO 2 et LOGO 3
 - 4/3.5.4 LOTO en LOGO 2 ou LOGO 3
 - 4/3.5.5 Histogrammes en LOGO 2 et LOGO 3

4/4 Turbo-PASCAL® : Définitions et rappels de base

- 4/4.1 Pourquoi utiliser le PASCAL et dans quels domaines ?
- 4/4.2 La programmation structurée en PASCAL
- 4/4.3 Les mots réservés de Turbo-Pascal
- 4/4.4 Identificateurs standard
- 4/4.5 Utilisation du Turbo-Pascal
 - 4/4.5.1 Optimisation d'écriture dans un fichier texte
 - 4/4.5.2 Définition de routines sonores
 - 4/4.5.3 Position du curseur sur l'écran
 - 4/4.5.4 Programmation d'un traitement de texte

4/5 Le langage Forth 83- Standard pour Amstrad 464, 664, 6128 et PCW

- 4/5.1 Le langage Forth sur les Amstrad
- 4/5.2 Le compilateur Forth
- 4/5.3 Contrôle de l'affichage
- 4/5.4 Edition des programmes écrits en Forth

4/6 Travail en Assembleur 8080 sous CP/M 2.2 ou CP/M Plus

- 4/6.1 Les instructions du 8080

5 Graphisme**5/0 Table des matières****5/1 Généralités****5/2 Tracé de points en BASIC****5/3 Déplacement du curseur graphique en BASIC****5/4 Tracé de droites en BASIC****5/5 Test de la couleur d'un point en BASIC****5/6 Tracé de points et de droites en ASSEMBLEUR****5/7 La mémoire d'écran****5/8 Caractères graphiques et signes spéciaux****5/9 Sprites : Définition de caractères par l'utilisateur**

- 5/9.1 Définition de caractères multiples

5/10 Logiciels

- 5/10.1 Programme de dessin

- 5/10.2 Utilitaires de manipulation de dessin

- 5/10.2.1 Reproduction de blocs graphiques

- 5/10.2.2 Miroir par rapport à un axe vertical

- 5/10.2.3 Miroir par rapport à un axe horizontal

- 5/10.3 Utilitaires de compactage

- 5/10.3.1 Compactage filiforme

- 5/10.3.2 Compacteurs monochromes en mode 1

- 5/10.4 Graphicomanies

- 5/10.4.1 Jeux de points

- 5/10.4.2 Jeux de lignes

- 5/10.4.3 Les espaces inconnus

6 Son**6/0 Table des matières****6/1 Définitions****6/2 Introduction au circuit sonore AY3-8912****6/3 Programmation du circuit sonore en BASIC****6/4 Programmation du circuit sonore en ASSEMBLEUR****6/5 Logiciels**

- 6/5.1 Fichiers musicaux sous interruptions

6/5.2 Emission d'un Beep sonore en utilisant la macro du firmware MC SOUND REGISTER

6/6 La synthèse vocale

6/6.1 Le synthétiseur vocal TECHNI-MUSIQUE

Tome 3

7 Logiciels à « caractère professionnel »

7/0 Table des matières

7/1 Traitements de texte

7/1.1 Pocket Wordstar

7/1.1.1 Les commandes de Wordstar

7/1.2 Tasword

7/1.2.1 Les fiches de référence

7/1.3 Tasprint

7/2 Tableurs

7/2.1 MULTIPLAN

7/2.1.1 Organisation de MULTIPLAN

7/2.1.2 Les commandes de MULTIPLAN

7/2.1.3 Les fonctions de MULTIPLAN

7/2.1.4 Ce qu'il faut savoir pour programmer sous MULTIPLAN 1,06

7/2.1.5 Applications de MULTIPLAN

7/2.1.5.1 Budget familial

7/2.2 Calcumat : le tableur/grapheur

7/2.2.1 Initiation à Calcumat

7/2.2.1.1 L'édition des cellules

7/2.2.1.2 Préparation du tableau et de ses commentaires

7/2.2.1.3 Définition des formules de calcul

7/2.2.1.4 Créer un graphe

7/2.2.1.5 Sauver le tableau avec ses paramètres de présentation

7/2.2.2 Calcumat par le détail

7/2.2.2.1 Menu Micro-Application

7/2.2.2.2 Menu Fichier

7/2.2.2.3 Menu Edition

7/2.2.2.4 Menu Saisie

7/2.2.2.5 Menu Nombres

7/2.2.2.6 Menu Choix divers

7/2.2.2.7 Menu Graphes

7/3 Gestionnaires de bases de données (SGBD)

7/3.0 Index

7/3.1 DBASE II

7/3.1.1 DBASE II en mode commande

7/3.1.1.1 Création de la structure du fichier

7/3.1.1.2 Vérification et modification de la structure

Partie 1 : Présentation générale

- 7/3.1.1.3 Saisie des fiches
- 7/3.1.1.4 Consultation du fichier
- 7/3.1.1.5 Correction des fiches
- 7/3.1.1.6 Suppression des fiches
- 7/3.1.1.7 Tri des fiches
- 7/3.1.1.8 Indexation du fichier
- 7/3.1.1.9 La production d'états
- 7/3.1.1.10 Totalisations sur le fichier
- 7/3.1.1.11 La gestion du disque
- 7/3.1.1.12 Modification des paramètres d'état
- 7/3.1.2 DBASE II en mode programme
- 7/3.1.2.1 Création d'un masque de saisie
- 7/3.1.2.2 Variables et macros
- 7/3.1.2.3 Traitement des variables numériques
- 7/3.1.2.4 Traitement des variables chaînes
- 7/3.1.2.5 Autres fonctions
- 7/3.1.3 Programmation sous DBASE II
- 7/3.1.3.1 Cadre de présentation
- 7/3.1.3.2 Effacement sélectif de l'écran
- 7/3.1.3.3 Menu principal
- 7/3.1.3.4 Ajout de fiches
- 7/3.1.3.5 Corriger le fichier
- 7/3.1.3.6 Tri du fichier
- 7/3.1.3.7 Edition du fichier
- 7/3.1.4 DBASE II programmation avancée
- 7/3.1.4.1 Le travail multifichier
- 7/3.1.4.2 L'intégration de modules binaires
- 7/3.1.4.3 Les échanges de données. L'option : DELIMITED
- 7/3.1.5 L'utilitaire Zip
- 7/3.1.6 Le compilateur dBase II DB COMPILER de Wordtech
- 7/3.2 Masterfile III
- 7/3.2.1 Présentation générale
- 7/3.2.2 Utilisation de Masterfile III
- 7/3.2.3 Fonctions avancées et adaptations

8 Périphériques

8/0 Table des matières

8/1 Les connecteurs de l'AMSTRAD

- 8/1.1 La prise d'extension
- 8/1.2 La prise « unité de disquette 2 »

8/2 Module péritel pour téléviseur couleur

- 8/2.1 Module commercialisé
- 8/2.2 Comment construire soi-même un module ?
- 8/2.3 Quelques branchements vidéo

8/3 Imprimantes

- 8/3.1 Quelle imprimante choisir ?
- 8/3.1.1 L'imprimante Citizen 120-D

- 8/3.2 L'interface CENTRONICS
- 8/3.3 Le câble de liaison : achat ou réalisation ?

8/4 La souris AMX Mouse

8/5 Systèmes et méthodes de transfert de fichiers

- 8/5.1 Copies et transmissions de cassettes
- 8/5.2 L'interface RS-232
 - 8/5.2.1 La liaison RS-232
 - 8/5.2.2 Le code ASCII
 - 8/5.2.3 La RS-232 Amstrad
 - 8/5.2.4 Connexions et programmes
 - 8/5.2.4.1 Connexion réduite
 - 8/5.2.4.2 Connexion à un Minitel
- 8/5.3 Les modems
 - 8/5.3.1 Le modem Digitelec DTL 2000
- 8/5.4 Réalisation d'un serveur télématique
 - 8/5.4.1 La structure matérielle
 - 8/5.4.2 La structure logicielle
 - 8/5.4.3 Les routines de base
 - 8/5.4.4 Le programme du serveur
- 8/5.5 OUT et les ports de sortie

8/6 Les mémoires de masse

- 8/6.1 Le lecteur de disquettes 5 1/4 Vortex en lecteur additionnel
- 8/6.2 Multiface II, sauvegarde mémoire
- 8/6.3 Brancher le lecteur du CPC 464 sur le CPC 6128

Tome 4

9 Programmes

9/0 Table des matières

9/1 Savoir programmer

9/2 Moniteur : Assembleur/Désassembleur/Debugger

- 9/2.1 Le Désassembleur
- 9/2.2 L'Assembleur
- 9/2.3 Le Debugger

9/3 Jeux d'esprit

- 9/3.1 Le jeu du taquin
- 9/3.2 Renservé
- 9/3.3 Tours de Hanoi
- 9/3.4 Jeu des allumettes
- 9/3.5 Awari
- 9/3.6 Jeu du Simon

9/4 Mathématiques

- 9/4.1 Nom d'un jour de la semaine
- 9/4.2 Calendrier perpétuel
- 9/4.3 Biorythmes

9/5 Gestion de fichiers**9/6 Jeux d'aventures**

- 9/6.1 Analyse syntaxique d'une phrase
- 9/6.2 Fonction LOCATE-INPUT
- 9/6.3 Fonction HELP
- 9/6.4 Création de jeux d'aventures
- 9/6.5 Exécution de jeux d'aventures

9/7 Jeux d'Arcade

- 9/7.1 Casse briques
- 9/7.2 Bataille navale
- 9/7.3 Danger piranhas

9/8 Utilitaires

- 9/8.1 Copie d'écran graphique
 - 9/8.1.1 Turbo copie d'écran graphique
- 9/8.2 Commande PIP en Basic
- 9/8.3 Transformation du clavier QWERTY en clavier AZERTY sous CP/M Plus
- 9/8.4 Checksum, vérificateur de données
- 9/8.5 Dump hexadécimal et ASCII
 - 9/8.5.1 Programme de Dump en Basic
 - 9/8.5.2 Programme de Dump en Assembleur
- 9/8.6 Récupération d'un fichier effacé par la commande IERA
- 9/8.7 Défilement d'un message alphanumérique sur l'écran
- 9/8.8 Driver d'imprimante DMP 2000
- 9/8.9 Instruction CAT évoluée
- 9/8.10 Edition et modification des secteurs d'une disquette
- 9/8.11 CAPS LOCK interactif
- 9/8.12 Protection écran (*Screen saver*)
- 9/8.13 Chargeur hexadécimal
- 9/8.14 Formattage des listings

9/9 Programmes divers

- 9/9.1 Générateur de signaux morse
- 9/9.2 Filtrage de fichiers ASCII
- 9/9.3 Transformez votre Amstrad CPC + DMP 2000 en machine à écrire

9/10 Gestion familiale

- 9/10.1 Gestion de compte bancaire
- 9/10.2 Gestion de logiciels

9/11 Traitement de texte

- 9/11.1 Mise en œuvre d'utilitaires

- 9/11.2 Le traitement de texte Weka
 - 9/11.2.1 Fonctions élémentaires
 - 9/11.2.2 Premier jeu de fonctions évoluées
 - 9/11.2.3 Second jeu de fonctions évoluées
- 9/12 **Correcteurs orthographiques**
 - 9/12.1 Correcteur orthographique de base

Tome 5**10 Fabrication de circuits additionnels pour AMSTRAD**

- 10/0 **Table des matières**
- 10/1 **Connexion de l'AMSTRAD au Minitel**
 - 10/1.1 Signaux de la prise péri-informatique
 - 10/1.2 De l'AMSTRAD vers le Minitel
 - 10/1.2.1 Programmation du Minitel
- 10/2 **Connexion des AMSTRAD CPC 664 et CPC 6128 à un magnétophone à cassettes**
- 10/3 **Commande de circuits TTL, CMOS et de puissance**
 - 10/3.1 Régulation de chauffage
 - 10/3.1.1 Réalisation d'une régulation de chauffage
 - 10/3.2 Commande d'un train électrique à partir de l'Amstrad
 - 10/3.3 Commande d'un projecteur de diapositives
 - 10/3.4 Commande de moteurs pas à pas
 - 10/3.5 Un relais de fréquence pour sortie audio
 - 10/3.6 Une carte à 8 entrées analogiques
- 10/4 **AMSTRAD et téléphonie**
 - 10/4.1 Un « mouchard » téléphonique
 - 10/4.2 Un composeur de numéros de téléphone
 - 10/4.3 Détecteur de sonnerie téléphonique
 - 10/4.4 Un redirecteur d'appels téléphoniques
 - 10/4.5 Centrale d'alarme à transmission téléphonique
 - 10/4.6 Un composeur acoustique de numéros de téléphone
 - 10/4.7 Un lecteur-enregistreur de télécartes usagées
- 10/5 **Mémoires d'ordinateur**
 - 10/5.1 Un programmeur de mémoires EPROM
- 10/6 **Construisez vos micro-ordinateurs**
- 10/7 **Prises et connecteurs**
- 10/8 **Technologie des montages électroniques**
 - 10/8.1 Le wrapping (connexions enroulées)

10/9 Amstrad et hi-fi

10/9.1 Un amplificateur de casque stéréo

10/10 Technologie de pointes

10/10.1 Pupitre de saisie

10/10.2 Support de moniteur

10/10.3 Support d'imprimante 80 colonnes avec récepteur de listing

10/10.4 Bras support pour moniteur

10/11 Amstrad et vidéo

10/11.1 Mariez votre unité centrale avec d'autres écrans de visualisation

11 Annexes

Annexe 1 : Codes de contrôle et caractères ASCII

Annexe 2 : Codes opératoires de l'Assembleur Z80
et leur codage en hexadécimal

Annexe 3 : Caractères standard et graphique de l'Amstrad CPC 464

12 Maintenance**12/0 Table des matières****12/1 Soyez votre propre dépanneur****12/2 Maintenance des unités centrales****12/3 Maintenance des périphériques**

12/3.1 Le lecteur de cassettes du CPC 464

12/3.2 Le lecteur de disquettes FD 1

12/4 Prenez soin de votre AMSTRAD

12/4.1 Entretien des claviers

12/4.2 Entretien des écrans

12/4.3 Entretien des lecteurs de disquettes

12/4.4 Entretien des imprimantes

12/5 Maintenance des moniteurs**12/6 La mécanique des AMSTRAD****13 Notions scientifiques de base****13/1 Introduction à l'électronique**

13/1.1 Electronique analogique

13/1.2 Electronique logique

13/2 Éléments de mathématiques générales

- 13/2.1 Langage des ensembles
 - 13/2.1.1 Ensembles des nombres
 - 13/2.1.2 Notions de numérotation
- 13/2.2 Notions générales de géométrie
- 13/2.3 Notions générales de trigonométrie
- 13/2.4 Notions d'analyse
 - 13/2.4.1 Aperçu sur les fonctions polaires et paramétriques en sinus et cosinus

Ecrivez-nous !

ces cartes vous permettent de joindre la rédaction de "Comment exploiter toutes les ressources et augmenter les performances de votre **AMSTRAD** pour

- poser des questions
- faire des suggestions
- émettre des souhaits
- et... critiquer

Carte de contact lecteur

Expéditeur

Prrière d'écrire lisiblement

Nom / Prénom

Profession

N° Rue

Code postal Ville

Code Client

Carte postale

Affranchir
au tarif
carte-postale

Editions WEKA

82, rue Curial

75935 PARIS Cedex 19

(France)

Carte de contact lecteur

Expéditeur

Prrière d'écrire lisiblement

Nom / Prénom

Profession

N° Rue

Code postal Ville

Code Client

Carte postale

Affranchir
au tarif
carte-postale

Editions WEKA

82, rue Curial

75935 PARIS Cedex 19

(France)

Carte de contact lecteur

Expéditeur

Prrière d'écrire lisiblement

Nom / Prénom

Profession

N° Rue

Code postal Ville

Code Client

Carte postale

Affranchir
au tarif
carte-postale

Editions WEKA

82, rue Curial

75935 PARIS Cedex 19

(France)

1

CARTE - CONTACT**"Comment exploiter toutes les ressources et augmenter les performances de votre AMSTRAD"**

Je me réfère à la partie Chapitre

Page à page et désire formuler à ce sujet la
remarque suivante :

2

CARTE - CONTACT**"Comment exploiter toutes les ressources et augmenter les performances de votre AMSTRAD"**

Je me réfère à la partie Chapitre

Page à page et désire formuler à ce sujet la
remarque suivante :

3

CARTE - CONTACT**"Comment exploiter toutes les ressources et augmenter les performances de votre AMSTRAD"**

Je me réfère à la partie Chapitre

Page à page et désire formuler à ce sujet la
remarque suivante :

1/1

Le système CPC : à quoi et à qui va-t-il servir ?

Pour un choix judicieux...

Quel que soit le CPC que vous possédiez, vous avez fait un bon choix, car cette gamme d'ordinateurs a su se différencier de ses concurrents en offrant des avantages non négligeables :

- rapport qualité/prix exceptionnel ;
- système écran/clavier/lecteur de cassettes ou écran/clavier/lecteur de disquettes livré complet dans la version de base ;
- clavier de bonne qualité ;
- aucune alimentation extérieure à l'ordinateur ;
- BASIC résident de très bonne qualité.

Si vous possédez un CPC 464 que vous voulez utiliser professionnellement, il vous faudra y adjoindre un lecteur de disquettes car la plupart des logiciels professionnels sont disponibles sur disquettes.

Les lecteurs de disquettes classiques trois pouces ont une capacité de stockage assez limitée, mais plusieurs constructeurs proposent de connecter à l'AMSTRAD un lecteur de disquettes au format trois pouces et demi d'une capacité approchant le méga-octet.

Si vous destinez votre ordinateur au jeu, vous aurez l'embarras du choix, car plus d'un éditeur s'est penché sur cette machine. Certains se sont même spécialisés dans l'édition de logiciels sur CPC. Les jeux proposés sont généralement de bonne qualité, d'un prix très abordable en version cassette mais un peu moins abordable en version disquette. A vous de savoir si vous aurez la patience d'attendre que le logiciel soit chargé en mémoire pour jouer...

Enfin, si vous débutez en informatique, les ordinateurs CPC sont d'excellents outils pour vous aider à démarrer sur les langages classiques : BASIC, LOGO, et même TURBO PASCAL. Un autre atout pour AMSTRAD : les documentations sont bien faites, et suivent une approche très pédagogiques. De plus, les langages développés sont tous de très bonne qualité, et la littérature qui traite des CPC abonde...

... Un ouvrage pratique

Vous trouverez 11 parties dans cet ouvrage :

- Présentation générale
- Conception matérielle des CPC,
- Systèmes d'exploitation des CPC,
- Langages du CPC,
- Graphisme,
- Son,
- Logiciels à caractère professionnel,
- Périphériques,
- Programmes,
- Fabrication de circuits additionnels pour AMSTRAD,
- Annexes.

Une table des matières générale (en partie 1), une table des matières détaillée par partie, un glossaire et un index (en partie 1) vous permettront d'avoir accès rapidement à l'information que vous recherchez.

Ce livre n'est pas un livre comme les autres : il progresse en même temps que la technique relative aux CPC.

Dans cet ouvrage, sont présentées les techniques développées sur les CPC.

Si vous débutez en programmation, nous vous conseillons de vous reporter à la partie 5 sur les langages.

Si vous désirez utiliser votre CPC de manière professionnelle, reportez-vous :

- à la partie 3 si vous souhaitez vous servir des commandes de CP/M 2.2 ou de CP/M plus,
- à la partie 7 si vous voulez choisir puis utiliser un progiciel de traitement de texte, un tableur ou un gestionnaire de base de données.

Si vous n'êtes pas un débutant, ou si la partie 2 n'a plus de secret pour vous, vous pouvez vous reporter aux parties 5 et 6 où l'utilisation graphique du CPC et la programmation du circuit sonore sont étudiées en détail.

Si vous vous sentez l'âme d'un bricoleur, la partie 2 :

- vous explique comment fonctionne l'AMSTRAD ;
- vous donne les différents schémas électroniques des cartes contenues dans le clavier, l'écran et le lecteur de disquette ;
- enfin, décrit le fonctionnement précis des principaux circuits intégrés utilisés dans les CPC.

la partie 8 :

- détaille le fonctionnement des périphériques classiques,
- vous permet de construire des cartes pour interfacer ces périphériques.

Enfin la partie 10 :

- vous donne la possibilité de construire des interfaces et de les piloter dans un logiciel correspondant.

Si ce sont plutôt les programmes qui vous intéressent, reportez-vous à la partie 9, où sont étudiés :

- des utilitaires,
- des jeux,
- des programmes à utilisation professionnelle,
- des programmes relevant de techniques spécifiques (interruptions, gestion de flux vers un périphérique, commande d'appareils électriques, etc.).

Et la réponse à tous vos problèmes !

Les « cartes-contacts » insérées à la fin de ce chapitre permettront d'établir une relation privilégiée avec vous. Vous pourrez ainsi formuler vos remarques, critiques ou suggestions, voire nous exposer les problèmes que vous avez rencontrés. Et nous nous efforcerons d'y répondre aussi précisément que possible. Alors, « écrivez-nous ! »

Et maintenant, passons aux choses sérieuses...

1/2

Glossaire

A

Accumulateur

Registre de 8 bits le plus utilisé par le Z80. C'est par lui que transitent la plupart des données qui sont impliquées dans des opérations arithmétiques, logiques ou des tests.

Adressage (mode d')

Manière dont on accède à l'information dans une opération d'un programme écrit en ASSEMBLEUR : directement, indirectement, par registre, avec pré ou post-incrémentation, etc..

Adresse

Emplacement d'une cellule mémoire RAM, ROM ou autre.

Amorce : (Voir Boot)**AMSDOS**

AMStrad Disk Operating System. Système d'exploitation simplifié intégré dans le BASIC AMSTRAD.

AMSOFT

Département "développement logiciel" d'AMSTRAD.

Analogique

Valeur continue, à opposer à une valeur digitale qui caractérise une donnée numérique (0 ou 1).

Architecture

Organisation matérielle (Hardware) de l'ordinateur.

ASCII

Codage numérique des caractères alphanumériques (entre 0 et 255 sur l'AMSTRAD)

ASSEMBLEUR

Langage de programmation de **bas niveau**, très rapide, et parfois obligatoire pour des raisons de vitesse d'exécution ou d'accès à des zones privilégiées non accessibles par les instructions d'un langage évolué.

Attributs

Qualification d'un fichier. Ces attributs peuvent être RO (Read Only ou lecture seule), RW (Read Write ou lecture écriture), SYS (SYStème), etc..

AZERTY

Type de clavier utilisé en France. Les premières lettres en haut à gauche du clavier sont, de gauche à droite A, Z, E, R, T, Y d'où, par extension, le nom de clavier AZERTY.

B**Base (de numération)**

Les nombres traités par un ordinateur sont exprimés dans une base de numération, par exemple en décimal (base 10), en hexadécimal (base 16) ou en binaire (base 2).

Base de données

Ensemble de données liées logiquement les unes aux autres. On peut retrouver une donnée à l'intérieur d'une base de données en définissant ses caractéristiques.

BASIC

Langage de programmation créé en 1964 aux Etats-Unis. Destiné aux programmeurs débutants ou confirmés, il est disponible sur la plupart des micro-ordinateurs.

Baud

Vitesse de communication sur les liaisons parallèles correspondant au nombre de bits par seconde sur les ordinateurs AMSTRAD.

BCD (Binary Coded Decimal)

Principe de codage permettant de représenter un chiffre décimal sur 4 bits.

BDOS (Basic Disk Operating System)

Système d'exploitation de base. Contient les programmes de gestion de disquettes.

Binaire

Base de numération à deux éléments (0 et 1).

BIOS (Basic Input Output System)

Système d'entrées/sorties de base. Contient le logiciel qui permet d'adapter un DOS (ici CPM+) à un ordinateur.

Bit (Binary digit ou élément binaire)

Nom donné au plus petit élément mémoire accessible par le microprocesseur. Un bit peut prendre deux valeurs : 0 ou 1. Un octet (ou byte) est composé de 8 bits.

Bruit blanc

Bruit aléatoire produit par la voie « bruit » du générateur sonore.

Bug (ou bogue)

Erreur dans un programme.

Boole

Logique binaire, manipulant des 0 et des 1, ou des TRUE et des FALSE.

Boot (ou secteur de configuration)

Secteur sur la disquette qui permet de démarrer un programme d'exécution automatique du type SUBMIT.

Buffer

Zone mémoire en RAM ou sur disque qui permet de stocker plusieurs informations, que le micro-processeur n'a pas le temps de traiter instantanément par exemple.

Bus

Connexions physiques permettant de relier le micro-ordinateur à un ou plusieurs périphériques. On distingue deux types de bus : les bus d'adresses (qui véhiculent des adresses) et les bus de données (qui véhiculent des données).

Byte : (voir Octet)**C****Canal**

Autre nom donné à une voie de communication entre micro-ordinateur et ses périphériques.

Caractère

Symbole numérique, alphabétique ou de contrôle.

Catalogue

Liste des fichiers de données ou de programmes présents sur une cassette ou une disquette.

CCP (Console command processor)

Processeur des commandes console : zone qui contient les commandes résidentes.

Chaîne de caractères

Souvent appelé chaîne alphanumérique, c'est un ensemble de caractères alphabétiques ou numériques qui sont manipulés par des instructions spéciales.

Code machine

Code hexadécimal généré par un compilateur auquel on a fourni un programme écrit en assembleur.

Compilateur

Nom donné à un langage qui traduit en code exécutable les instructions d'un programme avant de les exécuter dans une phase appelée compilation.

CP/M 2.2 (Control Program for Microcomputer Version 2.2)

Operating system proposé par Digital Research disponible sur CPC 664 et 6128.

CP/M plus (Control Program for Microcomputer Version plus)

Operating system proposé par Digital Research disponible sur CPC 6128.

CPU (Central Processing Unit)

Microprocesseur gérant l'ensemble des ressources du microordinateur.

Curseur

Pavé rectangulaire qui indique à quel endroit sera affiché le prochain caractère tapé au clavier ou fourni par l'instruction PRINT.

Curseur graphique

Pixel utilisé en mode graphique équivalent au curseur utilisé en mode texte : indique l'endroit où sera affiché le prochain point par une commande graphique.

D**Debugger (Debogger en français)**

Outil de mise au point de programmes écrits en ASSEMBLEUR.

Décimal

Base de numération courante : base 10.

Démarrage à chaud

Sous CP/M, relance du système par appui simultané sur les touches CONTROL et C. Cette opération ne détruit pas les données présentes en mémoire centrale.

Démarrage à froid

Entrée sous CP/M par la commande I CPM tapée en BASIC.

Digital Research

Société fondée par Gary Kildall en 1976 et qui commercialise les différentes versions de CP/M.

Directives d'assemblage

Mots clés utilisés en assembleur qui ne concernent que le compilateur et qui ne font pas partie des mnémoniques du Z80. Ces directives permettent de définir des zones de mémoire, d'assembler une partie du programme sous condition, d'implanter le code généré à une adresse particulière, etc. Les directives varient d'un assembleur à l'autre.

Directory : (voir catalogue).

DOS (Disk Operating System)

Système d'exploitation qui gère l'unité ou les unités de disquettes. Le DOS disponible sur CPC est CP/M 2.2 sur CPC 664 et 6128 et CP/M PLUS sur CPC 6128.

Dr. LOGO

Langage proposé par Digital Research pour programmeurs débutants.

Drive

Mot anglais désignant le lecteur de disquettes.

Driver disque

Programme et matériel gérant l'unité lecteur de disquettes.

Dump

Opération qui consiste à afficher le contenu hexadécimal d'un fichier ou d'une zone de mémoire.

E

Editeur de textes

Programme permettant de saisir un texte quelconque, par exemple une lettre, ou encore un programme qui sera compilé par la suite.

Enveloppe

de volume : courbe qui module le volume d'un son en fonction du temps.

de ton : courbe qui module le ton d'un son en fonction du temps.

Entrées/Sorties

Opération qui consiste à lire ou écrire des données sur un périphérique.

EROM (Erasable Programmable Read Only Memory)

Mémoire programmable par un programmeur de PROM et effaçable par rayons ultra-violets.

Erreur de syntaxe

Erreur apparaissant dans un langage de programmation lorsqu'un ordre a été mal utilisé ou dont l'orthographe n'a pas été respectée.

F

FALSE

Valeur que peut prendre un opérateur booléen. En logique positive, la valeur FALSE correspond au 0 logique.

Fichier

Collection de données situées en mémoire centrale, sur cassette ou disquette.

Firmware (ou micro-programme)

Nom donné aux programmes résidents contenus dans les ROM de la machine. Ils sont constitués d'un ensemble de sous-programmes souvent appelés routines.

Flags (drapeaux)

Indicateurs élémentaires utilisés en ASSEMBLEUR et rassemblés dans un registre interne au Z80 qui renseignent l'issue d'une opération.

Floppy disk

Dénomination anglaise d'un lecteur de disquettes.

Fonction

En PASCAL, nom donné aux procédures de calcul qui fournissent une valeur numérique lorsqu'on les appelle.

Formatage

Option qui consiste à arranger les pistes et secteurs d'une disquette vierge (ou non) pour qu'elle soit reconnue par le lecteur de disquettes. Cette opération est obligatoire avant toute écriture sur le disque.

G

Générateur de son

Référencé AY-3-8912. C'est le circuit sonore des CPC. Il est capable de gérer 3 voies indépendantes et une voie de bruit simultanément.

H**Hexadécimal**

Base de numération couramment utilisée en informatique. Il s'agit de la base 16.

Horloge

Quartz qui cadence l'exécution des instructions élémentaires stockées en mémoire.

I**Instruction**

Mot-clé d'un langage de programmation.

Intégration à grande échelle

Type de circuits intégrés (LSI = Large Scale Integration) renfermant un grand nombre de composants.

Interface parallèle

Connecteur situé derrière le clavier des AMSTRAD, aussi appelé interface CENTRONICS, capable de gérer des imprimantes parallèles au même standard.

Interface série

Connecteur inexistant sur l'AMSTRAD, mais qui peut y être implanté. Ce type d'interface permet de communiquer avec des périphériques bit par bit (d'où son nom).

Interpréteur

Nom donné à un langage qui traduit en code exécutable les instructions d'un programme pendant son exécution.

Interruption

De type software ou hardware une interruption provoque un débranchement immédiat ou temporisé (selon la priorité de l'interruption) à une adresse particulière où se trouve le code d'un traitement d'interruption.

I/O (Input/Output ou Entrée/Sortie)

Abréviation caractérisant un échange de données entre ordinateur et périphérique sur un port de communication.

J**Joker**

Notion utilisée sous CP/M permettant de faire appel à des fichiers de manière non explicite (en donnant leurs premières lettres seulement, par exemple).

Joystick (ou manette de jeu)

Nom donné à la manette de jeu qui permet de saisir un déplacement vers le haut, le bas, la droite ou la gauche, et l'appui sur un bouton-feu.

K**K ou KO**

Kilo-Octet : Unité de dimensionnement de la mémoire. Correspond à 1024 octets.

L**Langage**

Ensemble de mots-clés ou instructions destinés à manipuler les ressources hardware de l'ordinateur. On parle de langage évolué lorsque l'on fait référence au BASIC, au PASCAL ou au LOGO, et de langage non évolué lorsque l'on fait référence à l'ASSEMBLEUR.

Lecture/Ecriture

Attribut d'un fichier ou d'une disquette, signifiant que l'accès à ce programme ou à cette disquette peut se faire en lecture ou en écriture.

Lecture seule

Attribut d'un fichier ou d'une disquette, signifiant que l'accès à ce programme ou à cette disquette peut se faire en lecture seulement.

LOGO (voir Dr. LOGO)**LSB (Last Significant Byte)**

Octet de poids faible.

LSQ (Last Significant Quartet)

Quartet (paquet de 4 bits) de poids faible.

Lutin (ou Sprite)

Caractère graphique destiné à être déplacé sur l'écran sans altérer les caractères ou graphismes qui s'y trouvent.

M**Manette de jeu** : (Voir Joystick)**Mémoire**

Case mémoire capable de retenir une information binaire.

Microprogramme : (Voir Firmware).

Mode d'affichage

Les CPC possèdent trois modes d'affichage :

MODE 0 : 25 lignes de 20 caractères, 16 couleurs et de définition 160 × 200 pixels.

MODE 1 : 25 lignes de 40 caractères, 4 couleurs et de définition 320 × 200 pixels.

MODE 2 : 25 lignes de 80 caractères. 2 couleurs et de définition 640 × 200 pixels.

Moniteur

Ecran d'affichage dédié à un ordinateur.

Mot-clé

Mot d'un langage, parfois appelé « instruction » ou « primitive ».

MSB (Most Significant Byte)

Octet de poids fort.

MSQ (Most Significant Quartet)

Quartet (paquet de 4 bits) de poids fort.

O**Octet**

Ensemble de 8 bits consécutifs pouvant représenter un entier compris entre 0 et 255.

Op-Code (ou code opératoire)

Instruction élémentaire du langage ASSEMBLEUR.

Opérateur

Nom donné aux signes mathématiques "supérieur à", "différent de", etc..

Ordinogramme ou organigramme

Représentation schématique des diverses actions réalisées dans un programme.

P**PASCAL**

Langage de programmation de haut niveau.

TURBO PASCAL, commercialisé par BORLAND est disponible sur les CPC 664 et 6128.

Pavé numérique

Nom donné au bloc du clavier qui rassemble les chiffres et touches de fonctions.

Périphérique

Tout élément matériel connecté à l'ordinateur est appelé périphérique.

Pile

Élément mémoire externe au Z80 qui permet de stocker des données les unes au-dessus des autres. On parle de pile FIFO (First In First Out), FILO (First In Last Out), ou LIFO (Last In First Out).

Piste

Une piste est l'équivalent d'un sillon sur un disque. Il y en a 40 par face de disquette 3 pouces sur les ordinateurs AMSTRAD.

Pixel

Un pixel est le plus petit point élémentaire qui peut être affiché sur l'écran.

Primitive

Nom donné aux mots-clés du langage LOGO.

Procédure

Sous-programme inséré dans le programme principal qui peut être appelé une ou plusieurs fois.

Progiciel

Concaténation des mots PROfessionnel et loGICIEL : les progiciels sont des logiciels dont les applications se situent dans des domaines très particuliers, généralement réservés aux professionnels de l'informatique.

Programmation structurée

Type de programmation dont le but est de clarifier au maximum les programmes, et par là même, de diminuer ou supprimer la phase de mise au point.

PROM : (Programmable Read Only Memory)

Mémoire programmable par un programmeur de PROM et non effaçable par rayons ultra-violets (contrairement aux EPROM ou REPROM).

Prompt

Caractère particulier souvent affiché en début de ligne et destiné à rappeler à l'utilisateur qu'il se trouve sous un système d'exploitation ou sous un logiciel particulier. Par exemple, le prompt du CP/M est un signe supérieur (>).

Puce ou chip

Autre nom donné à un micro-processeur.

Q

Quartet

Ensemble de quatre bits consécutifs.

Queue sonore

Buffer dans lequel sont stockées des notes à destination du générateur sonore AY-3-8912.

QWERTY

Type de clavier utilisé aux Etats-Unis. Les premières lettres en haut à gauche du clavier sont, de gauche à droite, Q, W, E, R, T, Y d'où par extension le nom du clavier QWERTY.

R

Rafraîchissement

Opération qui consiste à réécrire périodiquement dans les RAM dynamiques pour que leur contenu ne soit pas perdu.

RAM (Random Access Memory)

Mémoire vive dans laquelle on peut écrire et lire. Les RAM des CPC sont de type « dynamique » et doivent être rafraîchies périodiquement.

Registre

Mémoire interne à un micro-processeur, à accès très rapide, utilisée pour effectuer des opérations élémentaires.

Rendez-vous

Technique utilisée pour synchroniser des sons émis sur des voies différentes.

Résolution

Nombre de pixels disponibles sur l'écran.

RESTART

Interruption particulière qui ramène le pointeur de programme à une adresse définie par construction.

ROM (Read Only Memory)

Mémoire à lecture seule. Les ROM des CPC contiennent le firmware.

Routine

Autre nom donné à un sous-programme

RS232C

Interface de communication série très répandue en micro-informatique.

S**Secteur**

Bloc élémentaire de données stockées sur disquettes. Les blocs gérés par les CPC font 512 octets.

Souris

Périphérique permettant d'entrer des commandes sans passer par le clavier.

SUBMIT

Qualifie les programmes exécutables de manière automatique sous CP/M. Ces programmes contiennent une ou plusieurs commande(s) qui pourraient être tapées au clavier, qui sont activées par programme.

T**Tableur**

Outil permettant d'effectuer rapidement des calculs élémentaires dans des cellules liées logiquement entre elles.

Tape

Appellation anglaise du magnétophone à cassettes.

Taux d'expansion

Rapport entre le nombre de codes machine produits par un compilateur et le nombre de codes machine réellement nécessaires pour produire une action donnée.

TPA (Transcient Program Area)

Zone des programmes temporaires où sont chargés les programmes non résidents d'extension « .COM ».

Traitement de texte

Programme permettant de saisir du texte ou des programmes qui seront compilés par la suite.

TRUE

Valeur que peut prendre un opérateur booléen. En logique positive, la valeur TRUE correspond au 1 logique.

U

User (ou utilisateur)

Définit une zone de stockage sur disque accessible sous un numéro d'utilisateur unique.

Z

Variable

Nom donné à un ou plusieurs emplacements en mémoire RAM. Ce ou ces emplacements sont accédés par leur nom pour augmenter la lisibilité des programmes.

Version

Un logiciel qui est édité peut subir des modifications. Chaque fois qu'une modification ou un groupe de modifications est/sont arrêtée(s), on parle de version. Par exemple, le CP/M disponible sur CPC 664 est CP/M 2.2. La version est 2.2

1/3

Index

A

Accumulateur	2/3.1, 4/2
Adressage (mode d')	4/2.2
Adresse	1/2, 4/2
Amorce	3/3, 3/4
AMSDOS	3/2
AMSOFT	1, 3
Analogique	8, 10
Architecture	2/2, 2/3.1
ASCII	4/1.2, 11 : Annexe 1
Assembleur	4/2
Attributs	3/3, 3/4
AZERTY	4/1.2

B

Base (de numération)	4/1.2, 4/4
Base de données	7/3.1
BASIC	4/1
Baud	8/1, 8/3
BCD (Binary Coded Decimal)	8/5
BDOS (Basic Disk Operating System)	3
Binaire	4/2
BIOS (Basic Input/Output System)	3
Bit (Binary digiT)	4/2
Bruit blanc	6/2
Bug (ou bogue)	4/1.2
Boole	4/2
Boot	3/3, 3/4
Buffer	4/1.2
Bus	8, 10
Byte	4/2, 4/1

Partie 1 : Présentation générale

C

Canal	4/1, 6/2
Caractère	4/1, 4/2, 4/4
Catalogue	4
CCP (Console Command Processor)	3
Chaîne de caractères	4/1, 4/4
Code machine	4/2
Compilateur	4/1, 4/4
CP/M 2.2	3/3
CP/M plus	3/4
CPU (Central Processing Unit)	4/2
Curseur	4
Curseur graphique	4/1, 4/3, 4/4

D

Debugger (ou Debogger)	4/2
Décimal	4/2
Démarrage à chaud	3
Démarrage à froid	3
Digital Research	4
Directives d'assemblage	4/2
Directory	3
DOS (Disk Operating System)	3
Dr. LOGO	4/3
Drive	4, 8/5
Driver disk (ou Driver disque)	4, 8/5
DUMP	4/1, 4/2, 4/4

E

Editeur de textes	4, 7/1
Enveloppe	6
Entrées/Sorties	4/1, 4/2, 4/4, 8, 10
EPROM (Erasable Programmable Read Only Memory)	2, 8, 10
Erreur de syntaxe	4/1

F

FALSE	4/1, 4/4
Fichier	4/1, 4/3, 4/4, 8/5
FIRMWARE (ou micro-programme)	4/2
Flags (ou drapeaux)	4/2
Floppy disk (ou lecteur de disquettes)	4, 8/5
Fonction	4/1, 4/4
Formatage	4, 8/5

G

Générateur de son	6
-------------------------	---

H

Hexadécimal	4/1, 4/2, 4/4
Horloge	2, 4/1, 4/4

I

Instruction	4
Intégration à grande Echelle (LSI)	2
Interface parallèle	2/3.4, 8, 10
Interface série	10
Interpréteur	4/1, 4/3
Interruption	4/2
I/O (Input/Output ou Entrée/Sortie)	2, 4/2, 8, 10

J

Joker	3
Joystick (ou manette de jeu)	2, 4/1, 4/2, 4/3, 4/4, 10

L

Langage	4
Lecture/Ecriture	3, 8/5
Lecture seule	3, 8/5
LOGO	4/3
LSB (Last Significant Byte)	4/2
LSQ (Last Significant Quartet)	4/2
Lutin (ou Sprite)	5, 4/1

M

Manette de jeu (ou Joystick)	2, 4, 10
Mémoire	4/2, 8, 10
Micro-programme (ou FIRMWARE)	4/2
Mode d'affichage	4
Moniteur	8
Mot-clé	4
MSB (Most Significant Byte)	4/2
MSQ (Most Significant Quartet)	4/2
Multiplan	7/2.1

O

Octet	4
Op-Code	4/2
Opérateur	4
Organigramme ou Ordinogramme	4

P

PASCAL	4/4
Pavé numérique	4
Périphérique	3, 4, 8, 10
Pile	4/2
Piste	3
Pixel	5
Primitive	4/3
Procédure	4/1, 4/4
Progiciel	7
Programmation structurée	4, surtout 4/4
PROM (Programmable Read Only Memory)	2, 4/2, 10
Prompt	4
Puce (ou chip)	2

Q

Quartet	4/2
Queue sonore	6
QWERTY	4

R

Rafraîchissement	2
RAM (Random Acces Memory)	2, 10
Registre	4/2
Rendez-vous	6
Résolution	5, 4/1
Restart	4/2
ROM (Read Only Memory)	2, 4/2, 10
Routine	4/1, 4/2
RS232C	10

S

Secteur	10
Souris	8/4
Submit	3

T

Tableur	7/2
Tape (ou Magnétophone)	4
Taux d'expansion	4/1
TPA (Transcient Program Area)	3
Traitement de texte	7/1
TRUE	4/1, 4/4

U

User (ou utilisateur)	3
-----------------------------	---

V

Variable	4
Version	4

1/4

Dictionnaire technique

Français-Anglais

Anglais-Français

Français	Anglais	Anglais	Français
A			
A	To	Absolute	Absolu
Absolu	Absolute	Access	Accès
Accès	Access	Add	Ajouter
Accord	Tune	Address	Adresse
Actionner	Drive	All	Tous
Adresse	Address	Analogue	Analogique
Afficher	Display	Array	Tableau
Ajouter	Add		
Aléatoire	Random		
Aller	Go		
Analogique	Analog		
Appeler	Call		
Arrêt	Stop		
Arrêt	Off		
Arrière	Back		
Attendre	Wait		
Autolancement	Boot		
Avec	With		
B			
Bande magnétique	Tape	Back	Arrière
Bas	Bottom	Binary	Binaire
Bidirectionnel	Duplex	Boot	Autolancement

Français

Binaire
Boîte

Anglais

Binary
Box

Anglais

Border
Border
Bottom
Box
Buffer
Busy
Byte

Français

Frontière
Limite
Bas
Boîte
Tampon
Occupé
Octet

C

Cadre
Caractère
Chaîne
Champs
Chercher
Clavier
Clé
Codage
Colonne
Compacter
Complet
Contrôler
Couches
CurseurFrame
Character
String
Fields
Search
Keyboard
Key
Encode
Column
Pack
Full
Check
Overlay
CursorCall
Carriage return
Change
Character
Check
Close
Cluster
Column
Computer
CursorAppeler
Retour chariot
Modifier
Caractère
Contrôler
Fermer
Groupe
Colonne
Ordinateur
Curseur

D

Décalage
Déclencher
Décodage
Demi
Dépanner
Départ
Dérouler
Diviser
Donnée
Drapeau
DurantOffset
Release
Decode
Half
Debug
Start
Scroll
Divide
Data
Flag
WhileData
Data
Debug
Debug
Decode
Digital
Display
Display
Divide
Do
Done
Dot
Dotted
Drive
Drive
DuplexDonnée
Information
Dépanner
Mettre au point
Décodage
Numérique
Visualiser
Afficher
Diviser
Faire
Fait
Point
Pointillé
Entraîner
Actionner
Bidirectionnel

Français	Anglais	Anglais	Français
E			
Echapper	Escape	Encode	Codage
Ecran	Screen	End	Fin
Ecrire	Write	End of file	Fin de fichier
Emplir	Fill	Entry	Entrée
En	In	Escape	Echapper
Encre	Ink	Event	Événement
Endroit	Location		
Enlever	Remove		
Enregistrement	Record		
Entraîner	Drive		
Entrée	Entry		
Envoyer	Send		
Essai	Test		
Événement	Event		
Externe	Out		
F			
Faire	Do	Fast	Rapide
Fait	Done	Fields	Champs
Fenêtre	Window	Fill	Emplir
Fermer	Close	Find	Trouver
Fil	Wire	Flag	Drapeau
File d'attente	Queue	Flag	Sémaphore
Fin	End	Float	Flotter
Fin de fichier	End of file	Floating	Flottant
Flottant	Floating	For	Pour
Flotter	Float	Frame	Cadre
Frontière	Border	Full	Plein
		Full	Complet
G			
Groupe	Cluster	Get	Obtenir
		Go	Aller
H			
Hauteur	Height	Half	Demi
Hors	Out	Half	Moitié
		Hardware	Matériel
		Height	Hauteur
		Hold	Maintenir

Français	Anglais	Anglais	Français
I			
Imprimante	Printer	In	En
Imprimer	Print	In	Interne
Indexer	Index	Index	Indexer
Information	Data	Initialize	Initialiser
Initialiser	Initialize	Ink	Encre
Interne	In	Invert	Inverser
Interroger	Poll		
Inverser	Invert		
J			
Jusqu'à	Until	Joystick	Manette de jeu
		Jump	Sauter
K			
		Key	Touche
		Key	Clé
		Keyboard	Clavier
L			
Largeur	Width	Less	Moins (le)
Lent	Slow	Limit	Limite
Libérer	Release	Line	Ligne
Lien	Link	Link	Lien
Lien	String	List	Lister
Ligne	Line	Locate	Localiser
Limite	Border	Location	Endroit
Limite	Limit		
Lire	Read		
Lister	List		
Localiser	Locate		
Logiciel	Software		
M			
Maintenir	Hold	Memory	Mémoire
Manette de jeu	Joystick	Most	Plus (le)
Marche	On	Motor	Moteur
Matériel	Hardware	Mouse	Souris

Français

Anglais

Anglais

Français

Mémoire	Memory
Mettre	Put
Mettre à	Set
Mettre au point	Debug
Mise à jour	Update
Modifier	Change
Moins (le)	Less
Moitié	Half
Moteur	Motor

N

Nouveau
Numérique

New
Digital

New
Next

Nouveau
Suivant

O

Obtenir
Occupé
Octet
Ordinateur
Ouvrir

Get
Busy
Byte
Computer
Open

Off
Offset
On
Only
Open
Out
Out
Overlay
Overlay

Arrêt
Décalage
Marche
Seulement
Ouvrir
Hors
Externe
Couches
Recouvrement

P

Papier
Passer
Pendant
Piste
Placer
Plein
Plume
Plus (le)
Point
Point écran
Pointillé
Pour
Précédent

Paper
Skip
While
Track
Place
Full
Pen
Most
Dot
Pixels
Dotted
For
Previous

Pack
Paper
Pen
Pixels
Place
Plotter
Poll
Previous
Print
Printer
Put

Compacter
Papier
Plume
Point écran
Placer
Traceur
Interroger
Précédent
Imprimer
Imprimante
Mettre

Français

Anglais

Anglais

Français

Q

Queue

File d'attente

R

Rang
Rapide
Recouvrement
Registre
Réinitialiser
Relatif
Remplacer
Répéter
Rétablir
Retour
Retour chariot
Rouler

Raw
Fast
Overlay
Register
Reset
Relative
Replace
Repeat
Restore
Return
Carriage return
Roll

Random
Raw
Read
Record
Register
Relative
Release
Release
Remove
Repeat
Replace
Reset
Restore
Return
Roll

Aléatoire
Rang
Lire
Enregistrement
Registre
Relatif
Déclencher
Libérer
Enlever
Répéter
Remplacer
Réinitialiser
Rétablir
Retour
Rouler

S

Sauter
Sémaphore
Seulement
Sommet
Son
Souris
Suivant
Synchrone

Jump
Flag
Only
Top
Sound
Mouse
Next
Synchronous

Screen
Scroll
Search
Send
Set
Skip
Slow
Software
Sort
Sound
Speed
Start
Stop
String
String
Synchronous

Ecran
Dérouler
Chercher
Envoyer
Mettre à
Passer
Lent
Logiciel
Trier
Son
Vitesse
Départ
Arrêt
Lien
Chaîne
Synchrone

T

Tableau
Tampon

Array
Buffer

Tape
Test

Bande magnétique
Essai

Français

Temps
Tonalité
Touche
Tous
Traceur
Trier
Trouver

Anglais

Time
Tone
Key
All
Plotter
Sort
Find

Anglais

Time
To
Tone
Top
Track
Tune

Français

Temps
A
Tonalité
Sommet
Piste
Accord

U

Utilisateur
Utiliser

User
Use

Until
Update
Use
User

Jusqu'à
Mise à jour
Utiliser
Utilisateur

V

Visualiser
Vitesse

Display
Speed

W

Wait
While
While
Width
Window
Wire
With
Write

Attendre
Durant
Pendant
Largeur
Fenêtre
Fil
Avec
Ecrire

Partie 2

Conception matérielle des CPC

2/0

Table des matières

2/1	Architecture interne
2/2	Schémas des cartes mères des AMSTRAD CPC 464, 664 et 6128
2/2.1	CPC 464
2/2.2	CPC 664
2/2.3	CPC 6128
2/2.4	Différences de conception
2/3	Les circuits intégrés spécialisés
2/3.1	Le microprocesseur Z80
2/3.2	Le contrôleur d'écran : CRTC et VGA
2/3.2.1	CRT Controller ou CRTC
2/3.3	Le circuit sonore AY3-8912
2/3.4	L'interface parallèle PIO 8255A
2/3.4.1	Exploitation du PIO 8255
2/3.5	Le contrôleur de disque μPD 765AC
2/4	Les mémoires
2/4.1	La mémoire morte (ROM)
2/4.2	La mémoire vive (RAM)
2/5	Les circuits intégrés standards
2/5.1	Les circuits intégrés linéaires
2/5.2	Les circuits intégrés digitaux
2/6	L'horloge interne



Éditions WEKA
82, rue Curial
75935 Paris Cedex 19
Tél. : (1) 40 37 01 00
Télex : 210 504 F
Télécopieur : (1) 40 37 02 17

URGENT



**NE PAS
AFFRANCHIR**

**CORRESPONDANCE
RÉPONSE**

VALABLE DU : 1.09.87
AU : 31.08.90

A utiliser seulement en
France métropolitaine
et dans les départements
d'outre-Mer
pour les envois
ne dépassant pas 20 g

MTP 1167

**EDITIONS WEKA
AUTORISATION N° 257975
75581 PARIS CEDEX 12**

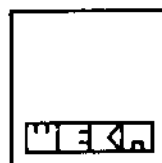
Ecrivez-nous !

Votre AMSTRAD est tombé en panne : vous l'avez réparé vous-même ou donné à dépanner à un professionnel.

Renvoyez-nous l'une de ces cartes en décrivant les défauts constatés, et en précisant si possible quelle était la cause de la panne.

En regroupant ces informations, nous pourrions enrichir cette rubrique au fil de nos compléments et mises à jour et faire profiter tous nos lecteurs de votre expérience personnelle.

Et si vous n'avez pas trouvé la panne, écrivez-nous tout de même : peut-être aurons-nous la réponse !



Éditions WEKA
82, rue Curial
75935 Paris Cedex 19
Tél. : (1) 40 37 01 00
Télex : 210 504 F
Télécopieur : (1) 40 37 02 17

URGENT



**NE PAS
AFFRANCHIR**

**CORRESPONDANCE
RÉPONSE**

VALABLE DU : 1.09.87
AU : 31.08.90

A utiliser seulement en
France métropolitaine
et dans les départements
d'outre-Mer
pour les envois
ne dépassant pas 20 g

MTP 1167

**EDITIONS WEKA
AUTORISATION N° 257975
75581 PARIS CEDEX 12**

CARTE - CONTACT

"Comment exploiter toutes les ressources et augmenter les performances de votre AMSTRAD"

Type de l'ordinateur N° de série

En service depuis

Panne constatée :

La panne était due à :

Pour la réparer, il a fallu :

CARTE - CONTACT

"Comment exploiter toutes les ressources et augmenter les performances de votre AMSTRAD"

Type de l'ordinateur N° de série

En service depuis

Panne constatée :

La panne était due à :

Pour la réparer, il a fallu :

2/1

Architecture interne

Les AMSTRAD CPC sont architecturés autour d'un micro-processeur 8 bits Z80 de chez INTEL.

Ce micro-processeur possède :

- un bus de 16 bits d'adresse. Sa possibilité d'adressage est donc de 2^{16} octets, soit 65536 octets ;
- un bus de données de 8 bits ;
- une entrée horloge de fréquence maximale 4 MHz (Mega Hertz), qui est à 4 MHz sur les CPC ;
- des entrées de gestion d'interruptions masquables et non masquables ;
- un signal de gestion de rafraîchissement des RAM dynamiques ;
- divers signaux de gestion de périphériques ;
- une entrée de remise à zéro ;
- une alimentation unique entre 0 et 5 volts.

Les *principaux circuits périphériques* sont les suivants :

PPI 8255 :

Circuit spécialisé dans la **gestion de données parallèles** sur 8 bits, commercialisé par INTEL.

Ce circuit possède 3 ports de 8 bits. Il s'occupe de la gestion :

- du lecteur de cassettes ;
- du clavier ;
- du circuit sonore ;
- du signal d'occupation de l'imprimante (BUSY).

CRTC 6845 :

Circuit spécialisé dans la **gestion de l'écran**.

VGA HSG 3130 :

Circuit hybride spécialisé dans la gestion de l'écran et des banques mémoires. Ses fonctions sont les suivantes :

- commutation des banques ROM ;
- choix du mode d'affichage sur l'écran ;
- sélection d'une encre ;
- affectation d'une encre à un stylo ;
- réinitialisation du compteur d'interruptions.

AY3-8912 :

Circuit sonore commercialisé par General Instruments. Il gère 3 voies indépendantes et mixables et une voie de bruit blanc mixable aux 3 voies sonores. Il permet de générer des sons simples, des enveloppes de ton et de volume. Grâce à ses registres internes, il peut être programmé par le CPU, et générer le son programmé sans requérir la présence du CPU. Ce dernier peut donc s'occuper d'autres tâches pendant qu'un son est généré.

Les ROM :

Référencées TMM 23256, ce sont des mémoires rapides (200 nanosecondes). Elles occupent 32 KO (kilo-octets) sur CPC 464 et 48 KO sur CPC 664 et 6128.

Les 32 KO sont implantés entre les adresses suivantes :

- #0000 à #3FFF : ROM BIOS
- #C000 à #FFFF : ROM BASIC

Les 16 KO supplémentaires sur 664 et 6128 sont implantés entre les adresses suivantes :

- #C000 à #FFFF : ROM DRIVER DISC

Comme leur nom l'indique :

- les **ROM BIOS** s'occupent de la gestion de l'ordinateur : entrées/sorties et circuits spécialisés ;
- les **ROM BASIC** contiennent l'interpréteur BASIC ;
- les **ROM DRIVER DISC** contiennent les primitives de gestion de l'unité lecteur de disquettes.

Les RAM :

Référencées 4864, elles contiennent 64 kilo bits (soit 8 KO) par circuit. Il y en a donc 8 sur les CPC 464 et 664 et 16 sur le CPC 6128. Elles sont implantées entre les adresses #0000 et #FFFF pour les CPC 464 et 664 et entre les adresses #0000 et #1FFFF pour le CPC 6128.

2/2

Schémas des cartes mères des AMSTRAD CPC 464, 664 et 6128

Nous donnons ici le schéma des cartes qui sont incorporées dans le clavier des CPC.

Ces schémas sont assez similaires, mis à part celui du CPC 6128 qui, au niveau de la mémoire RAM doit adresser 128 KO (kilo-octets), ce qui demande une petite logique supplémentaire.

Les grandes parties sont les suivantes :

- quartz à 16 Mhz (méga-hertz),
- microprocesseur Z80-A,
- circuit PIO spécialisé dans la gestion des périphériques parallèles,
- mémoire RAM dynamique,
- mémoire ROM,
- circuit contrôleur d'écran,
- circuit hybride de gestion de mémoires et d'écran GATE ARRAY,
- générateur sonore AY-3-8912,
- divers circuits de gestion comme latches, multiplexeurs, portes logiques, etc.

2/2.1

CPC 464

BE
F
S
EQ

A
G
E

641

100 100 100

100 100 100

100

100

100

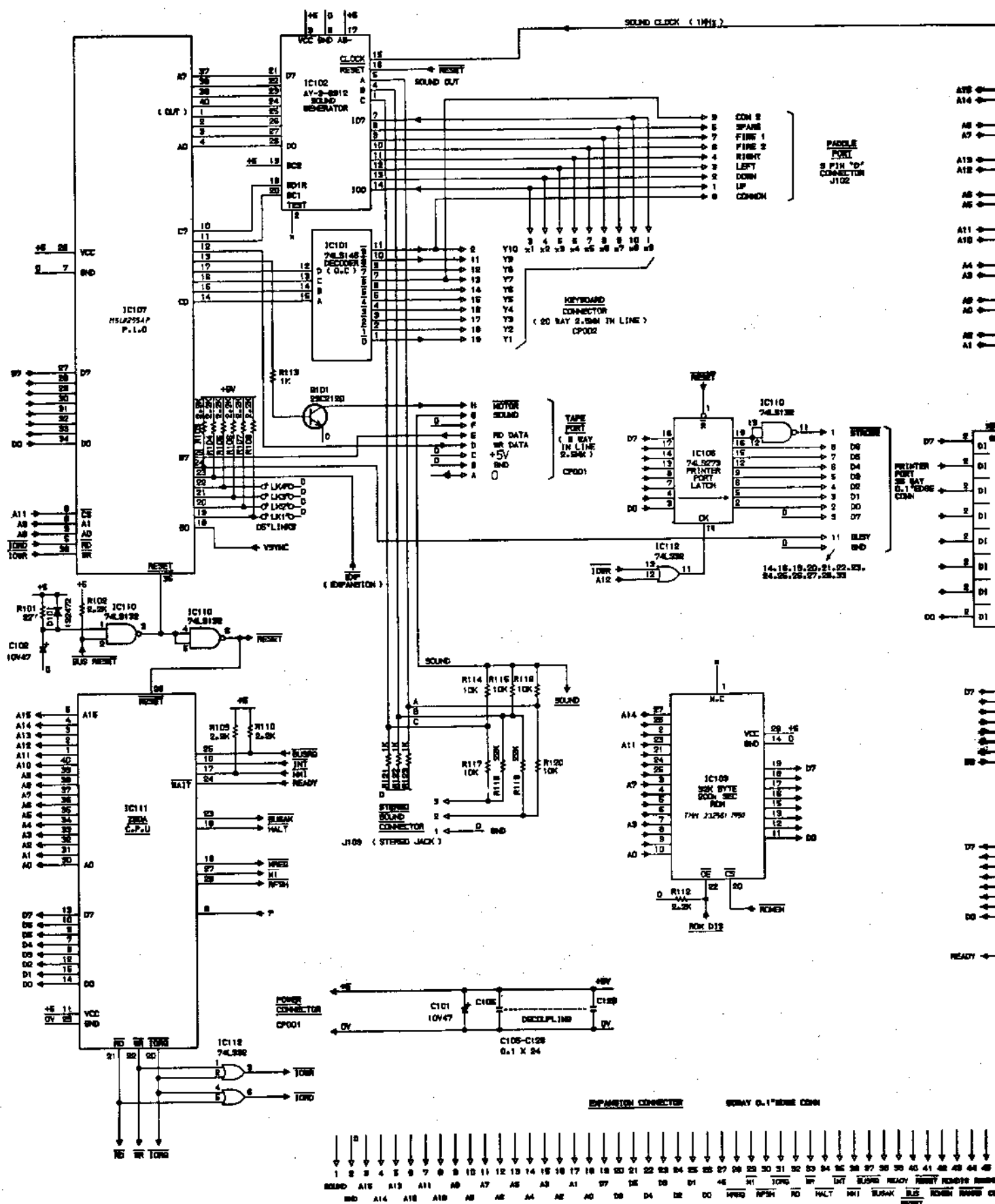
100

100
100

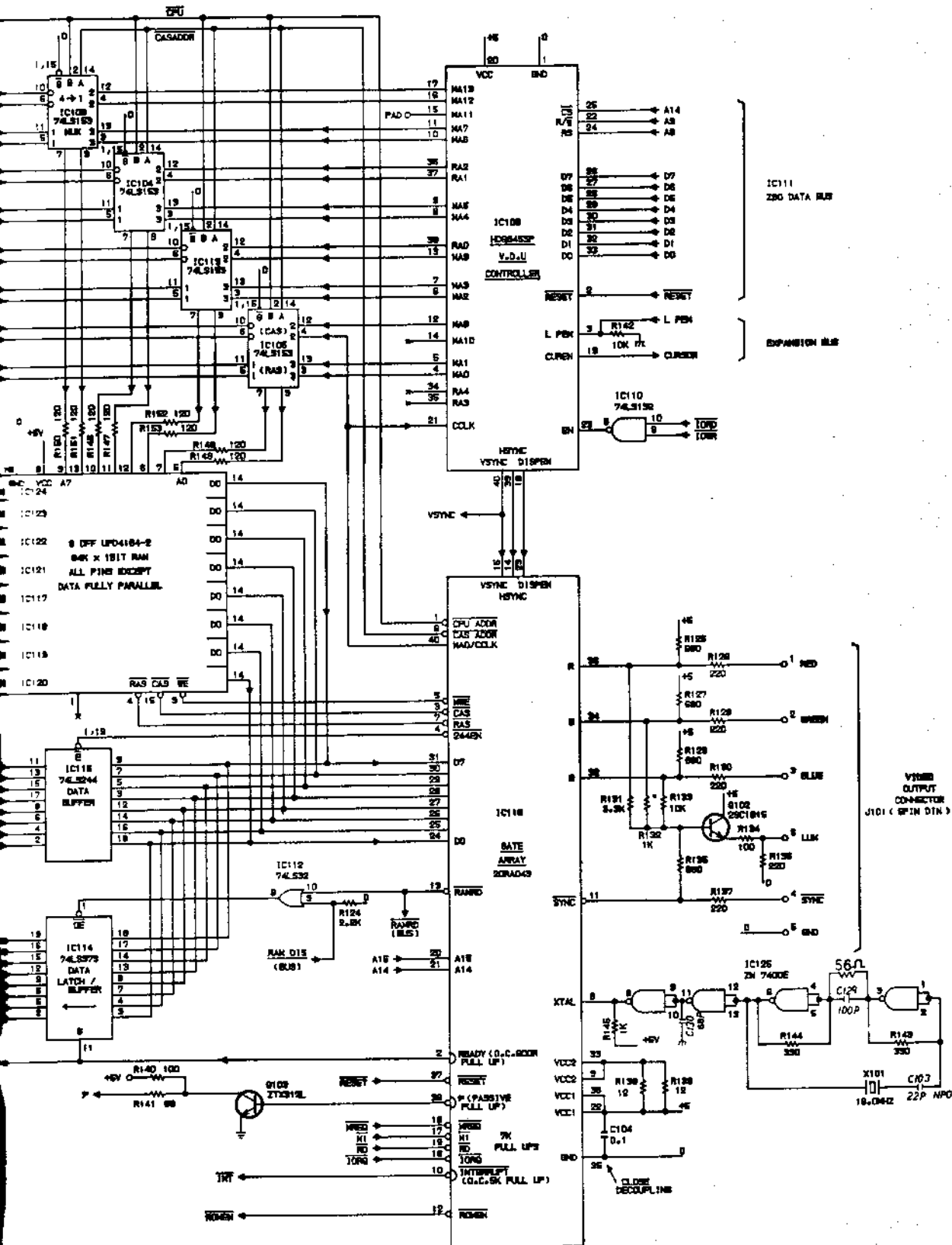
La liste des composants du CPC 464

Valeur	Référence
Résistances à film de carbone (1/4 de watt sauf si autre valeur précisée)	
56ohm	R306
68ohm	R141
100ohm	R134, 140
120ohm	R146-153
180ohm	R315
220ohm	R126, 128, 130, 136, 137
330ohm	R143, 144
560ohm	R154, 313, 325
680ohm	R125, 127, 129, 135, 301
820ohm	R324
1kohm	R113, 121-123, 132, 145
2k2ohm	R102-112, 124
3k3ohm	R321
4k7ohm	R320, 323
5k6ohm	R314
10kohm	R114-117, 120, 133, 142, 309, 312
12kohm	R318
18kohm	R308, 311
22kohm	R101, 118, 119, 304, 305
33kohm	R319, 322
47kohm	R302, 303, 307
180ohm	R317
1Mohm	R316
12ohm (1/2W)	R138, 139
Condensateurs électrolytiques	
1uF/50V	C309, 311, 314
10uF/16V	C324
22uF/10V	C308, 318
47uF/10V	C101, 102, 303, 306
100uF/10V	C301, 304
100uF/16V	C315
470uF/10V	C322
Condensateurs céramiques	
33pF/50 V	C316
200pF/50 V	C321
220pF/50 V	C310
270pF/50 V	C313
470pF/50 V	C307
0.001uF/50 V	C320
0.022uF/50 V	C317, 319, 323
0.1uF/50 V	C104-128
Condensateurs polystyrène	
0.001uF	C312
0.01uF	C305
0.068uF	C302

Référence	Description
Circuits Intégrés	
IC101	HD74LS145
IC102	AY-3-8912
IC103	TMM-23256P-1950
IC104, 105, 109, 113	HD74LS153
IC106	HD74LS273
IC107	M5L8255AP-5
IC108	HK6845SP
IC110, 112	HD74LS132
IC111	Z8400APS
IC114	HD74LS373
IC115	HD74LS244
IC116	20RA043
IC117-124	HM4864U-2
IC125	ZN7400E-D3
IC301	LA4140
IC302	LA6324
Transistors	
Q101	KTC2120Y
Q102, 301	KTC1815Y
Q103	ZTX312L
Diodes	
D101	SIL-IS2472-HL
D301	SIL-IN4002
D302	LED-Red.SLP145B
Self & Tx.	
L301	82uH
T101	C-12
Prises Jacks & Connecteurs	
J101	Prise DIN à 6 broches
J102	Port Joystick 9 broches
J103	Port d'entrée/sortie RCA (3.5mm)
J104	Prise Jack d'alimentation
J105-112, 116	Connecteurs 8 broches Dual In Line
J113	Connecteurs 14 broches Dual In Line
J1114,115	Connecteurs 20 broches Dual In Line
Switches (micro-interrupteurs)	
SW301	Interrupteur RIP
SW302	Interrupteur Marche/Arrêt
SW303	Interrupteur Clavier
Autres	
VR301	Résistance variable de contrôle de volume (20 kΩ)
RY301	Relai cassette
SP301	Interlocuteur
X101	Quartz HC-18/u 16Mhz



GRAM



2/2.2

CPC 664

664
1000
500

500

500
100
100
100

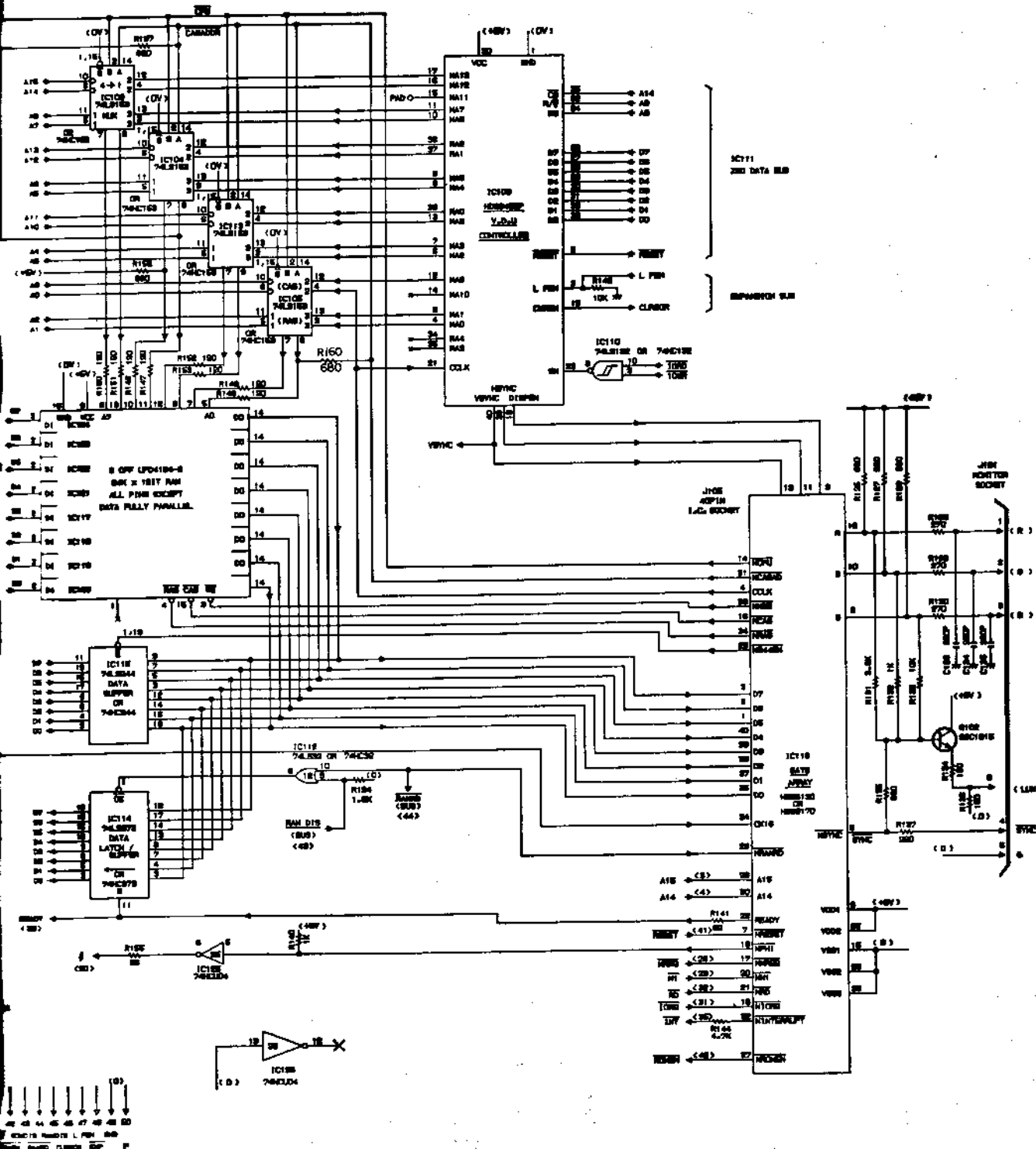
664
1000
500

La liste des composants du CPC 664

Valeur	Référence
4ohm7	R323
47ohm	R156, 159
56ohm	R155-306
120ohm	R146-153
150ohm	R136, 218
180ohm	R134
220ohm	R137
270ohm	R126, 128, 130
560ohm	R317
680ohm	R125, 127, 129, 135, 157, 158, 201-206, 215, 301
1kohm	R113, 121-123, 132, 140, 211, 212, 216, 313, 315, 321
1k5ohm	R124
2k2ohm	R102-110, 112, 214
3k3ohm	R131
4k7ohm	R144, 310
10kohm	R114-117, 120, 133, 142, 217, 309, 312
18kohm	R308, 311
22kohm	R118, 119, 304, 305
47kohm	R302, 303, 307, 319, 320
470kohm	R314
1Mohm	R111
3M3ohm	R101
10Mohm	R143, 145
4ohm7	R322 Fusible
100ohm	R316 1/2W
Condensateurs céramiques	
47pF/50 V	C132
68pF/50 V	C130, 131
220pF/50 V	C133-135, 310
270pF/50 V	C313
470pF/50 V	C307
0.1uF/16 V	C101-128, 201, 213
Condensateurs polystyrène	
0.001uF	C312
0.01uF	C305
0.047uF	C318
0.068uF	C302
0.1uF	C319
Condensateurs électrolytiques	
1uF/50V	C309, 311, 314, 316, 317
22uF/10V	C308
47uF/10V	C129, 303, 306
100uF/10V	C301, 304
100uF/16V	C315

Référence	Description
Circuits intégrés	
IC101	HD74LS145
IC102	AY-3-8912
IC103	TMM-23256P-1952
IC104-105, 109, 113	HD74LS153
IC106	HD74LS273
IC107	M5L8255AP-5
IC108	HD6845SP
IC110, 210	HD74LS132
IC111	Z8400APS
IC112, 207	HD74LS32
IC114	HD74LS373
IC115	HD74LS244
IC116	HSG3130/3170
IC117-124	HM4864U-2
IC125	TC74HCU04P
IC201	UPD765AC
IC202	FDC9216BT
IC203	SN74HC240N
IC204	TMM-23128P-1951
IC205	DN74LS08
IC206, 208	DN74LS38
IC209	DN74LS136
IC211	DN74LS27
IC212	DN74LS74
IC213	TC74HC161
IC301	LA4140
IC302	LA63585
Transistors	
Q101	KTC2120Y
Q102, 301, 302	KTC1815Y
Q303	KTC950Y
Diodes	
D101	IS2472-HS
D201	DS442XFA5
D301	IN4402
D302	SLP-145B
D303, 304	KDS1555
Autres	
J101	Jack DIN TCS4460-01-1011
J102	Connecteur D Sub 9
	HXC0730-01-010
J103	Jack RCA 3.5 HSJ1061-01-440
J104, 105	Connecteur IC DILB40P-8J
J201	Connecteur IC DILB28P-8J
J301	Jack DC HECO470-01-630
J302	Jack DIN TCS4450-01-101
SW301	Interrupteur ESD-3975
X101	Quartz HC-49/U
RY301	Relai cassette G4S-1112P-1-B-19
SP301	Interlocuteur CO40K01K2451
CP 201	Cordon d'alimentation du disque
CP 301	Cordon d'alimentation 14550401

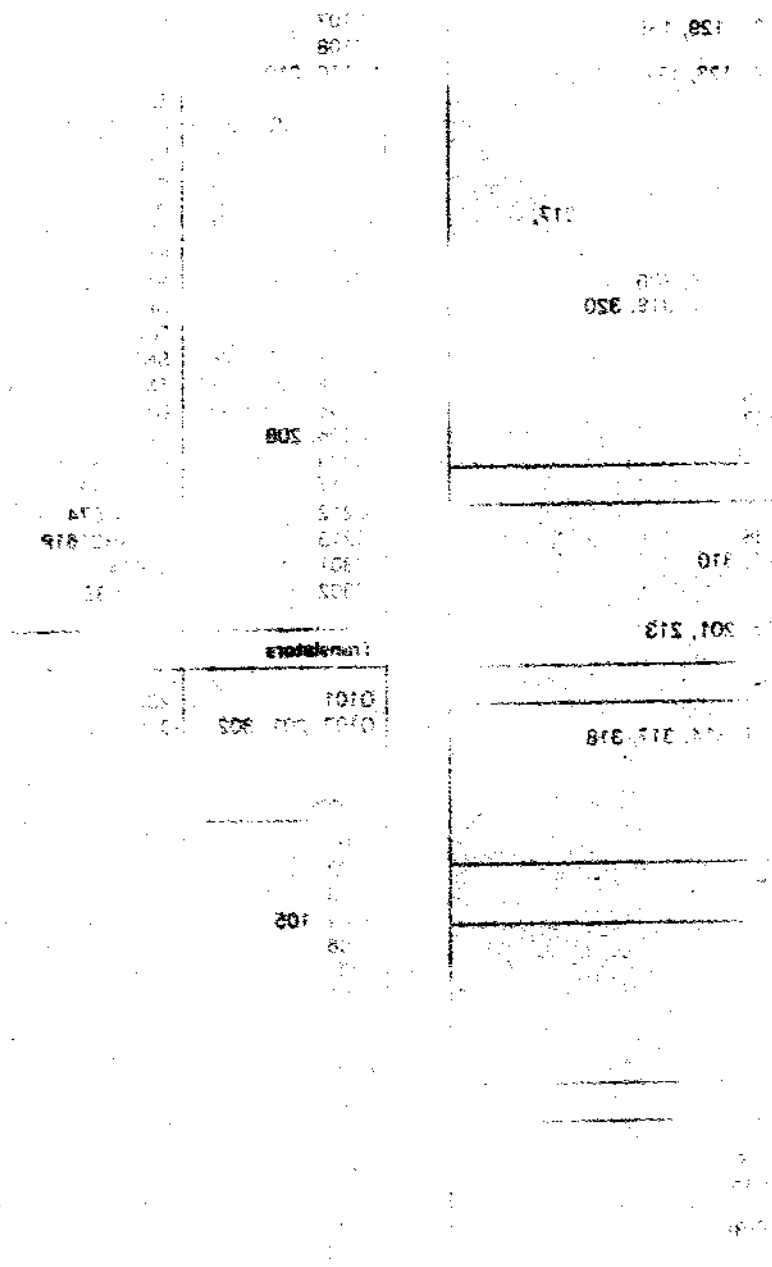
CIRCUIT DIAGRAM



2/2.3

Amstrad 6128

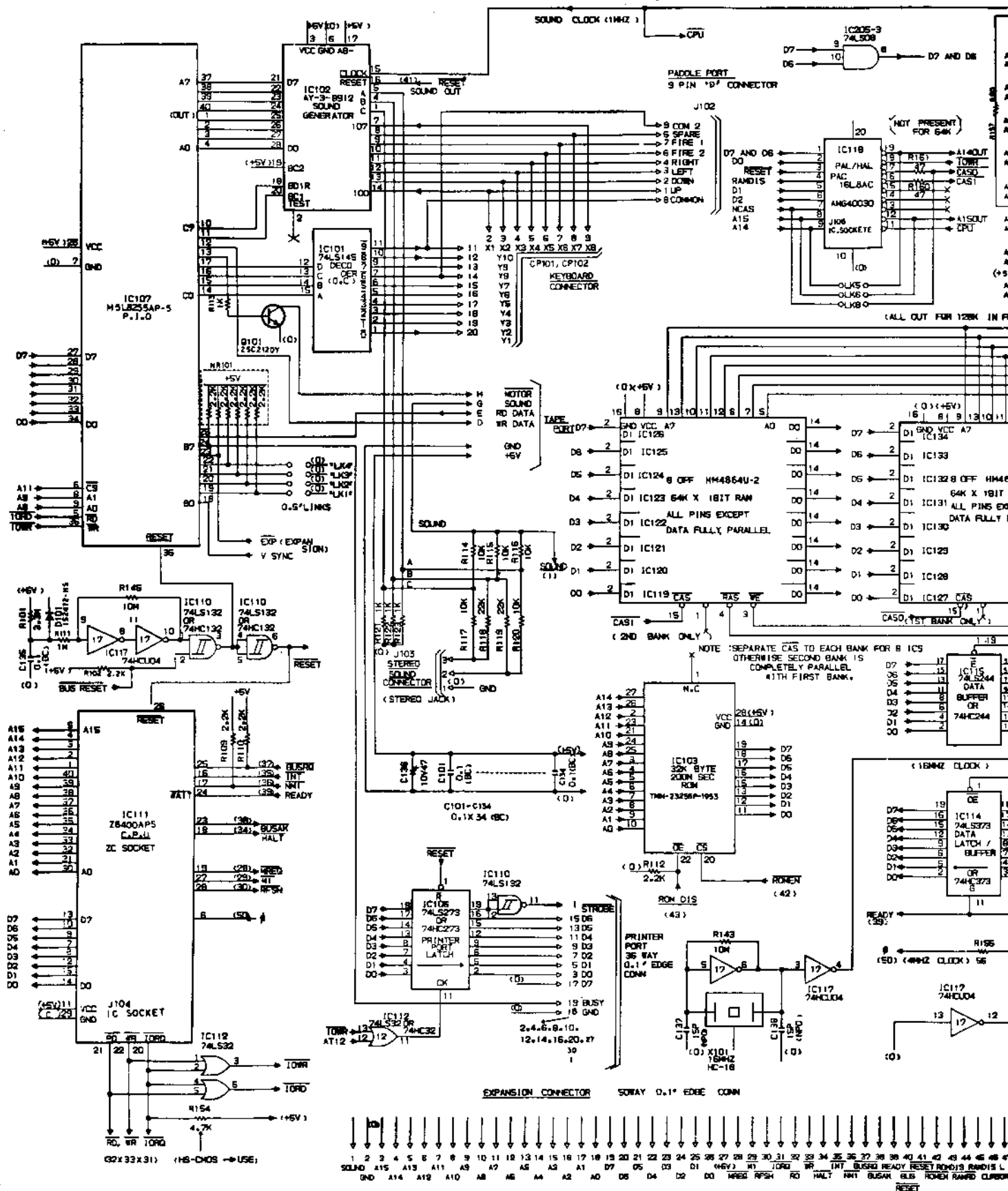
CPC 6128

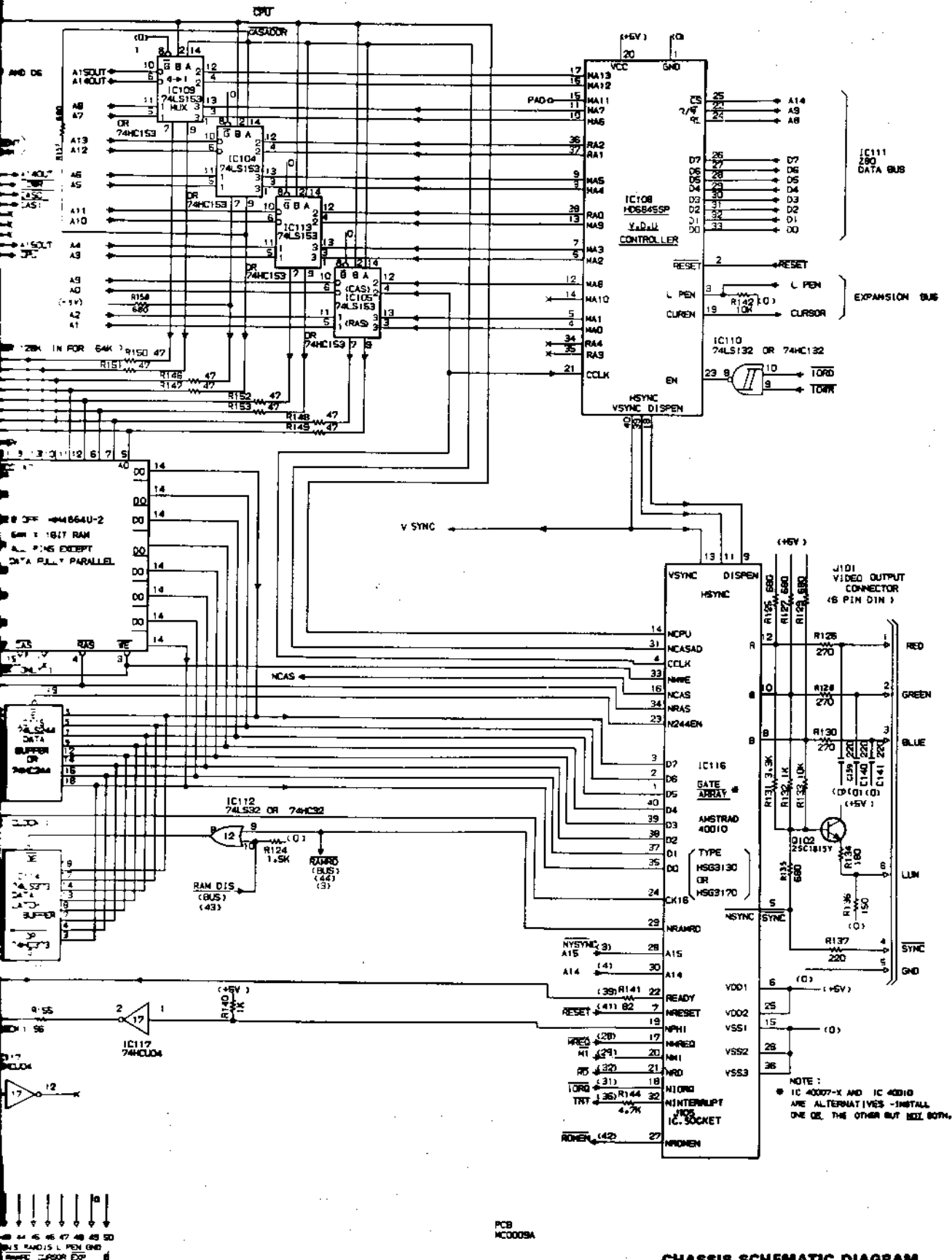


La liste des composants du CPC 6128

Valeur	Référence
Résistances à film de carbone	
47ohm	R146-153, 161, 162
56ohm	R155-306
82ohm	R141
150ohm	R136
180ohm	R134
220ohm	R137
270ohm	R126, 128, 130
560ohm	R317
680ohm	R125, 127, 129, 135, 157, 158, 301
1kohm	R113, 121-123, 132, 140, 211, 212, 216, 313, 315, 321
1k5ohm	R124
2k2ohm	R102, 109, 110, 112, 214
3k3ohm	R131
4k7ohm	R144, 310, 323
10kohm	R114-117, 120, 133, 142, 217, 309, 312
18kohm	R308, 311
22kohm	R118, 119, 304, 305
47kohm	R302, 303, 307, 319, 320
470kohm	R314
1Mohm	R111
3M3ohm	R101
10Mohm	R143, 145
4ohm 1/4W	Fuse R322
100ohm 1/2W	R316
Condensateurs céramiques	
15pF	C137, 138
220pF	C139-141, 310
270pF	C313
470pF	C306
0.1uF	C101-135, 201, 213
Condensateurs électrolytiques	
1uF/50V	C309, 311, 314, 317, 318
22uF/10V	C308
47uF/10V	C136, 303, 306
100uF/10V	C301, 304
100uF/16V	C315
Condensateurs polycarbonate (tension max. 50 V continu)	
0.001uF	C312
0.01uF	C305
0.047uF	C318
0.068uF	C302
0.1uF	C319, 320
Diodes	
D101, 303, 304	1S2472-HS
D201	DS442XFA5
D301	10E1
D302	SLP-155B(R)

Référence	Description
Circuits Intégrés	
IC101	HD74LS145
IC102	AY-3-8912
IC103	TMM-23256P-1953
IC104, 105, 109	HD74LS153
113	
IC106	HD74LS273
IC107	M5L8255AP-5
IC108	HD6845SP
IC110, 210	HD74LS132
IC111	Z8400APS
IC112, 207	HD74LS32
IC114	HD74LS373
IC115	HD74LS244
IC116	HSG3130/3170
IC117	TC74HCU04P
IC118	PAL 16L8AC
IC119-134	MSM3764-20RS
IC201	UPD765AC-2
IC202	FDC9216BT
IC203	SN74HC240N
IC204	TMM-23128P-1851
IC205	DN74LS08
IC206, 208	DN74LS38
IC209	DN74LS136
IC211	DN74LS27
IC212	DN74LS74
IC213	TC74HC161P
IC301	LA4140
IC302	LA6358S
Transistors	
Q101	2SC2120Y
Q102, 301, 302	2SC1815Y
W303	2SC950Y
Autres	
J101	Jack DIN
J102	Connecteur D Sub 9
J103	Jack RCA3.5
J104, 105	Connecteur IC 20 Pin DIL
J106	Connecteur IC 10 Pin DIL
J301	Jack D.C.
J302	Jack DIN
VR301	Vol. Rot. 20k
CD302	D.C. Cord
CD201	Cord Connector
FDD201	Lecteur de disquette EME155
NR101	R. Network Exb P86222J
NR201	R. Network Exb P87681J
RY301	Relai cassette G4S-1112P-1-B-19
SP301	Interlocuteur CO40K01K2451
X101	Quartz HC-18RW 16MHz





2/2.4

Différences de conception

Les trois CPC ont des différences minimales au niveau matériel.

Si nous partons du modèle de base CPC 464, le passage au CPC 664 a permis essentiellement :

- d'implanter un lecteur de disquettes interne à la place du lecteur de cassettes,
- d'améliorer légèrement le BASIC logé dans les ROM hautes (voir partie 4 chap. 1.3).

Le passage du CPC 664 au CPC 6128 a consisté à reprendre le circuit de base du 664 et à lui adjoindre la logique nécessaire (commutation de banques de mémoire) pour piloter 128 KO de RAM et 48 KO de ROM (au lieu de 64 KO de RAM et 48 KO de ROM).

2/3

Les circuits intégrés spécialisés

2/3.1

Le microprocesseur Z80

Le Z80 est un microprocesseur LSI (Large Scale Integration, ou intégration à grande échelle) 8 bits complètement compatible avec le 8080A (le Z80 accepte tous les ordres du 8080A) qui est né avant lui, il y a une dizaine d'années. Ces deux microprocesseurs ont été beaucoup utilisés, et sont encore beaucoup utilisés sur de petits systèmes industriels.

Le Z80 a une capacité d'adressage de 64 Kilo-octets de mémoire RAM ou ROM grâce à son bus d'adresse de 16 bits ($2^{16} = 65536$, soit 64 Kilo-Octets).

Son bus de données est de 8 bits, ce qui le classe dans la famille des micro-processeurs 8 bits.

Sa fréquence d'horloge typique est de 4 MHz (Méga Hertz), ce qui lui confère une place confortable parmi les processeurs 8 bits du même type installés sur les micro-ordinateurs familiaux, qui, souvent, acceptent une fréquence d'horloge maximum de 1 ou 2 MHz.

Sur les systèmes AMSTRAD CPC, le Z80 est cadencé par une horloge à 4 MHz, et adresse :

- pour le CPC 464 : 64 KO de RAM et 32 KO de ROM,
- pour le CPC 664 : 64 KO de RAM et 48 KO de ROM,
- pour le CPC 6128 : 128 KO de RAM et 48 KO de ROM.

Ces capacités d'adressage dépassent les 64 KO possibles, et on a recours à un artifice appelé « pagination » pour dépasser le seuil fatidique des 64 KO.

La pagination met en œuvre des pages mémoire (d'où son nom) couramment appelées *banques RAM* ou *banques ROM* suivant leur nature. Sur AMSTRAD, une banque fait 16 KO. A un moment donné, le microprocesseur pourra adresser 4 banques quelconques de mémoires RAM ou ROM. Pour adresser plus de 64 KO, il suffira de faire commuter les banques quand cela sera nécessaire (quand le Z80 voudra accéder à une banque différente des 4 courantes).

Le Z80 possède 26 octets de mémoire RAM interne répartis dans les registres suivants :

A	F	A'	F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'

Vecteur d'interruption I

Rafraîchissement mémoire R

Registres d'index IX et IY

Pointeur de pile SP

Compteur de programme PC

PC

Program Counter (ou compteur de programme) est un registre 16 bits qui pointe sur l'adresse de l'instruction courante.

SP

Stack Pointer (ou pointeur de pile) est un registre 16 bits qui donne l'adresse de la pile LIFO (Last In First Out, ou dernier entré premier sorti) située en RAM externe. Les instructions de manipulation de la pile sont POP (sortie de pile) et PUSH (mise en pile).

La désignation LIFO caractérise le type d'entrée/sortie des données dans un espace mémoire appelé *pile*. La dernière donnée entrée par l'instruction PUSH est la première sortie par l'instruction POP. Ainsi, si vous faites :

PUSH HL

PUSH BC

PUSH IX**POP BC**

Le registre pair BC contiendra la dernière valeur mise en pile par l'instruction PUSH, c'est-à-dire la valeur contenue dans IX. Si vous refaites un POP, c'est BC qui sera dépilé, et si vous refaites un troisième POP, c'est HL qui sera dépilé.

Les piles gérées en interne par la logique des microprocesseurs sont pratiquement toujours du type LIFO.

Cependant, il existe d'autres types de piles appelées LIFO et FIFO. La pile FILO (First In Last Out, ou premier entré dernier sorti) est assez rarement utilisée.

La pile FIFO (First In First Out, ou premier entré premier sorti) est surtout utilisée pour stocker des données que l'on ne peut pas traiter instantanément (pour des problèmes d'occupation du microprocesseur à d'autres tâches, par exemple). Ces données sont mémorisées, et la première donnée mémorisée est la première à être traitée.

IX**IY**

Ce sont des registres 16 bits utilisés en adressage indexé. Ils donnent la base de l'adressage. Reportez-vous à la partie 4 chap. 2.2 pour avoir plus de détails à ce sujet.

Nous ferons par exemple : LD A,(IX+4)

Cette instruction charge dans le registre A le contenu de la mémoire pointée par l'adresse IX+4.

I

Registre d'interruption qui donne la partie haute de l'adresse où doit se produire le débranchement en cas d'interruption.

R

Registre de rafraîchissement. Il contient l'adresse basse qui sera mise sur le bus d'adresse à chaque fois que le CPU est en train de lire et de décoder une instruction. Cette façon de procéder fait en sorte qu'aucune perte de temps n'est engendrée, puisque le bus est réquisitionné pour rafraîchir la mémoire pendant que le Z80 ne l'utilise pas.

Ce registre est automatiquement incrémenté à chaque instruction.

A

Accumulateur. C'est par lui que passent la plupart des opérations logiques ou arithmétiques sur 8 bits. Son accès est très rapide et les modes d'adressages qui le concernent sont multiples et variés.

F

Flags (ou drapeaux). Ils donnent l'état de la dernière opération effectuée.

Les registres A et F peuvent être manipulés simultanément sous la forme du registre pair AF.

Registres à usage général

B, C, D, E, H et L sont des registres 8 bits qui peuvent être utilisés comme registres pairs (16 bits) sous les dénominations suivantes : BC, DE et HL. Ils possèdent leurs homonymes :

- sur 8 bits : B', C', D', E', H' et L',
- sur 16 bits : BC', DE' et HL'.

LE CIRCUIT INTÉGRÉ Z80

C'est un circuit à 40 pattes dont le brochage est le suivant :

M1		A0
		A1
$\overline{\text{MREQ}}$		A2
		A3
$\overline{\text{IORQ}}$		A4
		A5
$\overline{\text{RD}}$		A6
		A7
$\overline{\text{WR}}$		A8
		A9
$\overline{\text{RFSH}}$		A10
		A11
$\overline{\text{HALT}}$	Z80	A12
	CPU	A13
$\overline{\text{WAIT}}$		A14
		A15
$\overline{\text{NMI}}$		
$\overline{\text{RESET}}$		Ø
$\overline{\text{BUSRQ}}$		D0
		D1
$\overline{\text{BUSAQ}}$		D2
		D3
GND		D4
		D5
+5V		D6
		D7
GND		

A0 à A15

Bus d'adresse tristate en sortie actif à l'état haut sur 16 bits. A0 est le bit le moins significatif (LSB), et A15 le bit le plus significatif (MSB).

D0 à D8

Bus de données tristate en entrée/sortie actif à l'état haut sur 8 bits. D0 est le bit le moins significatif (LSB), et D7 le bit le plus significatif (MSB).

M1 :

Sortie active à l'état bas, signale que le cycle horloge courant est consacré à une identification d'op-code, ou qu'un IORQ est actif.

MREQ :

Sortie tristate active à l'état bas. Signale que le bus d'adresse est en attente d'une adresse pour lire ou écrire en mémoire.

IORQ :

Sortie tristate active à l'état bas. Signale l'un des deux événements suivants :

- 1°) la partie de poids faible du bus d'adresse est en attente d'une adresse pour lire ou pour écrire en mémoire ;
- 2°) un vecteur d'interruption peut être placé sur le bus de données pour être acquis.

RD :

Sortie tristate active à l'état bas. Signale que le CPU veut lire des données en mémoire ou sur un périphérique.

WR :

Sortie tristate active à l'état bas. Signale que le bus de données contient une donnée qui peut être stockée en mémoire ou envoyée vers un périphérique.

RFSH :

Sortie active à l'état bas. Indique que les 7 bits de poids faible du bus d'adresse contiennent une adresse de rafraîchissement des RAM dynamiques.

HALT :

Sortie active à l'état bas. Indique qu'une instruction HALT a été exécutée par le CPU. Ce dernier est en attente d'une interruption. Pendant ce temps, il exécute des NOP pour pouvoir rafraîchir ses RAM dynamiques.

WAIT :

Entrée active à l'état bas. Indique que la mémoire ou le périphérique n'est pas prêt à être accédé.

INT :

Entrée active à l'état bas. INT est générée par un périphérique pour signaler qu'il désire qu'on s'occupe de lui. La requête prendra effet à la fin de l'exécution de l'instruction courante si le flag IFF est valide et si **BUSRQ** est inactif.

NMI :

Entrée active sur front bas. Cette demande d'interruption est plus prioritaire que INT. Elle est prise en compte quel que soit l'état du flag IFF. Cette interruption débranche le CPU à l'adresse 0066H. Le PC est automatiquement sauvegardé, et la sortie d'interruption ramènera le PC à l'endroit où il avait été sauvegardé.

RESET :

Entrée active à l'état bas. Le PC est mis à 0 et le CPU est initialisé : IFF = 1, I = 0, R = 0, IM = 0. Pendant la phase de reset, les bus de données et d'adresses sont dans l'état tristate.

BUSRQ :

Entrée active à l'état bas. Signale qu'un périphérique demande un accès aux bus de données, d'adresses et signaux de contrôles. Ces diverses pattes vont passer dans un état tristate.

BUSAQ :

Sortie active à l'état bas. Signale que les bus d'adresses, de données et les signaux de contrôle sont dans un état tristate, et que le périphérique qui a demandé de les contrôler peut le faire dès à présent.

 ϕ :

Horloge de cadencement du microprocesseur.

2/3.2

Le contrôleur d'écran : CRTC et VGA

2/3.2.1

CRT Controller ou CRTC

Le circuit CRTC de référence MC 6845 proposé par MOTOROLA est un circuit intégré à haute intégration comportant 40 broches, dont la fonction est la génération des signaux nécessaires pour produire une image qui sera affichée sur l'écran cathodique d'un moniteur ou d'un téléviseur.

L'organisation interne de ce circuit est la suivante :
(Voir schéma page suivante.)

Nous voyons que le CRTC contient 19 registres.

Ces registres sont chargés automatiquement par l'AMSTRAD au moment où vous le mettez sous tension. Ils ne doivent pas être modifiés sauf si vous désirez créer des effets spéciaux ou interfacer un light-pen.

Leur utilisation est la suivante :

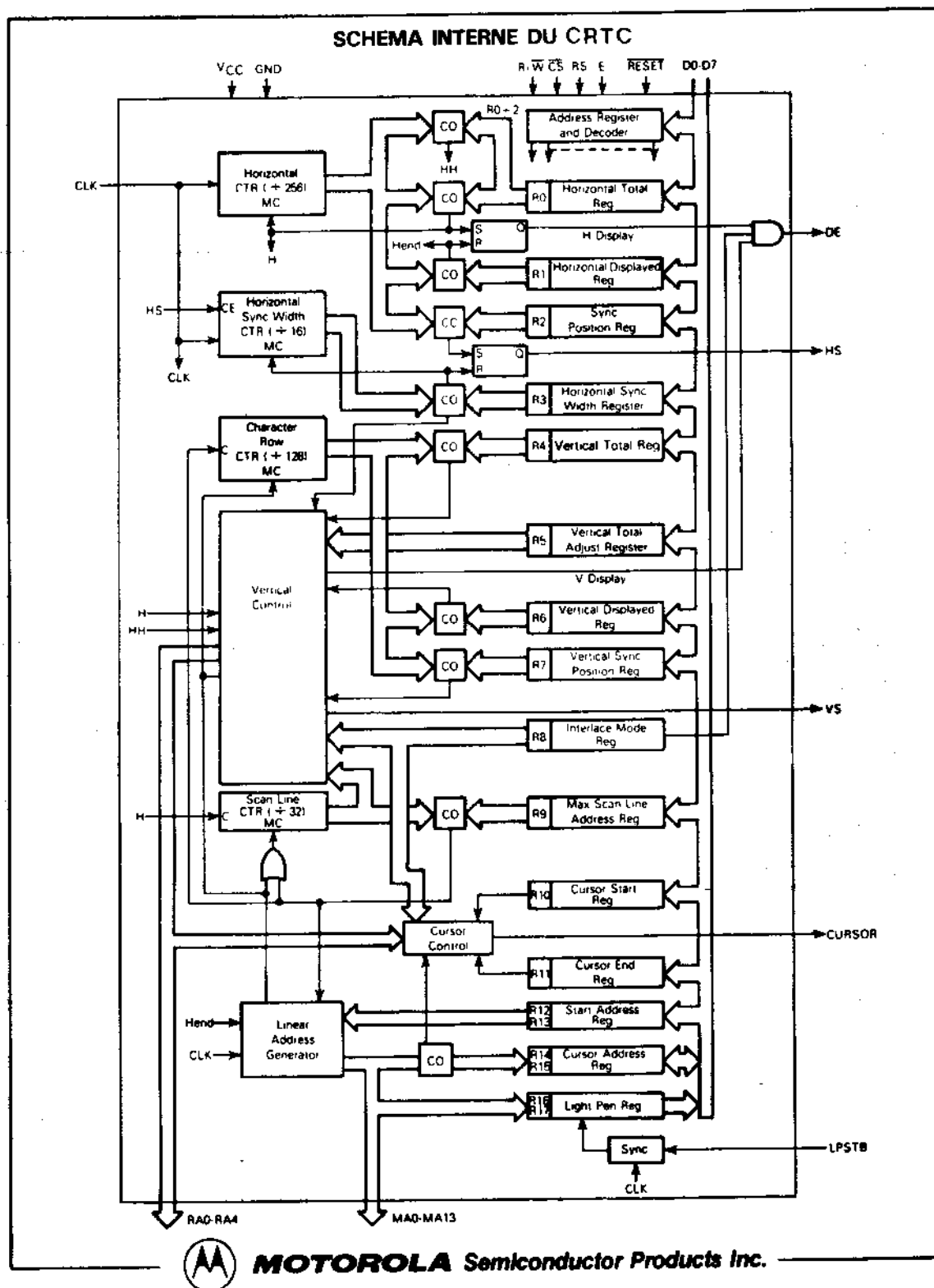
AR = ADDRESS REGISTER

(Registre d'adresse)

Ce registre est accédé à travers le bus de données.

Il permet de sélectionner un des 18 registres internes (R0 à R17).

Quand RS = 0 et CS = 0, le registre d'adresse est accessible.



Quand $RS = 1$ et $CS = 0$, le registre pointé par AR est accessible.

R0 = HORIZONTAL TOTAL REGISTER

(Fréquence horizontale totale)

Ce registre 8 bits à écriture seule définit la fréquence de synchronisation horizontale : temps nécessaire pour afficher une ligne horizontale + temps nécessaire pour le retour du spot - 1.

R1 = HORIZONTAL DISPLAYED REGISTER

(Nombre de caractères affichés par ligne)

Ce registre 8 bits à écriture seule définit le nombre de caractères affichés par ligne.

Remarque : R1 doit être inférieur ou égal à R0.

R2 = HORIZONTAL SYNC POSITION REGISTER

(Synchronisation horizontale)

Ce registre 8 bits à écriture seule permet de contrôler la synchronisation horizontale : délai de synchronisation (front porch) et délai de balayage horizontal (back porch).

Remarque : R2 doit être inférieur ou égal à R0 et supérieur à R1.

R3 = SYNC WIDTH REGISTER

(Largeur de la synchronisation horizontale)

Ce registre 8 bits à écriture seule dont 4 bits seulement sont utiles définit la largeur du top de synchronisation horizontale. Il doit avoir une valeur comprise entre 1 et 15. Il est exprimé en multiple du nombre de périodes d'horloge nécessaires à l'affichage d'un caractère.

R4 = VERTICAL TOTAL REGISTER

(Nombre de lignes par écran)

Ce registre 8 bits à écriture seule dont 7 bits seulement sont utiles détermine le nombre entier de lignes de caractères - 1 à partir duquel sera calculée la fréquence de synchronisation verticale.

R5 = VERTICAL TOTAL ADJUST REGISTER

(Ajustage du nombre de lignes par écran)

Ce registre 8 bits à écriture seule dont 5 bits seulement sont utiles déter-

mine le nombre de lignes de trames à ajouter à R4 pour calculer la fréquence de synchronisation verticale.

R6 = VERTICAL DISPLAYED REGISTER

(Nombre de lignes affichées à l'écran)

Ce registre 8 bits à écriture seule dont 7 seulement sont utiles définit le nombre de lignes de caractères affichés à l'écran.

Remarque : R6 doit être inférieur à R4.

R7 = VERTICAL SYNC POSITION

(Synchronisation verticale)

Ce registre 8 bits à écriture seule dont 7 seulement sont utiles permet de contrôler la synchronisation verticale. Il définit le nombre de lignes de trames à balayer avant d'activer le signal VSYNC.

Remarque : R7 doit être inférieur à R4 et supérieur à R6.

R8 = INTERLACE MODE AND SKEW REGISTER

(Mode normal ou entrelacé)

Ce registre 8 bits dont 2 seulement sont utiles permet de définir le mode de fonctionnement du CRTC :

Bit 0	Bit 1	Fonction
0/1	0	Mode normal (non entrelacé)
0	1	Mode entrelacé
1	1	Mode entrelacé et vidéo

Les modes définis ci-dessus ont la signification suivante :

Par exemple, sur la lettre T :

000000

00

00

00

00

00

00

Mode normal

000000

000000

00

00

00

00

00

00

Mode entrelacé

000000

000000

00

00

00

00

00

Mode entrelacé et vidéo

R9 = MAXIMUM SCAN LINE ADDRESS REGISTER

(Nombre de lignes élémentaires par caractère)

Ce registre 8 bits à écriture seule dont 5 bits sont utiles permet de définir le nombre de lignes élémentaires par caractère - 1.

R10 = CURSOR START REGISTER

(Début du curseur)

Ce registre 8 bits à écriture seule dont 7 bits sont utiles permet de définir la position de la première ligne élémentaire du curseur (bits 0 à 4) et la vitesse de clignotement du curseur.

Les bits 5 et 6 définissent la vitesse de clignotement comme suit :

Bit 6	Bit 5	Fonction
0	0	Pas de clignotement
0	1	Curseur invisible
1	0	Clignotement lent
1	1	Clignotement rapide

R11 = CURSOR END REGISTER

(Fin du curseur)

Ce registre 8 bits à écriture seule dont 5 bits sont utiles permet de définir la position de la dernière ligne élémentaire du curseur (bits 0 à 4).

R12-R13 = START ADDRESS REGISTER

(Adresse de la RAM d'écran)

Registres à lecture/écriture.

Première adresse de la RAM d'écran affichée après un « vertical blanking ». Les 6 bits de poids faible de R12 donnent le poids fort de cette adresse, et les 8 bits de R13 le poids faible de cette adresse.

R14-R15 = CURSOR REGISTER

(Position du curseur)

Registres à lecture/écriture.

Permet de positionner le curseur n'importe où sur l'écran. Les 6 bits de poids faibles de R14 donnent le poids fort de l'adresse, et les 8 bits de R15 le poids faible de l'adresse.

R16-R17 = LIGHT-PEN REGISTER

(Position du light-pen)

Registres à lecture seule.

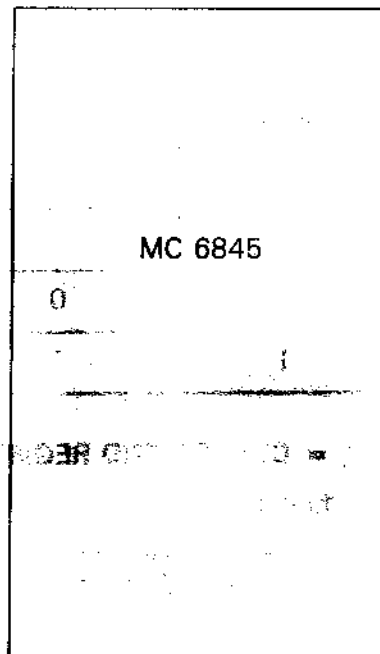
Permet de connaître la position du light-pen sur l'écran.

Les 6 bits de poids faible de R17 donnent le poids fort de l'adresse, et les 8 bits de R16 le poids faible de l'adresse.

LE CIRCUIT INTÉGRÉ MC 6845

C'est un circuit à 40 pattes dont le brochage est le suivant :

GND
 RESET
 LPSTB
 MA0
 MA1
 MA2
 MA3
 MA4
 MA5
 MA6
 MA7
 MA8
 MA9
 MA10
 MA11
 MA12
 MA13
 DE
 CURSOR
 VCC



VS
 HS
 RA0
 RA1
 RA2
 RA3
 RA4
 D0
 D1
 D2
 D3
 D4
 D5
 D6
 D7
 CS
 RS
 E
 R/W
 CLK

D0 à D7 :

(DATA BUS ou bus de données). Bus de données bidirectionnel tristate entre CRTC et micro-processeur.

E :

(ENABLE ou validation). Signal tristate compatible avec la logique TTL et CMOS. Un front descendant sur cette broche valide les données présentes sur le bus de données.

CS :

(CHIP SELECT ou validation du circuit). Signal tristate compatible avec la logique TTL et CMOS. Quand ce signal est bas, l'accès au CRTC est valide : les registres internes peuvent être lus ou écrits.

RS :

(REGISTER SELECT ou sélection de registre). Signal tristate compatible avec la logique TTL et CMOS. Si RS = 0, le registre d'adresse peut être sélectionné, si RS = 1, le registre de données ou un registre interne peut être sélectionné.

: 22V to 33V

R/W :

(READ/WRITE ou lecture/écriture). Signal tristate compatible avec la logique TTL et CMOS. Détermine si les registres internes vont être accédés en lecture ou en écriture.

VS et HS :

(VERTICAL SYNC and HORIZONTAL SYNC ou synchronisations verticale et horizontale). Sorties compatibles avec la logique TTL actives à l'état haut. Ces signaux sont destinés au processeur vidéo pour générer un signal composite.

DE :

(DISPLAY ENABLE ou affichage valide). Sortie compatible avec la logique TTL active à l'état haut. Indique si le CRTC est en train de manipuler une adresse en RAM écran.

MA0 à MA13 :

(REFRESH MEMORY ADDRESSES ou adresses de rafraîchissement de mémoire d'écran). Sorties destinées au rafraîchissement des mémoires RAM écran par bloc de 16 KO (Kilo-Octets).

RA0 à RA4 :

(ROW ADDRESSES ou adresses de lignes). Sorties destinées à lire les caractères à afficher dans la ROM du générateur de caractères.

CURSOR :

Sortie compatible avec la logique TTL active à l'état haut. Indique qu'une adresse valide est disponible pour la logique vidéo externe.

CLK :

(CLOCK ou horloge). Entrée comptable avec la logique TTL et CMOS active sur un front descendant. Permet de synchroniser toutes les fonctions d'écran.

LPSTB :

(LIGHT PEN STROBE ou lecture de la position du stylo lumineux). Un front descendant sur cette broche positionne l'adresse courante de rafraîchissement de RAM d'écran dans le registre « light pen ».

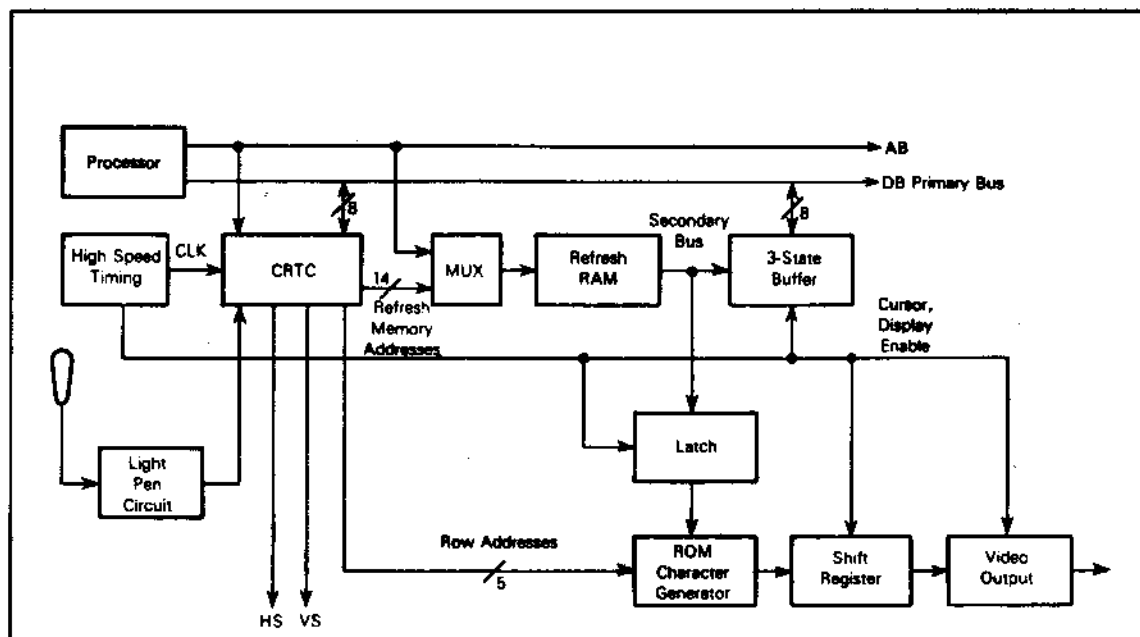
VCC et VSS :

Alimentation du CRTC : $VCC = 5V \pm 5\%$. $VSS = 0V$.

RESET :

Un niveau bas sur cette broche permet d'effacer le contenu de tous les registres internes du CRTC.

Le CRTC s'interface avec les divers circuits d'un ordinateur de la manière suivante :



(Document Motorola)

2/3.3

Le circuit sonore AY3-8912

Ce circuit LSI (Large Scale Integration ou intégration à grande échelle) proposé par GENERAL INSTRUMENTS permet de générer des sons sur 3 voies et du bruit blanc sur une voie mixable avec les trois autres. Il est prévu pour s'interfacer facilement avec toutes sortes de microprocesseurs 8 ou 16 bits, ce qui explique sa popularité sur les microordinateurs familiaux.

Il est disponible en boîtier de 40 broches sous la référence AY-3-8910A et en boîtier 28 broches sous la référence AY-3-8912A. Le brochage de ces deux boîtiers est donné ci-dessous :

VSS	AY-3-8910 A	VCC	AY-3-8912A	DA0
NC		NC		DA1
Analog B		Analog C		DA2
Analog A		DA0	Analog B	DA3
NC		DA1	Analog A	DA4
IOB7		DA2	VSS	DA5
IOB6		DA3	IOA7	DA6
IOB5		DA4	IOA6	DA7
IOB4		DA5	IOA5	BC1
IOB3		DA6	IOA4	BC2
IOB2		DA7	IOA3	BDIR
IOB1		BC1	IOA2	A8
IOB0		BC2	IOA1	RESET
IOA7		BDIR	IOA0	CLOCK
IOA6		NC		
IOA5		A8		
IOA4		A9		
IOA3		RESET		
IOA2		CLOCK		
IOA1		IOA0		

DA0 à DA7 :

Entrées/Sorties tristate. Ces 8 broches véhiculent

- les données et adresses envoyées par le microprocesseur au circuit sonore,
- les données envoyées par le circuit sonore à destination du microprocesseur.

Quand le mode « adresse » est actif, les bits DA0 à DA3 sélectionnent un des registres internes, et les bits DA4 à DA7 servent de « chip select » avec les broches A9 et A8 :

A9 et A8 :

Leur fonction est équivalente à un « chip select », c'est-à-dire que lorsque A9 est à l'état bas et A8 à l'état haut, le circuit AY3 peut être accédé. Si A9 et A8 ne sont pas utilisées dans ce sens, il est recommandé de les relier respectivement à la masse et au + VCC à travers une résistance pour éviter les problèmes de sélection/désélection intempestives qui pourraient arriver en environnement bruité.

Remarque : Ces deux broches ne sont pas disponibles sur la version à 28 broches AY-3-8912A.

RESET :

Permet d'effacer tous les registres internes du circuit sonore lorsqu'on lui applique un état bas.

CLOCK :

Entrée compatible avec le standard TTL. Permet de cadencer les générateurs de ton, de bruit et d'enveloppe.

BDIR, BC2 et BC1 :

Bus DIRection, Bus Control 1 et Bus Control 2. Ces entrées permettent de définir la fonction courante du AY3 comme suit :

BDIR	BC2	BC1	Fonction du AY3
0	1	0	Inactif
0	1	1	Lecture
1	1	0	Ecriture
1	1	1	Adresse sur le bus à mémoriser par AY3

Analog Channel A, B and C :

Sorties des convertisseurs digital/analogique A, B et C. Ils produisent des signaux d'un volt crête-crête maximum.

VCC et VSS :

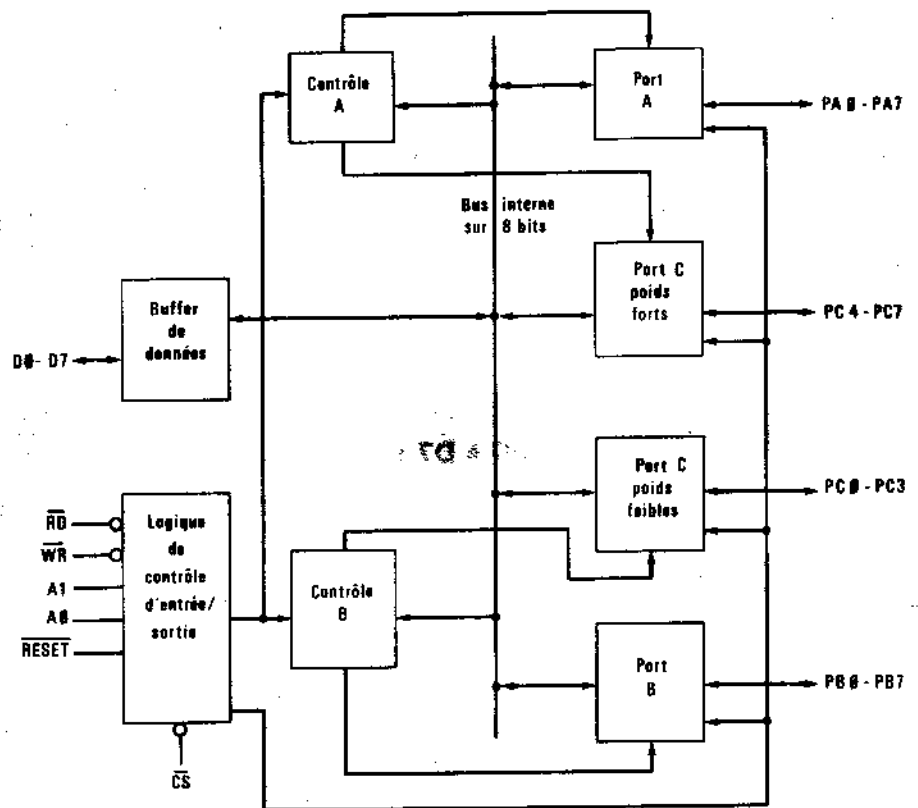
Alimentation du AY-3.

2/3.4

L'interface parallèle PIO 8255A

Le circuit 8255 proposé par INTEL est un circuit LSI (Large Scale Integration ou intégration à grande échelle) dont les fonctions concernent l'interfaçage de l'unité centrale avec divers équipements périphériques qui possèdent un bus d'entrée/sortie parallèle sur 4 ou 8 bits.

L'organisation interne de ce circuit est la suivante :

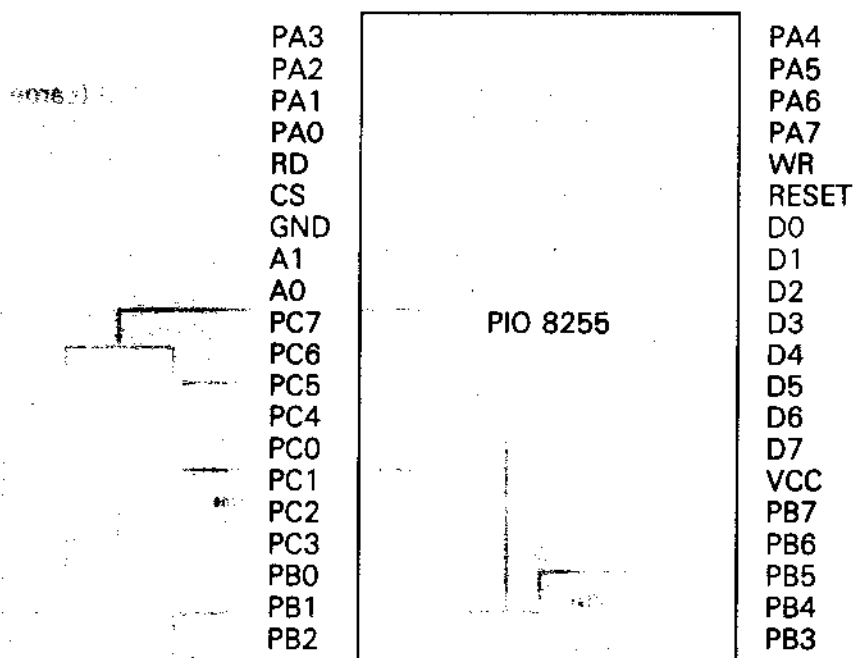


Nous voyons qu'il comporte trois ports d'entrée/sortie : A, B et C. Chaque port est relié au bus interne du 8255, lequel est relié au bus de données externe général (D0 à D7).

Les ports A, B et C sont connectés à deux registres de contrôle. Ces registres déterminent le mode de fonctionnement des trois ports. Les registres de contrôle sont accédés en écriture par le CPU. Ainsi, le PIO est entièrement configurable par programmation. Le port C est divisé en deux demi-ports de 4 bits (C UPPER et C LOWER). C UPPER est affecté au registre de contrôle A, et C LOWER au registre de contrôle B.

LE CIRCUIT INTÉGRÉ PIO 8255

C'est un circuit à 40 pattes dont le brochage est le suivant :



D0 à D7 :

Entrée/sortie : 8 bits de données connectés au CPU. Toutes les données ou ordres provenant du CPU transitent sur ces connections.

PA0 à PA7 :

Données sur 8 bits concernant le port d'entrée/sortie A sur 8 bits. Sur ce port est connecté le générateur sonore AY3-8912.

PB0 à PB7 :

Données sur 8 bits concernant le port d'entrée/sortie B sur 8 bits. Sur ce port sont connectés le lecteur de cassettes et le signal BUSY de l'imprimante.

PC0 à PC7 :

Données sur 8 bits concernant le port d'entrée/sortie C sur 8 bits. Sur ce port sont connectés les ports clavier et joystick.

 $\overline{CS}$:

Entrée active à l'état bas. Le PIO est accessible quand cette entrée est à l'état bas, c'est-à-dire qu'une donnée transitant sur un des ports A, B, C ou D ne sera accessible que si $\overline{CS}$ est à 0.

 $\overline{RD}$:

Entrée active à l'état bas. Si cette entrée est active en même temps que $\overline{CS}$ ($\overline{RD} = 0$ et $\overline{CS} = 0$), les données disponibles sur le port D sont lues par le CPU.

 $\overline{WR}$:

Entrée active à l'état bas. Si cette entrée est active en même temps que $\overline{CS}$ ($\overline{WR} = 0$ et $\overline{CS} = 0$), les données présentes sur le port de données D et envoyées par le CPU sont lues par le PIO.

A0 et A1 :

Entrées. Désignant le registre interne du PIO dans lequel vont se faire les opérations de lecture/écriture.

 $\overline{RESET}$:

Entrée active à l'état bas. Provoque une initialisation du PIO.

2/3.4.1

Exploitation du PIO 8255

Le système d'exploitation des CPC fait très largement appel aux possibilités du PIO 8255 pour des tâches aussi variées que la gestion du clavier, la programmation du synthétiseur de sons, le pilotage du lecteur de cassettes et la supervision de l'imprimante.

Moyennant certaines précautions visant à éviter les conflits avec ces opérations, le programmeur peut détourner à son profit une partie des vastes possibilités de ce composant spécialisé dans les entrées/sorties.

Il devra cependant savoir exactement ce qu'il fait, certaines instructions malencontreuses pouvant non seulement bloquer la machine, mais aussi endommager certains de ses composants.

Mode de sélection du PIO 8255

La figure 1 montre comment le 8255 est matériellement incorporé dans le schéma des CPC : la broche de sélection du PIO est directement reliée à la ligne A11 du bus d'adresses du Z80, ce qui signifie que pour communiquer avec le 8255, le microprocesseur devra faire passer cette ligne A11 à zéro, on consultera également la Partie 2 chapitres 2.1 p. 3, 2.2 p. 3, 2.3 p. 3.

Simultanément, le PIO doit être positionné en écriture ou en lecture selon que l'on veut exécuter une opération de sortie ou d'entrée. Pour mettre le 8255 en écriture, il faut appliquer un zéro logique à sa broche $\overline{WR}$, tandis que pour le mettre en lecture, c'est sa broche $\overline{RD}$ qu'il faut amener à zéro.

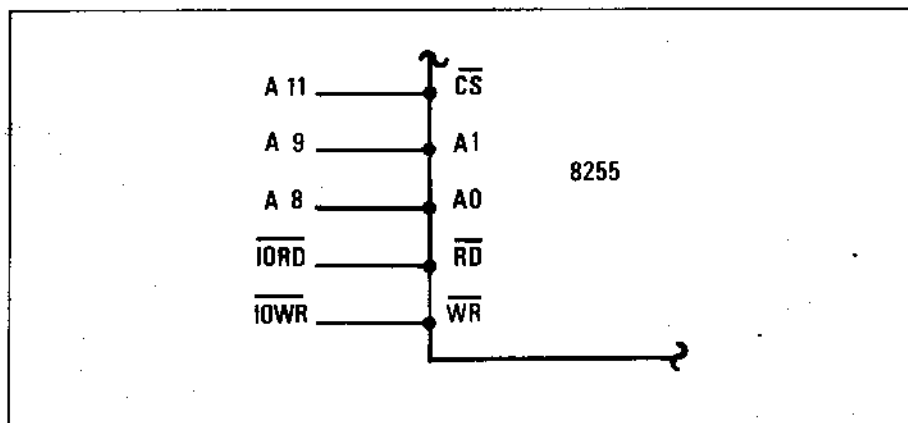


Fig. 1

La figure 2 montre comment les signaux $\overline{\text{IORQ}}$ et $\overline{\text{IOWR}}$, respectivement appliqués à $\overline{\text{RD}}$ et $\overline{\text{WR}}$ du PIO, sont obtenus à partir des lignes $\overline{\text{RD}}$, $\overline{\text{WR}}$, et $\overline{\text{IORQ}}$ du bus de commande du microprocesseur.

La ligne $\overline{\text{RD}}$ passe à zéro toutes les fois que le Z80 exécute une instruction de lecture, soit en mémoire, soit sur un port. Inversement, il positionne $\overline{\text{WR}}$ à zéro lorsqu'il exécute une instruction d'écriture en mémoire ou sur un port.

Parallèlement, il met la ligne $\overline{\text{IORQ}}$ à zéro s'il s'adresse à un port, et la ligne $\overline{\text{MREQ}}$ s'il s'adresse à la mémoire (ce qui nous ne nous concerne pas ici).

On constate ainsi que $\overline{\text{IORQ}}$ se positionne à zéro seulement en cas de lecture d'un port, et $\overline{\text{IOWR}}$ en cas d'écriture sur un port : ces deux lignes restent insensibles à toute opération en mémoire.

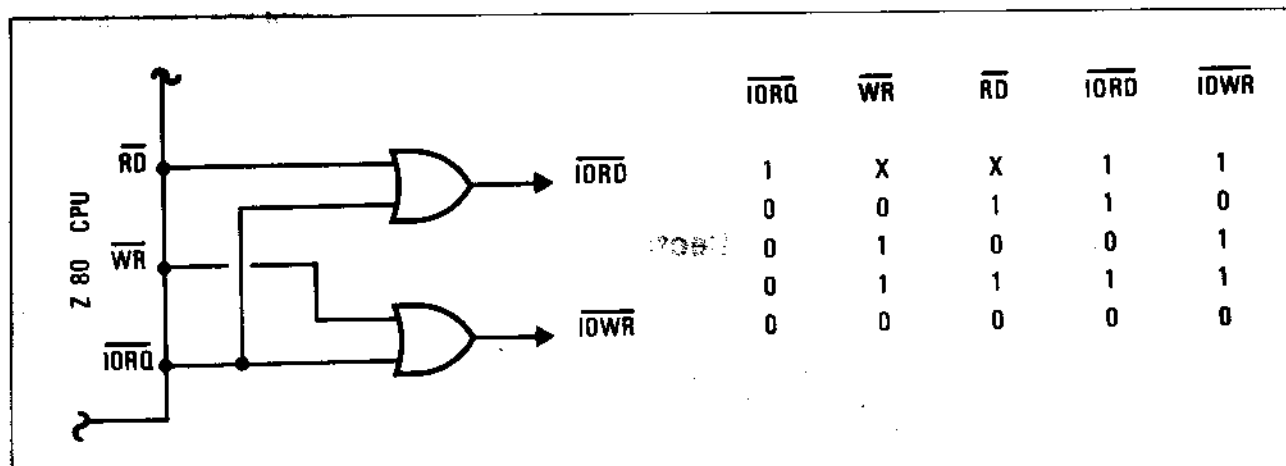


Fig. 2

Mode d'adressage du PIO 8255

Le 8255 est donc informé de la direction des échanges qui le concernent, mais doit encore savoir lequel de ses trois ports est « adressé ». Le microprocesseur transmet cette information sur les lignes A8 et A9 de son bus d'adresses, lesquelles rejoignent les entrées A0 et A1 du 8255.

Le tableau 1 résume les différentes combinaisons possibles entre les états de toutes les lignes desservant le PIO. Grâce au tableau 2, vous comprendrez aisément comment sont calculées les adresses de ports des instructions Basic INP et OUT permettant de communiquer avec le 8255.

En théorie, l'état des lignes d'adresse ne desservant pas le 8255 pourrait sembler indifférent, ce qui inciterait à croire que d'autres adresses mèneraient aux mêmes résultats, ce qui est exact en ce qui concerne le seul 8255. Cependant, d'autres dispositifs d'entrée/sortie sont desservis par les mêmes bus, et réveillés par des états zéro sur d'autres lignes d'adresse.

Tableau 1 : Fonctionnement des ports d'entrée et de sortie du 74180

Basic	A ₁ (A ₉)	A ₀ (A ₈)	RD	WR	CS	Action
ENTRÉE INP (29951) INP (30207) INP (30463)	0	0	0	1	0	A → Bus de données
	0	1	0	1	0	B → Bus de données
	1	0	0	1	0	C → Bus de données
	1	1	0	1	0	Combinaison illégale
SORTIE OUT 62719 OUT 62975 OUT 63231 OUT 63487	0	0	1	0	0	Bus de données → A
	0	1	1	0	0	Bus de données → B
	1	0	1	0	0	Bus de données → C
	1	1	1	0	0	Bus de données → registre contrôle
Repos	X	X	X	X	1	Bus de données en haute impédance
	X	X	1	1	0	Bus de données en haute impédance

Tableau 2

A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	OUT
1	2	4	8	16	32	64	128	256	512	1 024	2 048	4 096	8 192	16 384	32 768	/
1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	62 719
1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	62 975
1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	63 231
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	63 487

L'imprimante, par exemple, est sélectionnée par un niveau zéro sur la ligne A12.

Tenons-nous en donc strictement aux adresses indiquées au tableau 1 sous peine de risquer de commander plusieurs périphériques à la fois.

Attention :

Par ailleurs, il faut veiller à n'écrire que sur les ports de sortie, et, bien que cela soit moins important pour la santé du matériel, à ne lire que sur les ports d'entrée : écrire sur un port d'entrée risque de causer un conflit de données susceptible de détériorer des composants.

Imaginons que nous tentions de sortir un 1 logique sur une ligne maintenue à zéro par un autre dispositif : nous créons un court-circuit qui va faire circuler un courant nettement plus important qu'en temps normal, d'où un échauffement excessif et dangereux dans les circuits intégrés...

Exploitation du PIO 8255 dans les Amstrad

En temps normal, le système d'exploitation des CPC affecte comme suit les trois ports d'entrée/sortie du 8255 :

- port A : tantôt en entrée, tantôt en sortie (communications bidirectionnelles avec le AY-3-8912 utilisé tour à tour en synthétiseur de son et en port d'entrée/sortie pour le clavier et les manettes de jeu) ;
- port B : toujours en entrée (données lues sur cassettes, synchronisation vidéo, broche d'expansion du connecteur, cavaliers de configuration LK1 à LK4) ;
- port C (séparé en deux moitiés) : toujours en sortie (données à écrire sur cassette, moteur du magnétophone, commande du AY-3-8912).

Sans modifier ces choix effectués par le système d'exploitation, nous allons illustrer ces informations par un exemple pratique.

Le schéma électrique du CPC 464 nous montre que le moteur du magnétophone est mis en service par une tension positive (1 logique) sur la ligne C4 du port C du PIO. Le tableau 3 rappelle par ailleurs que le « poids » décimal de toute ligne de données numérotée « 4 » est de 16 (2 à la puissance 4).

Tableau 3

Bus de données								
D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	Ligne n°
1	2	4	8	16	32	64	128	Poids décimal

Le tableau 1 nous permet de déterminer que pour positionner à 1 la ligne C4, il faut programmer, en Basic, **OUT 63231,16**. Essayons, et constatons que cette commande fait bien démarrer le moteur, à condition évidemment que la touche **PLAY** soit enfoncée afin de débloquer la mécanique.

Vérifions inversement que **OUT 63231,0** arrête bien le moteur. Nous disposons dorénavant de commandes identiques aux **MOTOR ON** et **MOTOR OFF** présentes sur des machines d'autres marques (Thomson par exemple).

A titre d'exercice, essayez de lire, par un INP approprié, l'état de la broche **BUSY** du connecteur d'imprimante (reliée à la ligne B6 du 8255)...

Possibilités supplémentaires de programmation

Même s'il nous faut insister sur les risques que suppose toute expérimentation hasardeuse à ce niveau, nous devons, pour être complets, donner les informations permettant de modifier les modalités d'utilisation des ports A, B et C du 8255.

Précisons bien que de telles modifications ne sont guère envisageables que sous langage machine, le système d'exploitation se chargeant généralement, sous Basic, d'annuler toute commande ne lui convenant pas.

Il faut savoir que le 8255 possède trois modes de fonctionnement pouvant être sélectionnés par programmation :

- le mode 0, qui permet d'effectuer des opérations d'entrée/sortie directes sur chacun des trois ports, sans aucun contrôle. Le port C peut, pour sa part, être séparé en deux ports à quatre lignes ;

- le mode 1, dans lequel les entrées/sorties peuvent être contrôlées par le port C, dont quatre bits sont affectés aux signaux handshake (poignée de main) entre le 8255 et les périphériques avec lesquels il communique ;

- le mode 2, permettant des liaisons bidirectionnelles sur le port A : cinq bits du port C contrôlent ces échanges, conformément aux affectations résumées aux tableaux 4 et 5.

Insistons encore une fois sur le fait que, bien que le registre de contrôle défini au tableau 4 soit accessible en Basic par **OUT 63487**, la modification de son contenu est une affaire de programmeur averti, possédant une bonne connaissance du matériel. Cela d'autant plus que ce registre ne peut pas être lu : il n'est pas possible, dès lors, de prendre connaissance de son contenu afin de pouvoir le restaurer après une modification.

La lecture des registres du AY-3-8912, à l'inverse, est possible. Et comme ce composant dialogue avec le 8255 par l'intermédiaire du port A, il y a là une voie à explorer, avec l'aide de la Partie 6.

Tableau 4

D ₀	4 bits de poids faible de C : 1 = entrée, 0 = sortie				Groupe II
D ₁	B 1 = entrée - 0 = sortie				
D ₂	mode : 0 ou 1				
D ₃	4 bits de poids fort de C : 1 = entrée, 0 = sortie				Groupe I
D ₄	A 1 = entrée - 0 = sortie				
D ₅	0	0	1	1	
	mode 0	mode 1	mode 2	mode 2	
D ₆	0	1	0	1	
D ₇	1 : sélection de mode - 0 : positionnement de bits				

Tableau 5

Tableau 5

D ₀	Valeur du bit : 1 ou 0								
D ₁	N° du bit	0	1	2	3	4	5	6	7
D ₂		0	1	0	1	0	1	0	1
		0	0	1	1	0	0	1	1
D ₃		0	0	0	0	1	1	1	1
D ₄	Inutilisés								
D ₅									
D ₆									
D ₇	1 : sélection de mode - 0 : positionnement de bits.								

2/3.5

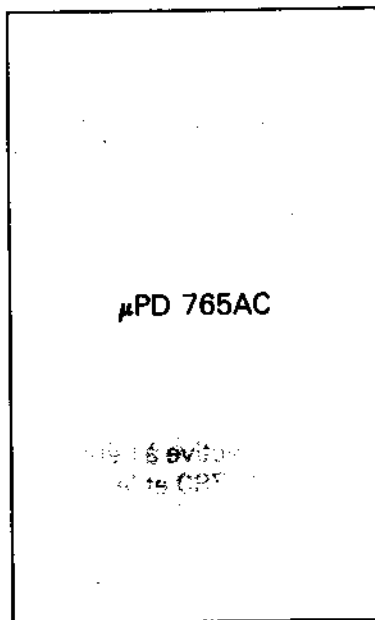
Le contrôleur de disque μ PD 765AC

Ce circuit intégré à 40 broches se trouve logé dans les CPC 664 et 6128, ou dans le connecteur qui se branche sur un CPC 464.

Le μ PD 765 constitue l'interface entre le lecteur de disquettes et le Z80.

Le brochage du contrôleur de disque est le suivant :

RESET
RD
WR
CS
RO
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
DRO
DACK
TC
INDEX
INT
CLK
GND



VCC
RW/SEEK
LCT/DIR
FLTR/STEP
HDLD
READY
WPRT/2 SIDE
FLT/TRK0
PS0
PS1
WDA
US0
US1
(HD) SIDE
MFM
WE
VCO
RDA
RDW
WCK

RESET :

Entrée active sur niveau haut. Cette broche permet de réinitialiser le contrôleur. Dans l'état de fonctionnement normal, elle est dans un état bas.

$\overline{RD}$:

Entrée active à l'état bas. Cette broche est reliée au $\overline{RD}$ du Z80. Elle est active lorsque le microprocesseur désire lire des données sur une disquette.

 $\overline{WR}$:

Entrée active à l'état bas. Cette broche est reliée au $\overline{WR}$ du Z80. Elle est active lorsque le microprocesseur désire écrire des données sur une disquette.

 $\overline{CS}$:

Entrée active à l'état bas. Cette broche valide les informations présentes sur les entrées $\overline{RD}$ et $\overline{WR}$ et plus généralement l'état actif du contrôleur.

DB0 à DB7 :

Bus de données relié au bus de données du Z80. C'est par ces 8 broches que sont acheminées les commandes et les données lues ou à écrire sur disquette.

DRQ :

Sortie active à l'état haut. Cette broche signale au contrôleur de DMA (Direct Memory Access) qu'un accès à la mémoire va se produire.

 $\overline{DACK}$

Entrée active à l'état bas. Cette broche signale au contrôleur de disque que le contrôleur de DMA a commencé le transfert des données.

TC :

Entrée active à l'état haut. Permet d'interrompre le transfert de données entre le Z80 et le contrôleur de disque. Cette broche est surtout utilisée en mode DMA.

INDEX :

Entrée active à l'état haut. Cette broche est reliée au circuit qui détecte le début d'une piste dans le lecteur de disquettes.

INT :

Sortie active à l'état haut. Cette broche permet de déclencher une interruption du Z80. Sur les AMSTRAD, cette possibilité n'est pas utilisée.

CLK :

Entrée d'horloge. Elle reçoit une fréquence de 4 MHz pour les lecteurs de disquettes de 5" 1/4 ou plus petits et de 8 MHz pour les autres.

GND : Masse.

WCK :

Entrée qui détermine la vitesse de transmission des données entre le lecteur de disquettes et le contrôleur de disque.

RDW :

Entrée active à l'état haut. Permet de séparer les données lues en fonction de leur type.

RDA :

Entrée active à l'état haut. Cette broche permet d'acquérir les données lues sur une disquette.

VCO :

Entrée active à l'état haut. Cette broche permet de synchroniser le VCO par rapport au séparateur de données. Le signal VCO n'est pas utilisé sur les ordinateurs AMSTRAD.

WE :

Sortie active à l'état haut. Permet de valider l'écriture de données sur une disquette.

MFM :

Sortie active à l'état haut. Donne le mode de travail du lecteur de disquettes (simple ou double densité).

(HD) SIDE :

Sortie. Indique si le lecteur piloté est simple ou double face (toujours simple face dans le cas des lecteurs standard AMSTRAD).

US0, US1 :

Permet de sélectionner un lecteur pour une opération de lecture/écriture parmi les quatre qui peuvent être connectés sur le même contrôleur de disque.

WDA :

Entrée active à l'état haut. Cette broche véhicule les données envoyées au lecteur de disquettes.

PS1, PS0 :

Sorties qui indiquent le type de transmission de données pour un lecteur de disquettes à double face (EARLY, NORMAL ou LATE).

FLT/TRKO :

Entrée qui permet d'attendre le passage à 1 du signal TRKO ou la détection d'une erreur par le lecteur.

WPRT/2 SIDE :

Entrée active à l'état haut. Détermine l'état de protection de la disquette contenue dans le lecteur (WRITE PROTECT).

READY :

Entrée active à l'état haut. Indique qu'une disquette se trouve insérée dans le lecteur et que ce dernier tourne à une vitesse qui permet des opérations de lecture/écriture.

HDLD :

Sortie active à l'état haut. Sur les lecteurs de 8'', cette broche signale une action à effectuer sur la tête de lecture.

FLTR/STEP :

Sortie qui permet d'envoyer des impulsions au lecteur de disquettes à chaque déplacement de la tête de lecture, et de remettre à zéro la bascule d'erreur du lecteur de disquette.

LCT/DIR :

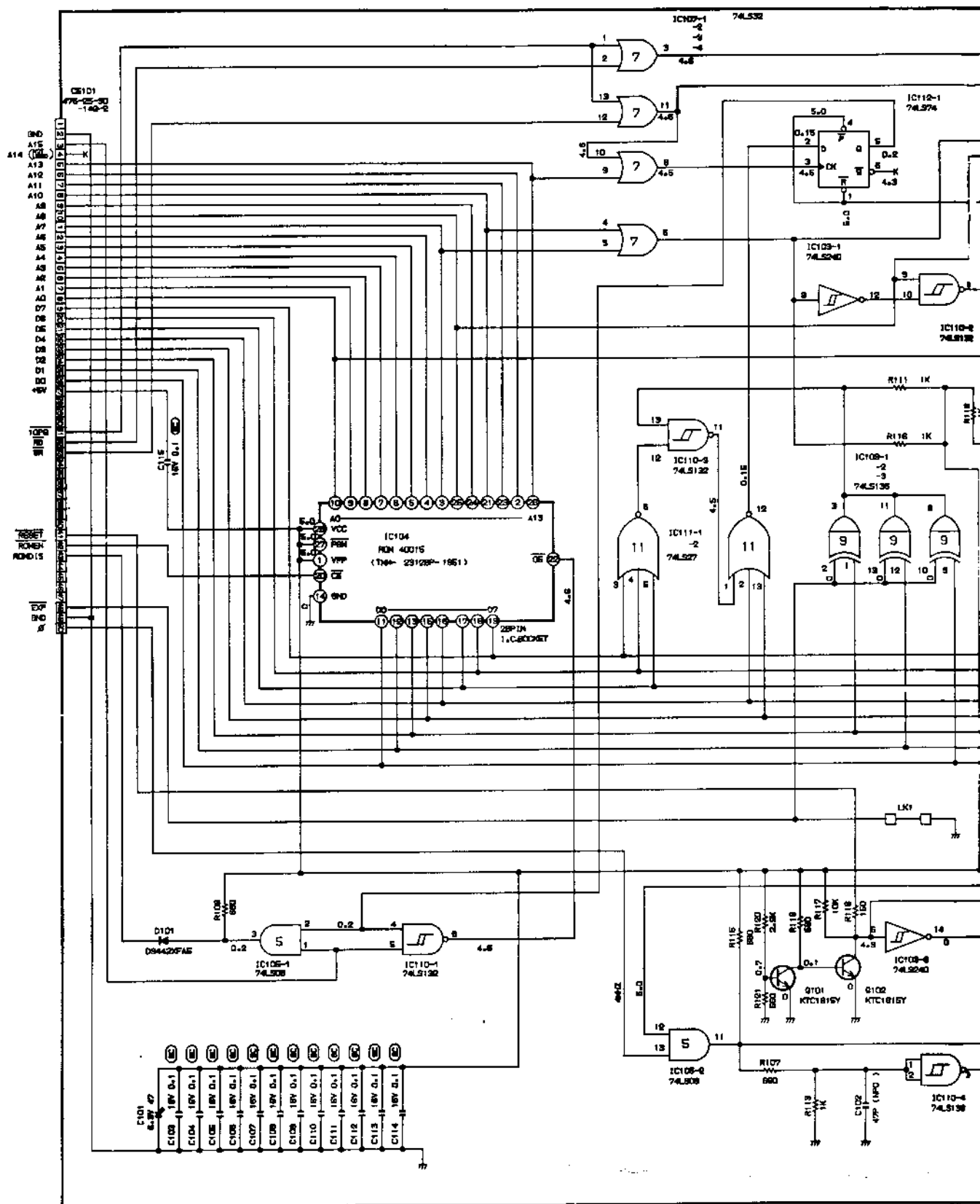
Sortie qui détermine la direction que doit prendre la tête de lecture pour effectuer la prochaine lecture/écriture.

RW/SEEK :

Sortie active à l'état bas. Sélectionne les données nécessaires au contrôleur parmi toutes celles fournies par le lecteur de disquettes.

VCC : Alimentation 5 V + 5 %.

Le schéma de la carte supportant le contrôleur de disque est le suivant :



CONDITION: MAIN SWITCH OF FLOPPY DISK IS SWITCHED ON,
BE READY TO WORK,
AND THE DISKET IS NOT IN DRIVE.

NOTE: THIS SCHEMATIC DIAGRAM IS THE LATEST AT THE TIME OF PRINTING AND SUBJECT TO CHANGE WITHOUT NOTICE.

2/4

Les mémoires

Comme tout micro-ordinateur, les AMSTRAD renferment des *mémoires*, circuits intégrés chargés d'emmagasiner des informations de toutes sortes. Les informations indispensables au fonctionnement de la machine (système d'exploitation, etc.) sont figées en usine dans la *mémoire morte* ou *ROM* (Read Only Memory).

La *mémoire vive* ou *RAM* (Random Access Memory), dont la taille varie selon les modèles, sert de « bloc-notes » aussi bien au microprocesseur qu'au programmeur : on peut à volonté y *écrire* et y *lire* des informations.

L'information élémentaire est l'élément binaire (eb) ou « bit », mais presque toutes les opérations informatiques s'effectuent sur des *octets* (ou *bytes*), ensembles indivisibles de huit bits. L'ensemble de la mémoire de n'importe quel AMSTRAD est donc divisé en octets, physiquement répartis entre les différents boîtiers de RAM et de ROM.

2/4.1

La mémoire morte (ROM)

La mémoire ROM de base des différents modèles d'AMSTRAD est une TMM 23256 (ou un équivalent), qui contient 256 k-bits organisés en 32 k-octets (rappelons que 1 k = 1024).

La figure 1 montre que le boîtier à 28 broches abritant cette mémoire comprend la matrice de $8 \times 32\,768$ « cellules », un « décodeur d'adresse » permettant de choisir l'octet que l'on veut lire, et un « buffer de sortie » ne connectant la mémoire au bus que lorsque cela est nécessaire (uniquement pendant la lecture). Quinze lignes d'adresse sont donc prévues (puisque $2^{15} = 32\,768$) aux côtés des huit lignes de données.

Outre les broches d'alimentation (+5 V et masse ou Vcc et Vss), on trouve encore les broches de commande $\overline{CS}$ et $\overline{OE}$ (Chip Select et Output Enable) : les données lues ne peuvent atteindre le bus que si ces deux broches sont en même temps à l'état bas (0 V). Dans l'AMSTRAD, $\overline{OE}$ est mise à la masse par une résistance, mais un niveau haut (+5 V) peut être « forcé » de l'extérieur par l'intermédiaire de la borne ROMDIS (ROM DISable) du connecteur : une ROM externe peut alors être substituée à celle d'origine, avec toutes les possibilités que cela suppose pour le programmeur averti...

C'est donc finalement le signal $\overline{ROMEN}$ (ROM ENable) généré par le circuit « gate array », qui déclenche le transfert des données sur le bus de la machine. Un synchronisme rigoureux doit évidemment être respecté entre toutes ces opérations afin d'éviter tout « conflit » sur les bus.

Les ROM supplémentaires présentes sur certaines versions d'AMSTRAD fonctionnent exactement de la même façon, bien que leur capacité puisse être différente : une ROM de 16 k-octets, par exemple, sera tout simplement démunie de la ligne d'adresse A14.

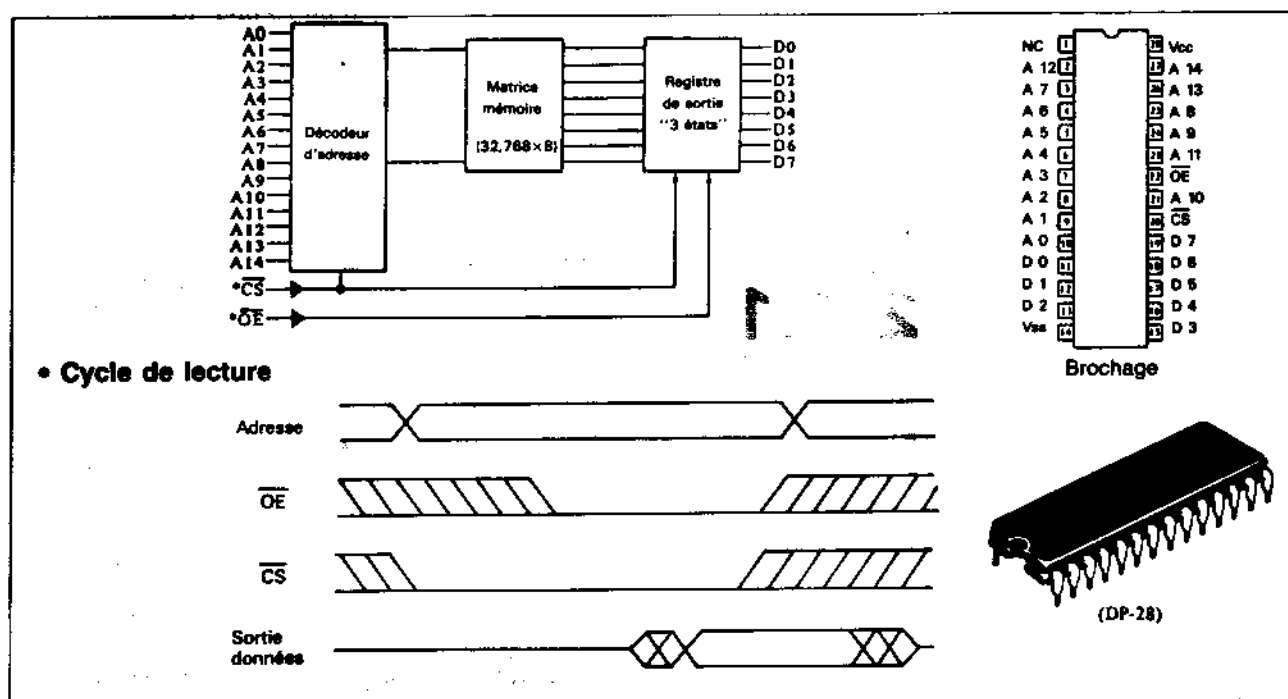


Fig. 1 : ROM TM 23256.

2/4.2

La mémoire vive (RAM)

Les différentes versions d'AMSTRAD utilisent le même composant de base pour leur mémoire RAM : le circuit intégré HM 4864 ou un équivalent. Il s'agit d'une mémoire *dynamique* (DRAM) de 64 k-bits, dont il faut évidemment huit exemplaires pour stocker 64 k-octets. La figure 2 résume ses principales caractéristiques. Pour permettre l'utilisation d'un boîtier à 16 broches seulement, le fabricant n'offre que huit lignes d'adresse au lieu des seize théoriquement nécessaires : l'adresse complète est donc scindée en deux mots de huit bits, l'adresse « colonne » (column) et l'adresse « rangée » (row). Ces deux moitiés sont appliquées l'une à la suite de l'autre sur les lignes d'adresse de la mémoire, ce qui complique notablement les circuits annexes : une batterie de « multiplexeurs » 74 LS 153 est nécessaire, mais le contrôleur d'écran en profite aussi !

Parallèlement, ce type de mémoire « dynamique » est plus lourd à utiliser que des mémoires « statiques » (SRAM) : il est clair que le constructeur a préféré employer des mémoires aussi peu coûteuses que possible, quitte à compliquer la conception des autres parties de la machine.

Les mémoires dynamiques « oublient » en effet très vite les données qui y sont inscrites, ce qui oblige à les « rafraîchir » (refresh) à intervalles réguliers. Fort heureusement, le Z 80 est particulièrement bien adapté à ce type de fonctionnement.

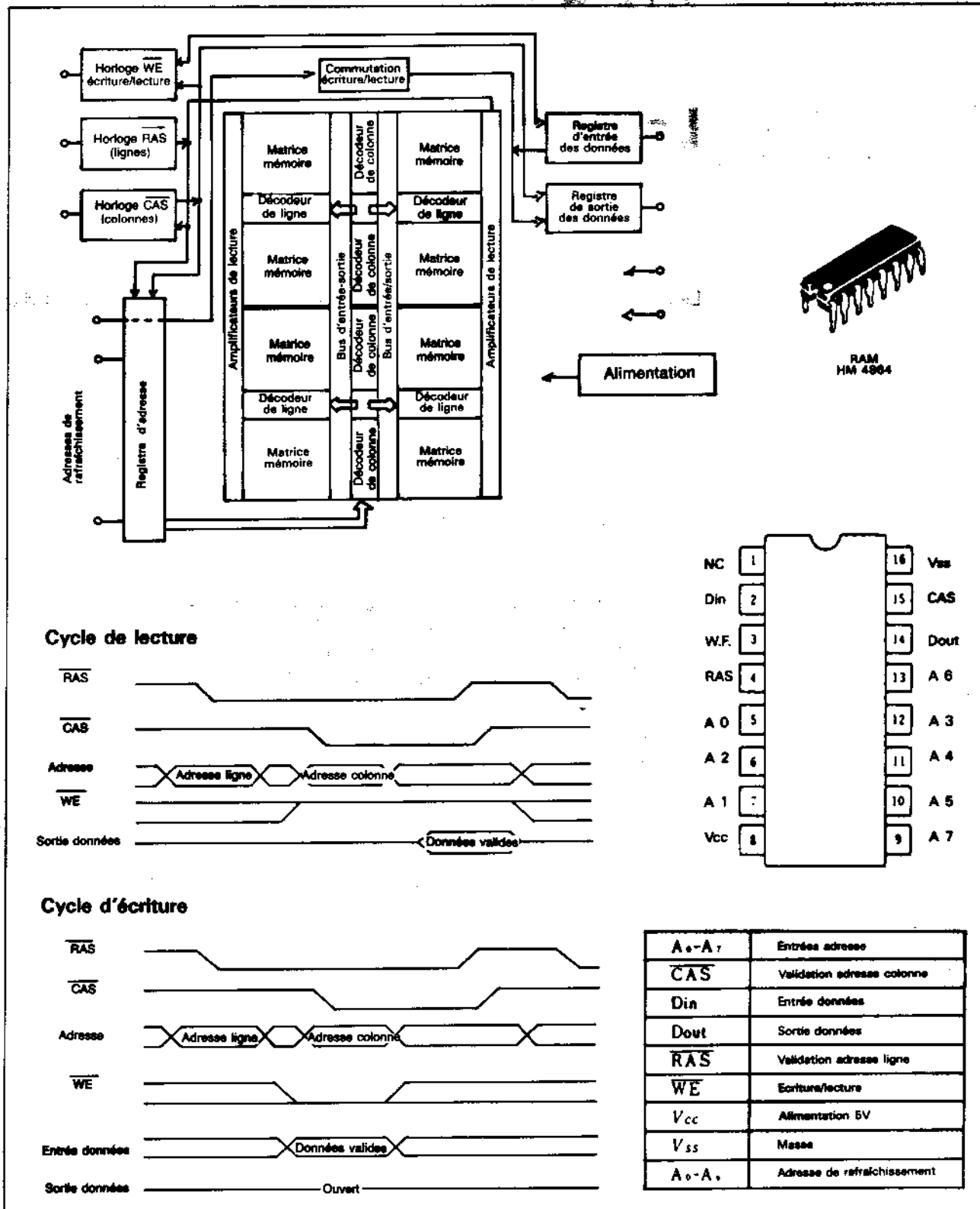


Fig. 2 : RAM HM 4864.

2/5

1.3.5

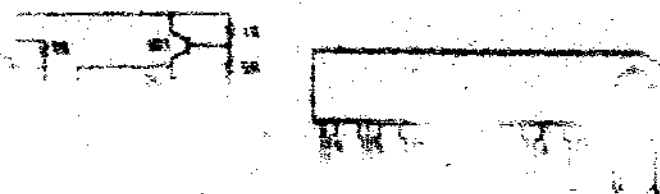
Les circuits intégrés standards

A côté des circuits spécialisés que sont, par exemple, le microprocesseur, les mémoires, et les périphériques, les AMSTRAD contiennent de nombreux circuits intégrés tout à fait classiques qu'il importe de bien connaître pour deux raisons :

- assimilation du fonctionnement profond de la machine avant de tenter de développer des extensions « maison » ;
- dépannage non limité aux défaillances bénignes.

Seul le circuit « gate array »¹⁾, spécialement fabriqué pour AMSTRAD selon ses propres plans, gardera une bonne partie de ses secrets et ne pourra pas être acheté chez votre revendeur habituel...

La plupart des circuits intégrés standards sont des composants *digitaux* (TTL ou CMOS rapide), mais il y a aussi quelques circuits *linéaires*, notamment au niveau du lecteur de cassettes qui intègre aussi l'amplificateur audio.



1) Encore appelé « ULA » (Uncommitted Logic Array).

Les circuits intégrés linéaires

Le LA 4140, dont les caractéristiques apparaissent à la figure 2, est un amplificateur de puissance servant à attaquer le haut-parleur. Il est logé dans un boîtier SIL (Single In Line) ne possédant qu'une seule rangée de broches, mais muni d'un coin coupé évitant toute confusion d'orientation.

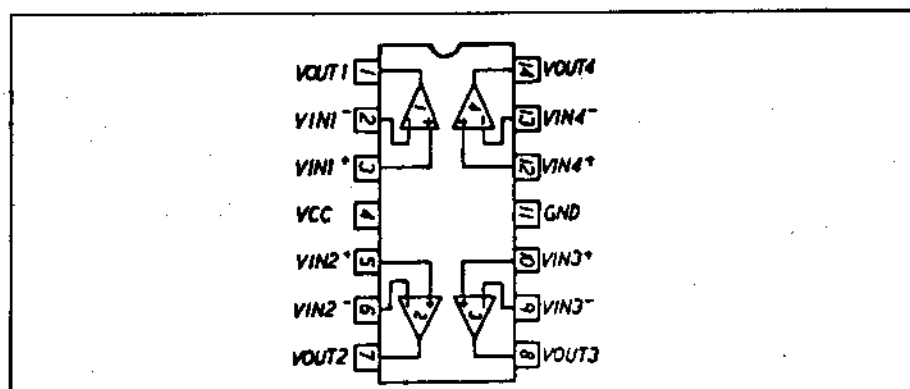


Fig. 1 : LA 6324.

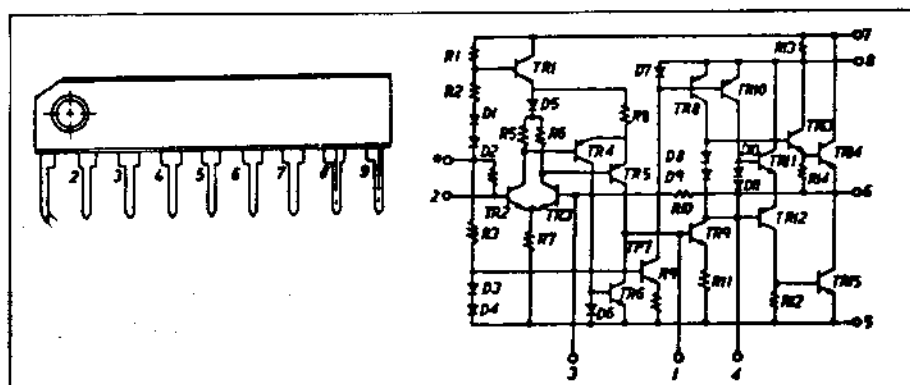


Fig. 2 : LA 4110.

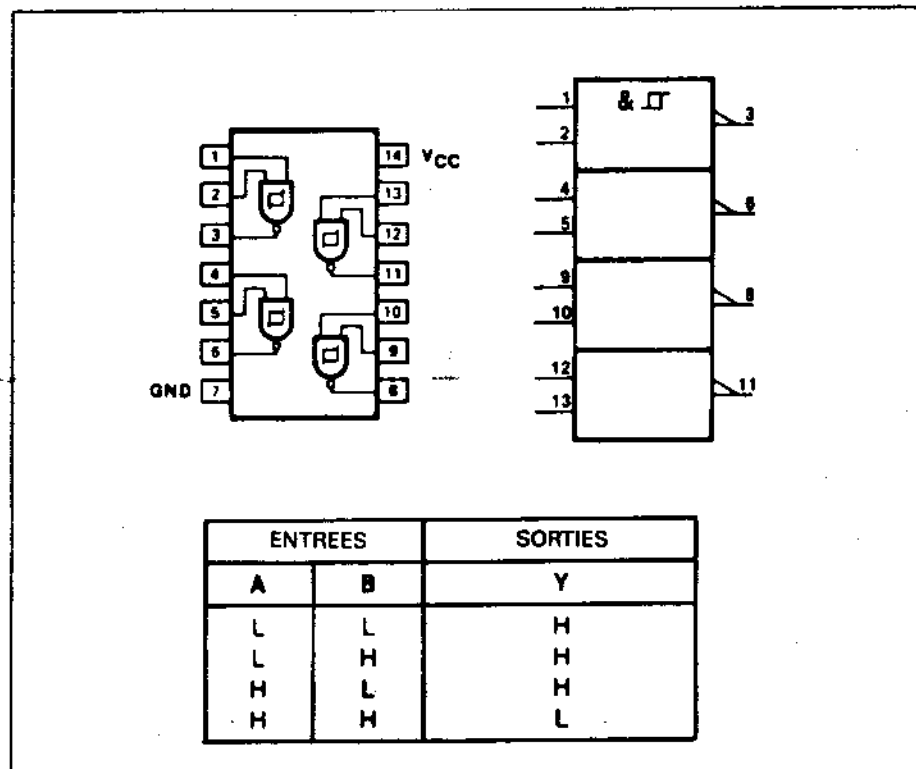
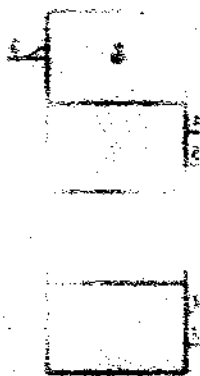


Fig. 2 : LS 132.



A la figure 3, le 74LS08 réalise la fonction ET (AND), complémentaire de la précédente : la sortie de chaque porte à deux entrées ne passe au niveau haut que si les deux entrées sont elles-mêmes hautes (à + 5 V).

Le 74LS32 de la figure 4 est une quadruple porte OU (OR) à deux entrées : la sortie de chaque porte passe à un niveau haut dès qu'au moins une des entrées est haute : on parle d'une fonction « OU inclusif » par opposition au 74LS136 de la figure 5 qui contient quatre portes « OU exclusif » à deux entrées : la sortie est haute si une entrée est basse et l'autre haute. Elle est basse si les deux entrées sont au même niveau, quel qu'il soit.

A la figure 6, nous sommes en présence d'une triple porte NOR (fonction complémentaire du OU) à 3 entrées 74LS27 : la sortie n'est haute que si les trois entrées correspondantes sont basses en même temps.

Le 74HCU04 de la figure 7 est un *sextuple inverseur* : chaque élément inverse le niveau appliqué à son entrée.

Contrairement à tous les circuits précédents dont la sortie délivrait du + 5 V à l'état haut, le 74LS38 de la figure 8 contient quatre portes NAND à deux entrées avec sorties *collecteur ouvert* : lorsque les deux entrées sont hautes, la sortie correspondante est reliée à la masse par un transistor NPN saturé. Dans les autres cas, le transistor est bloqué et la sortie reste dans un état « haute impédance » : son niveau électrique est fixé par le circuit extérieur.

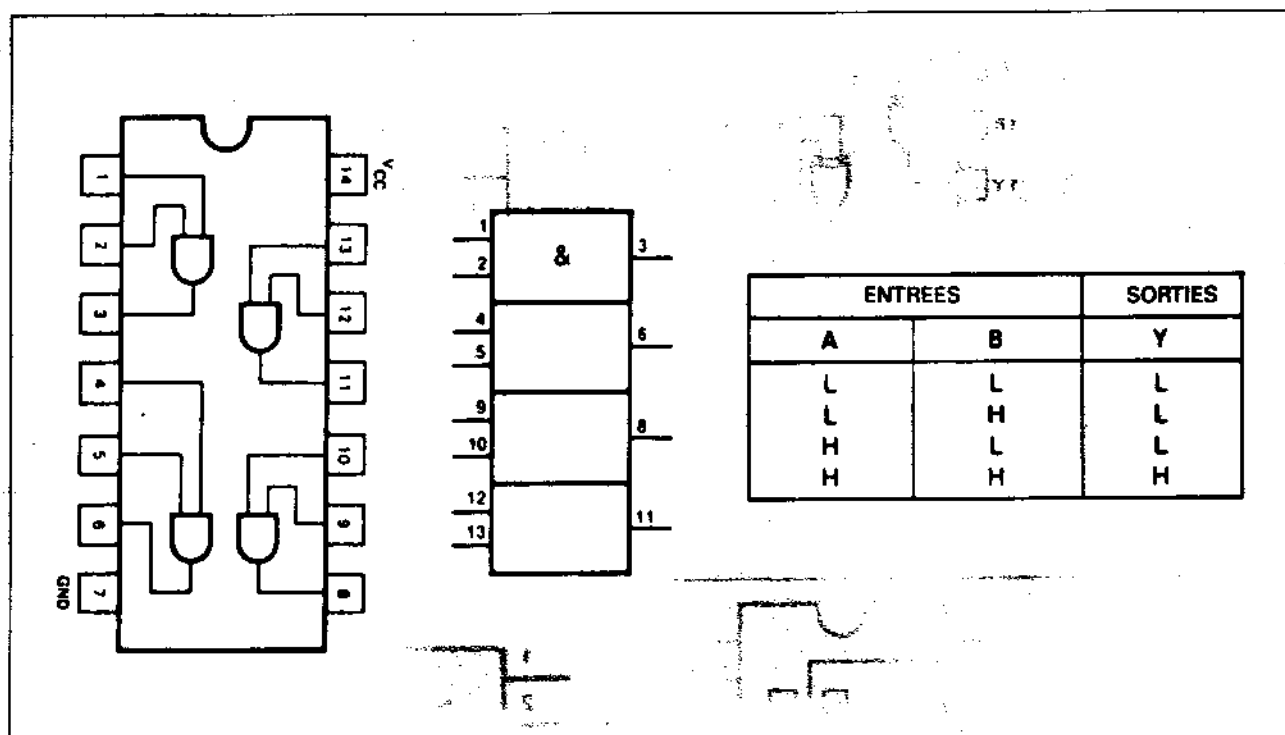


Fig. 3 : 74 LS 08.

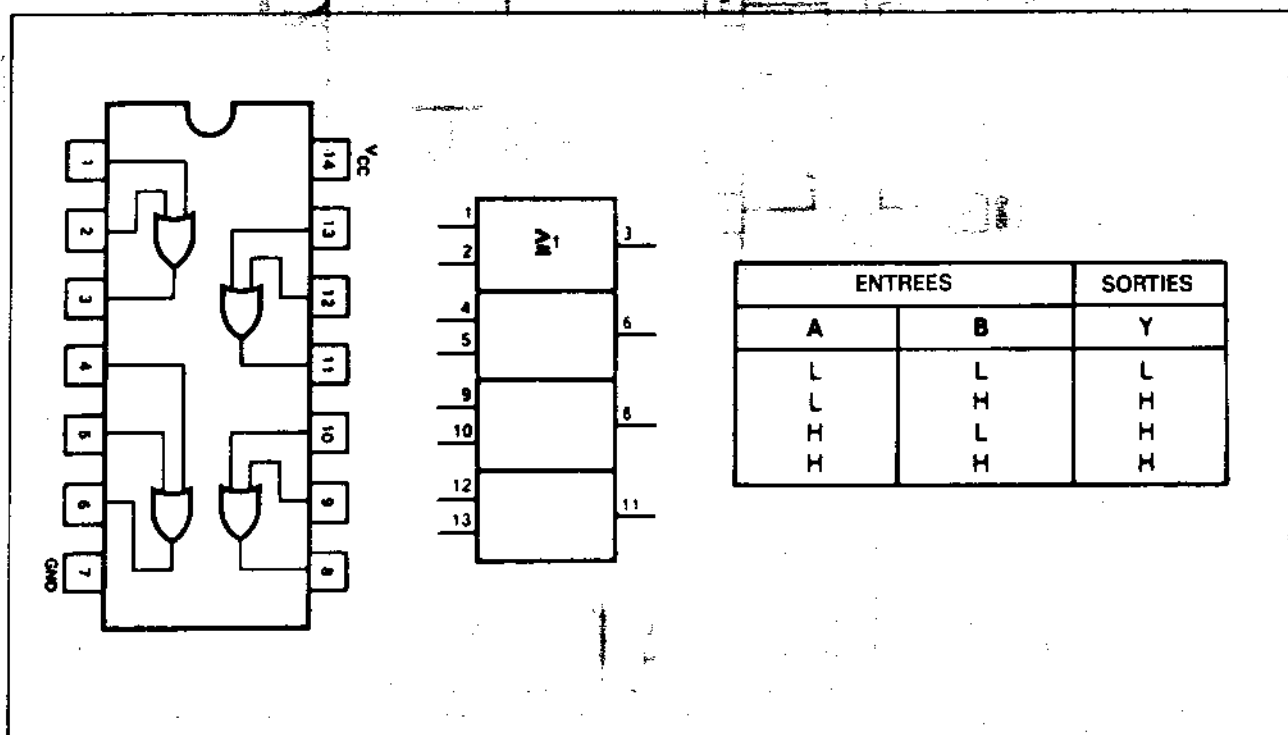


Fig. 4 : 74 LS 32.

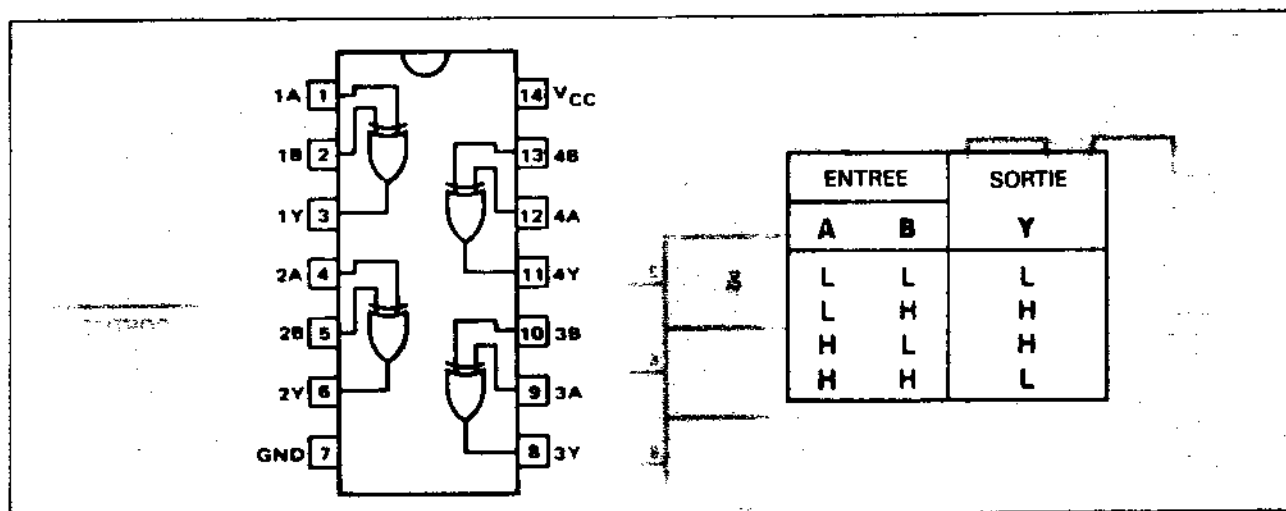


Fig. 5 : 74 LS 136.

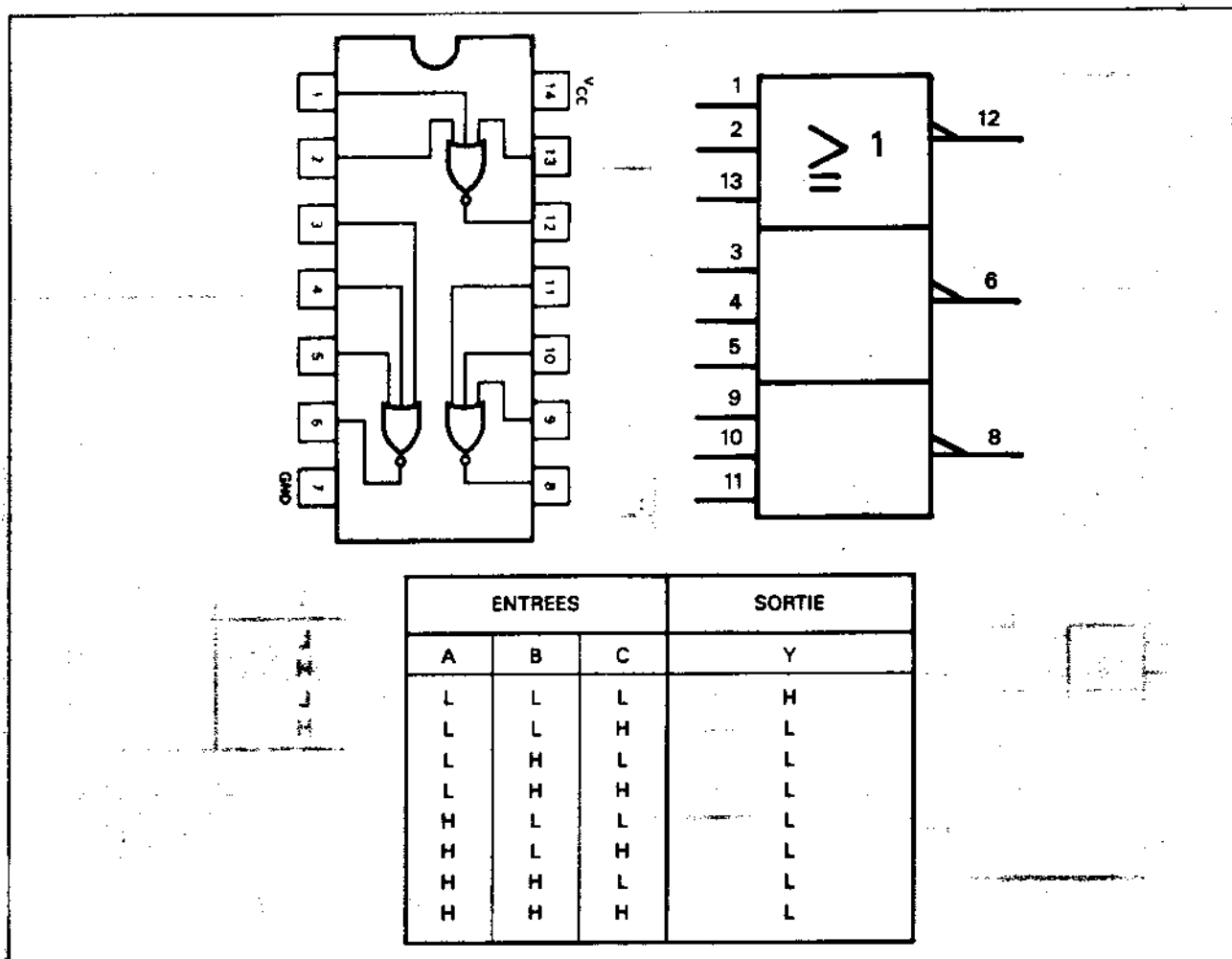


Fig. 6 : 74 LS 27.

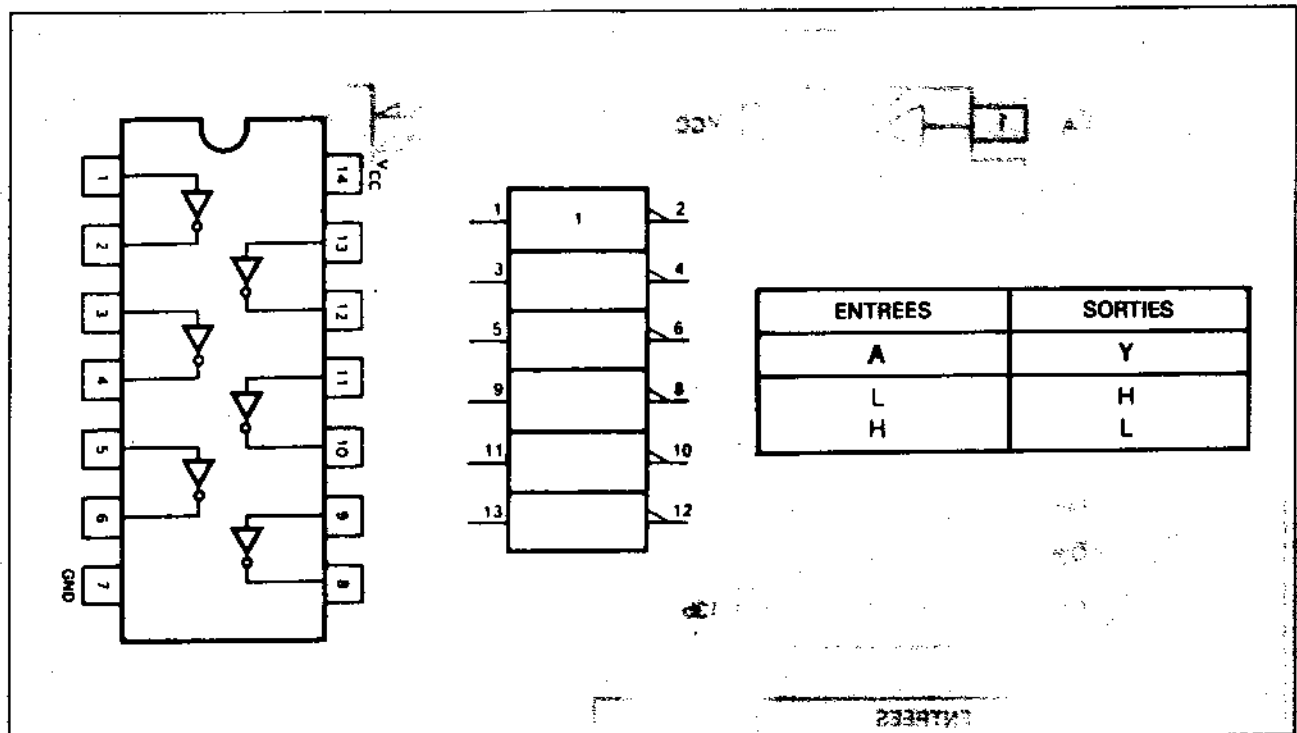


Fig. 7 : 74 HCU 04.

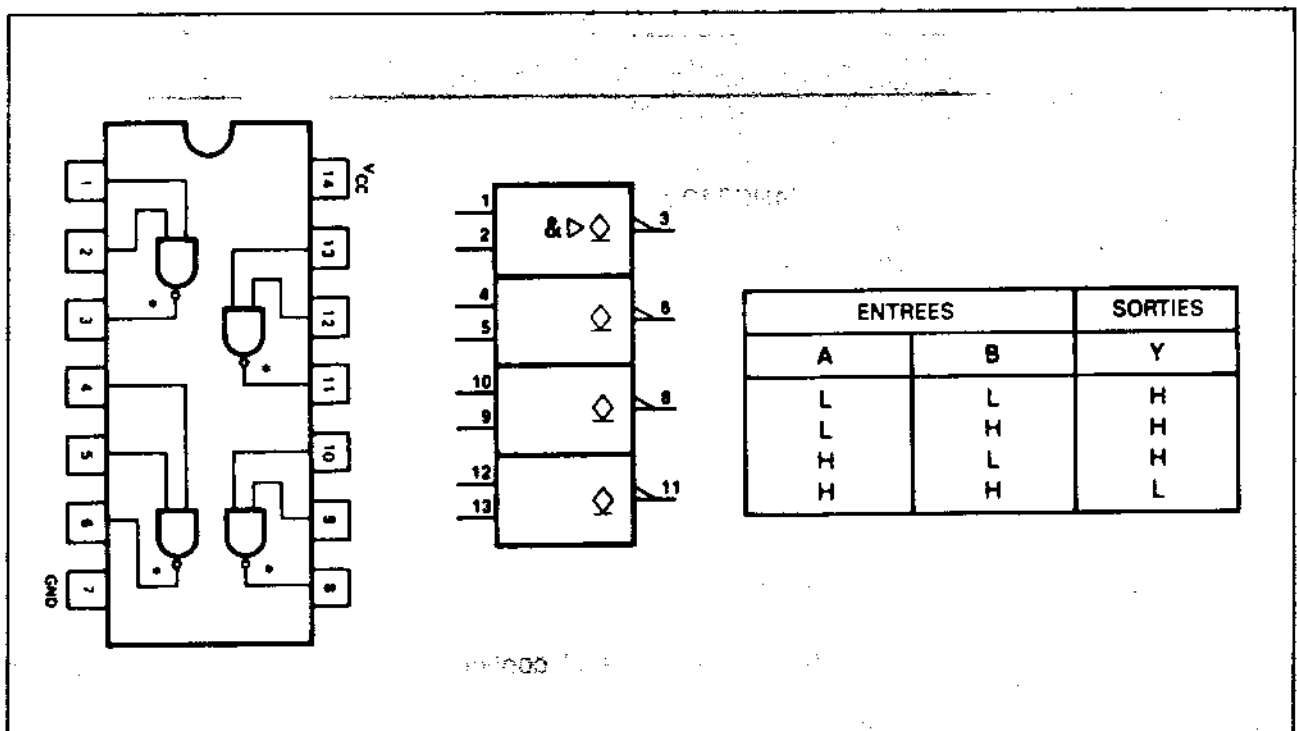


Fig. 8 : 74 LS 38.

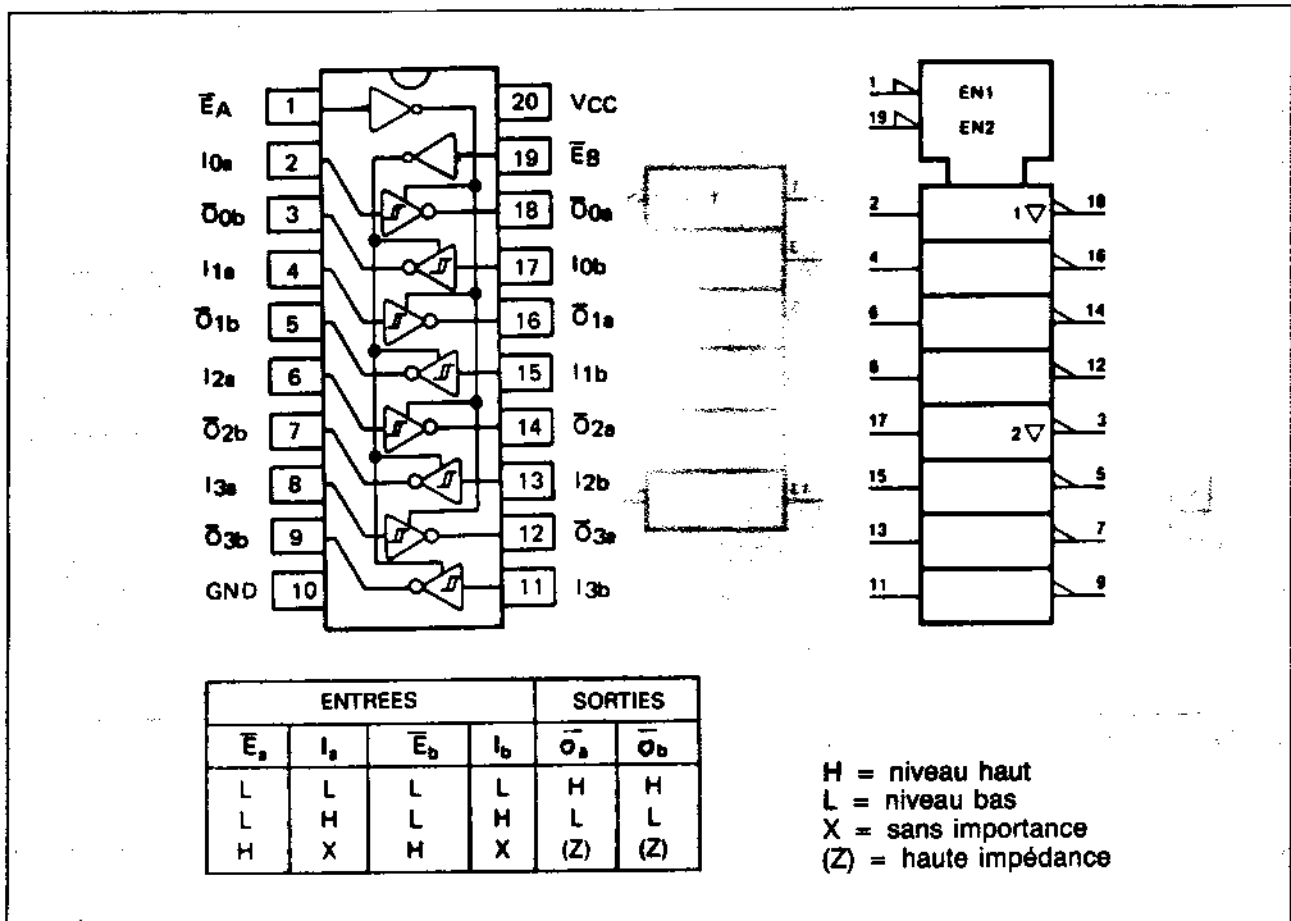


Fig. 9 : 74 HC 240.

Avec le 74HC240 de la figure 9, nous abordons le domaine des circuits intégrés de moyenne complexité : les huit sections de ce composant sont en effet des *buffers trois états* inverseurs. Lorsque l'entrée $\bar{E}$ correspondant à un buffer donné est dans un état haut, la sortie du buffer est en « haute impédance » (Z). L'état de l'entrée de « donnée » I est alors sans importance (X). Lorsque cette entrée $\bar{E}$ passe au niveau bas, les données appliquées à l'entrée I d'un buffer se retrouvent sur la sortie O correspondante, mais *inversées*.

Le 74LS244 de la figure 10 fonctionne de la même façon, mais n'inverse pas les données entre l'entrée et la sortie.

Ces deux composants sont séparés en deux groupes A et B de quatre buffers chacun, mais on utilise généralement les huit buffers ensemble, en réunissant $\bar{E}_a$ et $\bar{E}_b$.

Le 74LS373 de la figure 11 contient des *bascules D* en amont des huit buffers qui cette fois sont indissociables au niveau de la commande $\bar{OE}$. Cette disposition permet de « saisir au vol » un groupe de huit états sur les entrées, puis de le recopier plus tard sur les sorties, sans inversion cette fois encore. Un état bas sur l'entrée de commande E provoque la

mise en mémoire des états des huit entrées, tandis qu'un état bas sur $\overline{OE}$ déclenche le transfert des états mémorisés vers les sorties. Bien entendu, les huit sorties restent en « haute impédance » tant que $\overline{OE}$ est à l'état haut.

Ce type de composant sert souvent à réaliser des *ports de sortie*, et les simples buffers des *ports d'entrée*.

Le 74LS273 de la figure 12 n'est pas équipé de buffers : il ne contient que huit bascules qui peuvent mémoriser l'état des entrées lorsqu'une impulsion positive est appliquée à l'entrée d'horloge CP. Toutes les bascules peuvent être remises à zéro simultanément par un niveau bas sur l'entrée $\overline{MR}$, ou « Master Reset ».

Les entrées du 74LS273 sont évidemment compatibles avec un bus « trois états » comme le bus de données du Z80, mais ses sorties ne le sont pas directement. On peut cependant les utiliser pour créer un *port de sortie*, par exemple pour l'imprimante.

Logé dans un boîtier à 14 broches seulement, le 74LS74 de la figure 13 ne contient que deux bascules, mais équipées d'entrées individuelles de remise à zéro et à un, et aussi de sorties directes et complémentées.

A ceci près, le fonctionnement est semblable : le niveau présent sur l'entrée D (Donnée) est transféré et mémorisé sur les sorties pendant le front montant de l'impulsion appliquée à l'entrée C (Clock).

Assemblage complexe de bascules, le 74HC161 est un *compteur binaire* à quatre étages, pouvant donc compter de zéro à quinze. Différentes entrées de commande permettent toute une variété de fonctionnements spéciaux : prépositionnement à un état quelconque grâce aux entrées « D », remise à zéro, mise en attente, etc. La figure 14 résume les principales caractéristiques de ce composant.

Décodeur BCD-décimal, le 74LS145 de la figure 15 fait passer à l'état bas celle de ses dix sorties qui correspond à la valeur décimale du mot binaire de quatre bits appliquée à ses entrées. Les combinaisons supérieures à 9 sont ignorées.

Le 74LS153 de la figure 16 est pour sa part un *multiplexeur* capable de réaliser deux fois l'aiguillage vers une sortie unique d'une entrée choisie parmi quatre possibles, si $\overline{E}$ est à 0.

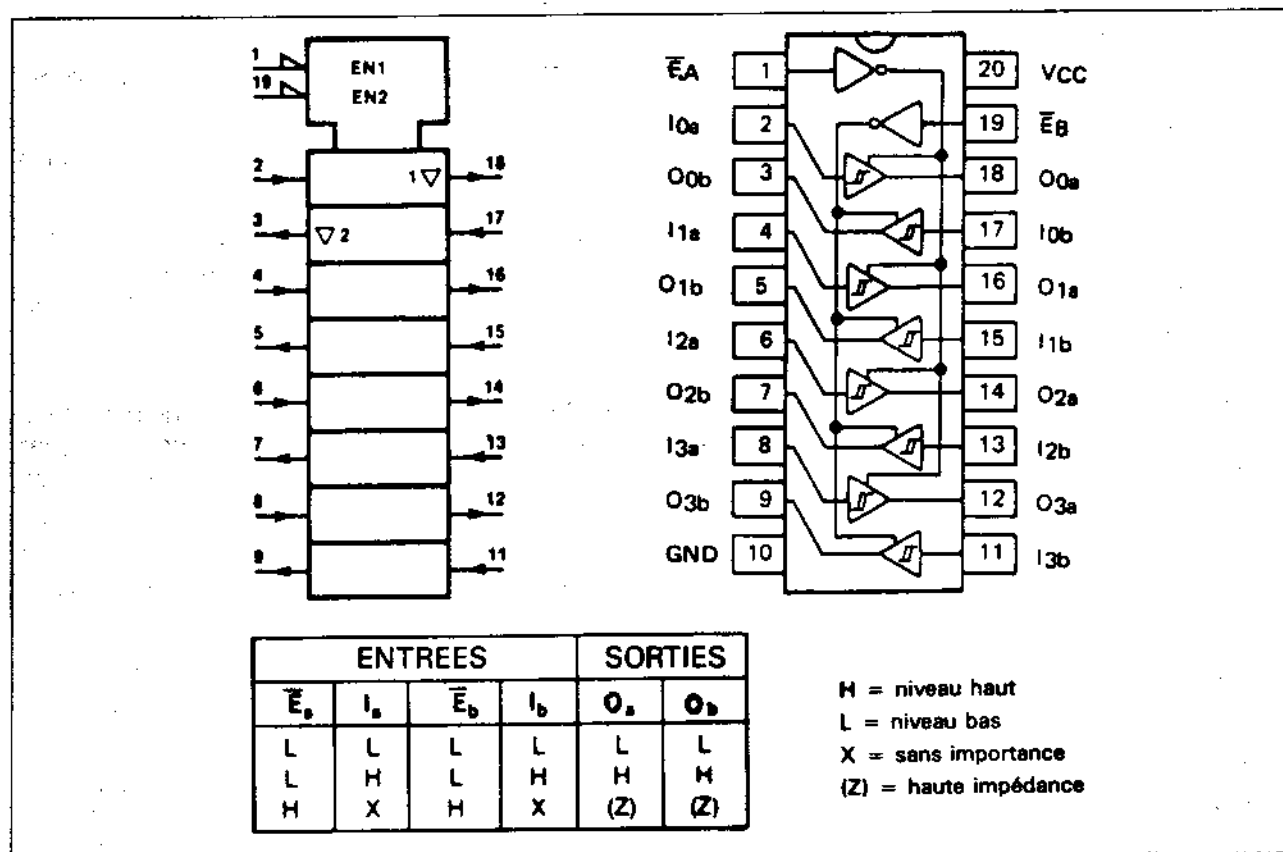


Fig. 10 : LS 244.

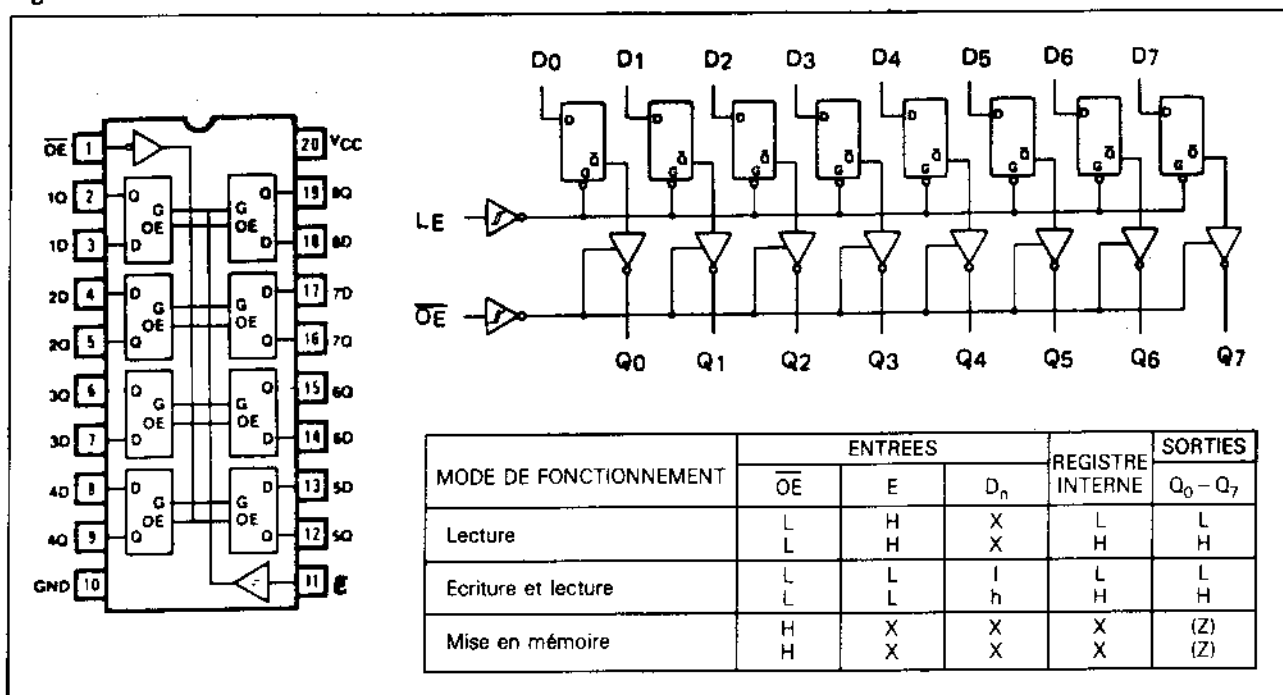


Fig. 11 : LS 373.

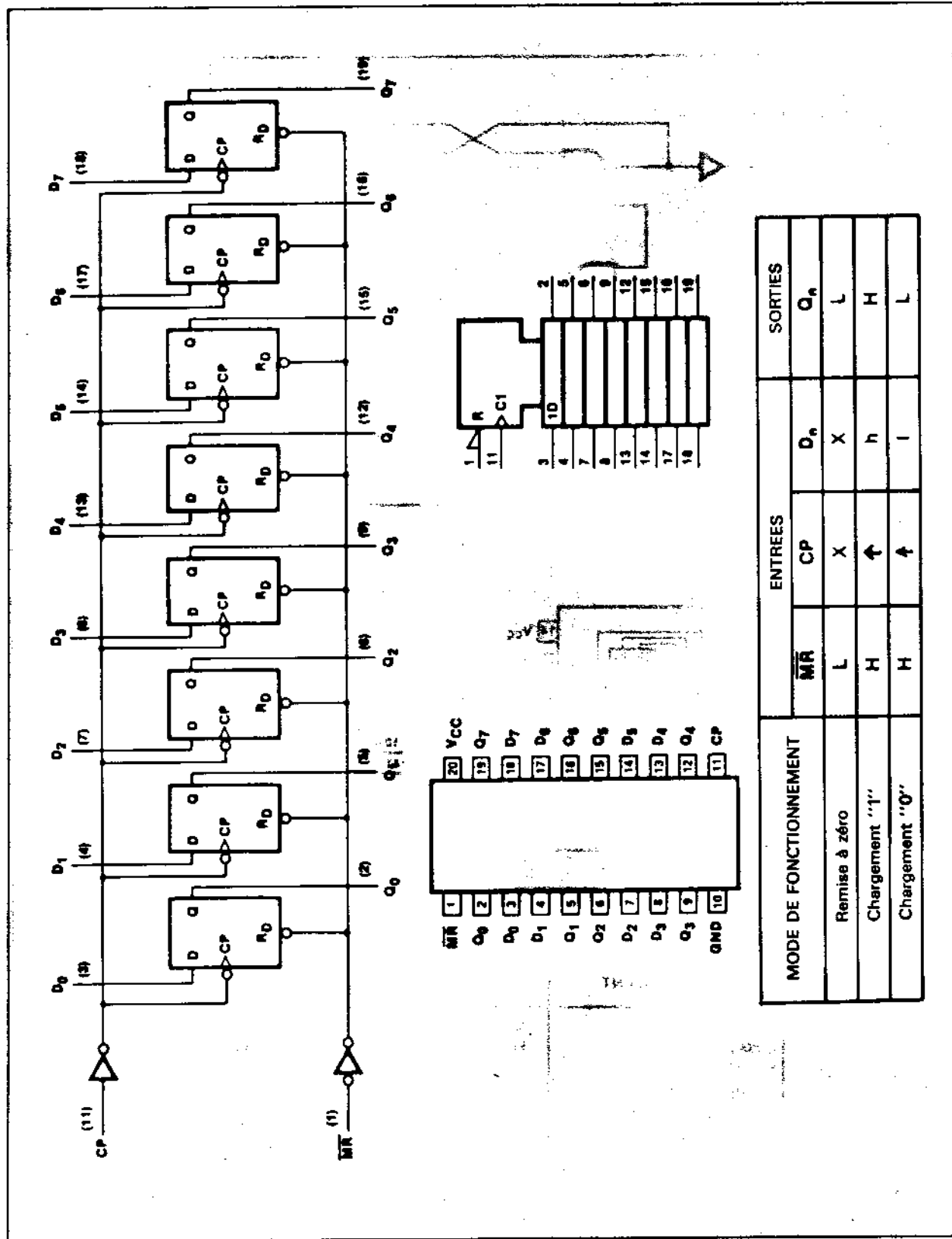
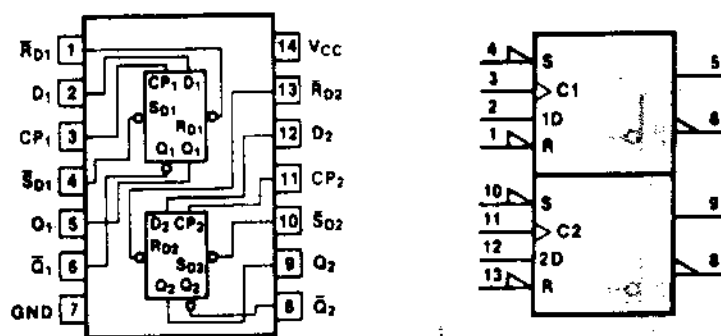
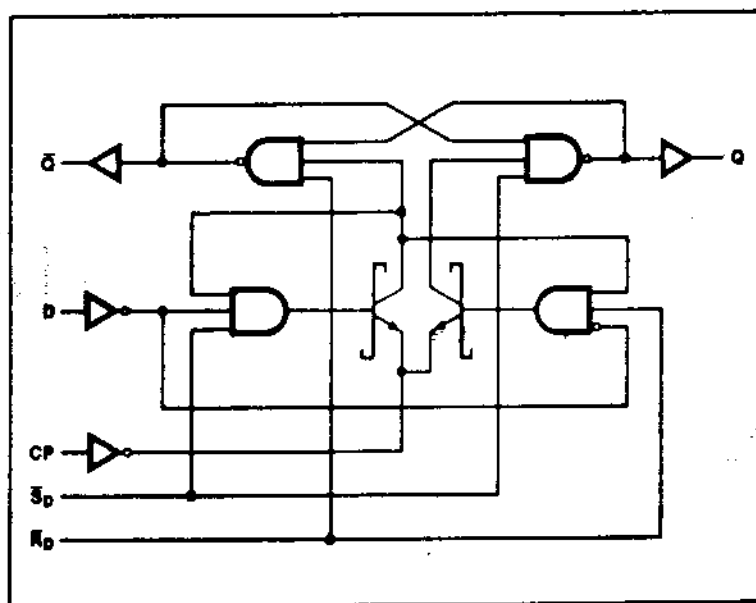
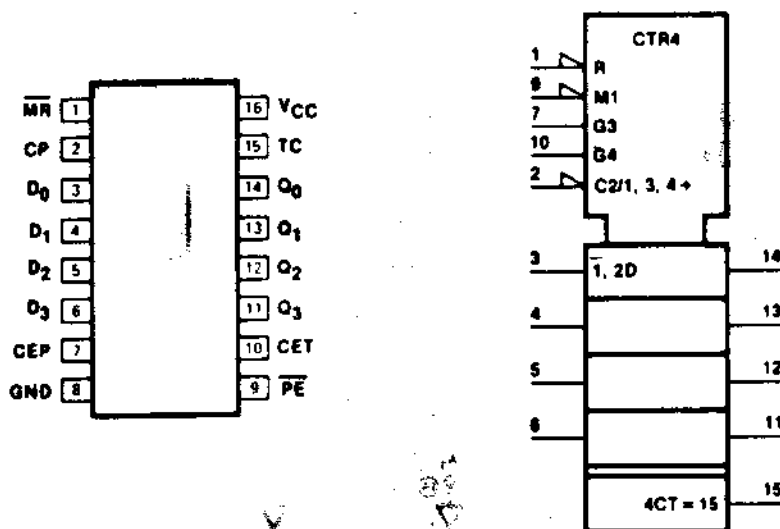


Fig. 12 : LS 273.



MODE DE FONCTIONNEMENT	ENTREES				SORTIES	
	$\bar{S}_D$	$\bar{R}_D$	CP	D	Q	$\bar{Q}$
Forçage à 1	L	H	X	X	H	L
Forçage à 0 (Clear)	H	L	X	X	L	H
Indéterminé	L	L	X	X	H	H
Chargement "1"	H	H	↑	h	H	L
Chargement "0"	H	H	↑	l	L	H

Fig. 13 : 74 LS 74.



MODE DE FONCTIONNEMENT	ENTREES						SORTIES	
	$\overline{MR}$	CP	CEP	CET	PE	D_n	Q_n	TC
Remise à zéro	L	X	X	X	X	X	L	L
Chargement	H	L	X	X	L	L	L	L
Comptage	H	L	h	h	h ^(c)	X	count	(a)
Attente	H	X	L ^(b)	X	h ^(c)	X	q_n	(a)
	H	X	X	L ^(b)	h ^(c)	X	q_n	L

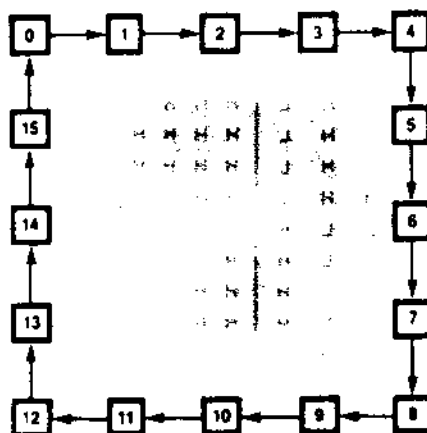


Fig. 14 : 74 HC 161.

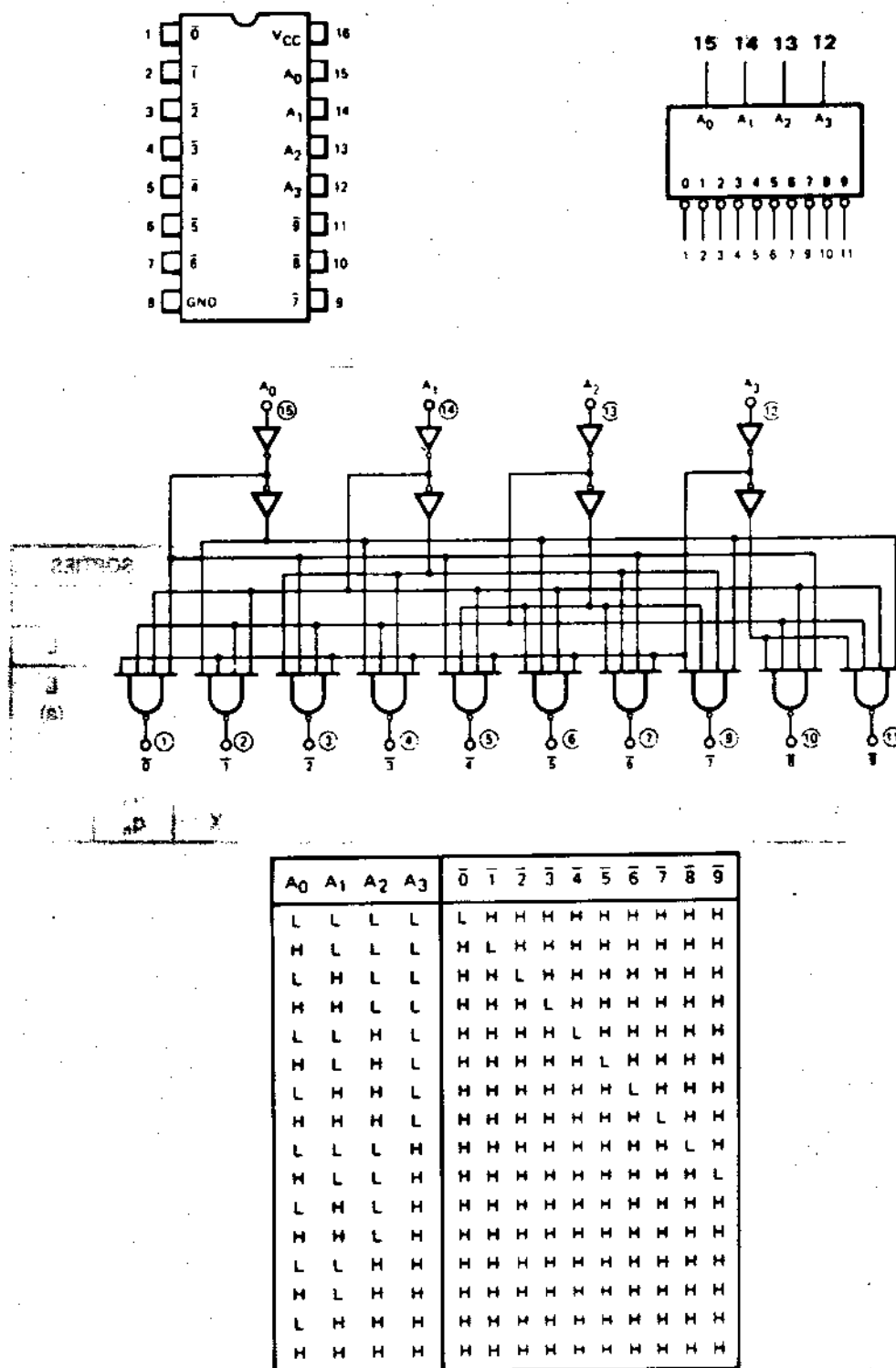


Fig. 15 : LS 145.

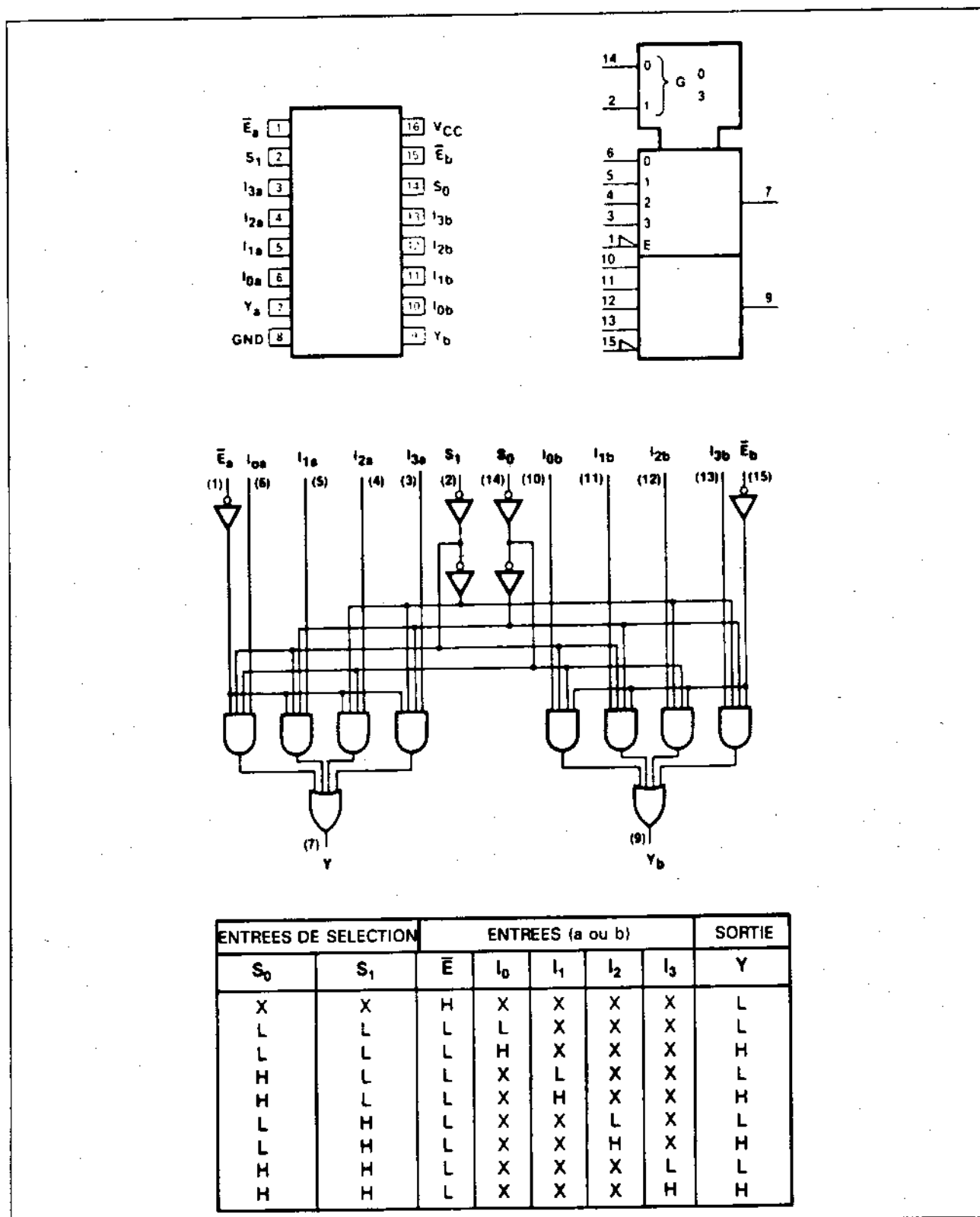


Fig. 16 : LS 153.

2/6

L'horloge interne

Les AMSTRAD CPC 464, 664 et 6128 sont bâtis autour d'un microprocesseur Z 80 A « tournant » à 4 MHz, ce qui est relativement rapide : beaucoup d'ordinateurs personnels équipés d'autres microprocesseurs se contentent de 1 MHz.

La famille Z 80 comprend cependant des composants plus rapides (Z 80 B : 6MHz) et plus lents (Z 80 CPU : 2,5 MHz), ce qui fait que le choix effectué se situe dans une très bonne moyenne.

Le Z 80 A doit donc recevoir sur sa broche 6 (CLK ou clock) un signal rectangulaire de 4 MHz, c'est-à-dire quatre millions d'impulsions ou *périodes d'horloge* par seconde.

Cette fréquence est dérivée d'un signal à 16 MHz produit par un oscillateur à quartz équipé de deux inverseurs appartenant soit à un 74LS00 (TTL Schottky), soit à un 74HCU04 (CMOS rapide).

La division par quatre est opérée dans le circuit « Gate Array » ou « ULA », qui en profite pour élaborer d'autres fréquences à partir de cette référence très stable.

En particulier, le signal 1 MHz $\overline{\text{CPU}}$ est utilisé par le synthétiseur de son AY-3-8912 et par le contrôleur vidéo 6845.

La fréquence de tous les sons pouvant être émis par l'AMSTRAD est donc obligatoirement un *sous-multiple de 1 MHz*, puisque le synthétiseur fonctionne par division de fréquence. En même temps, la précision obtenue sur la sortie « audio » sera celle du quartz, c'est-à-dire extrêmement bonne.

Par contre, il ne sera pas possible de synthétiser n'importe quelle fréquence : les notes de musique seront « arrondies » au plus proche sous-multiple de 1 MHz, ce qui peut être perceptible pour une oreille très « juste ».

Il ne pourra pas non plus être question de produire, par exemple, les paires de fréquences utilisées pour composer des numéros de téléphone en code « multifréquences » car la « résolution » offerte est très inférieure à la tolérance acceptée par les centraux.

Retenons donc que la *stabilité* des fréquences générées est excellente, mais que l'*erreur* par rapport à la fréquence désirée peut être relativement importante, particulièrement aux fréquences les plus hautes qui correspondent à de plus petits rapports de division.

Pour produire des fréquences inaccessibles au générateur de son, on peut toutefois songer à piloter un port de sortie (par exemple le STROBE de l'imprimante) par une routine en langage machine soigneusement « accordée ».

Ce procédé, qui peut être employé pour toutes sortes d'autres « chronométrages », exploite le fait que la fréquence d'horloge est le véritable « chef d'orchestre » du microprocesseur : chaque instruction exécutée, chaque opération effectuée, dure un nombre entier de périodes d'horloge (0,25 microseconde pour 4 MHz). Pour étalonner une routine (ou une partie de routine), il suffit d'additionner le nombre de périodes d'horloge correspondant à chaque instruction, en tenant compte des différents « chemins » possibles (par exemple selon le résultat de comparaisons).

Comme il n'est généralement pas possible d'accélérer une routine (sauf si elle a été maladroitement écrite), le « réglage » se fait par ralentissement : pour perdre « beaucoup » de temps (des millisecondes ou des secondes), on fait « tourner en rond » le programme dans des boucles utilisant généralement l'instruction DJNZ. L'ajustement fin se fait ensuite par introduction d'instructions NOP, qui ne font rien d'autre que « perdre » quatre périodes d'horloge soit une microseconde (un millionième de seconde).

On peut donc dire que l'unité de temps dans laquelle compte le microprocesseur est la microseconde.

Pour calculer la durée d'exécution d'une routine, la connaissance du nombre de cycles machine (voir annexe 2 de la partie 11) ne suffit pas : certains cycles comportent plus de périodes d'horloge que d'autres.

Le tableau 1 donne donc la durée d'exécution des principales instructions, exprimée en périodes d'horloge de 0,25 μ S.

En théorie, on pourrait envisager d'altérer la durée d'exécution des instructions et la fréquence des sons émis par le synthétiseur, en modifiant légèrement la fréquence du quartz de l'oscillateur d'horloge.

En pratique, ce genre d'intervention risque de fausser la fréquence des signaux vidéo produits par le contrôleur d'écran, qui travaille lui aussi par division de la fréquence auxiliaire de 1 MHz : l'affichage écran pourrait alors se trouver sérieusement perturbé.

Cette solution n'est toutefois pas à écarter à 100 % lorsqu'il s'agit de résoudre des problèmes très spéciaux, pour lesquels le moniteur n'est pas forcément indispensable.

On se souviendra cependant que les routines de gestion des différents autres périphériques (lecteur de cassettes ou de disquettes, imprimante, etc.) sont évidemment cadencées par la même horloge, et ne supportent guère d'importants écarts.

Tableau 1

LD r, r'	LD r, n	LD r, (HL)	LD r, (IX + d)	LD r, (IY + d)	LD (HL), r	LD (IX + d), r	LD (IY + d), r	LD (HL), n	LD (IX + d), n	LD (IY + d), n	LD A, (BC)	LD A, (DE)	LD A, (nn)	LD (BC), A	LD (DE), A	LD (nn), A	LD A, I	LD A, R	LD I, A	LD R, A	LD dd, nn	LD IX, nn	LD IY, nn	LD HL, (nn)	LD dd, (nn)	LD IX, (nn)	LD IY, (nn)	LD (nn), HL	LD (nn), dd	LD (nn), IX	LD (nn) IY				
4	7	7	19	19	7	19	19	10	19	19	7	7	13	7	7	13	9	9	9	9	10	14	14	16	20	20	20	16	20	20	20				
LD SP, HL	LD SP, IX	LD SP, IY	PUSH qq	PUSH IX	PUSH IY	POP qq	POP IX	POP IY	EX DE, HL	EX AF, AF'	EXX	EX (SP), HL	EX (SP), IX	EX (SP), IY	LDI	LDIR	LDD	LDDR	CPI	CPH	CPI	CPH	CPD	CPDR	RLCA	RLA	RRCA	RRD	RRCA	RRD					
6	10	10	11	15	15	10	14	14	4	4	4	19	23	23	16	21	16	21	16	18	21	16	18	21	21	16	16	21	18	18					
ADD A, r	ADD A, n	ADD A, (HL)	ADD A, (IX + d)	ADD A, (IY + d)	INC r	INC (HL)	INC (IX + d)	INC (IY + d)	DAA	CPL	NEG	CCF	SCF	NOP	HALT	DI	EI	IM 0	IM 1	IM 2	ADD HL, ss	ADC HL, ss	SBC HL, ss	ADD IX, pp	ADD IY, rr	INC ss	INC IX	INC IY	DEC ss	DEC IX	DEC IY	RLCA	RLA	RRCA	RRD
4	7	7	19	19	4	11	23	23	4	4	4	4	4	4	4	4	8	8	8	8	11	15	15	15	15	6	10	10	6	10	10	4	4	4	
RLC r	RLC (HL)	RLC (IX + d)	RLC (IY + d)	RLD	RRD	OTDR	BIT b, r	BIT b, (HL)	BIT b, (IX + d)	BIT b, (IY + d)	SET b, r	SET b, (HL)	SET b, (IX + d)	SET b, (IY + d)	JP nn	JP cc, nn	JRe	JR C, e	JR NC, e	JP Z, e	JR NZ, e	JP (HL)	JP (IX)												
8	15	23	23	23	18	18	21	16	8	12	20	20	20	8	15	23	23	10	10	12	7	12	7	12	7	12	7	12	7	12	8				
ADC A, r	ADC A, n	ADC A, (HL)	ADC A, (IX + d)	ADC A, (IY + d)	SBC A, r	SBC A, n	SBC A, (HL)	SBC A, (IX + d)	SBC A, (IY + d)	AND r	AND n	AND (HL)	AND (IX + d)	AND (IY + d)	XOR r	XOR n	XOR (HL)	XOR (IX + d)	XOR (IY + d)	CP r	CP n	CP (HL)	CP (IX + d)	CP (IY + d)	DEC r	DEC (HL)	DEC (IX + d)	DEC (IY + d)	RL r	RL (HL)	RL (IX + d)	RL (IY + d)			
4	7	7	19	19	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7			
RR r	RR (HL)	RR (IX + d)	RR (IY + d)	SRA r	SRA (HL)	SRA (IX + d)	SRL r	SRL (HL)	SRL (IX + d)	SRL (IY + d)	RES r	RES (HL)	RES (IX + d)	RES (IY + d)	INI	INIR	IND	INDR	OUT (n), A	OUT (C), r	OUTI	OTIR	OUTD												
8	15	23	23	23	8	15	23	23	8	15	23	23	8	15	23	23	10	10	12	7	12	7	12	7	12	7	12	7	12	7	12	8			

Tableau 1 : Durée d'exécution (à 4 MHz) des principales instructions du Z 80, exprimées en périodes d'horloge de 0,25 μ s. Lorsque deux durées sont indiquées, elles correspondent aux deux fonctionnements possibles de l'instruction : par exemple, RET cc prend 2,75 μ s si la condition cc est fautive, et seulement 1,25 μ s si cc est vraie. INIR, pour sa part, dure 5,25 μ s si B $\neq$ 0, et 4 μ s si B = 0.

Partie 3

Systèmes d'exploitation des 664 et 6128

3/0

Table des matières

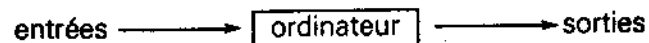
3/1	Introduction au DOS
3/2	AMSDOS : Définitions, rappels et utilisations I. Gestion des unités de disquettes II. Gestion de l'unité de cassettes III. Utilitaires
3/2.1	Liste alphabétique des mots clés
3/3	CP/M 2.2 : Définitions, rappels et utilisations I. Entrées/Sorties sur disque ou cassette II. Utilitaires III. Ordres évolués
3/4	CP/M plus : Définitions, rappels et utilisations I. Entrées/Sorties sur disque II. Entrées/Sorties sur périphériques III. Utilitaires IV. Ordres évolués
3/4.1	Liste alphabétique des mots clés
3/4.2	Programmation sous CP/M plus
3/4.2.1	Organisation mémoire du 6128
3/4.2.1.1	FCB (<i>File Control Bloc</i>)
3/4.2.1.2	SCB (<i>System Control Bloc</i>)
3/4.2.2	Table d'indirection des fonctions BDOS en page zéro

3/1

Introduction au DOS

Pour introduire le DOS (Disk Operating System), voyons sommairement ce qu'est un ordinateur.

Pour cela, utilisons la technique des boîtes noires qui permet de formaliser un grand nombre de systèmes, informatiques ou non. Considérons un ordinateur comme une boîte noire dans laquelle entrent et sortent des informations.



Interrogeons-nous sur la nature des informations à fournir (fournies) à (par) l'ordinateur.

Entrées : Clavier, interfaces de communication, etc.

Sorties : Ecran, imprimante, lecteurs de disquettes, interfaces de communication, etc.

Un ordinateur apparaît donc comme un système :

- capable de traiter des commandes issues d'organes d'entrée comme des ordres utilisateur provenant du clavier, des ordres ou données arrivant par une interface de communication, etc. ;
- capable d'afficher le résultat des traitements sur divers périphériques : écran, imprimante, etc., de les stocker sur support magnétique, de les émettre sur interface de communication, etc.

Pour réaliser ces opérations, un ordinateur est constitué d'une carte mère regroupant des composants que nous répartirons en trois grandes familles :

- 1) Le ou les microprocesseur(s) ;
- 2) Les mémoires (RAM, ROM, autres...) ;
- 3) Les circuits de gestion de périphériques (clavier, écran, imprimante, etc.).

Les composants des types 2 et 3 suivent les directives du ou des microprocesseur(s) exprimées dans un langage de communication de bas niveau : le binaire. Pour faciliter l'utilisation des ressources de l'ordinateur, des langages semi-évolués ou évolués ont été créés : ASSEMBLEUR, BASIC, PASCAL, etc.

Le volume des données à traiter n'a cessé d'augmenter. Alors est apparu le besoin de stocker ces données, d'abord sur unités de cassettes, puis sur unités de disquettes et disques durs.

Devant la grande taille de stockage des lecteurs de disquettes ou disques durs, les constructeurs ont réagi en créant des systèmes d'exploitation (DOS = Disk Operating System, ou, en français SED = Système d'Exploitation de Disques).

Un DOS est destiné à faciliter :

- la gestion de l'espace disque ;
- la gestion des entrées/sorties.

Remarque :

Les systèmes d'exploitation sont des programmes constitués d'un ensemble de fonctions représentées par des mots-clés. Chaque fonction fait l'objet d'un programme d'extension « .COM » stocké sur disque.

Facilité et rapidité d'accès aux fichiers sur disque :

Pour accéder à un fichier disque, il suffira de donner son nom au DOS. C'est le DOS qui se chargera de le convertir en une adresse physique et d'extraire les données demandées parmi les milliers ou millions d'octets disponibles sur le support magnétique.

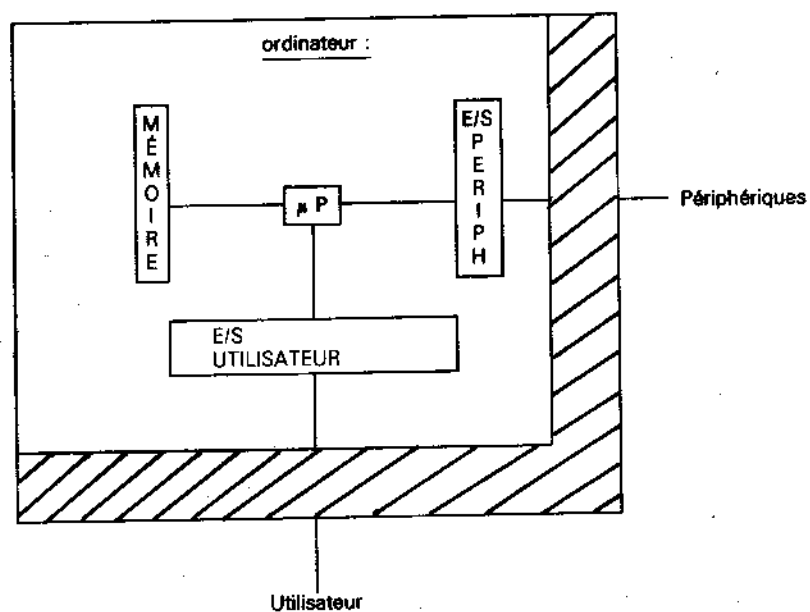
De même, le DOS sera capable de stocker des données sur disque, rapidement, et là où aucune autre donnée n'est déjà stockée. Il pourra même morceler le fichier soumis pour occuper les « trous » libres sur la disquette si celle-ci est relativement pleine.

De plus, le DOS permettra à l'utilisateur de faire le répertoire (DIRECTORY) des fichiers présents sur le support, de supprimer ou de renommer un ou plusieurs fichier(s).

Facilité et rapidité d'accès aux périphériques :

Les organes périphériques sont essentiellement le clavier, l'écran, l'imprimante et le lecteur de disquettes. Ils sont reliés à l'ordinateur par des connexions standard, et échangent avec lui des messages selon un protocole de communication. Les différences de protocole entre deux périphériques donnés sont très importantes, et, sans l'aide du DOS, l'utilisateur devrait effectuer une gymnastique intellectuelle pour passer d'un périphérique à un autre. Heureusement, le DOS permet de banaliser les entrées/sorties, de telle sorte que, par exemple, une demande d'impression sur écran ou sur imprimante se fait selon le même format, en précisant seulement quel est l'organe de sortie choisi. De même, la lecture au clavier ou sur lecteur de disquettes est identique, mis à part le nom du périphérique qui émet les données.

En conclusion, nous voyons que le DOS est une « couche logicielle » d'interfaçage entre la carte mère et les périphériques utilisant un protocole d'échange de haut niveau.



 Couche logicielle DOS

3/2

AMSDOS : Définitions, rappels et utilisations

AMSDOS n'est pas un système d'exploitation en tant que tel, puisqu'il ne gère qu'une des deux fonctions d'un système d'exploitation : les entrées/sorties disques.

AMSDOS se compose d'un ensemble de mots-clés activables depuis le langage BASIC, en mode direct comme en mode programme. Pour les distinguer des autres mots-clés, ils sont précédés d'une barre verticale. Ils permettent de gérer deux lecteurs de disquettes et un lecteur de cassettes (effacer et renommer un fichier ou un ensemble de fichiers disque, et définir des « USER »).

MOTS-CLÉS ET LEUR UTILISATION

Voyons en détails la fonction de chaque mot-clé, et comment les utiliser en nous appuyant sur des exemples précis.

Nous avons réparti les mots-clés d'AMSDOS en trois groupes :

I. Gestion des unités de disquettes :

IA, IB, IDISC, IDISC.IN, IDISC.OUT, IDRIVE

II. Gestion de l'unité de cassettes :

ITAPE, ITAPE.IN, ITAPE.OUT

III. Utilitaires :

IDIR, IERA, IREN, IUSER

Leur sens est indiqué ci-après.

I. Gestion des unités de disquettes

IA

Définit le lecteur de disquettes vers lequel s'effectueront les entrées/sorties comme étant le drive interne.

Synonyme : IDRIVE, "A"

Si vous ne possédez qu'un lecteur de disquettes, il sera forcément identifié par :A et cette commande sera implicite à la mise en marche de la machine.

IB

Définit le lecteur de disquettes vers lequel s'effectueront les entrées/sorties comme étant le lecteur externe.

Synonyme : IDRIVE, "B"

Si vous ne possédez qu'un lecteur de disquettes, une telle commande provoquera l'apparition du message d'erreur « Drive B: disk missing » dès la chute du « time-out » du driver disque. Ce message indique qu'aucun lecteur n'est connecté sur l'unité B.

IDISC

Les entrées/sorties des données se feront sur disquette (à opposer à ITAPE). Cette commande équivaut aux commandes « IDISC.IN » et « IDISC.OUT » cumulées.

IDISC.IN

La lecture des données se fera sur disquette.

Cette commande est à utiliser plutôt que « : DISC » si vous désirez différencier les supports magnétiques d'entrée et de sortie.

Par exemple, pour lire sur lecteur de disquettes et pour écrire sur lecteur de cassettes, il faudra faire : IDISC.IN et ITAPE OUT.

IDISC.OUT

L'écriture des données se fera sur disquette.

Cette commande est à utiliser plutôt que « : DISC » si vous désirez différencier les supports magnétiques d'entrée et de sortie.

Par exemple, pour lire sur lecteur de cassettes et pour écrire sur lecteur de disquettes, il faudra faire : ITAPE.IN et IDISC.OUT

IDRIVE, < Chaîne alphanumérique >

Cette commande a pour équivalents :

	IDRIVE, "A"	IA
et	IDRIVE, "B"	IB

Utilisation

De même que pour IB, si vous possédez un seul lecteur de disquettes, la commande IDRIVE, « B » provoquera l'apparition du message d'erreur « Drive B: disk missing » à la chute du « time-out » du driver disque.

Sur votre CPC 664 ou CPC 6128, vous avez connecté deux lecteurs de disquettes et un lecteur de cassettes, et vous voulez définir la direction des entrées/sorties.

```
10  MODE 2:PRINT "Choix de la direction de l'I/O sur le support"
20  INPUT "Entrez 1)K7 ou 2)Disquette" ; I
30  IF I = 1 THEN ITAPE.IN:GOTO 60
40  INPUT "Disquette A ou B" ; I$
50  IF I$ = "A" THEN IA:IDISC.IN ELSE IB:IDISC.IN
60  INPUT "Sortie 1)K7 ou 2)Disquette" ; I
70  IF I = 1 THEN ITAPE.OUT ELSE IDISC.OUT
```

Ligne 30 : Entrée sur cassette,

Ligne 50 : Entrée sur disque A: ou B:

Ligne 70 : Sortie sur cassette ou sur disquette.

II. Gestion de l'unité de cassettes

ITAPE

Les entrées/sorties des données se feront sur cassette après cette commande (à opposer à IDISC). Equivaut aux commandes « ITAPE.OUT » et « ITAPE.IN » cumulées.

ITAPE.IN

La lecture des données se fera sur cassette à la suite de cette commande.

Cet ordre est à utiliser plutôt que ITAPE si vous désirez différencier les supports magnétiques d'entrée et de sortie (Cf. exemple ci-dessous).

ITAPE.OUT

L'écriture des données se fera sur cassette à la suite de cette commande.

Cet ordre est à utiliser plutôt que ITAPE si vous désirez différencier les supports magnétiques d'entrée et de sortie (Cf. exemple ci-dessous).

Remarque :

Reportez-vous au chapitre 4/1 pour avoir des détails sur l'utilisation des ordres BASIC de lecture/écriture sur cassette (CAT, CHAIN, PRINT #, OPENIN, LOAD, etc.).

III. Utilitaires

IDIR[, <Chaîne alphanumérique>]

Affiche le catalogue (DIRectory) de la disquette selon l'ordre d'entrée des fichiers sur la disquette.

La chaîne alphanumérique peut prendre les valeurs suivantes :

— Inexistante :

Donne la liste de tous les fichiers sur USER 0

— Nom de lecteur, par exemple « B: » :

Donne la liste de tous les fichiers présents sur « B: » et retourne sur le lecteur courant en fin d'affichage.

— *.COM : Donne la liste de tous les fichiers d'extension « .COM »

— PI*.COM : Donne la liste de tous les fichiers commençant par « PI » et d'extension « .COM »

— P?E.EXE : Donne la liste de tous les fichiers de trois lettres dont la première est « P », la deuxième est quelconque, la troisième est « E » et dont l'extension est « .EXE »

— AMSDOS.COM : Fait apparaître le fichier « AMSDOS.COM » s'il existe sur la disquette.

Remarque :

La place disponible sur la disquette est systématiquement donnée en fin d'affichage sous la forme :

« XXX K Free »

IERA, <Chaîne alphanumérique>

Efface le ou les fichier(s) correspondant à la chaîne alphanumérique citée.

Par exemple, IERA, "*.BAK" effacera tous les fichiers d'extension « .BAK ».

Si le ou les fichier(s) cité(s) n'existe(nt) pas sur le disque, le message suivant apparaîtra : « ??????.ext not found »

Si la disquette concernée est protégée, le message suivant apparaît :

« Drive A: disk is write protected »

« Retry, Ignore or Cancel ? »

IREN, <Chaîne alphanumérique 1>, <Chaîne alphanumérique 2>

Rebaptise le nom d'un fichier. Les chaînes 1 et 2 peuvent avoir l'allure suivante :

NOM.EXT,
ou USER n:NOM.EXT

Ainsi, il est possible de changer l'« USER » d'un fichier par cette commande.

Exemple :

Soit le fichier « 0:FILE.BAS » (donc sous USER 0) que nous voulons transférer sous USER 15. Il faudra faire : IREN, "15:FILE.BAS", "0:FILE.BAS"

Le « 0 » est optionnel ; s'il est omis, il prendra par défaut la valeur de l'USER courant.

Pour visualiser le fichier « renommé », il faudra faire :
IUSER,15 et IDIR.

IUSER, <Nombre entier entre 0 et 15>

Définit un « sous-directory » dans lequel les commandes d'entrées/sorties disquettes vont être restreintes. Ainsi, par exemple, si nous passons sous IUSER,3 et que nous stockons un programme ou un fichier, il ne sera visible que sous USER 3. Si nous repassons sous USER 0 (USER affecté à la mise sous tension de la machine), et que nous faisons :DIR, le programme ou fichier stocké sous USER 3 n'apparaîtra pas. Cette commande est utile pour séparer des applications qui se trouvent sur un même disque.

Par exemple, USER 0 contiendra un traitement de textes, USER 1 des jeux, USER 2 des fichiers de données, etc.

Remarque sur les RSX (Résident System eXtension) :

Les RSX sont des commandes définies par l'utilisateur et précédées d'un « I » pour les différencier des commandes internes au BASIC comme « PRINT », « GOTO », etc.

Ces commandes sont à différencier des commandes AMSDOS, et nous verrons comment les utiliser (voir Partie 4 chap. 2.4)

PROGRAMMATION EN AMSDOS

Les possibilités d'AMSDOS sont assez limitées. Mais nous voyons qu'elles représentent une bonne extension au langage BASIC des 464, 664 et 6128. En effet, elles peuvent être incorporées dans un programme et être utilisées comme des instructions internes au BASIC.

3/2.1

Liste alphabétique des mots-clés

mots-clés		numéro de page
IA	Passage sur le disque A	2
IB	Passage sur le disque B	2
IDIR	Catalogue d'une unité de disque ou de cassette	3
IDISC	Le disque est défini par défaut en lecture/écriture	2
IDISC.IN	Le disque est défini par défaut en lecture	2
IDISC.OUT	Le disque est défini par défaut en écriture	2
IDRIVE	Passage sur le disque A ou sur le disque B	2
IERA	Effacement d'un fichier disque	4
IREN	Renomme un fichier disque	4
ITAPE	L'unité de cassette est définie par défaut en lecture/écriture	3
ITAPE.IN	L'unité de cassette est définie par défaut en lecture	3
ITAPE.OUT	L'unité de cassette est définie par défaut en écriture	3
IUSER	Définit le numéro d'utilisateur courant	5

3/3

CP/M 2.2 : Définitions, rappels et utilisations

Cette partie concerne les possesseurs de CPC 464 qui ont un lecteur de disquettes, les possesseurs de CPC 664, et éventuellement les possesseurs de CPC 6128. Ces derniers pourront utiliser soit CPM + (voir chapitre 4 de cette partie) soit CP/M 2.2 qui en est une sous-version.

HISTORIQUE

CP/M est un système d'exploitation monotâche, mono-utilisateur très répandu sur les micro-ordinateurs utilisant un microprocesseur 8080, 8085, Z80 ou même un 16 bits.

CP/M a été conçu en 1973 par Gary Kildall avec les objectifs suivants :

- fonctionner sur tout ordinateur à base de 8080, 8085 ou Z80 ;
- faciliter la gestion des lecteurs de disquettes ;
- fournir un ensemble d'ordres « système » simples pour aider le programmeur dans sa tâche.

En 1976, Kildall fonde Digital Research et CP/M est commercialisé.

Depuis, on peut dire que CP/M est devenu un standard des systèmes d'exploitation. Le nombre approximatif de ses utilisateurs est évalué à trois cent mille sur des ordinateurs familiaux et professionnels.

DÉFINITIONS ET RAPPELS

Lorsque vous faites un « démarrage à froid » de CP/M 2.2 en tapant la commande AMSDOS « ICPM », un programme situé dans les ROM du CPC charge quatre modules en mémoire RAM : TPA, CCP, BDOS et BIOS.

Remarque :

Par opposition, nous parlerons de démarrage à chaud lorsque seulement CCP et BDOS sont chargés en mémoire RAM. Cette opération est activée par l'appui simultané des touches CTRL et C et doit impérativement être effectuée après chaque changement de disquette pour éviter les erreurs du type « BDOS Error on A : R/O » qui signalent que CP/M ne reconnaît pas la disquette insérée dans le lecteur.

TPA :

(Transient Program Area) : Zone des programmes temporaires.

Dans cette zone sont en particulier chargés les programmes non résidents d'extension « .COM » qui composent les ordres de CP/M 2.2 et, en général, tout programme exécuté sous CP/M.

CCP :

(Console Command Processor) : Processeur des commandes consoles.

C'est la zone qui contient les commandes résidentes (voir *définitions*) : DIR, ERA, REN, SAVE, TYPE et USER.

BDOS :

(Basic Disk Operating System) : Système d'exploitation de base.

Contient les programmes servant à la gestion des unités de disquettes.

BIOS :

(Basic Input Output System) : Système d'entrée/sortie de base

Contient le logiciel de base qui permet d'adapter CP/M 2.2 aux ordinateurs CPC.

Le message « CP/M 2.2 Amstrad Consumer Electronics plc » et le prompt « A> » vous signalent que CP/M 2.2 est en mémoire et attend une commande.

Définitions :

Une commande est dite *résidente* quand elle est toujours présente en mémoire centrale RAM ou ROM.

Une commande est dite *transitoire* quand elle est présente sur disquette sous la forme d'un fichier d'extension « .COM » qui est chargé en RAM dans la zone TPA pour être exécutée.

- Caractères de contrôle reconnus par CP/M 2.2 :

Comme nous venons de le voir, l'appui simultané sur les touches CTRL et C produit un démarrage à chaud. D'autres touches peuvent être action-

nées en même temps que la touche CTRL pour produire ce que l'on appelle des caractères de contrôle. Ainsi :

- ^E** produit un passage à la ligne suivante
- ^H** un déplacement du curseur vers la gauche (identique à la touche DEL)
- ^I** une tabulation (identique à la touche TAB)
- ^J** un saut à la ligne (identique à la touche ENTER)
- ^M** un retour chariot (identique à la touche ENTER)
- ^P** un début ou une fin d'écho sur imprimante
- ^R** une réimpression de la ligne courante
- ^S** un arrêt de l'affichage sur l'écran
- ^U** un abandon de la ligne courante, et le début d'une autre ligne
- ^X** un abandon de la ligne courante sans commencer une autre ligne.

Dans la suite, nous allons analyser la ou les utilisation(s) possible(s) de chaque ordre CP/M 2.2.

MOTS-CLÉS ET LEUR UTILISATION

Nous avons divisé ces fonctions en trois grands groupes :

I. Entrées/sorties sur disque ou cassette :

CLOAD, COPYDISC, CSAVE, DIR, DISCOPY, DISCCHK, ERA, FILECOPY, FORMAT, LOAD, PIP, SAVE, STAT et USER.

II. Utilitaires :

BOOTGEN, DUMP, MOVCPM, REN, SETUP, SYSGEN, TYPE et XSUB.

III. Ordres évolués :

AMSDOS, CHKDISC, DISCKIT2, ED et SUBMIT.

Conventions d'écriture :

- [] Encadrent une ou plusieurs option(s)
- < > Encadrent un paramètre obligatoire
- () Plage finie de valeurs possibles
- | Soit l'un, soit l'autre
- n Substitution du paramètre par un entier
- s Substitution du paramètre par une chaîne alphanumérique

I. Entrées/Sorties sur disque ou cassette

CLOAD

CLOAD[<"Nom fichier cassette">][<Nom fichier disquette>]

Permet de copier un fichier d'une cassette vers une disquette.

Si les deux paramètres sont omis, cette commande chargera le premier fichier rencontré sur cassette et le sauvera sur disquette sous le même nom.

Format

Fonction

Utilisation

Si le premier nom est omis (celui entre cotes), le premier fichier rencontré sur cassette est chargé. Il est ensuite sauvegardé sur disquette sous le nom donné dans la commande.

Remarques :

a) Le premier nom peut être précédé d'un point d'exclamation pour éviter l'affichage des commentaires lors du chargement sur cassette.

b) La vitesse à laquelle a été enregistré le programme sur cassette n'est pas à considérer, car l'ordinateur choisira automatiquement la vitesse de lecture (1 000 ou 2 000 bauds).

Exemple :

Pour charger le programme cassette PROG.BIN et le sauvegarder sur disquette sous le nom PROG1.BIN, il faudra faire :

CLOAD "PROG.BIN" PROG1.BIN

COPYDISC

Format

COPYDISC

Fonction

Copie la totalité d'une disquette sur un CPC qui possède deux lecteurs de disquettes.

Utilisation

Tapez COPYDISC

L'ordinateur affiche « **Please insert source disc into drive A and destination disc into drive B then press any key :** ».

Une fois la copie terminée, le message suivant apparaît : « **Do you want to copy another disc (Y/N) :** ». Si vous répondez « **N** », CP/M affichera le message suivant : « **Please insert a CP/M system disc into drive A then press any key :** », vous invitant à placer une disquette CP/M dans le lecteur A.

Remarques :

a) La disquette destination n'a pas besoin d'être formatée. Si elle est vierge, CP/M saura le détecter et la formatera automatiquement.

b) Si une erreur se produit pendant la copie, un message d'erreur « **Failed...** » suivi de la cause de l'erreur sera affiché, et l'opération pourra être recommencée avec une autre disquette (dans le cas où c'est la destination qui est mise en défaut).

CSAVE

Format

CSAVE<nom de fichier disquette>
[<"Nom de fichier cassette">][<n>]

Fonction

Permet de copier un fichier d'une disquette vers une cassette.

Utilisation

Si le paramètre <"Nom de fichier cassette"> est omis, le fichier sauvegardé sur cassette portera le même nom et la même extension que le fichier disquette lu.

Si le paramètre <n> est omis, le fichier sera sauvegardé sur cassette à la vitesse de 2 000 bauds. Dans le cas contraire, il peut prendre deux valeurs pour spécifier la vitesse de sauvegarde : 0 pour 1 000 bauds et 1 pour 2 000 bauds.

Remarque :

Le deuxième nom peut être précédé d'un point d'exclamation pour éviter l'affichage des commentaires lors de l'écriture sur cassette.

Exemple :

Pour charger le programme disquette PROG1.BIN et le sauvegarder sur cassette sous le nom PROG.BIN à une vitesse de 2 000 bauds, il faudra faire :

CSAVE PROG1.BIN "PROG.BIN", 1

DIR

Format

DIR[<Nom de fichier>] ou DIR[<Nom d'unité>]

Fonction

Donne le répertoire (DiRectory) ou une partie du répertoire de l'unité de disque spécifiée ou de l'unité par défaut.

Utilisation

Dans le cas où aucune option n'est précisée, cette commande donne la liste de tous les fichiers de l'unité par défaut.

L'option <Nom d'unité> précise sur quel lecteur doit être fait le DiRectory.

L'option <Nom de fichier> permet :

— de vérifier qu'un fichier en particulier est bien sur une unité. Par exemple, si nous voulons vérifier que le fichier UTIL.PRG est bien sur l'unité B, nous taperons : **DIR B : UTIL.PRG**

Si le fichier est présent, son nom apparaîtra suite à la commande. Dans le cas contraire, le message « No file present » sera affiché.

— de lister un ensemble de fichiers en utilisant les signes « * » et « ? ».

Le signe « * » (appelé « Joker ») permet de remplacer tout ou partie d'un nom de fichier ou d'extension. Par exemple :

DIR *.BAS affichera tous les fichiers d'extension .BAS

DIR D.* affichera tous les fichiers de nom « D » et d'extension quelconque.

DIR D*.* affichera tous les fichiers dont la première lettre est D et d'extension quelconque.

Le signe « ? » permet de remplacer un caractère alphanumérique quelconque. La recherche des fichiers est faite sans considérer ce caractère mais en tenant compte du fait qu'il occupe une place.

« ? » peut être combiné avec le joker.

Exemple :

DIR AT?R*. * donnera la liste des fichiers dont les deux premières lettres sont A et T, dont la troisième lettre est quelconque, dont la quatrième est « R », suivi de 0, 1 ou plusieurs lettres et d'extension quelconque.

DISCCOPY**Format**

DISCCOPY

Fonction

Copie la totalité d'une disquette sur un CPC qui possède un lecteur de disquettes.

Utilisation

Tapez DISCCOPY puis ENTER.

CP/M affiche « **Please insert disc into drive A then press any key:** » Vous devez introduire la disquette source dans le lecteur et presser sur une touche quelconque. Un message vous indique que CP/M est en train de mémoriser une partie de la disquette. Quand la zone TPA est pleine, le message « **Please insert destination disc into drive A then press any key:** » apparaît. Vous devez éjecter la disquette source puis la remplacer par la disquette destination et enfin appuyer sur une touche quelconque.

Si la disquette destination n'est pas formatée, le message « **Formatting whilst copying** » vous indiquera qu'un formatage sera fait pendant la copie.

Cinq manipulations de disquettes seront nécessaires pour arriver à copier une disquette. Le message « **Do you want to copy another disc (Y/N):** » vous fera savoir que la copie est terminée et qu'une autre copie peut être effectuée si vous le désirez.

Si vous avez fini vos copies, il vous faudra insérer une disquette CP/M pour pouvoir retourner sous le système d'exploitation.

Remarque :

Pendant la copie, plusieurs messages d'erreur peuvent apparaître pour signaler que vous avez inversé la disquette source et la disquette destination, que la disquette destination est protégée ou encore qu'une partie de la disquette source ou destination est inexploitable.

DISCCHK

non

Format

DISCCHK

Fonction

Si vous possédez un seul lecteur de disquettes, DISCCHK (DISC CHeck ou vérification de disque) vous permettra de vérifier qu'une disquette copiée par DISCCOPY est bien la copie conforme de la disquette source.

Utilisation

Tapez DISCCHK.

Le message « **DISCCHK V2.0** »

« **Please insert source disc into A Then press any key:** » vous invite à placer le disque source dans l'unité de disquette et à appuyer sur une touche quelconque.

Huit pistes (sur les quarante) sont chargées en mémoire, et le message « **Please insert destination disc into drive A then press any key** » vous demande de retirer le disque source, de placer le disque à vérifier dans le lecteur et d'appuyer sur une touche quelconque.

Lorsque la vérification est terminée (après 5 échanges source/destination) le message suivant apparaît :

« **Copy checking complete** »
« **Do you want to check another disc (Y/N):** »

Ce message vous indique que la vérification s'est bien passée et vous demande s'il y a une autre disquette à vérifier. Si oui, recommencez les mêmes opérations avec la nouvelle disquette. Si non, insérez une disquette CP/M pour revenir sous le système d'exploitation.

Remarque :

Si les deux disquettes ne sont pas identiques, l'ordinateur donne le numéro de secteur et de piste différent dans un message du type suivant :

« **Failed to verify the destination disc correctly : track 1 Sector 5** »

ERA

Format

ERA [<Nom d'unité>:] <Nom de Fichier>

Fonction

Efface un fichier ou un groupe de fichiers sur disque.

Utilisation

Tapez ERA suivi du nom du fichier ou des fichiers à effacer. Faites précéder ce nom du nom de l'unité s'il ne s'agit pas de l'unité par défaut.

Remarque :

Les messages d'erreur suivants peuvent apparaître :

a) « **Bdos Err On a: File R/O** »

Ce message signale que le fichier que vous avez tenté d'effacer a un attribut Read Only, ce qui interdit son effacement. Reportez-vous à l'ordre STAT pour modifier l'attribut de ce fichier.

b) « **Drive A: disc is write protected** »
« **Retry, Ignore, Cancel** »

Ce message signale que la disquette est protégée en écriture. Si vous désirez tout de même effacer le ou les programme(s) spécifié(s), déprotégez la disquette et tapez R(etry).

Exemples :

ERA AB?T*.COM effacera tous les fichiers dont les deux premières lettres sont A et B, dont la troisième est quelconque, la quatrième est T, les lettres suivantes quelconques ou inexistantes et l'extension .COM.

Si vous tapez « ERA*.* », CP/M vous demandera si vous voulez bien effacer tout le répertoire de la disquette en affichant le message « **ALL (Y/N)** ».

FILECOPY**Format**

FILECOPY <Nom de fichier>, ou
FILECOPY <Nom de fichier>/S<Numéro de zone>, ou
FILECOPY <Nom de fichier>/D<Numéro de zone>

Fonction

Sur un CPC ne comportant qu'un lecteur de disquettes, copie un ou plusieurs fichier(s) d'une disquette sur une autre, ou d'un « USER » sur un autre.

Utilisation

Insérez une disquette contenant le programme « FILECOPY.COM », tapez « FILECOPY <Nom de fichier(s)> » pour copier <Nom de fichier(s)> d'une disquette sur une autre.

Le programme FILECOPY.COM est chargé en TPA, et le message suivant apparaît :

« FILECOPY V2.1 »

« Please insert SOURCE disc into drive A then press any key : »

Insérez la disquette contenant le ou les fichiers à copier dans le drive A et tapez sur une touche quelconque. Le message suivant apparaît :

« Copying started... »

« Please insert DESTINATION disc into drive A then press any key : »

pour vous signaler que le programme a bien été chargé en mémoire et peut être copié sur la disquette DESTINATION. Insérez-la et tapez sur une touche quelconque. Le message suivant apparaît :

« <Nom de fichier> Copied »

« Copying complete »

« Please insert a CP/M system disc into drive A then press any key : »

Retirez la disquette « destination » et insérez une disquette « système » pour retourner sous CP/M.

Remarque :

Si vous utilisez un joker ou un ? dans la spécification des fichiers à copier, le message suivant apparaîtra :

« Ambiguous file name : Confirm individual files (Y/N) ? »

Si vous répondez **N(o)**, tous les fichiers seront copiés. Si vous répondez **Y(es)**, une confirmation sera demandée avant la copie de chaque fichier. Le message suivant sera alors affiché :

« <Nom de fichier> Copy (Y/N) ? »

Une autre utilisation possible de FILECOPY consiste à utiliser les options <D> ou <S>.

L'option <D> sert à copier un fichier ou un groupe de fichiers de l'USER courant à un USER défini par D.

L'option <S> sert à copier un fichier ou un groupe de fichiers de l'USER défini par S à l'USER courant.

Exemples :

FILECOPY ST * . * copie tous les fichiers dont les deux premières lettres sont S et T et d'extension quelconque d'une disquette sur une autre.

FILECOPY PROGRAM.COM/D4 copie le programme PROGRAM.COM de l'USER courant sous USER 4. Le message suivant apparaît :

« Copying will be to USER 4 » vous signalant que le fichier PROGRAM.COM sera copié sur l'USER 4.

FILECOPY PROGRAM.COM/ S12 copie le programme PROGRAM.COM de l'USER 12 sous l'USER courant. Le message suivant apparaît :

« Copying will be from USER 12 » vous signalant que le fichier PROGRAM.COM sera lu sur l'USER 12.

FORMAT**Format**

FORMAT [<S>] ou
FORMAT [<D>] ou
FORMAT [<I>] ou
FORMAT [<V>]

Fonction

Formate une disquette vierge ou déjà utilisée.

Utilisation

L'option <S> est l'option par défaut. Elle correspond au format AMS-TRAD système standard (le même que celui où a été lue la commande FORMAT).

L'option <D> vous permet de créer une disquette au format « Données ». CP/M ne pourra pas être copié sur une telle disquette. Ce format sert essentiellement aux CPC possédant deux lecteurs de disquettes.

L'option <I> est comparable à l'option <D>, à ceci près que le format de configuration des pistes suit le standard IBM.

L'option <V> (ou vendeur) permet de créer une disquette système (sur laquelle on pourra insérer CP/M). Reportez-vous aux commandes SYSGEN et BOOTGEN pour avoir plus de détails (voir p. 15 et 22).

Placez la disquette contenant le programme « FORMAT.COM » dans l'unité par défaut et tapez « FORMAT » éventuellement suivi d'une des options décrites ci-dessus. Le message suivant apparaît : « Please insert disc to be formatted into drive A then press any key : ». Ce message vous invite à placer le disque à formater dans l'unité par défaut et à appuyer sur une touche quelconque.

Lorsque le formatage est terminé, le message suivant apparaît : « Do you want to format another disc (Y/N) : ».

Si vous répondez N(o), placez un disque système pour retourner sous CP/M.

Remarques :

a) Le message d'erreur suivant peut être affiché :

« Drive A; Read fail n »

« Retry, Ignore or Cancel ? »

Une erreur s'est produite sur la piste n. Essayez à nouveau en choisissant l'option R(etry). Si le même message est affiché, la disquette que vous tentez de formater possède un défaut et doit impérativement rester inutilisée.

b) Si vous tentez de formater un disque protégé, le message suivant apparaîtra :

« The disc to be formatted in drive A must be write enabled » « Please insert disc to be formatted into drive A then press any key : »

vous invitant à déprotéger le disque à formater, et à appuyer sur une touche quelconque.

LOAD

Format

LOAD <Nom de fichier>

Fonction

Permet de transformer un fichier d'extension .HEX (compilé par le programme « ASM » ou un autre compilateur) en fichier exécutable sous CP/M 2.2 d'extension .COM.

Utilisation

Tapez LOAD suivi du nom du fichier .HEX à transformer. Lorsque l'opération est terminée, le message suivant apparaît :

FIRST ADDRESS	<Adresse de chargement du programme .COM>
LAST ADDRESS	<Dernière adresse du programme .COM>
BYTES READ	<Nombre d'octets lus>
RECORDS WRITEN	<Nombre d'enregistrements de 256 octets écrits>

Remarque :

Le message d'erreur suivant peut apparaître pour signaler que des codes incorrects ou inconnus ont été rencontrés :

INVALID HEX DIGIT	
LOAD ADDRESS	<Adresse de chargement du programme .COM>
ERROR ADDRESS	<Adresse où l'erreur a été rencontrée>
BYTES READ	<Liste des octets ayant provoqué l'erreur>

PIP

Format

PIP [<ligne de commande>]

Fonction

PIP = (Peripheral Interchange Program) : programme d'échange entre périphériques.

Cet ordre permet de transférer un ou plusieurs fichier(s) d'une unité de disquette sur une autre, ou plus généralement d'un périphérique sur un autre, de concaténer plusieurs fichiers en un seul ou encore d'affecter un attribut à un fichier.

Utilisation

Vous pouvez utiliser la commande PIP de deux manières différentes :

- en direct, en tapant une ligne de commande après le mot-clé PIP,
- en changeant PIP en TPA (voir partie 3 chapitre 3). Pour cela, tapez « PIP », puis <cr>. Un astérisque apparaît pour signaler que PIP est en mémoire et attend une commande. Ce procédé a pour avantage de ne pas obliger la présence du programme PIP sur la disquette qui contient le programme à manipuler.

La <ligne de commande> d'un ordre PIP est la suivante :

PIP <destination> = <source 1>[, <source 2>, ...[, <source n>]]
[, <paramètre(s)>]

où <destination> est un nom de fichier ou d'unité.

<source i> est un nom de fichier ou d'unité.

Si plusieurs fichiers « sources » sont présents, ils seront concaténés, dans l'ordre où ils sont cités, dans le fichier « destination ».

- Les unités suivantes peuvent être utilisées :

Unités logiques :

CON : CONsole : Unité d'écran/clavier (entrée/sortie)

RDR : REaderR : Unité de lecture (entrée)

PUN : PUNch : Unité d'écriture (sortie)

LST : LiSTer : Unité de listage (sortie)

Unités physiques :

CRT : Ecran/clavier

LPT : Interface parallèle

TTY : Unité d'entrée/sortie

UC1 : Unité d'entrée/sortie

- Les paramètres suivants peuvent être utilisés :

- B Le transfert des données n'est effectué que lors de la réception du caractère XOFF. Cette option permet de stocker plusieurs fichiers en mémoire, à concurrence de la taille RAM disponible.
- Dn Efface tous les caractères occupant une colonne supérieure à n
- E Echo à l'écran des données copiées.
- F Efface les « Form Feed » (saut de page) du fichier copié. Cette option est utile pour éviter les pertes de papier lorsque la sortie se fait sur imprimante.
- Gn Utilisation d'un USER différent de l'USER courant.
- H Vérifie les données transférées. Cette option est utilisée lorsque le ou les fichier(s) transféré(s) ont une extension .HEX.
- I Supprime les octets à 0 d'un fichier d'extension .HEX.
- L Met en minuscules les caractères du fichier manipulé.
- N Numérote les lignes du fichier transféré.
- O Permet d'ignorer le caractère « fin de fichier » d'un fichier non ASCII.

- Pn** Définit la longueur de la page à n lignes.
- Qs^Z** Finit la copie quand le caractère s est rencontré.
- R** Permet de lire des fichiers d'attribut SYS.
- Ss^Z** Commence la copie à partir du caractère s.
- Tn** Les tabulations sont étendues à n espaces.
- U** Met en majuscules les caractères du fichier manipulé.
- V** Vérification des données copiées.
- W** Permet d'écrire sur un fichier protégé sans qu'une confirmation soit demandée.
- Z** Les caractères copiés ont leur bit de parité à 0.

Exemples :

PIP TOTAL.COM = PA1.COM, PA2.COM, B:PA3.COM concatène les fichiers PA1.COM et PA2.COM qui se trouvent sur l'unité par défaut avec le fichier PA3.COM qui se trouve sur B dans le fichier TOTAL.COM ; ce dernier sera créé sur l'unité par défaut.

PIP CON: = PROG.ASM copie le programme PROG.ASM sur l'écran.

PIP COMMAND.SUB = CON: copie les caractères tapés au clavier dans le fichier COMMAND.SUB. Pour passer à la ligne, il faudra taper « CTRL + J ».

Quand la saisie sera terminée, il faudra taper « CTRL + Z ».

PIP LST: = PROG.ASM envoie le fichier PROG.ASM sur l'imprimante.

PIP PROG.COM[g3] = PP.COM copie le fichier PP.COM lu sur l'USER 0, sur l'USER 3 sous le nom PROG.COM.

SAVE**Format**

SAVE <Nombre de pages de 256 octets> <Nom de fichier>.

Fonction

Sauvegarde sur disquette une zone mémoire commençant en 0100H dans le fichier spécifié. La taille de la zone mémoire sauvegardée est déterminée modulo 256 par le paramètre <Nombre de pages>.

Cette commande est à rapprocher de la commande MOVCPM (si vous voulez sauvegarder les caractéristiques de l'image CP/M générée par cet ordre). Reportez-vous à cette commande pour avoir plus de détails (voir p. 16).

Exemple :

SAVE 34 IMAGE.COM sauvegarde 34 pages de 256 octets, soit 768 octets à partir de l'adresse 0100H dans le fichier IMAGE.COM.

STAT**Format**

STAT[<Unité de disquette :> [= <Attribut>]
 ou STAT[<Nom de fichier> [<\$Attribut>]
 ou STAT[<Unité logique> := <Unité physique>]
 ou STAT DEV :
 ou STAT VAL :
 ou STAT USR :
 ou STAT[<Unité de disquette :>] [DSK:]

Fonction

Donne des renseignements sur les périphériques connectés ou sur les disquettes utilisées, et permet d'affecter un attribut à un fichier.

Utilisation

STAT <Unité de disquette :>
 donne la place disponible sur le disque sous la forme :

Bytes Remaining on <unité> : XXk

STAT <Unité de disquette :> [= <\$Attribut>]

STAT B: = R/W autorise la lecture et l'écriture sur la disquette de l'unité B: soumise à la commande.

STAT A: = R/O autorise la lecture seule sur la disquette de l'unité A: soumise à la commande.

STAT[<Nom de fichier> [<\$Attribut>]

STAT PIP.COM donnera les informations suivantes sur le fichier PIP.COM:

Recs	Bytes	Ext	Acc
58	8k	1	R/W A:PIP.COM

Bytes Remaining on A: 12k (par exemple).

STAT PIP.COM \$R/O permet d'interdire la modification, l'écriture ou l'effacement du fichier PIP.COM.

STAT PIP.COM \$R/W permet aux commandes ERA et REN d'agir sur le fichier PIP.COM. Il pourra être effacé, renommé ou modifié avec un tel attribut.

STAT PIP.COM \$SYS permet de faire disparaître le fichier PIP.COM du répertoire des fichiers listables par la commande DIR. Cependant, un fichier système (d'attribut \$SYS) sera listable par la commande STAT PIP.COM.

STAT PIP.COM \$DIR permet de rétablir l'affichage du label du fichier PIP.COM dans le répertoire des fichiers.

STAT DEV: donne les affectations des périphériques logiques aux périphériques physiques.

A>stat dev:

CON: is CRT:

RDR: is TTY:

PUN: is TTY:

LST: is LPT:

STAT VAL : donne des renseignements sur l'utilisation de STAT

A>stat val:

Temp R/O Disk : d: = R/O

Set Indicator : d:filename.typ \$R/O \$R/W \$SYS \$DIR

Disk Status : DSK: d:DSK:

User Status: USR:

IObyte Assign :

CON : = TTY: CRT: BAT: UC1:

RDR : = TTY: PTR: UR1: UR2:

PUN : = TTY: PTP: UP1: UP2:

LST : = TTY: CRT: LPT: UL1:

STAT USR : donne la zone « utilisateur » courante et les zones contenant au moins un fichier.

A>stat usr :

Active User : 0

Active Files : 0

STAT DSK : donne les caractéristiques de l'unité par défaut.

A>stat disk:

A : Drive Characteristics
1368 : 128 Byte Record Capacity
171 : Kilobyte Drive Capacity
64 : 32 Byte Directory Entries
64 : Checked Directory Entries
128 : Records/Extent
8 : Records/Block
36 : Sectors/Track
2 : Reserved Tracks

USER

USER <Numéro d'utilisateur>

Définit le numéro d'utilisateur courant.

Le numéro d'utilisateur doit être compris entre 0 et 15. L'utilisateur par défaut est 0.

Format

Fonction

Utilisation

Cette commande permet de définir sur un même disque des zones de stockage indépendantes pour la commande DIR et les diverses commandes de lecture/écriture sur disque.

Exemple :

« USER 4 » fait passer l'utilisateur en zone 4.

II. Utilitaires

BOOTGEN

Format

BOOTGEN.

Fonction

Permet de copier le secteur de configuration d'une disquette (Piste 0, Secteur 49).

Utilisation

Insérez la disquette CP/M et tapez BOOTGEN.

Le message suivant apparaît :

BOOTGEN V2.0

Please insert SOURCE disc into drive A then press any key :

Placez la disquette contenant le secteur de configuration à copier dans le lecteur A: et pressez une touche. Le message suivant apparaît :

Please insert DESTINATION disc into drive A then press any key :

Placez la disquette qui doit recevoir le secteur de configuration dans le lecteur A: et pressez une touche.

Le message suivant apparaît, vous signalant que la copie est terminée et qu'une autre copie peut être lancée :

Do you wish to reconfigure another disc (Y/N) :

Insérez une disquette système pour retourner sous CP/M quand vous aurez fini vos transferts.

DUMP

Format

DUMP [<unité>:]<Nom de fichier>

Fonction

Affiche sur l'écran ou sur l'imprimante (si vous tapez ^P) le contenu d'un fichier en hexadécimal.

Utilisation

Tapez DUMP suivi du nom du fichier à examiner. A gauche de l'écran apparaît l'adresse examinée A suivie de 16 valeurs hexadécimales sur 8 bits qui indiquent les valeurs des octets d'adresse A, A + 1, ... A + 15.

Remarque :

Si le fichier n'est pas sur la disquette indiquée, le message d'erreur suivant apparaîtra :

NO INPUT FILE PRESENT ON DISC.

MOVCPM**Format**

MOVCPM <taille allouée> * ou

MOVCPM * *

Fonction

Chargement de CP/M à une adresse différente de celle qui est habituellement utilisée.

Utilisation

Si la taille est précisée, elle indique le nombre de blocs de 256 octets (entre 64 et 179) alloués à CP/M.

Si aucune taille n'est précisée, CP/M occupera 48 kilo-octets (version maximale).

Le message suivant sera affiché :

CONSTRUCTING XXk CP/M vers 2.2
READY FOR "SYSGEN" OR
"SAVE 34 CPMXX.COM"

Il vous indique que CP/M est actuellement en RAM, et peut être sauvegardé sur disquette par la commande « SYSGEN », ou sauvegardé dans le fichier CPMXX.COM en tapant :

« SAVE 34 CPMXX.COM ».

REN**Format**

REN [<unité>:]<Nouveau nom> = [<unité>:]<Ancien nom>

Fonction

Donne un nouveau nom à un fichier disque.

Exemple :

Si vous voulez renommer le fichier PIP.COM en COPY.COM, il faudra faire :

REN COPY.COM = PIP.COM**Remarques :**

La commande REN est résidente. Elle se trouve donc en permanence en mémoire et il n'est pas nécessaire d'utiliser la disquette CP/M pour y avoir accès.

Il est impossible de renommer un fichier présent sur une unité n en un fichier sur une unité p.

Si vous essayez par exemple : REN B:COPY.COM = A:PIP.COM, un message d'erreur apparaîtra.

Si le fichier PIP.COM n'est pas trouvé sur le disque par défaut, le message d'erreur suivant apparaîtra : « NO FILE » et la commande ne sera pas utilisée.

Si le fichier COPY.COM existe déjà sur le disque par défaut, le message d'erreur suivant apparaîtra : « FILE EXISTS » et la commande ne sera pas exécutée.

Si le fichier **PIP.COM** possède un attribut « Read/Only » (voir commande **STAT** p. 22), il ne sera pas renommé, et le message d'erreur suivant apparaîtra : « Bdos Err On A: File R/O » si le lecteur par défaut est le lecteur A:

SETUP

Format

SETUP.

Fonction

Cet ordre permet de modifier :

- les configurations clavier, lecteur(s) de disquettes, interface série,
- le contenu du programme qui sera exécuté lors d'un démarrage à froid dès que CP/M aura été chargé.

Utilisation

Les informations de configuration des périphériques et l'« auto-start CP/M » sont inscrits sur la disquette CP/M, piste 0, secteur 42H. La commande **SETUP** permet d'accéder à ces informations et éventuellement de les modifier de manière interactive.

Phase 1 :

Entrez la commande « **SETUP** ».

Le premier message affiché par la commande **SETUP** indique le contenu de l'éventuel programme « batch auto-start » (programme lancé automatiquement sur un démarrage à froid de CP/M).

Si le programme « auto-start » n'existe pas, le message suivant apparaîtra à l'écran :

**** Initial command buffer empty
Is this correct (Y/N) :**

Si le programme « auto-start » existe et s'il contient, par exemple, l'activation d'un programme de nom « PROG », le message suivant apparaîtra à l'écran :

**Initial command buffer :
PROG^M
Is this correct (Y/N) :**

Le ^M (control M) est le code ENTER.

Reportez-vous à l'annexe 1, partie 11, pour avoir la liste des codes de contrôle de l'AMSTRAD et leurs fonctions.

Dans les deux cas, tapez Y si le programme « auto-start » vous convient tel qu'il est actuellement. Tapez N si vous voulez le modifier et entrez la ou les commandes que vous désirez activer sur un démarrage à froid, en les séparant par le code ^M (Enter).

Phase 2 :

Le deuxième message affiché par la commande **SETUP** concerne le message de « loggon », c'est-à-dire le message qui sera affiché à chaque

démarrage à froid. Ce message peut comporter toute combinaison de caractères alphanumériques et de contrôle. Reportez-vous à l'annexe pour avoir la liste des codes de contrôle de l'AMSTRAD et leurs fonctions. Le message standard de « loggon » est le suivant :

Sign-on string :

```
^ \@ww^a@@^]wwCP/M 2.2 Amstrad Consumer Electronics
plc^J^M
```

Is this correct (Y/N) : __

Si vous vous reportez à l'annexe concernant les codes de contrôle, vous pourrez voir que :

^ \ est l'équivalent CP/M de la commande BASIC INK. Ce code devra donc être suivi de trois paramètres indiquant :

- pour le premier, le numéro d'encre modulo 16,
- pour le second, le numéro de première couleur modulo 32,
- pour le troisième, le numéro de seconde couleur modulo 32. Par exemple : ^ \@ww est à interpréter de la façon suivante :

Partant du fait que,

- le code ASCII de @ est 64 en codage décimal, et $64 \text{ MOD } 16 = 0$,
- le code ASCII de w est 119 en codage décimal, et $119 \text{ MOD } 32 = 23$,

il s'ensuit que la commande CP/M « ^ \@ww » est équivalente à la commande BASIC INK 0, 23, 23.

En appliquant le même raisonnement à la commande CP/M « ^ a@@ », on obtient un équivalent BASIC INK 1, 0, 0.

En ce qui concerne la commande ^]ww, et en raisonnant comme ci-dessus, nous voyons que la commande BASIC équivalente est :

BORDER 23, 23

Les caractères ^J et ^M correspondent respectivement aux codes « passage à la ligne suivante » et « retour charriot ».

Nous voyons donc que le message standard de « loggon » a pour équivalent les commandes BASIC suivantes :

INK 0, 23, 23: INK 1, 0, 0.

PRINT "CP/M 2.2 Amstrad Consumer Electronics plc"

Si vous répondez Y à la question « Is this correct (Y/N) », ce message sera conservé. Si vous répondez N, vous pourrez définir un nouveau message en utilisant des caractères alphanumériques et de contrôle comme il vient d'être indiqué.

Phase 3 :

Le troisième message affiché par la commande SETUP est le suivant :

**** Printer power-up string empty**
Is this correct (Y/N) :

Ce message signale qu'aucun caractère de commande n'est envoyé à l'imprimante (dans le but de la configurer) sur un démarrage à froid. Consultez le manuel technique de votre imprimante si vous désirez lui envoyer de tels caractères, et répondez N à la question.

Phase 4 :

Le quatrième message affiché par la commande SETUP est le suivant :

No keyboard translation set

Is this correct (Y/N) : _

Ce message signale qu'aucune touche n'est actuellement redéfinie. Répondez Y à la question si cela vous convient, et N si vous désirez définir une ou plusieurs touches. Dans le deuxième cas, le menu suivant apparaît :

Enter required command from :

A - Add key translation (key number, normal, shift, control)

D - Delete key translation (key number)

C - Clear all translations

F - Finish translations

Command :

La redéfinition d'une touche du clavier suit les mêmes règles que pour la commande BASIC « KEY DEF ».

Ainsi, supposons que vous vouliez affecter à la lettre W majuscule (code 59) la lettre Z majuscule (code 90), à la lettre w minuscule la lettre z minuscule (code 122) et au code de contrôle ^W le code de contrôle ^Z (code 26).

Choisissez l'option A et tapez :

Command : A 59,122,90,26

Reportez-vous à la commande BASIC « KEY DEF » pour avoir plus de détails sur la manière dont est codée la commande CP/M équivalente (voir partie 4, chap. 1.2, p. 21).

L'option D permet d'effacer une affectation de la manière suivante :

— Si vous voulez effacer la redéfinition précédente concernant la touche W de code 59, tapez :

Command : D 59.

— L'option C efface toutes les redéfinitions de touches.

— L'option F permet de passer à la phase suivante.

Phase 5 :

Le cinquième message affiché par la commande SETUP est le suivant :

No keyboard expansions set

Is this correct (Y/N) :

Ce message vous signale qu'aucune touche de fonction n'est actuellement dans le secteur de configuration de CP/M.

Répondez Y si cela vous convient. Répondez N si vous désirez définir une ou plusieurs touche(s) de fonction.

Dans ce cas, reportez-vous à la commande BASIC équivalente « KEY » pour avoir la liste des touches programmables (voir partie 4, chap. 1.2, p. 21).

Si vous avez répondu N à la question précédente, un menu du même genre que celui décrit dans la phase 4 apparaît et vous pouvez définir une ou plusieurs touche(s) de fonction.

Remarque :

Terminez la définition d'une touche par le code ^M si vous désirez que la commande soit effectivement exécutée. (^M est le code de contrôle équivalent à ENTER.)

Phase 6 :

Le sixième message affiché par la commande SETUP est le suivant :

Default IO byte settings are
 CON : is assigned to CRT: (keyboard and VDU)
 RDR : is assigned to TTY: (special IO device 0)
 PUN : is assigned to TTY: (special IO device 0)
 LST : is assigned to LPT : (centronics printer)
 Is this correct (Y/N) :

Cette phase permet d'associer un périphérique logique (le nom que l'on donne à un appareil) à un périphérique physique (l'appareil). Reportez-vous à la commande STAT pour avoir plus de détails concernant les périphériques physiques et logiques possibles (voir p. 13). Répondez Y si cette affectation vous convient, et N si vous désirez modifier ou créer une affectation.

Le message suivant apparaît alors :

Enter required IO byte setting :

vous invitant à entrer le périphérique logique suivi du périphérique physique. Par exemple :

Enter required IO byte setting : PUN : TTY :

Phase 7 :

Le septième message affiché par la commande SETUP est le suivant :

Default : slow mode set
 Is this correct (Y/N) :

Cette option permet de définir si les registres secondaires du Z80 seront sauvegardés lorsque CP/M accédera aux ROM (mode lent) ou non (mode rapide). Par sécurité, nous vous conseillons de choisir le mode lent, d'autant plus qu'il ne ralentit pratiquement pas CP/M.

Phase 8 :

Le huitième message affiché par la commande SETUP est le suivant :

Default : BIOS message disabled

Is this correct (Y/N) :

Cette option permet d'afficher ou non les messages d'erreur du CP/M.

Phase 9 :

Le neuvième message affiché par la commande SETUP est le suivant :

Default : Clear initial command buffer on keyboard input

Is this correct (Y/N) :

Cette option permet d'effacer par CP/M (ou de ne pas effacer : Preserve initial command buffer on keyboard unit) le buffer de commande.

Phase 10 :

Le dixième message affiché par la commande SETUP est le suivant :

Default motor on delay is 50 1/50 seconds units

Is this correct (Y/N) :

Cette option permet de régler le temps maximum d'accès au disque en lecture. Si vous répondez Y(es), le message suivant apparaîtra :

Enter new motor on delay in 1/50 seconds units :

Phase 11 :

Le onzième message affiché par la commande SETUP est le suivant :

Default motor off delay is 250 1/50 seconds units

Is this correct (Y/N) :

Cette option permet de régler le temps maximum d'accès au disque en écriture. Si vous répondez Y(es), le message suivant apparaîtra :

Enter new motor off delay in 1/50 seconds units :

Phase 12 :

Le douzième message affiché par la commande SETUP est le suivant :

Default stepping rate is 12 milliseconds

Is this correct (Y/N) :

Cette option permet de régler le délai de mise en route et d'arrêt du lecteur de disquettes. Si vous répondez Y(es), le message suivant apparaîtra :

Enter new stepping rate in milliseconds :

Phase 13 :

Le treizième message affiché par la commande SETUP est le suivant :

Z80 SIO channel A: 9600 tx baudrate, 9600 rx baudrate, 8 data bits

No parity, 1 stop bit

Is this correct (Y/N) :

Cette option permet de modifier le protocole de communication en émission et réception pour le port de communication série RS-232, canal A.

Si vous désirez modifier ces paramètres, répondez N à la question et donnez dans l'ordre :

— vitesse d'écriture :

Une des valeurs suivantes : 50, 75, 110, 150, 200, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 9600 ou 19600 bauds.

— vitesse de lecture :

Mêmes valeurs possibles que la vitesse d'écriture.

— nombre de bits de données :

Une des valeurs suivantes : 5, 6, 7 ou 8.

— parité :

Une des valeurs suivantes : ODD (impaire), EVEN (paire) ou NONE (pas de parité).

— nombre de bits stop :

Une des valeurs suivantes : 1, 1.5 ou 2.

Phase 14 :

Identique à la phase 13 mais pour le canal B :

Phase 15 :

Les quatorze phases étant initialisées, le message suivant apparaît :

Do you want to update your system disc (Y/N) :

Si vous répondez Y, les options définies seront enregistrées sur la disquette CP/M. Elles seront perdues dans le cas contraire.

Enfin, si vous avez répondu Y(es), le message suivant apparaît, vous permettant de faire un démarrage à froid de CP/M afin de tester les nouvelles options que vous venez de définir :

Do you want to restart CP/M (Y/N) :

SYSGEN

Format :

SYSGEN ou SYSGEN * ou SYSGEN <Nom de fichier>

Fonction :

Permet de copier le système CP/M d'une disquette sur une autre.

Utilisation :

Si aucune option n'est précisée, le système est copié à partir d'une disquette CP/M normale. Si l'option * est précisée, le système est copié à partir d'une disquette CP/M modifiée par MOVCPM. Enfin, si un nom de fichier est précisé, c'est uniquement ce fichier qui sera copié.

Tapez la commande « SYSGEN » avec ou sans argument.

Le message suivant apparaît :

SYSGEN V2.0

Please insert SOURCE disc into drive A then press any key:

Insérez la disquette contenant le système à copier en A et appuyez sur une touche.

La configuration de CP/M est chargée en TPA, puis le message suivant apparaît :

Please insert DESTINATION disc into drive A then press any key:

Insérez la disquette sur laquelle doit s'effectuer la copie, puis tapez une touche quelconque.

La disquette est configurée puis le message suivant apparaît :

Do you wish to reconfigure another disc (Y/N) :

Si vous désirez copier une nouvelle configuration CP/M, répondez Y, sinon répondez N, puis insérez une disquette système pour retourner sous CP/M.

Remarque :

La disquette sur laquelle CP/M va être copié doit avoir été formatée en système (option S ou V).

TYPE

Format

TYPE[<drive>:]<Nom de fichier>

Fonction

Affiche à l'écran le contenu d'un fichier.

Utilisation

Entrez la commande TYPE suivie du nom du fichier à visualiser.

Remarques :

a) L'affichage peut être provisoirement interrompu par l'appui simultané des touches CONTROL et S et peut reprendre par l'appui sur une touche quelconque.

b) L'affichage peut être définitivement interrompu par l'appui sur une touche quelconque (sauf ^S).

c) Si le fichier listé contient des caractères de tabulation, ils sont automatiquement étendus à 8 espaces.

d) Le fichier peut être listé sur imprimante en même temps que sur l'écran. Pour cela, tapez ^P (CONTROL et P simultanément) avant d'entrer la commande TYPE.

XSUB

Format

XSUB.

Fonction

Permet de lancer des commandes à un ou plusieurs programme(s) activé(s) par une commande SUBMIT.

Utilisation

Entrez cette commande sous CP/M ou incluez-la dans un fichier SUBMIT. Dès qu'elle est exécutée, cette commande reste valide jusqu'au prochain démarrage à froid.

Exemple :

Vous voulez lancer le programme de mise au point DDT avec l'argument ENCOURS, et afficher le contenu des registres. Si vous devez répéter plusieurs fois cette opération, il sera judicieux d'incorporer les commandes suivantes dans un fichier SUBMIT :

```
XSUB  
DDT ENCOURS  
X
```

III. Ordres évolués

AMSDOS

Format : AMSDOS.

Fonction :

Permet de revenir sous BASIC et AMSDOS.

L'écran affiche : **BASIC 1.1 Ready.**

CHKDISC

Format

CHKDISC.

Fonction

Vérification de la similitude de deux disquettes sur un AMSTRAD possédant deux lecteurs de disquettes.

Utilisation

Insérez la disquette CP/M, et tapez CHKDISC. Le message suivant apparaît :

Please insert source disc into drive A and destination disc into drive B then press any key:

Insérez le disque source dans le lecteur A: et le disque à vérifier dans le lecteur B:.

Pressez une touche quelconque.

Le message « **Copy checking complete** » vous indique que la vérification a été faite et que les deux disques sont identiques.

Le message « **WARNING : Failed to compare disc correctly** » vous indique que la comparaison a été interrompue car au moins un octet est différent entre les disquettes source et destination.

DISKIT2

Format

DISKIT2.

Fonction

Utilitaire disque de haut niveau permettant de :

→ copier,

Utilisation

- formater,
- vérifier un disque.

Tapez DISCKIT2.

Tapez F7 pour copier un disque sur un autre.

Divers messages vous indiqueront d'insérer successivement les disques SOURCE et DESTINATION, vous informeront que le disque DESTINATION n'est pas formaté et qu'il le sera pendant la copie, ou encore qu'un secteur du disque est défectueux et que la copie ne peut être poursuivie sans risque de défectuosité.

Tapez F4 pour formater un disque vierge ou reformater un disque déjà utilisé. Cette phase doit précéder toute utilisation de disque vierge. Le disque peut être formaté en :

- système (pistes 0 et 1 réservées pour recevoir CP/M),
- données (disquette ne pouvant pas recevoir un CP/M),
- vendeur (identique au format système, mais les pistes 0 et 1 ne sont pas initialisées).

Tapez F1 pour vérifier qu'une disquette copiée par l'option F7 est bien conforme à la disquette SOURCE.

Le message suivant vous indique que les disquettes sont identiques :

Verify completed
Remove disc
Press any key to continue

Par contre, le message suivant vous indique qu'une anomalie a été rencontrée :

Disc not completely verified
Remove disc
Press any key to continue

ED**Format**

ED [<drive>:]<Nom de fichier>[<drive>]

Fonction

Editeur de lignes. Permet la saisie d'un texte quelconque, ligne par ligne, Par exemple, vous pourrez saisir un programme source en assembleur ou un programme batch activé par SUBMIT.

Utilisation

Reportez-vous au même mot-clé partie 3 chapitre 4 page 27 où les diverses commandes sont commentées.

SUBMIT**Format**

SUBMIT [<drive>:]<Nom de fichier>[<param 1>[<param 2>...[<param n>]]]

Fonction Active un fichier « batch », c'est-à-dire un fichier contenant des commandes CP/M ou/et des commentaires qui auraient pu être tapés au clavier.

Utilisation Si vous utilisez des paramètres (9 au maximum), ceux-ci doivent être repérés par \$1 à \$9 dans le fichier batch. Le paramètre 1 remplacera \$1, ..., le paramètre 9 remplacera \$9.

Exemple :

Vous voulez créer un programme source assembleur sous ED, puis le compiler sous ASM. Le programme « submit » suivant qui comporte un paramètre réalisera ces actions :

ED \$1
ASM \$1

Supposons que le fichier que vous voulez créer porte le nom de « ESSAI » et que le programme submit s'appelle SCOM. SUB.

Pour lancer le programme submit, il faudra faire :

SUBMIT SCOM ESSAI.

3/3.1

Liste alphabétique des mots-clés

mots-clés		numéro de page
AMSDOS	Retour sous AMSDOS	24
BOOTGEN	Copie la liste 0, secteur 49	15
CHKDISK	Vérification de la similitude de 2 disquettes si 2 lecteurs	24
CLOAD	Copie cassette → disquette	3
COPYDISC	Copie disquette → disquette si 1 lecteur	4
CSAVE	Copie disquette → cassette	4
DIR	Répertoire d'une cassette ou d'une disquette	5
DISCCOPY	Copie disquette → disquette si 2 lecteurs	6
DISCCHK	Vérification d'une disquette copiée par COPYDISC	6
DISCKIT2	Utilitaire disque	24
DUMP	Affiche le contenu d'un fichier en hexadécimal	15
ED	Editeur de lignes	25
ERA	Effacement d'un fichier sur disquette	7
FILECOPY	Copie d'un fichier sur disquette	8
FORMAT	Formatage d'une disquette	9
LOAD	Transforme un fichier .HEX en .COM	10
MOVCPM	Déplacement de CP/M	16
PIP	Transfert entre deux périphériques	10
REN	Donne un nouveau nom à un fichier	16
SAVE	Sauvegarde de n pages sur disquette	12
SETUP	Modifie la configuration des périphériques	17
STAT	Donne des informations sur une disquette ou un fichier	13
SUBMIT	Lancement d'un fichier batch	25
SYSGEN	Copie CP/M d'une disquette sur une autre	22
TYPE	Affiche le contenu d'un fichier	23
USER	Définit le numéro d'utilisateur courant	14
XSUB	Simule l'entrée au clavier de paramètres	23

3/4

CP/M Plus : définitions, rappels et utilisations

Cette partie concerne uniquement les possesseurs de CPC 6128. En effet, CP/M + est écrit pour exploiter un ordinateur de plus de 64 kilo-octets de RAM comme l'AMSTRAD CPC 6128.

HISTORIQUE

Comme nous l'avons vu précédemment, (voir partie 3, chap. 1) un système d'exploitation est un ensemble de sous-programmes destinés à :

- faciliter les entrées/sorties sur disquettes,
- faciliter les accès aux périphériques.

Après la conception du premier système d'exploitation CP/M en 1973 par Gary Killdall, Digital Research fut fondé en 1976 et CP/M commercialisé.

Depuis, de nombreuses versions ont vu le jour. Dernière en date, CP/M+ (aussi appelé CP/M 3.1) vous permettra de :

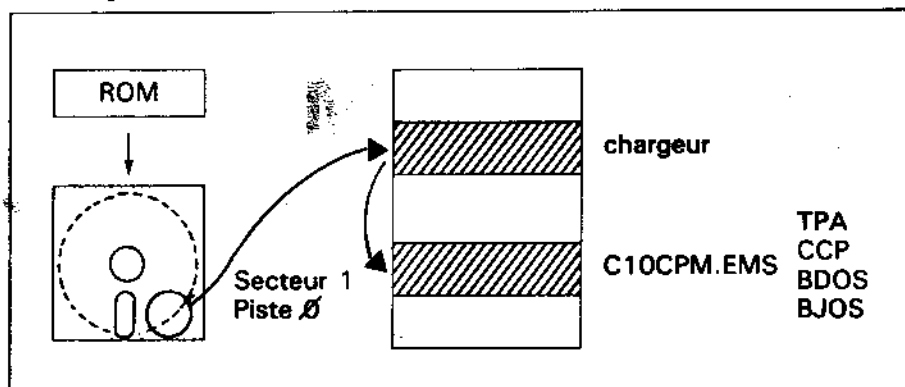
- gérer les ressources usuelles de votre machine comme le faisaient ses prédécesseurs (mémoire, disque, écran, clavier) ;
- grâce à plusieurs commandes additionnelles, vous pourrez facilement redéfinir le clavier, faire des copies de disquettes ou encore choisir les paramètres caractéristiques de l'interface de communication.

DÉFINITIONS ET RAPPELS

Pour pouvoir fonctionner sous CP/M+, le CPC 6128 est pourvu d'un programme en ROM qui effectue le chargement en RAM de la piste 0 du premier secteur de la disquette CP/M+.

Ce secteur contient un programme qui est exécuté et qui provoque le chargement en RAM du fichier « C10CPM3.EMS ».

Ce chargement s'effectue quand l'utilisateur tape la commande « :CPM ».



Le programme « C10CPM3.EMS » contient :

- le **TPA** (Transcient Program Area) : Zone de programme temporaire,
- le **CCP** (Console Command Processor) : Interpréteur de commandes entrées au clavier,
- le **BDOS** (Basic Disk Operating System) : Système d'exploitation de base,
- le **BIOS** (Basic Input Output System) : Système d'entrées sorties de base.

TPA : C'est la zone RAM des programmes utilisateur ou système. Elle sera utilisée pour stocker le ou les programme(s) à exécuter. Sur CPC 6128, cette zone fait un peu plus de 61 kilo-octets.

CCP : Cette zone analyse les ordres entrés au clavier ou issus des programmes différés (batch) de type « SUBMIT ». Elle contient les commandes résidentes (voir définition) du CP/M+ et charge dans la zone TPA les commandes transitoires (voir définition) lorsque cela est nécessaire.

BDOS : Système de gestion des fichiers disquette. Ce système contient, par exemple, les routines de conversion des fichiers à lire/écrire en pistes et secteurs du disque.

BIOS : Contient les « drivers » des équipements périphériques à piloter.

Définition :

Une commande est dite *résidente* quand elle est toujours présente en mémoire RAM ou ROM.

Par opposition, une commande est dite *transitoire* quand elle est présente sur disquette sous la forme d'un fichier d'extension « .COM » qui est chargé en RAM en zone « TPA » pour être exécuté.

Une fois chargé le programme « C10CPM3.EMS », le DOS cherche le programme « PROFILE.SUB » et l'exécute s'il est présent. Ce programme

est un fichier de commandes « batch », c'est-à-dire qu'il contient une ou plusieurs commandes CP/M+ (écrites sous traitement de textes, par exemple) qui s'exécuteront séquentiellement comme si l'utilisateur les rentrait au clavier en mode direct. Ce fichier « auto-exec » permet de personnaliser chaque disquette en exécutant directement par la commande « :CPM » un des programmes qui s'y trouvent. Pour avoir plus de détails sur le fichier « PROFILE.SUB », reportez-vous à la description de l'ordre CP/M+ « SUBMIT » (voir p. 29 de ce chapitre).

Dans la suite, nous allons analyser la ou les utilisation(s) possible(s) de chaque ordre CP/M+.

MOTS-CLÉS ET LEUR UTILISATION

Nous avons divisé ces fonctions en quatre grands groupes :

- I. Entrées/Sorties sur disque :**
DIR, DIRS, ERA, INITDIR, PIP, REN, SET, SETDEF, SHOW et USER.
- II. Entrées/Sorties sur périphériques :**
DEVICE, SETSIO, SETLST.
- III. Utilitaires :**
Ecran : DATE, LANGUAGE, PALETTE, SETDEF, SET24 x 80, TYPE.
Clavier : SETKEYS.
Autres : GET, PAPER, PUT.
- IV. Ordres évolués :**
DISKIT3, ED, HELP, SUBMIT.

Conventions d'écriture :

Dans la suite, nous utiliserons les signes ou écritures suivants pour décrire les ordres CP/M+ :

- ()** Encadre une ou plusieurs déclaration(s) optionnelle(s).
- |** Soit l'un, soit l'autre.
- <cr>** Carriage Return.
- ^** Caractère de contrôle. Par exemple ^A = CTRL + A.
- n** Substitution du paramètre par un entier.
- s ou ch** Substitution du paramètre par une chaîne alphanumérique.
- []** Encadre une ou plusieurs option(s).
- ()** Plage finie de valeurs possibles.
- RW** Attribut « Read/Write ».
- RO** Attribut « Read Only ».
- SYS** Attribut « SYSteme » à opposer à « DIR ».
- DIR** Attribut « DIRectory ».
- *** Remplace un groupe de lettres par une chaîne quelconque.
- ?** Remplace une lettre par une autre quelconque.

I. Entrées/Sorties sur disque

DIR

Listage du répertoire des fichiers d'une disquette.

S'utilise de deux manières : avec ou sans options.

Les arguments de cette commande peuvent être :

- Aucun : Donne la liste de tous les fichiers de l'unité et de l'USER par défaut.
- Lettre du lecteur : Exemple « DIR A: » donne la liste de tous les fichiers de l'unité spécifiée sous l'USER courant et retourne sous le prompt courant.
- Utilisation du « joker » possible :

Par exemple :

DIR *.BAS Affiche tous les fichiers d'extension « .BAS ».

DIR D.* Affiche tous les fichiers de nom « D » et d'extension quelconque.

DIR D*.* Affiche tous les fichiers de nom commençant par « D » et d'extension quelconque.

- Utilisation du signe « ? » :

Le signe « ? » remplace un caractère alphanumérique quelconque.

La recherche est faite sans considérer ce caractère, mais en tenant compte du fait qu'il occupe une place.

Exemples :

DIR SET?ST.COM donnera tous les programmes de 6 lettres commençant par « SET », finissant par « ST » et d'extension « .COM », en l'occurrence « SETLST.COM » sur la disquette CP/M + face 1.

« ? » peut être combiné avec le joker :

DIR S?T*.* donnera la liste des programmes dont la première lettre est « S », la troisième lettre est « T », de longueur quelconque et d'extension quelconque.

- Utilisation d'options :

DIR (d:) [OPTIONS] ou
DIR (Fichier)(Fichier)... [OPTIONS]

Les options peuvent être les suivantes :

ATT	: Affiche les attributs des fichiers (DIR, SYS, RO, RW)
DATE	: Affiche la date et l'heure sur les fichiers datés
DIR	: Affiche les fichiers d'extension « .DIR »
DRIVE=ALL	: Affiche les fichiers de tous les lecteurs connectés
DRIVE=(A, B, ...N)	: Affiche les fichiers des lecteurs spécifiés

Format

DRIVE:d	: Affiche les fichiers du lecteur spécifié
EXCLUDE	: Affiche tous les fichiers sauf ceux spécifiés
FF	: Envoie un saut de page (« form feed ») à l'imprimante
FULL	: Affiche toutes les informations possédées sur les fichiers du lecteur courant
LENGTH=n	: Un en-tête est inséré après chaque n lignes, n compris entre 5 et 65536.
MESSAGE	: Affiche le nom des lecteurs et l'« USER » actif
NO SORT	: Affiche les fichiers dans l'ordre trouvé sur le disque
RO	: Affiche les fichiers à lecture seule
RW	: Affiche les fichiers à lecture/écriture
SIZE	: Affiche la taille des fichiers en KO (kilo-Octets)
SYS	: Affiche les fichiers d'extension « .SYS »
USER=ALL	: Affiche les fichiers de tous les « USER » du disque courant
USER-n	: Affiche les fichiers de l'« USER » spécifié sur le disque courant
USER=(0,1..15)	: Affiche les fichiers des « USERS » spécifiés sur le disque courant.

DIRS ou DIRSYS

Affiche le répertoire des fichiers d'une disquette possédant l'extension « .SYS ». Cet ordre est équivalent à « DIR*.SYS ». DIRS peut être utilisé comme les jockers et « ? ». Par exemple, « DIRS B* » listera l'ensemble des fichiers dont le nom commence par « B » et d'extension « .SYS ».

ERA ou ERASE

Permet d'effacer un fichier ou un ensemble de fichiers stocké(s) sur disque. Cette commande supporte les jockers et les « ? », mais la commande « ERA *.* », qui va détruire tous les fichiers de la disquette, demande deux confirmations avant d'être exécutée : la première qui est générale à l'utilisation du jocker, (« ERASE *.* (Y/N) ») et la seconde qui est particulière à l'utilisation du double jocker (« Confirm delete all user files (Y/N) ? »).

Remarques :

a) Les fichiers protégés (à lecture seule : voir commande « SET ») ne sont pas effacés. Si vous tentez d'effacer un fichier d'attribut « RO » (lecture seule), le message suivant apparaîtra : « Fichier.ext Not erased, Read Only » et la commande « ERA » sera avortée.

b) Si vous demandez d'effacer un fichier ou un groupe de fichiers qui n'existe pas, le message suivant sera affiché « No file » et la commande « ERA » sera avortée.

c) L'utilisation du joker tant pour le nom que pour l'extension provoquera l'affichage d'un message de confirmation :

Exemple : « ERA*.COM » provoquera l'affichage de « ERASE *.COM (Y/N) », et « ERA DATE.* » provoquera l'affichage de « ERASE DATE.* (Y/N) ».

d) Si vous tentez d'effacer un fichier ou un groupe de fichiers alors que le disque n'est pas prêt (cordon d'alimentation débranché, disquette sortie) ou hors service, un bip sonore signalera le problème, et un message défilera sur la 25^e ligne de l'écran : « drive not ready - Retry, Ignore or Cancel ? ».

Les options « Retry » et « Ignore » essayeront d'accéder à nouveau à la disquette, alors que l'option « Cancel » affichera le message d'erreur « CP/M error on A: Disk I/O »

« BDOS Function = 20 File = ERASE .COM »
et l'accès disque ne sera pas réitéré.

INITDIR

« Marque » une disquette pour que les créations, accès ou mises à jour de fichiers soient datés sur le directory.

Remarques :

a) Les fichiers datés sont listables par plusieurs options de la commande « DIR ». Pour plus de détails, reportez-vous à cette commande (voir p. 5).

b) INITDIR s'utilise avec les commandes « SET[CREATE=ON/OFF] », « SET[ACCESS=ON/OFF] » et « SET[UPDATE=ON/OFF] ».

Reportez-vous à ces commandes pour avoir plus de détails (voir p. 17).

Exemples d'utilisation :

Si vous tentez d'affecter « CREATE=ON », « ACCESS=ON » ou « UPDATE=ON » à la commande « SET » sans que « INITDIR » ait été fait, vous obtiendrez le message suivant :

set [update=on]

ERROR : Directory needs to be re-formatted for time/date stamps.

(ERREUR : le répertoire doit être reformaté pour pouvoir y inclure les informations d'heure et de date.)¹⁾

Please see INITDIR.

(reportez-vous à INITDIR)

Pour valider les options de marquage, tapez :

A>initdir a:

INITDIR WILL ACTIVATE TIME STAMPS FOR SPECIFIED DRIVE.

(INITDIR va valider les informations relatives aux fichiers sur le disque spécifié.)

¹⁾ Chaque message apparaissant à l'écran est traduit en français.

Do you want to re-format the directory on drive : A (Y/N) ? y
 (Voulez-vous reformater le répertoire sur le disque A)

Pour valider (par exemple) l'écriture de la date à chaque remise à jour des fichiers disques, tapez :

A>set[update = on]

Label for drive A:

(étiquette)

Directory (répertoire)	Passwds (mot de passe)	Stamp (marque de création)	Stamp (marque d'accès)	Stamp (marque de mise à jour)
Label	Reqd	Create	Access	Update
A:LABEL	off	off	off	on

La demande de l'option [DATE] sur un « DIR » produira l'en-tête suivant :

a>dir[date]

Scanning Directory...

(Répertoire en cours de balayage)

Sorting Directory...

(Répertoire en cours de tri)

Directory For Drive A: User 0

(Répertoire du disque A : Utilisateur 0)

Name (Nom)	Bytes (Octets)	Recs (Enregistrement)	Attributes (Attributs)	Prot (Protection)	Update (Mise à jour)	Create (Création)
---------------	-------------------	--------------------------	---------------------------	----------------------	-------------------------	----------------------

Il est également possible d'effacer les attributs « UPDATE » et « CREATE » des fichiers du disque en faisant :

A>initdir a:

INITDIR WILL ACTIVATE TIME STAMPS FOR SPECIFIED DRIVE.

Do you want to re-format the directory on drive: A (Y/N) ? y

Directory already re-formatted.

Do you want to recover time/date directory space (Y/N) ? n

Do you want the existing time stamps cleared (Y/N) ? y

(Voulez-vous effacer les marquages existants ?)

Pour dévalider l'affichage des options de marquage, tapez :

INITDIR WILL ACTIVATE TIME STAMPS FOR SPECIFIED DRIVE.

Do you want to re-format the directory on drive: A (Y/N) ? y

Directory already re-formatted.

(Répertoire déjà reformaté)

Do you want to recover time/date directory space (Y/N) ? y

(Voulez-vous conserver les anciens marquages ?)

La demande de l'option « DATE » sur un « DIR » produira alors :

A>dir[date]

Scanning Directory...

Sorting Directory...

ERROR : Date and Time Stamping Inactive

(Marquages de date et heure inactifs)

Enfin, si vous oubliez de spécifier le nom du lecteur sur lequel vous voulez faire l'« INITDIR », le message suivant apparaîtra :

initdir

ERROR: Unrecognized drive

(lecteur inconnu)

DRIVE: A

Enter Driver: a

(Nom du lecteur)

PIP

Peripheral Interchange Program = Programme d'échange entre périphériques.

PIP [<drive>:]<Destination> = [<drive>:] <Source>

où PIP <Destination> est un nom de fichier ou un organe de sortie tel que CONsole, AUXiliaire ou LST (imprimante).

<Source> est un nom de fichier ou un organe d'entrée tel que CONsole, AUXiliaire ou EOF (marque la fin d'un fichier).

PIP B:=A:*;* copie tous les fichiers de A: sur B:

PIP B:=A: MOVCPM.COM copie le fichier MOVCPM de A: sur B:

PIP TOTAL.ASM=PA1.ASM,PA2.ASM,B:PA3.ASM concatène les fichiers PA1, PA2 et PA3 qui se trouve sur B: dans le fichier TOTAL.ASM qui sera stocké sur A:

PIP LST:=PROG.ASM copie le fichier PROG.ASM sur l'imprimante.

PIP CON:=PROG.ASM copie le fichier PROG.ASM sur l'écran.

PIP PROFILE.SUB=CON : fabrique le fichier PROFILE.SUB à partir d'une saisie au clavier. Pour passer à la ligne, il faudra taper « CTRL + J ». Quand la saisie sera terminée, tapez « CTRL + Z ».

Remarques :

a) La commande « PIP » s'utilise de deux manières différentes :

— En direct comme nous venons de le voir, en tapant « PIP <Commande> ». Les fichiers disque manipulés doivent être présents sur la ou les disquettes spécifiées.

Format

Utilisation

— En chargeant « PIP » en zone TPA (Transcient Program Area) : (voir chap. 3 de cette partie pour plus de détails). Pour cela, tapez « PIP ».

Un astérisque apparaît pour signaler que PIP est en mémoire et attend vos instructions. Ce procédé a pour avantage de permettre un changement de disquette courante. En effet, le programme que l'on désire transférer ne se trouve pas forcément sur la disquette système où réside la commande « PIP ».

Pour sortir de l'utilitaire PIP, tapez <cr> derrière le « * ».

b) La phase de copie peut être interrompue à tout moment en appuyant sur une touche quelconque du clavier. De même, s'il se produit une erreur pendant le transfert, la commande PIP sera annulée, et un prompt indiquera que l'on est revenu sur l'unité par défaut.

Les options suivantes peuvent être employées avec PIP :

- A : Archive : Copie uniquement les fichiers dont au moins un octet a changé depuis la dernière sauvegarde.
- C : Confirm : Demande la confirmation pour chaque fichier copie.
- Dn : Efface tous les caractères occupant une colonne > n.
- E : Echo transféré à la visu.
- F : Efface les « form feed » du fichier copie ; utile pour éviter les pertes de papier lorsque la sortie se fait sur imprimante.
- Gn : Utilisation d'un USER différent de l'USER courant.
- K : Interdiction des commentaires à la visu.
- L : Met en minuscules les caractères du fichier manipulé.
- N : Numérote les lignes du fichier transféré.
- Pn et à et ne : Longueur de la page : n lignes.
- Qs^Z : Arrête la copie sur le caractère « s ».
- R : Lit les fichiers d'extension « .SYS ».
- Ss^Z : Commence la copie à partir du caractère « s ».
- Tn : Les tabulations sont étendues à n espaces.
- U : Met en majuscules les caractères du fichier manipulé.
- V : Vérification de la sauvegarde.
- W : Permet d'écrire sur un fichier protégé sans qu'une confirmation soit demandée.
- Z : Les caractères copies ont leur bit du parité à 0.

Exemple :

PIP SET.COM[g3] = SET.COM copie le fichier « SET.COM » présent sur l'USER 0 sur l'USER 3 en lui donnant le même nom.

REN ou RENAME

REN <Ancien nom> = <Nouveau nom>

Cette commande permet de changer le nom d'un fichier ou d'un groupe de fichiers existant sur disque. Les jokers et « ? » sont autorisés dans le préfixe. Par exemple, « REN *.TY1 = *.TY2 » qui renommera tous les fichiers d'extension « .TY2 » en « .TY1 ».

Format

Partie 3 : Systèmes d'exploitation des 664 et 6128

Si le fichier à renommer n'existe pas, le message « Not found » sera affiché. De même, si le nouveau nom correspond à un fichier déjà présent sur la disquette, le message « File exists » sera affiché.

Remarques :

a) Un programme peut être protégé contre le changement de nom s'il possède l'attribut adéquat (RO = Read Only). Le message suivant apparaîtra alors : « Bdos Error A : File R/O » et le fichier ne sera pas renommé. Reportez-vous à l'ordre « SET » pour modifier l'attribut d'un fichier.

b) RENAME peut être utilisé de deux manières différentes :

— En direct comme nous venons de le voir en tapant
REN <New> = <Old>. Le fichier manipulé doit être présent sur la disquette où se trouve RENAME.

— En chargeant RENAME en TPA (Transcient Program Area : voir définitions et rappels, (Partie 3, chap. 3). Pour cela, tapez « RENAME » ou « REN ». Mettez la disquette qui contient le fichier à manipuler et répondez aux questions posées. Par exemple, si vous voulez modifier le nom du programme « SET.COM » en « SETT.COM », tapez :

A>rename

Enter New Name: sett.com

(Entrez le nouveau nom)

Enter Old Name: set.com

(Entrez l'ancien nom)

Cette commande est équivalente à la suivante si « SET.COM » est sur la même disquette que « RENAME » :

A>rename sett.com=set. com

c) La désignation d'un fichier qui n'est pas présent sur la disquette provoquera l'apparition du message « Error : No such file to rename » et l'abandon du RENAME.

SET

Modification des attributs d'un fichier.

La commande SET peut utiliser les extensions suivantes :

SET[ATTRIBUTES]

SET[PASSWORD]

SET[DEFAULT]

SET[TIME-STAMPS]

SET[DRIVES]

SET[ATTRIBUTES] Les attributs peuvent être :

RO : Lecture seule permise,

RW : Lecture et écriture permises,

SYS : Force l'extension à « .SYS »,
 DIR : Force le fichier à être listable par la commande « DIR »,
 ARCHIVE=ON/OFF : Voir l'ordre « PIP »,
 F1 à F4=ON/OFF : Valide ou dévalide l'attribut de fichier utilisateur F1 à F4.

SET[PASSWORD] Les options sont les suivantes :

SET[PROTECT=ON/OFF] permet l'activation des options de protection pour tous les fichiers disque
A>set[access=on]

Label for drive A:

Directory Label	Passwds Reqd	Stamp Create	Stamp Access	Stamp Update
A:LABEL	off	off	on	off

SET Nom Fichier[PASSWORD = Mot de passe]
 Par exemple SET *.COM[PASSWORD = PCOM] assigne le mot de passe « PCOM » à tous les fichiers d'extension « .COM ».

SET Nom Fichier[PROTECT = READ]
 Un mot de passe sera demandé pour lire, écrire, copier, effacer ou renommer un tel fichier.

SET Nom Fichier[PROTECT = WRITE]
 Un mot de passe sera demandé pour écrire, effacer ou renommer un tel fichier.

SET Nom Fichier[PROTECT = DELETE]
 Un mot de passe sera demandé pour effacer ou renommer un tel fichier.

SET Nom Fichier[PROTECT = NONE]
 Aucun mot de passe n'est nécessaire. Si un mot de passe a été créé par une précédente option « SET », cet ordre le détruit.

SET[DEFAULT = dd] Demande à l'ordinateur d'utiliser le mot de passe « dd » par défaut si aucun mot de passe n'est entré lors de l'accès (lecture, écriture, copie, effacement et renommage) à un fichier.

Exemple :

A>set[default = secret]

Default password = SECRET
 (Mot de passe par défaut)

SET[TIME-STAMPS] Les options possibles sont les suivantes :

SET[CREATE=ON] : A la création d'un fichier, la date sera enregistrée et apparaîtra si vous demandez un « DIR[FULL] ».

SET[ACCESS=ON] : Après chaque accès à un fichier, la date sera enregistrée et apparaîtra si vous demandez un « DIR[FULL] ».

SET[UPDATE=ON] : A chaque mise à jour d'un fichier, la date sera enregistrée et apparaîtra si vous demandez un « DIR[FULL] ».

Exemple :

La demande de prise en compte de l'attribut « ACCESS » se fait comme suit :

A> set[access = on]

Label for drive A:

Directory Label	Passwds Reqd	Stamp Create	Stamp Access	Stamp Update
A:LABEL	off	off	on	off

Utilisation

SET[DRIVES] SET d:[RO] donne une protection logique au disque et empêche toute écriture dessus.

set a: [ro]

Drive A: set to Read only (RO)

(Lecteur A positionné en lecture seule).

Si vous essayez d'écrire sur le disque (par exemple, par « RENAME »), le message suivant apparaîtra :

A> rename sett.com = set.com

ERROR: Drive read only.

(Lecture seule autorisée)

SET d:[RW] autorise lecture et écriture sur disque.

A> set a: [rw]

Drive A: set to Read Write (RW)

(Lecteur A positionné en lecture/écriture)

SHOW**Format**

8

SHOW (d:) ((SPACE:LABEL:USERS:DIR:DRIVE)) donne les informations suivantes sur le disque :

- Mode d'accès (RO ou RW) et espace libre sur le disque,
- Label du disque,
- « USER » courant,
- Nombre de fichiers existant pour chaque « USER »,
- Nombre de directories libres sur le disque,
- Caractéristiques du disque.

Les options sont les suivantes :

SHOW
ou **SHOW[SPACE]** Donne l'espace libre et le mode d'accès aux drives loggés.

A>show

A: RW, Space: 23k

(Espace restant sur le disque A d'attribut lecture/écriture)

SHOW[B:] Identique à **SHOW** mais sur le drive B: seulement.

SHOW[A:LABEL] Donne le label du drive A:

A>show[label]

Label for drive A:

Directory Label	Passwds Reqd	Stamp Create	Stamp Update	Label Created	Label Updated
LABEL	off	on	off	12/15/82 00:02	12/15/82 00:03

SHOW[USER] Donne le numéro d'« USER » courant, tous les « USERS » utilisés sur le lecteur A: et le nombre de fichiers assignés à chaque fichier.

A>show[user]

A: Active User : 0

(Utilisateur actif)

A: Active Files : 0

(Fichiers actifs)

A: # of files : 28

(Nombre de fichiers)

A: Number of time/date directory entries : 16

(Nombre de marques sur le répertoire)

A: Number of free directory entries : 18

(Nombre de places libres dans le répertoire)

SHOW[DIR] : Nombre de directories libres sur le disque.

A>show[dir]

A: Number of time/date directory entries : 16

A: Number of free directory entries : 18

SHOW[DRIVE] : Donne les caractéristiques du drive A:

A>show[drive]

A: Drive Characteristics

(Caractéristiques du lecteur)

1,368: 128 Byte Record Capacity

(128 octets par enregistrement)

171: Kilobyte Drive Capacity

(171 kilo-octets sur le lecteur A)

64: 32 Byte Directory Entries

(64 x 32 octets d'entrée)

64: Checked Directory Entries

(64 entrées vérifiées)

128: Records / Directory Entry

(128 enregistrements par entrée)

8: Records / Block

(8 enregistrements par bloc)

36: Sectors / Track

(36 secteurs par piste)

2: Reserved Tracks

(2 pistes réservées)

512: Bytes / Physical Record

(512 octets par enregistrement physique)

USER <n>

Changement de zone utilisateur avec n compris entre 0 et 15. Permet d'affecter sur un même disque des zones de stockage indépendantes pour la commande « DIR » et les diverses commandes d'entrée/sortie disque.

Exemple d'utilisation :

« USER 3 » fera passer l'utilisateur en zone 3. Le nouveau prompt de CP/M + sera « 3A> ».

II. Entrées/Sorties périphériques

DEVICE (NAMES:VALUES:EQUIPT PHYSIQUE:EQUIPT LOGIQUE) ou

DEVICE Eqt log = Eqt phys (option) (.Eq phy (option), ...) ou

DEVICE Eqt log = NULL ou

DEVICE Eqt Phys = (option) ou

DEVICE CONSOLE[PAGE:COLUMNS = Nb colonnes: LINES-Nb lignes]

L'ordre « DEVICE » produit les actions suivantes :

- donne la correspondance entre un équipement logique (sa désignation informatique) et un équipement physique (le périphérique) ;
- définit le protocole de communication et la vitesse d'un périphérique ;
- définit la taille de l'écran.

L'ordre « DEVICE » employé sans arguments donne :

- le nom de code des périphériques ;
- la liste des correspondances Physique/Logique installées, et demande si l'on veut créer une correspondance.

A>device

Physical Devices :
(Equipements physiques)

I = Input, O = Output, S = Serial, X = Xon - Xoff

CRT NONE IO LPT NONE O

Current Assignments :
(Assignations courantes)

CONIN: = CRT

CONOUT: = CRT

AUXIN: = Null Device

AUXOUT: = Null Device

LST: = LPT

Enter new assignment or hit RETURN

(Entrez une nouvelle assignation tapez RETURN)

DEVICE[NAMES] donne le nom des codes des périphériques acceptés par l'ordre « DEVICE ».

A>device[names]

Physical Devices :

I = Input, O = Output, S = Serial, X = Xon - Xoff

CRT NONE IO LPT NONE O

DEVICE[VALUES] donne la correspondance courante entre périphériques logiques et périphériques physiques.

A>device[valuesdevice[values]]

Current Assignments :

CONIN: = CRT

CONOUT: = CRT

AUXIN: = Null Device

AUXOUT: = Null Device

LST: = LPT

DEVICE CRT donne les caractéristiques du périphérique « CRT ».

A>device crt

Physical Device : CRT

(Equipement physique)

Baud Rate : NONE

(Vitesse de communication)

**Characteristics : INPUT
OUTPUT
PARALLEL**

de même,

DEVICE CON donne les caractéristiques du périphérique « CON »**A>device con****CONIN: = CRT****CONOUT: = CRT**

et

DEVICE LPT Donne les caractéristiques du périphérique « LPT »**A>device lpt****Physical Device : LPT****Baud Rate : NONE****Characteristics : OUTPUT
PARALLEL**

Définition des caractéristiques d'un périphérique physique :

Pour délimiter la taille de l'écran à 40 colonnes et 20 lignes, il faudra faire :

A>device console[columns=40 lines=20]**Console width set to 40 columns**

(Largeur de la console : 40 colonnes)

Console page set to 20 lines

(20 lignes par page à la console)

La commande suivante donne les caractéristiques (nombre de lignes et de colonnes) de l'écran.

A>device console[page]**Console width set to 80 columns**

(largeur de la console : 80 colonnes)

Console page set to 25 lines

(25 lignes par page à la console)

Affectation de la sortie écran à l'écran et à l'imprimante :

A>device conout:=lpt, crt**PPhhyysiiiccaall DDDeevviicceess::****II==Ilnnpppuutt,, OO==OOuuttppuutt,, SS==SSeerriiaall,,****XX==XXoonn--XXooffff****CCRRTT NNOONNEE IIOO LLPPTT NNOONNEE OO**

CCuurrrreenntt AAAssssiiggnnmmeennttss::

CCOONNIINN:: == CCRRTT

CCOONNOUUTT:: == CCRRTT LLPPTT

AAUUXXIINN:: == NNUuIII DDeevviicce

AAUXXOOUUTT:: == NNUuIII DDeevviicce

LLSSTT:: == LLPPTT

Notez le dédoublement des caractères qui s'explique par l'utilisation du « hard-copy » d'écran.

Définition du périphérique « AUXOUT » selon le protocole XON/XOFF à une vitesse d'échange de 9600 bauds:

DEVICE AUXOUT(XON,9600)

Remarque :

XON fait référence au protocole de communication XON/XOFF qui ne permet d'envoyer des données sur une ligne que lorsque le périphérique récepteur est prêt. L'option « NOXON » permet d'envoyer les données à la vitesse spécifiée, que le périphérique soit prêt ou non.

Les vitesses possibles sont :

50, 75, 110, 135, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600 et 19200 bauds.

SETSIO

Gestion d'une interface série monocanal.

SETSIO (Bauds:Bits:Stop:Parite:XON:Handshake)

Remarque :

SIO signifie « Serial Input Output » ou, en français : Interface d'entrée/sortie série.

Le paramètre « Baud » caractérise la vitesse de transmission en bits/seconde et doit faire partie des valeurs suivantes : 50, 75, 110, 135, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600 et 19200.

Le paramètre « Bits » indique le nombre de bits de données compris entre le ou les bit(s) start et le ou les bit(s) stop. Il doit être égal à 5, 6, 7 ou 8.

Le paramètre « Stop » donne le nombre de bits stop à la fin de chaque donnée. Ce nombre peut être 1, 1.5 ou 2.

Le paramètre « Parité » peut être « EVEN » (pair), « ODD » (impair) ou « NONE » (sans parité),

Le paramètre « XON » peut valoir « ON » (Mode XON/XOFF validé) ou « OFF » (Mode XON/XOFF dévalidé).

Format

Le paramètre « Handshake » peut valoir « ON » (Poignée de main valide) ou « OFF » (poignée de main dévalidée).

Remarque :

Pour les paramètres émission, réception, stop, bits, parité, XON et handshake, seule la première lettre du mot (Exemple « S » pour STOP) est nécessaire pour être identifiée par SETSIO.

Si l'interface série n'est pas connectée et que vous tapez « SETSIO », le message suivant apparaîtra : « SIO not found ».

SETLST <Nom de fichier>

Envoi de caractères de contrôle à l'imprimante.

Les caractères du fichier compris entre #20 et #FF sont envoyés tels quels. Pour envoyer des caractères de contrôle, il est nécessaire de les faire précéder du caractère « ^ ».

Par exemple, pour envoyer le caractère « tabulation horizontale » de code ASCII 9, les codes suivants seront équivalents :
^HT' ou ^I' ou ^ &9' ou ^9'.

Un caractère précédé de « ^ » est masqué par #1F, ce qui revient à dire que, par exemple, ^A est envoyé sous la forme :

(Code ASCII de 'A') ET logique #1F, c'est-à-dire CTRL + A.

```

      A  0 1 0 0 0 0 0 1
      AND
      #1F 0 0 0 1 1 1 1 1
      = 0 0 0 0 0 0 0 1
  
```

Les caractères de contrôle disponibles sont :

NUL, SOH, STX, ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT, FF, CR, SO, SI, DLE, DC1, DC2, DC3, DC4, NAK, SYN, ETB, CAN, EM, SUB, ESC, FS, GS, RS, US, SP, DEL, XON, XOFF.

Remarque :

Le fichier contenant les caractères de contrôle à émettre sera créé sous traitement de textes.

Si nous appelons ce fichier « FI », il suffira de faire « SETLST FI » pour envoyer les caractères de contrôle.

III. Utilitaires

ECRAN

DATE

Format

DATE (C) ou DATE (CONTINUOUS) ou
DATE (Time Specifications) ou
DATE SET ou
DATE

DATE donne la valeur de la date courante.

date

Wed 12/15/82 00:07:57

DATE SET permet de définir une nouvelle date.

A>date set

Enter today's date (MM/DD/YY) : 02/14/87

Entrez la date (mois/jour/année)

Enter the time (HH:MM:SS) : 10.12.30

Entrez l'heure (heure:minute:seconde)

Press any key to set time

Appuyez sur une touche pour initialiser date et heure.

DATE CONTINUOUS affiche date et heure en continu jusqu'à l'appui sur une touche quelconque du clavier.

LANGUAGE

Modification du jeu de caractères standard obtenus sur l'écran et sur l'imprimante pour obtenir les caractères spécifiques des pays suivants : Danemark, Etats-Unis d'Amérique, France, Allemagne, Italie et Espagne selon le tableau ci-dessous :

Danemark

USA

91	[	ae lié	majuscule
92	\	O	majuscule
93	]	A	majuscule
123	{	ae lié	minuscule
124	:	o	minuscule
125	}	à	minuscule

Caractères obtenus à la mise sous tension de l'ordinateur.

France				Allemagne			
64	@	à	minuscule	64	@	§	
91	[	°	degré	91	[	Ä	majuscule
92	\	ç		92	\	Ö	majuscule
93	]	§	section	93	]	Ü	majuscule
123	{	é		123	{	ä	minuscule
124	:	ù		124	:	ö	minuscule
125	}	è		125	}	ü	minuscule
126	~		tréma	126	~	ß	stz allemand

Grande-Bretagne				Suède			
35	#		livre	64	@	É	majuscule
				91	[	Ä	majuscule
				92	\	Ö	majuscule
				93	]	Å	majuscule
				94	^	Ü	majuscule
				96	'	é	
				123	{	ü	

Italie				Espagne			
91	[	°	degré	35	#	Pt	péséta
93	]	é	minuscule	91	[	í	
123	{	à	minuscule	92	\	ñ	
124	:	ò	minuscule	123	{		
125	}	è	minuscule	124	:	ñ	
126	~	ì	minuscule				

Remarques :

A la mise sous tension, les caractères standard sont les caractères américains.

Pour utiliser l'ordre « LANGUAGE », voici comment procéder :

- 1) Positionner le switch sur l'imprimante qui correspond à la langue choisie.
- 2) Sélectionner le jeu de caractères correspondant en faisant : « LANGUAGE n ».

Remarque :

Cette commande peut être insérée dans « PROFILE.SUB » ou n'importe quel autre fichier SUBMIT.

PALETTE

PALETTE <n1> (<n2> ... <n15>)

<n1> : représente la couleur du fond de l'écran

Format

<n2> : représente la couleur des caractères INK 1.

<ni> : représente la couleur des caractères INK i-1

Le codage des couleurs est le suivant :

<ni> doit avoir une valeur comprise entre 0 et 63. Ce nombre représente trois fois deux bits correspondant aux couleurs de base R(ed, rouge) G(reen, vert) B(lue, bleu).

Les bits 4 et 5 pour la couleur verte, les bits 2 et 3 pour la couleur rouge et les bits 0 et 1 pour la couleur bleue.

Trois niveaux d'intensité sont possibles selon le code suivant :

Valeur du groupe de 2 bits :	0	1	2	3
Intensité AMSTRAD :	0	1	1	2

(0 = faible, 1 = moyenne, 2 = forte)

Remarques :

- Tout nombre supérieur à 63 est masqué par 63 (AND 63) et ne produit donc pas d'erreur.
- Si plus de seize paramètres sont fournis, les derniers entrés sont ignorés.

Exemple :

PALETTE 0 48 → bits 5 et 6 à 1 → Vert brillant

_____ Fond noir

SETDEF

Format

SETDEF (d:(,d:(,d:)))([temporary = d:]
:[ORDER = (typ(,typ))]) ou
SETDEF [DISPLAY:NO DISPLAY] ou
SETDEF [PAGE:NO PAGE]

La première utilisation de SETDEF permet de déclarer de un à quatre disque(s) où seront cherchés les programmes chargés en mode direct ou par un fichier SUBMIT.

La deuxième utilisation permet de valider/dévalider le système d'affichage.

Enfin, la troisième permet de valider/dévalider l'arrêt en fin de page.

SETDEF Affiche tous les paramètres courants :

A >setdef

A:SETDEF COM

Drive Search Path :

(Nom des disques par ordre de priorité)

1st Drive	— A:
2nd Drive	— Default
Search Order (Ordre de recherche)	— SUB, COM
Temporary Drive (Lecteur temporaire)	— B:
Console Page Mode (Mode console)	— On
Program Name Display (Affichage du nom du programme)	— On

SETDEF[TEMPORARY = B:] Définit le disque B : pour stocker les fichiers temporaires.

A>setdef[temporary = b:]

A:SETDEF COM

Temporary Drive — B:

SETDEF B, * Tous les programmes seront d'abord cherchés sur B:

S'ils ne sont pas trouvés, la recherche se poursuivra sur le disque courant.

A>setdef b, *

Drive Search Path :

1st Drive	— B;
2nd Drive	— Default

SETDEF[ORDER = (SUB,COM)] Lorsque aucune extension n'est indiquée pour accéder à un fichier, CP/M + cherchera à exécuter en premier lieu un fichier d'extension « SUB » et en deuxième lieu un fichier d'extension « .COM » (si le .SUB n'existe pas).

A>setdef[order = (sub,com)]

Search Order — SUB, COM

SETDEF[DISPLAY] Les noms des fichiers exécutés en mode direct ou SUBMIT seront affichés au fur et à mesure de leur utilisation.

A>setdef[display]

A:SETDEF COM

Program Name Display — On

SETDEF[NO DISPLAY] Les noms des fichiers exécutés en mode direct ou en mode SUBMIT n'auront aucun écho à l'écran. C'est l'option par défaut à l'allumage de l'ordinateur.

SET24X80 ON ou **SET24X80 OFF** La première commande **SET24X80 ON** permet de limiter la taille de l'écran à 24 lignes et 80 colonnes (au lieu de 25 lignes et 80 colonnes que peut gérer l'AMSTRAD.)

La plupart des programmes standard fonctionnant sous CP/M utilisent le mode 24 lignes et 80 colonnes.

La deuxième commande SET24X80 OFF permet de restituer la taille maximale de l'écran : 25 lignes et 80 colonnes.

TYPE

Format

TYPE (Nom fichier ([PAGE:NO PAGE]))

Affiche le contenu d'un fichier ASCII à l'écran.

L'option « PAGE » permet l'arrêt automatique du listage dès que la ligne 24 de l'écran est atteinte. Le listage redémarre en appuyant sur une touche quelconque et s'arrête en appuyant sur CTRL + C.

L'option « NO PAGE » liste le fichier d'un bout à l'autre sans s'arrêter.

Remarque :

Vous pouvez stopper le listage en cours en appuyant sur « CTRL + S » et le reprendre en appuyant sur « CTRL + Q »

Exemples :

TYPE PROFILE.SUB Affiche le contenu du fichier batch « PROFILE.SUB ».

TYPE B:TEXT[PAGE] Affiche le contenu du fichier « TEXT » lu sur le drive B: en faisant un arrêt à chaque fin de page.

CLAVIER

SETKEYS

Format

SETKEYS <Nom de fichier>

Reconfigure le clavier selon les informations contenues dans le fichier.

Ces données doivent être organisées comme suit :

- Chaque ligne contient la définition d'une touche ou de son code d'expansion.

- Pendant l'exécution du fichier de redéfinition, chaque ligne mal décrite fera l'objet d'une erreur affichée sur l'écran, la ligne sera ignorée et la prochaine ligne exécutée.

La définition d'une touche se fera comme suit :

Numéro de touche suivi éventuellement des états SHIFT suivi du caractère assigné entre cotes.

Les états SHIFT sont les suivants :

S=SHIFT C=CONTROL N=NOTHING (ni SHIFT, ni CONTROL)

Les caractères de code ASCII compris entre #20 et #FF et différents de « ^ » et de « ' » sont pris tels quels.

Le caractère « ^ » définit une séquence ESCAPE (codes de contrôle) : « ^ » suivi d'un caractère de code compris entre #40 et #FF masque le caractère par #1F.

Par exemple, ^A sera équivalent au code CTRL + A.

Remarques :

^^ est équivalent au code ^

^^ est équivalent au code ''

^'ESC' est équivalent au code ESCAPE

^ #Hexa' est équivalent au code CTRL + le nombre exprimé en hexadécimal.

Les codes de contrôle sont les suivants : NUL, SOH, STX, ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT, FF, CR, SO, SI, DLE, DC1, DC2, DC3, DC4, NAK, SYN, ETB, CAN, EM, SUB, ESC, FS, GR, RS, US, SP, DEL, XON, XOFF.

Tous les autres caractères sont interprétés comme des commentaires.

Examinons le contenu du fichier « KEYS.CCP » présent sur la disquette CP/M+. C'est un fichier de redéfinition qui peut être utilisé avec la commande SETKEYS :

A>type keys.CCP

0 N S C ^^ #1F' '' CCP cursor up

1 N S ^^F'' cursor right

1 C ^^ #9F' ''

2 N S C ^^ #1E' '' cursor down

8 N S ^^A'' cursor left

8 C ^^ #9E' ''

9 N S C ^^W'' copy

16 N S ^^G'' clr

16 V ^^K''

18 C ^^E'' enter

66 N S ^^A'27' '' esc

66 C ^^C''

79 C ^^X'' del

E #8C ^^R'' ctrl enter

E #9E ^^F^B''

E #9F ^^F^B^B''

Ligne 1 : Caractère redéfini 0 en normal, shift et control = #1F, soit caractère de contrôle « curseur vers le haut ».

Ligne 2 : Caractère redéfini en normal et shift comme CTRL + F, soit le caractère de contrôle « curseur vers la droite ».

Ligne 8 : Caractère redéfini 16 en normal et shift comme CTRL + G, soit le caractère de contrôle « CLS ».

- Utilisation des codes d'expansion :

Par définition, ils sont représentés par un « E » suivi d'un numéro de token succédé éventuellement d'un commentaire.

Les numéros de token sont compris entre #80 et #9F.

Par exemple, en ligne 14 du précédent listing, le caractère d'expansion #8C est défini comme CTRL + R (soit CTRL + ENTER).

AUTRES :**GET****Format**

GET <Fichier> [<Options>] ou
GET CONSOLE

La forme GET <fichier> indique que les prochaines commandes CP/M seront lues dans le fichier spécifié.

Si aucune option n'est précisée, les commandes sont lues dans le fichier dès qu'un programme lancé par l'opérateur les nécessite. Les options possibles sont :

ECHO : Toutes les commandes du fichier sont affichées à l'écran au moment de leur utilisation.

NO ECHO : Les commandes du fichier ne sont pas affichées à l'écran.

SYSTEM : Le fichier de commandes est lu immédiatement.

Remarque :

La commande GET est typiquement utilisée avec la commande SUBMIT lorsque le fichier batch lancé par SUBMIT est paramétré. Les paramètres peuvent être insérés dans un fichier pour éviter à l'utilisateur de les entrer à chaque utilisation du batch.

PAPER**Format**

PAPER <un ou plusieurs paramètres>

Cette commande permet de configurer l'imprimante.

Les paramètres possibles sont les suivants :

CONTINUOUS STATIONERY pour utiliser l'imprimante avec du papier en continu.

DEFAULTS les paramètres courants sont envoyés vers l'imprimante pour la configurer. Ils deviennent donc des paramètres par défaut jusqu'à l'extinction de l'imprimante.

FORM LENGTH <n> où <n> est compris entre 6 et 99.
Définit le nombre de lignes par page imprimée.
Ce paramètre positionne l'interligne à 6 et le saut entre deux pages à 0.

GAP LENGTH <n> où <n> est compris entre 0 et 99.
définit le nombre de lignes entre deux pages.

Remarque :

si <n> est différent de 0, l'interligne est positionné à 6.

LINE PITCH <n> où <n> vaut 6 ou 8.

Définit le nombre de lignes par pouce.

PAPER OUT DEFEAT ON active le détecteur de fin de papier.

PAPER OUT DEFEAT OFF désactive le détecteur de fin de papier.

SINGLE SHEET pour utiliser l'imprimante feuille à feuille.

PUT

Format

PUT CONSOLE FILE <fichier> [<options>] ou

PUT PRINTER FILE <fichier> [<options>] ou

PUT CONSOLE CONSOLE ou

PUT PRINTER PRINTER

La première forme envoie les données affichées à l'écran vers le fichier spécifié.

La deuxième forme envoie les données affichées à l'imprimante vers le fichier spécifié.

Les options possibles pour la première et la deuxième formes sont :

ECHO pour avoir un affichage sur l'écran

NO ECHO pour ne pas avoir d'affichage sur l'écran.

FILTER pour avoir une représentation des caractères de contrôle en leur correspondant ASCII sous la forme ^ASCII (par exemple ^I).

NO FILTER pour annuler l'option FILTER.

SYSTEM pour stocker également sur le fichier les messages envoyés par le système.

La troisième forme annule la première, et la quatrième annule la deuxième.

IV. Ordres évolués

DISKIT3

Utilitaire disque de haut niveau permettant de :

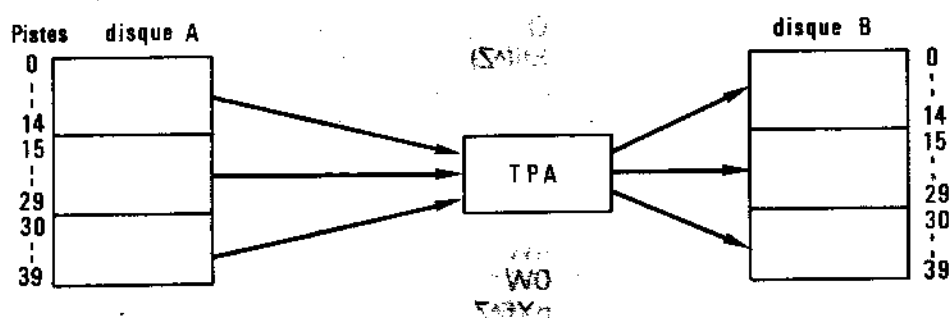
- copier,
- formater,
- vérifier un disque.

La copie se fait d'une unité de disque sur une autre ou d'une unité de disque sur elle-même.

Le formatage est la phase préalable à toute utilisation d'un disque vierge. Le disque peut être formaté en données ou en système.

La vérification permet de comparer deux disques entre eux pour vérifier s'ils sont identiques. Pratiquement, cette étape est dans la plupart des cas inutile, car le lecteur de disquettes AMSTRAD est très fiable.

DISCKIT3 utilise la zone TPA (transcient program area, voir 3/3) pour minimiser les accès disques de la manière suivante :



La copie d'une face de disquette se fait en 3 fois.

ED

Format

ED (Fichier en entrée) ((d:) Fichier en sortie)

Editeur de lignes.

ED permet de saisir des fichiers textes. Sans avoir la prétention d'être un traitement de textes performant, il permet cependant de saisir facilement des fichiers SUBMIT, par exemple.

Les commandes utilisées dans ED sont les suivantes :

nA	définit un buffer RAM de n lignes.
OA	définit un fichier RAM jusqu'à ce qu'il soit à moitié plein.
#A	définit un fichier RAM jusqu'à ce qu'il soit plein.
B, -B	Pointeur de ligne en début (B) ou en fin (-B) de buffer.
nC, -nC	Pointeur de ligne n caractères plus loin (nC) ou n caractères plus haut (-nC).
nD, -nD	Efface n caractères après la position courante (nD), ou avant la position courante (-nD) du pointeur.
E	Sauvegarde le buffer et retourne sous CP/M +.
Fch (^Z)	Cherche la chaîne ch.
H	Sauve le buffer et retourne sous éditeur.
I<cr>	Passe en mode insertion.
Ich(^Z)	Insère une chaîne à la position courante du pointeur ligne.
Jch1^Zch2^Z	Juxtapose les chaînes ch1 et ch2.
nK, -nK	Efface n lignes à partir de la position courante du pointeur vers le bas (nK) ou vers le haut (-nK).
nL, -nL, OL	Déplace n lignes à partir de la position courante du pointeur de lignes.
nM Cde	Exécute la commande « Cde » n fois.
n, -n	Déplace le pointeur de ligne de n lignes vers le bas (n) ou vers le haut (-n) et affiche la nouvelle ligne pointée.
n:	Déplace le pointeur à la ligne n et l'affiche.
:n Cde	exécute la commande jusqu'à la ligne n.

Nch (^Z)R Recherche étendue
 O Retourne au fichier original.
 np, -nP Déplace le pointeur de lignes 23 lignes vers le bas (nP) ou vers le haut (-nP) et affiche ces 23 lignes.
 Q Abandonne le fichier en mémoire et retourne sous CP/M +.
 Rfi(^Z) Lit le fichier « fi » et le place dans le buffer.
 Sch1^Zch2 Substitue ch2 à ch1.
 nT, -nT, OT Affiche n lignes.
 U, -U Convertit les caractères d'une ligne en majuscules.
 V, -V Numérotation/non numérotation des lignes.
 OV Affiche l'espace libre dans le buffer.
 nW Ecrit n lignes sur le fichier disque.
 OW Ecrit sur disque jusqu'à ce que le buffer soit à moitié plein.
 nXfi^Z Ecrit n lignes sur le fichier « fi » spécifié.
 OXfi^Z Efface le fichier « fi » spécifié.
 nZ Attend n secondes.

Nous allons créer un fichier sous ED pour nous habituer à manipuler ses commandes.

ED TEST.COM (nom du fichier sur lequel on va travailler)

NEW FILE (ce fichier n'existe pas sur le disque)

: *I (entrée dans le mode « insertion »)

1 : CE FICHIER VA SERVIR
 2 : DE TEST POUR LE
 3 : PROGRAMME ED.COM
 4 :

: *B (passage en ligne 1)

1 : * - P (affichage des 23 premières lignes)

1 : CE FICHIER VA SERVIR
 2 : DE TEST POUR LE
 3 : PROGRAMME ED.COM

1 : *Q (sortie du programme)

Q - (Y/N)?N

1 : *3T (affichage des 3 prochaines lignes)

1 : CE FICHIER VA SERVIR
 2 : DE TEST POUR LE
 3 : PROGRAMME ED.COM

1 : *OV (mémoire libre ?)

39969/40027 (mémoire libre/mémoire totale en octets)

1 : * - V (pas de numérotation de ligne)

*V (numérotation de ligne)

1 : * - 10 (remonte de 10 lignes)

1 : CE FICHIER VA SERVIR

1 : *2 (descend de 2 lignes)

3 : PROGRAMME ED.COM
3 : * - 10
1 : CE FICHIER VA SERVIR
1 : *FPOUR^Z (cherche la chaîne « POUR »)
2 : *1 (chaîne « POUR » trouvée en ligne 2)
3 : PROGRAMME ED.COM

HELP

Donne des informations sur les mots-clés de CP/M+.

HELP fonctionne par niveaux hiérarchiques.

Un mot-clé peut avoir une ou plusieurs options qui, elles-mêmes, peuvent avoir une ou plusieurs options, etc.

Par exemple :

SET possède les options LABEL, PASSWORD, ATTRIBUTES, DEFAULT, TIME-STAMPS et DRIVES.

Pour avoir des informations sur l'option LABEL de SET, tapez SET LABEL.

LABEL nous indique que des exemples peuvent être obtenus en tapant EXAMPLES. Il faudra taper SET LABEL EXAMPLES pour avoir les informations concernant les exemples proposés pour SET LABEL.

SUBMIT

SUBMIT <Fichier> (Argument)...(Argument).

Permet d'exécuter un fichier différé (souvent, batch ou submit) en lui passant 0, 1 ou plusieurs arguments dans la limite de 10.

Une ligne d'un fichier SUBMIT peut contenir :

- une commande CP/M+ avec 0, 1 ou plusieurs paramètres repérés par \$0 à \$9,
- une ligne de demande d'entrée,
- un nom de programme avec des paramètres repérés par \$0 à \$9.

La longueur d'une ligne doit être inférieure ou égale à 135 caractères.

Le fichier « PROFILE.SUB » est un SUBMIT particulier : à chaque initialisation (boot) de l'ordinateur ou lors de l'activation de CP/M+, le fichier PROFILE.SUB est exécuté s'il est présent sur la disquette courante.

Le passage de paramètres se fait de la façon suivante :

Appel du SUBMIT « ESSAI » en passant les paramètres « DIR » et « B » :
SUBMIT ESSAI DIR B:

Format

Dans le programme SUBMIT :

RENAME \$2TOTO=TITI Renomme TITI en TOTO sur disque B:
\$1 Affiche le directory pour vérifier que le fichier
a bien été renommé.

HELP

SUBMIT

3/4.1

Liste alphabétique des mots-clés

mots-clés		numéro de page
DATE	Valeur de la date courante	19
DEVICE	Correspondance entre équipement physique et logique	14
DIR	Répertoire d'une cassette ou d'une disquette	4
DIRS	Répertoires des fichiers d'extension .SYS	5
DISCKIT3	Utilitaire disque	26
ED	Editeur de lignes	27
ERA	Effacement d'un fichier sur disquette	5
GET	Lecture des commandes CP/M dans un fichier	25
HELP	Informations sur les mots-clés	29
INITDIR	Marquage d'une disquette	6
LANGUAGE	Modification du jeu de caractères standard	19
PALETTE	Définition de la palette des couleurs	20
PAPER	Configuration de l'imprimante	25
PIP	Transfert entre deux périphériques	8
PUT	Envoie des données vers un fichier	26
REN	Renomme un fichier	9
SET	Modification des attributs	10
SET24 x 80	Configuration de l'écran en 24 ou 25 lignes	22
SETDEF	Ordre de recherche sur disque	21
SETKEYS	Reconfigure le clavier	23
SETLST	Caractères de contrôle à l'imprimante	18
SETSIO	Gestion d'une interface série	17
SHOW	Espace disponible sur un disque	13
SUBMIT	Exécute d'un fichier batch	29
TYPE	Liste le contenu d'un fichier	23
USER	Numéro d'utilisateur	14

3/4.2

Programmation sous CP/M plus

3/4.2.1

Organisation mémoire du 6128

Le 6128, comme ses deux prédécesseurs (464 et 664) possède un Z80 qui lui permet d'adresser 64 Koctets de mémoire. Pourtant sa mémoire RAM est de 128 Koctets et sa mémoire ROM de 48 Koctets.

Pour ce faire, l'espace mémoire est divisé en trois banques superposées que nous appellerons banque 0, banque 1 et banque 2.

La banque 0 est utilisée par le BDOS (Basic Disk Operating System) et le BIOS (Basic Input Output System) comme le montre le schéma ci-dessous :

#FFFF	Bloc 7
#C000	Bloc 2 : BDOS, BIOS, table d'indirection du firmware
#8000	Bloc 1 : Ecran
#4000	Bloc 0 : ROM basse table d'indirection du BIOS
#0000	

La banque 1 est la TPA (Transcient Program Area). Elle contient les blocs 4, 5, 6 et 7 entre les adresses #0000 et #FFFF.

La banque 2 occupe les mémoires comprises entre #4000 et #8000 d'une part de #C000 et #FFFF d'autre part.

L'espace compris entre #4000 et #8000 contient le CCP (Console Command Processor), divers buffers et tables concernant le lecteur de disquettes.

L'espace compris entre #C000 et #FFFF est le même pour les trois banques. Il contient le bloc 7 (parties résidentes des BDOS, BIOS et TPA).

CP/M plus se compose de plusieurs modules fonctionnels : BIOS, BDOS, LOADER, RSX, TPA et CCP.

Le BIOS (Basic Input Output System) contient un ensemble de routines qui permettent de communiquer avec les périphériques.

Le BDOS (Basic Disk Operating System) contient un ensemble de routines que nous verrons en détails par la suite. Ces routines permettent à un programme situé en TPA d'accéder aux divers modules de CP/M+.

Le LOADER n'est souvent pas résident en RAM lorsqu'un programme en TPA s'exécute. Il contient des routines de gestion pour les modules CCP et TPA.

Les RSX (Resident System eXtension) sont des modules qui viennent s'ajouter au système d'exploitation de base. Lorsque de tels modules sont présents, le LOADER est résident et peut être accédé par la fonction BDOS 59.

Le TPA (Transcient Program Area) est la zone mémoire réservée pour l'exécution des programmes CP/M+ et CCP.

Le CCP (Console Command Processor) contient toutes les routines d'interface entre opérateur et CP/M+.

A noter également que la page zéro (#0000 à #0100) constitue une zone d'interface (table d'indirection) entre le BDOS d'une part, les CCP et TPA d'autre part.



3/4.2.1.1**FCB (File Control Bloc)**

Le bloc de contrôle de fichiers (FCB) est une zone en mémoire utilisée comme interface entre BDOS et le logiciel de base de CP/M+. Cette zone contient de nombreuses informations qui concernent les entrées/sorties disquettes. Nous allons les passer en revue.

La longueur du FCB est souvent 33 octets, et parfois 36 octets. Les champs du FCB seront repérés comme suit :

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	...	31	32	33	34	35
dr	f1	f2	f3	f4	f5	f6	f7	f8	t1	t2	t3	ex	s1	s2	rc	d0	...	dn	cr	r0	r1	r2

dr : Identification du lecteur de disquettes :

→ 0 : lecteur par défaut,

→ 1 : lecteur A

→ 2 : lecteur B

→ 16 : lecteur P

f1...f8 : Nom du fichier en ASCII et en majuscules, bits de poids forts à zéro.

t1...t3 : Type du fichier en ASCII et en majuscules, bits de poids forts à zéro selon le codage suivant :

t1 = 1 → Fichier à lecture seule

t2 = 1 → Fichier système

t3 = 1 → Fichier archive

ex : Nombre d'extension du fichier. Ce nombre est souvent mis à zéro par le programme appelant. Il évolue entre 0 et 31 pendant les accès disquette.

s1, s2 : Réservé pour CP/M+.

rc : Nombre d'enregistrements accédés en une fois (entre 0 et 255).

d0..dn : Réservé pour CP/M+.

cr : Numéro d'enregistrement à lire ou à écrire. Ce numéro est positionné à zéro lors d'une ouverture ou d'une création de fichier.

r0..r2 : Numéro d'enregistrement accédé (entre 0 et #3FFFF). Ce champ est uniquement utilisé pour les accès directs.

3/4.2.1.2

SCB (System Control Bloc)

Le bloc de contrôle système est une structure de données (# 63 octets) qui se trouve dans le BDOS. Cette structure est utilisée par CP/M Plus pour communiquer entre le BDOS, le BIOS, les programmes et les RSX. Elle contient les données suivantes :

Octet	Signification
#00 à #04	Réservé.
#05	Numéro de version BDOS.
#06 à #09	Indicateurs réservés à l'utilisateur.
#0A à #0F	Réservé.
#10 à #11	Code d'erreur d'un programme ou valeur passée à un programme chaîné.
#12 à #19	Réservé.
#1A	Nombre de colonnes par ligne affichée (79 dans la plupart des cas).
#1B	Abscisse du curseur.
#1C	Nombre de lignes par page (24 dans la plupart des cas).
#1D à #21	Réservé.
#22 à #2B	Indicateurs de redirection des 5 périphériques de type « caractère ». Deux octets par entrée. Le bit 15 représente le périphérique physique 0. Le Bit 4 représente le périphérique physique 11.
#22 à #23	Redirection de CONIN.
#23 à #25	Redirection de CONOUT.
#26 à #27	Redirection de AUXIN.
#28 à #29	Redirection de AUXOUT.
#2A à #2B	Redirection de LSTOUT.
#2C	Affichage par page. Si cet octet vaut 0, l'affichage des données issues de la plupart des fonctions de CP/M Plus se fait par page. La page suivante est visualisée par l'appui sur une touche quelconque du clavier. Si cet octet est non nul, l'affichage est continu. Les touches Ctrl S et Ctrl Q peuvent dans tous les cas être utilisées pour suspendre et reprendre l'affichage.
#2D	Réservé.

Octet	S.I.S.A/E Signification
#2E	Lorsque cet octet est nul, Ctrl H est actif (effacement du caractère précédent). Il n'est pas actif dans le cas contraire.
#2F	Lorsque cet octet est nul, la fonction « rubout » est active. Elle est inactive s'il vaut #FF.
#30 à #32	Réservé.
#33 à #34	Mode d'affichage (voir fonction BDOS 109).
#35 à #36	Réservé.
#37	Caractère délimiteur en sortie (\$ par défaut). Ce caractère peut être changé à l'aide de la fonction BDOS 110.
#38	Si cet octet vaut 1, tout ce qui est affiché sur l'écran est également imprimé. Si cet octet vaut 0, les caractères affichés sur l'écran n'ont aucun écho sur l'imprimante.
#39 à #3B	Réservé.
#3C à #3D	Adresse de DMA courante (#0080 par défaut). Reportez-vous aux fonctions BDOS 13 et 26 qui agissent sur cette adresse.
#3E	Disque par défaut (0 = A, ..., 15 = P). Cette valeur ne peut être que lue. Reportez-vous également à la fonction BDOS 25.
#3F à #43	Réservé.
#44	Numéro d'utilisateur (0 à 15). Cette valeur ne peut être que lue. Reportez-vous également à la fonction BDOS 32.
#45 à #49	Réservé.
#4A	Nombre de secteurs lus en accès direct. Reportez-vous également à la fonction BDOS 44.
#4B	Mode d'erreur BDOS. Reportez-vous à la fonction BDOS 45.
#4C à #4F	Lecteurs de recherche. Dans le cas où moins de 4 lecteurs sont utilisés, le dernier code lecteur doit être suivi de la valeur #FE. Les lecteurs sont codés entre 0 et 16. 0 correspond au lecteur par défaut, 1 au lecteur A:, ..., 16 au lecteur P:.
#50	Numéro du lecteur des fichiers temporaires (0 à 16. 0 pour le lecteur par défaut, 1 pour le lecteur A:, ..., 16 pour le lecteur P:).

Octet	Signification
#51	Numéro du lecteur sur lequel s'est produite la dernière erreur physique.
#52 à #56	Réservé.
#57	Indicateurs BDOS. Bit 7 = 1 lorsque les banques mémoire sont utilisées. Bit 6 = 1 lorsque les banques mémoire ne sont pas utilisées.
#58 à #59	Date (depuis le 01/01/1978).
#5A	Heures au format BCD.
#5B	Minutes au format BCD.
#5C	Secondes au format BCD.
#5D à #5E	Mémoire de base commune (0 lorsque les banques mémoire ne sont pas utilisées, sinon différent de 0).
#5F à #63	Réservé.

3/4.2.2

Table d'indirection des fonctions BDOS en page zéro

Les fonctions BDOS sont accessibles à travers la page zéro par un numéro de fonction. Elles sont activées en assembleur en chargeant dans le registre C le numéro de la fonction désirée. Eventuellement, un ou plusieurs paramètres peuvent être échangés à travers les autres registres.

Nous allons donner la liste des fonctions BDOS de CP/M + de 0 à 152.

Fonction 0

SYSTEM RESET

Initialisation du système

Entrée : C = #00

Sortie : Aucune

L'appel à cette fonction a le même effet qu'un saut à l'adresse #0000. Le système est initialisé et le contrôle est donné au CCP.

Sous CP/M+, le système disque n'est pas initialisé par l'appel à cette fonction. Le programme appelant peut passer un code de retour au CCP en activant la fonction 108 avant de faire appel à la fonction BDOS 0.

Fonction 1

CONSOLE INPUT

Lecture d'un caractère alphanumérique sur le clavier

Entrée : C = #01

Sortie : A = Caractère ASCII lu

Cette fonction attend qu'un caractère soit entré sur l'unité logique CONIN: (clavier) pour rendre le contrôle au programme appelant et stocke le code ASCII de ce caractère dans le registre A.

Les caractères CR, LF, BS et CTRL-H ont un écho sur l'écran. Les tabulations CTRL-I sont converties en déplacements de 8 caractères et ont un écho sur l'écran.

Lorsque le mode d'affichage est celui par défaut, les caractères CTRL-S (stop scroll), CTRL-Q (start scroll) et CTRL-P (start/stop print echo) sont interceptés par la fonction BDOS 1 et ont un écho sur l'écran. Cependant, si les fonctions associées à ces caractères sont dévalidées par le mode d'affichage, CTRL-S, CTRL-Q et CTRL-P ne sont pas interceptés par la fonction BDOS 1 et n'ont donc aucun écho sur l'écran.

Fonction 2**CONSOLE OUTPUT**

Affichage d'un caractère sur l'unité CONOUT:.

Entrée : C = #02
E = Caractère ASCII

Sortie : Aucune

Cette fonction envoie le caractère dont le code ASCII se trouve dans le registre E sur l'unité CONOUT: (écran).

Lorsque le mode d'affichage est celui par défaut, la fonction BDOS 2 le caractère CTRL-I est converti en une tabulation de 8 caractères, et les caractères CTRL S, CTRL-Q et CTRL P sont interprétés.

Fonction 3**AUXILIARY INPUT**

Lecture d'un caractère sur l'entrée auxiliaire.

Entrée : C = #03

Sortie : A = Caractère lu

Cette fonction permet de lire un caractère sur l'entrée AUXIN:.

Elle rend le contrôle au programme appelant lorsqu'un caractère se présente sur cette entrée. Le caractère se trouve dans le registre A.

Fonction 4**AUXILIARY OUTPUT**

Envoi d'un caractère sur l'unité AUXOUT:.

Entrée : C = #04
E = Caractère ASCII

Sortie : Aucune

Cette fonction envoie le caractère dont le code ASCII se trouve dans le registre E sur l'unité logique AUXOUT:.

Fonction 5**LIST OUTPUT**

Envoi d'un caractère sur l'unité LST:

Entrée : C = #05
E = Caractère ASCII

Sortie : aucune

Cette fonction envoie le caractère dont le code ASCII se trouve dans le registre E sur l'unité logique LST:.

Fonction 6**DIRECT CONSOLE I/O**

Lecture ou écriture directe d'un caractère sur l'unité CONIN:.

Entrée : C = #06
 E = #FF (Entrée ou Etat),
 #FE (Etat),
 #FD (Entrée),
 caractère (Sortie)

Sortie : A = caractère lu le cas échéant.

Cette fonction BDOS permet de lire (avec ou sans attente) ou d'écrire un caractère sur l'unité CONIN:, en fonction de la valeur contenue dans le registre E:

E = #FF Lecture d'un caractère. A contient la valeur zéro si aucun caractère n'est disponible.

E = #FE Lecture de l'état de CONIN:. A contient la valeur zéro si aucun caractère n'est disponible, #FF sinon.

E = #FD Attente de l'entrée d'un caractère sur l'unité CONIN:. Le code ASCII du caractère se trouve dans le registre A.

E = ASC Lorsque le registre E contient la valeur ASCII d'un caractère, ce dernier est envoyé sur l'unité CONOUT:

Fonction 7

AUXILIARY INPUT STATUS

Lecture de l'état de l'entrée auxiliaire.

Entrée : C = #07

Sortie : A = Etat de l'entrée auxiliaire

Cette fonction renvoie dans le registre A l'état de l'entrée auxiliaire AUXIN:. Le registre A prend la valeur #FF si un caractère n'est pas disponible sur AUXIN: et #00 si aucun caractère n'est disponible sur AUXIN:.

Fonction 8

AUXILIARY OUTPUT STATUS

Lecture de l'état de la sortie auxiliaire.

Entrée : C = #08

Sortie : A = Etat de la sortie auxiliaire.

Cette fonction renvoie dans le registre A l'état de la sortie auxiliaire AUXOUT:. Le registre A prend la valeur #FF si la sortie auxiliaire AUXOUT: est prête à recevoir un caractère et #00 si AUXOUT: n'est pas prête à recevoir un caractère.

Fonction 9

PRINT STRING

Affichage d'une chaîne de caractères

Entrée : C = #09

DE = Adresse de début de chaîne

Sortie : Aucune.

Cette fonction envoie une chaîne de caractères sur la sortie logique CONOUT: jusqu'à ce que le caractère délimiteur \$ soit rencontré dans la chaîne. La fonction BDOS 110 permet de changer la valeur ASCII du caractère délimiteur.

Lorsque le mode d'affichage est le mode par défaut, la fonction BDOS 9 convertit les caractères tabulations CTRL-I en déplacements de 8 caractères et les caractères CTRL-Q, CTRL-S et CTRL-P sont interprétés.

Fonction 10 **READ CONSOLE BUFFER**

Lecture du buffer de la console CONIN:

Entrée : C = #0A

DE = Adresse du buffer ou #0000

Sortie : Le buffer est rempli.

Cette fonction lit une « ligne » sur l'unité logique CONIN: et range les caractères lus dans le buffer dont l'adresse de début est donnée dans le registre DE. La « ligne » est considérée terminée lorsqu'un des caractères suivants est rencontré : CTRL-M, CTRL-J ou CR. Un caractère BEL (CTRL-G) est envoyé sur CONOUT: lorsque le buffer est plein.

L'adresse pointée par le contenu du registre DE doit être positionnée avant l'appel à la fonction BDOS 10. Son contenu est la longueur maximum du buffer de lecture (entre 1 et 255). Les données lues sont rangées en (DE+2), (DE+3), etc. La mémoire (DE+1) contient le nombre de caractères lus (entre 0 et (DE)).

ob 1010

10101010

Si DE = #0000, la fonction BDOS 10 suppose que le buffer de lecture est initialisé ((DE+2) à (DE+n+1) contiennent les valeurs désirées et (DE+n+2) le terminateur #00) et se trouve par défaut à l'adresse DMA courante. Lorsque le terminateur est rencontré, BDOS 10 attend une entrée de caractère(s) sur CONIN: . Cette méthode permet à l'opérateur d'accepter la chaîne qui est affichée (il presse alors ENTER) ou de la modifier en utilisant au besoin les caractères de contrôle suivants :

- CTRL-A déplace le curseur vers la gauche d'une position
- CTRL-B déplace le curseur au début de la ligne ou à la fin de la ligne s'il se trouve déjà au début de la ligne.
- CTRL-C annule la modification de la chaîne
- CTRL-E provoque un passage à la ligne des caractères qui se trouvent à la droite du curseur.
- CTRL-F déplace le curseur vers la droite d'une position
- CTRL-G efface le caractère qui occupe la position courante
- CTRL-H efface le caractère qui se trouve à la gauche du curseur
- CTRL-J marque la fin de la lecture

CTRL-K efface les caractères qui se trouvent à la droite du curseur
CTRL-M identique à CTRL-J
CTRL-P écho sur l'équipement logique LST:
CTRL-R le caractère à gauche du curseur est réaffiché sur la nouvelle ligne
CTRL-U La ligne précédente du buffer contient le caractère à gauche du curseur, la ligne courante est effacée et il y a un passage à la nouvelle ligne
CTRL-W Réaffiche la ligne précédente si la ligne courante est vide ou déplace le curseur en fin de ligne dans un autre cas
CTRL-X Efface tous les caractères à gauche du curseur
DEL Efface le caractère à gauche du curseur.

Fonction 11**GET CONSOLE STATUS**

Donne l'état de la console CONIN:

Entrée : C = #0B

Sortie : A = Etat de la console

Cette fonction indique si un caractère a été tapé sur l'unité logique CONIN:.

Lorsque le mode d'affichage est le mode par défaut, le registre A contient la valeur 1 lorsqu'un caractère a été tapé et la valeur 0 dans le cas contraire.

Lorsque le mode d'affichage est « CTRL-C », le registre A contient la valeur 1 uniquement si la séquence CTRL-C a été tapée.

Fonction 12**RETURN VERSION NUMBER**

Numéro de version du fichier système BDOS

Entrée : C = #0C

Sortie : HL = Numéro de version

Cette fonction retourne dans le registre HL le numéro de version du fichier système BDOS. De cette manière les logiciels CP/M peuvent exploiter les possibilités particulières de chaque CP/M et ainsi fonctionner sur des CP/M de versions différentes.

Fonction 13**RESET DISK SYSTEM**

Etat initial du système de fichiers BDOS.

Entrée : C = #0D

Sortie : Aucune

Fonction 14

Cette fonction positionne le système de fichiers BDOS dans un état initial (disque par défaut = A:, adresse de DMA par défaut : #0080, lecteur(s) de disques positionné(s) en lecture/écriture).

SELECT DISK

Définition du lecteur de disques par défaut

Entrée : C = #0EH
E = Disque par défaut (0 à 15)

Sortie : A = Indicateur d'erreur
H = Indicateur d'erreur physique

Cette fonction définit le disque qui sera accédé par défaut pour toutes les opérations CP/M à venir. Le registre E désigne ce disque : 0 pour le lecteur A jusqu'à 15 pour le lecteur P.

En retour de fonction, le registre A est à zéro si la définition du disque par défaut s'est déroulée correctement. Dans le cas contraire, la valeur de A dépend du mode d'erreur BDOS:

- mode par défaut : un message identifiant le type d'erreur est affiché,
- autre mode : A = #FF
et H = #01 pour une erreur Disk I/O Error
#04 pour une erreur Invalid drive

Fonction 15**OPEN FILE**

Ouverture d'un fichier disque

Entrée : C = #0F
DE = Adresse de FCB

Sortie : A = Code répertoire
H = Code d'erreur

Cette fonction active le FCB (File Control Block) d'un fichier disque qui appartient au numéro d'utilisateur courant (ou zéro).

L'adresse du FCB est donnée dans le registre DE. Elle pointe sur une zone mémoire qui respecte les conventions suivantes :

Octet 0 : Référence du lecteur (0 = par défaut, 1 = A, 2 = B, etc.)

Octets 1 à 8 : Nom du fichier en minuscule ou majuscule. Toutes les lettres sont converties en majuscules

Octets 9 à 11 : Extension du fichier en minuscule ou majuscule. Toutes les lettres sont converties en majuscules.

Octets 12 à 15 : A zéro

Si le fichier que l'on tente d'accéder est protégé en lecture, le mot de passe correct doit se trouver dans les huit premiers octets de la zone DMA courante ou avoir été défini par la fonction BDOS 106.

Lorsque le numéro d'utilisateur courant n'est pas zéro, et que le fichier que l'on tente d'ouvrir par la fonction 15 n'existe pas sous le numéro d'utilisateur courant, le fichier spécifié est cherché sous le numéro d'utilisateur zéro. Pour que l'ouverture aboutisse, il est nécessaire que le fichier ait l'attribut « système » positionné.

Les données renvoyées par la fonction BDOS 15 sont les suivantes :

Code répertoire dans le registre A:

#00 si l'ouverture s'est déroulée correctement ;

#FF si le fichier n'a pas été trouvé.

Code d'erreur dans le registre H:

#00 si l'ouverture s'est déroulée correctement ou si le fichier n'a pas été trouvé,

Un message d'erreur est affiché si le mode d'erreur est celui par défaut. Sinon, le registre A est positionné à #FF et le registre H contient :

#01 pour une erreur « Disk I/O Error »,

#04 pour une erreur « Invalid Drive Error »,

#07 pour une erreur « File Password Error »,

#09 pour une erreur sur les champs nom ou extension du FCB.

Fonction 16

CLOSE FILE

Fermeture d'un fichier

Entrée : C = #10

DE = Adresse du FCB

Sortie : A = Code répertoire

H = Code d'erreur

Cette fonction ferme un fichier ouvert par une des fonctions BDOS 15 ou 22.

L'attribut d'interfaçage f5 indique si la fermeture du fichier est permanente (f5=0) ou partielle (f5=1). Dans ce dernier cas, le fichier est maintenu ouvert mais le repertoire est mis à jour.

Les données renvoyées par la fonction BDOS 16 sont les suivantes :

Code répertoire dans le registre A:

#00 si la fermeture s'est déroulée correctement ;

#FF si le fichier n'a pas été trouvé.

Code d'erreur dans le registre H:

#00 si l'ouverture s'est déroulée correctement ou si le fichier n'a pas été trouvé,

Un message d'erreur est affiché si le mode d'erreur est celui par défaut. Sinon, le registre A est positionné à #FF et le registre H contient :

- #01 pour une erreur « Disk I/O Error »,
- #02 pour une erreur « Read/Only Disk »,
- #04 pour une erreur « Invalid Drive Error ».

Fonction 17

Inamer

SEARCH FOR FIRST

Recherche du premier fichier qui satisfait le FCB

Entrée : C = #11

DE = Adresse de FCB

Sortie : A = Code répertoire

H = Code d'erreur

Cette fonction recherche dans le répertoire courant le premier fichier qui s'accorde avec les éléments fournis dans le FCB (File Control Bloc). L'adresse du FCB est spécifiée dans le registre DE. Rappelons la structure du bloc FCB :

Octet 0 : Référence du lecteur (0 = par défaut, 1 = A, 2 = B, etc.)

Octets 1 à 8 : Nom du fichier en minuscule ou majuscule. Toutes les lettres sont converties en majuscules.

Octets 9 à 11 : Extension du fichier en minuscule ou majuscule. Toutes les lettres sont converties en majuscules.

Octets 12 à 15 : A zéro

Octets 16 à 23 : Mot de passe en minuscule ou majuscule ou caractères espaces si aucun mot de passe n'est désiré.

Octets 24 à 31 : Réservés pour CP/M

La fonction BDOS 17 peut être complétée par la fonction BDOS 18 (SEARCH FOR NEXT) qui détermine le ou les fichiers suivants qui satisfont les données indiquées dans le bloc FCB.

Les données renvoyées par la fonction BDOS 17 sont les suivantes :

Code répertoire dans le registre A:

Ce code contient le nom du lecteur (0 à 3) si la recherche a abouti, et #FF sinon.

Code d'erreur dans le registre H:

Si le code renvoyé dans le registre A est 0, 1, 2, 3 ou #FF, le registre H contient la valeur zéro.

Un message d'erreur est affiché si le mode d'erreur est celui par défaut. Sinon, le registre A est positionné à #FF et le registre H contient :

- #01 pour une erreur « Disk I/O Error »,
- #04 pour une erreur « Invalid Drive Error ».

Fonction 18**SEARCH FOR NEXT**

OS

Recherche du prochain fichier qui satisfait le FCB

Entrée : C = #12

Sortie : A = Code répertoire
H = Code d'erreur

Cette fonction est identique à la fonction BDOS 17, à ceci près qu'elle poursuit la recherche de fichier qui avait commencé par une fonction 17.

Fonction 19**DELETE FILE**

Effacement d'un fichier

Entrée : C = #13

DE = Adresse du FCB

Sortie : A = Code répertoire
H = Code d'erreur

Cette fonction permet d'effacer le fichier dont le FCB correspond à celui pointé par le registre DE. La structure du FCB est donnée dans la fonction BDOS 17. Les champs « nom du fichier » et « extension » peuvent contenir des jockers « ? ». L'attribut d'interfaçage f5 définit le type d'effacement désiré.

f5 = 0 pour un effacement standard :

Le FCB du répertoire et le fichier sont effacés et la place disque ainsi libérée est disponible pour y stocker d'autres informations.

f5 = 1 pour un effacement de FCB dans le répertoire seulement.

Dans ce cas, le fichier n'est pas effacé mais n'est plus accessible car le FCB du répertoire n'existe plus. De cette manière, le fichier est rendu « invisible » lors des opérations DIR.

Les données renvoyées par la fonction BDOS 19 sont les suivantes :

Code répertoire dans le registre A:

Ce code contient zéro si l'effacement s'est bien déroulé, et #FF sinon.

Code d'erreur dans le registre H:

Si le code renvoyé dans le registre A est 0 ou #FF, le registre H contient la valeur zéro.

Un message d'erreur est affiché si le mode d'erreur est celui par défaut. Sinon, le registre A est positionné à #FF et le registre H contient :

- #01 pour une erreur « Disk I/O Error »,
- #02 pour une erreur « Read-Only Disk »,
- #03 pour une erreur « Read-Only File »,
- #04 pour une erreur « Invalid Drive Error »,
- #07 pour une erreur « File Password Error »

Fonction 20**READ SEQUENTIAL**

804. a Lecture d'un ou plusieurs enregistrement(s)

Entrée : C = #14H
DE = Adresse FCB

Sortie : A = Code d'erreur
H = Erreur physique

Cette fonction transfère de 1 à 128 enregistrements de 128 octets d'un fichier disque ouvert vers la mémoire centrale à partir de l'adresse DMA courante.

Reportez-vous à la fonction BDOS 44 pour définir le nombre d'enregistrement(s) à lire.

Les données renvoyées par la fonction BDOS 20 sont les suivantes :

— Code d'erreur dans le registre A:

Ce code est nul si l'opération de lecture s'est bien déroulée, sinon, le registre A contient un des codes d'erreur suivants :

#01 : End Of File rencontré

#09 : FCB invalide

#0A : Changement de disquette entre l'ouverture et la lecture

#FF : Erreur physique pendant la lecture.

— Code d'erreur physique dans le registre H:

Ce registre signifie qu'une erreur physique s'est produite uniquement quand le registre A contient #FF. Dans ce cas, H peut avoir une des deux valeurs suivantes :

#01 : Disk I/O Error

#04 : Invalid Drive Error

Si A <> #FF, le registre H contient le nombre de secteurs lus avant que ne se produise une erreur.

Fonction 21**WRITE SEQUENTIAL**

Ecriture d'un ou plusieurs enregistrement(s)

Entrée : C = #15
DE = Adresse FCB

Sortie : A = Code d'erreur
H = Erreur physique

Cette fonction transfère de 1 à 128 enregistrements de 128 octets de la mémoire centrale (à partir de l'adresse DMA courante) vers un fichier ouvert.

Reportez-vous à la fonction BDOS 44 pour définir le nombre d'enregistrement(s) à écrire.

Les données renvoyées par la fonction BDOS 21 sont les suivantes :

— Code d'erreur dans le registre A:

Ce code est nul si l'opération d'écriture s'est bien déroulée, sinon, le registre A contient un des codes d'erreur suivants :

#01 : Espace non disponible dans le répertoire

#02 : Bloc non disponible

#09 : FCB invalide

#0A : Changement de disquette entre l'ouverture et l'écriture

#FF : Erreur physique pendant la lecture.

— Code d'erreur physique dans le registre H:

Ce registre signifie qu'une erreur physique s'est produite uniquement quand le registre A contient #FF. Dans ce cas, H peut avoir une des quatre valeurs suivantes :

#01 : Disk I/O Error

#02 : Read-Only disk

#03 : Read-Only file

#04 : Invalid Drive Error

Si $A \neq \text{FF}$, le registre H contient le nombre de secteurs écrits avant que ne se produise une erreur.

Fonction 22

MAKE FILE

Création d'une entrée dans un répertoire

Entrée : C = #16

DE = Adresse FCB

Sortie : A = Code répertoire

H = Code d'erreur physique ou étendue

Cette fonction crée :

— une entrée dans le répertoire courant ;

— un XFCB si le répertoire courant autorise les mots de passe. Un mot de passe est alors affecté au fichier.

L'adresse du FCB est passée à la fonction 22 dans le registre DE. Le premier octet (octet 0) du FCB contient le numéro du lecteur de disquettes, les octets suivants (1 à 11) indiquent le nom du fichier et son type. L'octet 12 indique le numéro d'extension (souvent zéro). L'attribut f6 indique si un mot de passe doit ou non être attribué au fichier (f6=0 pour qu'aucun mot de passe ne soit attribué, f6=1 pour qu'un mot de passe soit attribué). Lorsque l'attribut f6 est positionné, le mot de passe doit être placé dans les huit premiers octets du buffer courant de DMA.

La fonction MAKE FILE renvoie un code d'erreur si le FCB fait référence à un nom de fichier déjà existant dans le répertoire courant.

MAKE FILE initialise également les champs « date de création » et « date de mise à jour » du fichier si :

- le nombre d'extension est nul ;
- le répertoire courant demande une telle datation.

En sortie, la fonction MAKE FILE retourne un code répertoire :

- nul dans le registre A si tout s'est bien passé ;
- égal à #FF s'il n'y a plus de place dans le répertoire.

Le registre H est nul dans ces deux cas. Si une erreur d'ordre physique ou étendu survient, un message est affiché et le programme appelant est avorté dans le cas où le mode d'erreur BDOS est le mode standard. Dans un autre cas, le retour vers l'appelant se fait avec un des codes d'erreur suivant :

- 01 : Erreur d'entrée/sortie disque ;
- 02 : Disque à lecture seule ;
- 04 : Lecteur invalide ;
- 08 : Le fichier existe déjà ;
- 09 : Caractère « ? » dans le nom du fichier ou dans le type.

Fonction 23

RENAME FILE

Renomme un fichier

MAKE FILE

Fonction 23

Entrée : C = #17

DE = Adresse FCB

Sortie : A = Code répertoire

H = Code d'erreur physique ou étendue

Cette fonction renomme un fichier ou un groupe de fichiers en passant par le FCB dont l'adresse se trouve dans le registre DE. Les seize premiers octets du FCB contiennent le nom du fichier à renommer, et les seize suivants, le nouveau nom du fichier. Lorsque l'accès au fichier nécessite un mot de passe, celui-ci doit être placé dans les huit premiers octets du buffer de DMA courant, ou encore doit avoir été déclaré comme mot de passe courant grâce à la fonction BDOS 106.

En sortie, RENAME FILE renvoie un code répertoire nul dans A si tout s'est bien passé, et un code répertoire égal à #FF si le fichier à renommer n'a pu être trouvé dans le répertoire courant. Le registre H est nul dans ces deux cas. Si une erreur physique ou étendue se produit, la fonction RENAME FILE réagit différemment en fonction du mode d'erreur BDOS. Si c'est le mode standard, un message d'erreur est envoyé à l'écran, et le programme est avorté. Sinon, un code d'erreur est retourné au programme appelant selon la convention suivante :

01 : Erreur d'entrée/sortie ;

02 : Disque à lecture seule ;

03 : Fichier à lecture seule ;

04 : Disque invalide ;

noir : ... du ...

07 : Erreur sur le mot de passe ;

08 : Le fichier existe déjà ;

09 : Caractère « ? » dans le nom ou dans le type.

Fonction 24**RETURN LOGIN VECTOR**

Identification du lecteur de disque

Entrée : C = #18

Sortie : HL = Vecteur LOGIN

niar : ...

Cette fonction indique quel est le lecteur de disque actif. La convention de codage du registre HL en sortie de fonction est la suivante : le bit 0 du registre HL représente le lecteur A, le bit 15 du registre HL représente le lecteur P. Un des bits de HL positionné à un indique quel est le lecteur actuellement actif.

Fonction 25**RETURN CURRENT DISK**

Identification du lecteur de disque par défaut

Entrée : C = #19

Sortie : A = Disque courant

Cette fonction indique le numéro du lecteur de disque par défaut. Pour ce faire, le registre A prend une valeur comprise entre 0 et 15, correspondant respectivement aux lecteurs A à P.

Fonction 26**SET DMA ADDRESS**

Initialisation de l'adresse d'accès direct à la mémoire (DMA)

Entrée : C = #1A

Sortie : Adresse de DMA

Sous CP/M +, une zone en mémoire est définie dans le but de stocker un enregistrement (ou plus) issu d'une lecture sur disque ou destiné à être écrit sur disque. La fonction BDOS 44 (SET MULTI-SECTOR COUNT) permet de définir la taille de cette zone mémoire. L'adresse de DMA la plus fréquente est #0080. C'est également l'adresse initialisée par la fonction BDOS 13. La fonction SET DMA ADDRESS modifie cette adresse qui reste ainsi initialisée jusqu'à ce qu'elle soit changée par un autre appel à la fonction 26 ou jusqu'à ce qu'un RESET se produise.

Fonction 27**GET ADDR**

Lecture de l'adresse du vecteur d'allocation

Entrée : C = #1B

Sortie : HL = Adresse du vecteur

Cette fonction permet de lire l'adresse de base du vecteur d'allocation du lecteur de disque courant. Ce vecteur se trouve en mémoire RAM et permet entre autres de déterminer la quantité de mémoire libre sur une disquette. Lorsqu'aucune erreur physique ne se produit, le registre HL contient l'adresse du vecteur. Dans le cas contraire, le registre HL contient #FFFF.

Fonction 28**WRITE PROTECT DISK**

Positionne un lecteur de disque en lecture seule

Entrée : C = #1C

Cette fonction permet de positionner temporairement (jusqu'au prochain « DRIVE RESET ») un lecteur de disquettes ou de disque dur en lecture seule. De cette manière, tout accès en écriture sur ce lecteur (pour renommer ou effacer un fichier, écrire des données, etc.) sera refusé.

Fonction 29**GET READ-ONLY VECTOR**

Identification des lecteurs autorisés en lecture seulement

Entrée : C = #1D

Sortie : HL = Vecteur d'identification

Cette fonction renvoie un vecteur dans HL dont chaque bit à un identifie un lecteur autorisé à lecture seule. Le bit de poids le plus faible correspond au lecteur A; le bit de poids le plus fort correspond au lecteur P.

Fonction 30**SET FILE ATTRIBUTES**

Modification des attributs d'un fichier

Entrée : C = #1E

DE = Adresse FCB

Sortie : A = Code répertoire

H = Code d'erreur physique ou étendu

Cette fonction permet de modifier les attributs d'un fichier et de positionner les derniers compteurs d'octets. Les attributs que peut modifier cette fonction sont les suivants :

- f1' à f4' ;
- Read-Only ;
- t1' ;

- system ;
- t2' ;
- archive ;
- t3'.

Le FCB accédé par l'adresse contenue dans le registre DE doit contenir le nom du fichier dont les attributs vont être modifiés ainsi que les attributs positionnés comme nécessaire. Si le fichier est protégé par un mot de passe, ce dernier doit se trouver dans les huit premiers octets du buffer de DMA ou avoir été spécifié comme mot de passe par défaut par la fonction 106.

L'attribut f6 indique si le compteur du dernier enregistrement doit être modifié (f6 = 0 pour que le compteur soit inchangé ; f6 = 1 pour que le compteur soit modifié).

En sortie, SET FILE ATTRIBUTES renvoie un code répertoire nul dans le registre A si tout s'est bien passé, et un code répertoire égal à #FF dans le registre A si le fichier à accéder n'a pu être trouvé dans le répertoire courant. Le registre H est nul dans ces deux cas. Si une erreur physique ou étendue se produit, la fonction SET FILE ATTRIBUTES réagit différemment en fonction du mode d'erreur BDOS. Si c'est le mode standard, un message d'erreur est envoyé à l'écran, et le programme est avorté. Sinon, un code d'erreur est retourné au programme appelant selon la convention suivante :

- 01 : Erreur d'entrée/sortie ;
- 02 : Disque à lecture seule ;
- 04 : Disque invalide ;
- 07 : Erreur sur le mot de passe ;
- 09 : Caractère « ? » dans le nom ou dans le type.

Fonction 31

GET ADDR

Lecture de l'adresse du DPB

Entrée : C = #1F

Sortie : HL = Adresse du DPB

Cette fonction lit l'adresse du DPB (Disk Parameter Block) du disque courant et la stocke dans le registre HL.

Si une erreur physique se produit alors que le mode d'erreur BDOS est un des modes de retour, la valeur #FFFF est renvoyée dans le registre HL. (Cf. Fonction 45).

Fonction 32

SET/GET USER CODE

Lecture/modification du numéro d'utilisateur courant

Entrée : C = #20

E = #FF (lecture) / numéro (écriture)

Sortie : A = code courant (dans le cas d'une lecture)
valeur quelconque dans le cas d'une écriture

Lorsque le registre E vaut #FF, le numéro d'utilisateur est renvoyé dans le registre A en sortie de la fonction. Si le registre E est différent de #FF, le numéro utilisateur devient celui indiqué dans le registre E (modulo 16).

Fonction 33

READ RANDOM

Lecture en accès direct dans un fichier

Entrée : C = #21
DE = Adresse FCB

Sortie : A = Code d'erreur
H = Code d'erreur physique

Cette fonction est identique à la lecture séquentielle, à ceci près que le pointeur de lecture est placé sur un enregistrement particulier avant d'effectuer la lecture. La position de lecture est indiquée dans les octets r0 (poids faible), r1 et r2 (poids fort) qui se trouvent dans le FCB à partir de la position 33. Avant d'utiliser la fonction READ RANDOM, le fichier concerné doit avoir été ouvert avec une dimension de base 0.

Lorsque le compteur de secteurs BDOS est plus grand que 1, la fonction READ RANDOM lit un nombre d'enregistrements en conséquence. Les champs r0, r1 et r2 sont automatiquement incrémentés.

En sortie de la fonction READ RANDOM, le registre A contient 0 si tout s'est déroulé correctement. Si ce n'est pas le cas, A contient un des codes d'erreur suivants :

- 01 : Fin de fichier rencontrée ;
- 02 : L'enregistrement courant ne peut être fermé ;
- 04 : Positionnement sur un enregistrement non initialisé ;
- 06 : Numéro d'enregistrement en dehors des limites permises (r2 > 3) ;
- 10 : Changement de disquette entre l'ouverture et la lecture ;
- 255 : Erreur physique.

Lorsqu'une erreur physique est détectée (A=255), le registre H contient 01 si une erreur d'entrée/sortie s'est produite et 04 si le lecteur de disquette ou de disque dur ne peut être accédé.

Fonction 34

WRITE RANDOM

Ecriture en accès direct dans un fichier

Entrée : C = #22
DE = Adresse FCB

Sortie : A = Code d'erreur
H = Code d'erreur physique

Cette fonction est identique à la précédente, à ceci près qu'elle provoque une écriture et non une lecture dans le fichier ouvert.

En sortie de la fonction WRITE RANDOM, le registre A contient 0 si tout s'est déroulé correctement. Si ce n'est pas le cas, A contient un des codes d'erreur suivants :

- 02 : Bloc de données non valide ;
- 03 : L'enregistrement courant ne peut être fermé ;
- 05 : Pas de place dans le répertoire ;
- 06 : Numéro d'enregistrement en dehors des limites permises ($r2 > 3$) ;
- 10 : Changement de disquette entre l'ouverture et la lecture ;
- 255 : Erreur physique.

Lorsqu'une erreur physique est détectée ($A = 255$), le registre H contient :

- 01 si une erreur d'entrée/sortie s'est produite ;
- 02 si le disque est à lecture seule ;
- 03 si le fichier est à lecture seule ;
- 04 si le lecteur de disquette ne peut être accédé.

Fonction 35

COMPUTE FILE SIZE

Détermination de la taille d'un fichier

Entrée : C = # 23
DE = Adresse FCB

Sortie : A = Code d'erreur
H = Code d'erreur physique

Cette fonction détermine la taille d'un fichier. Cette taille est la taille réelle si le fichier est accédé en séquentiel, mais elle est inférieure à la taille réelle si le fichier est accédé en direct. Pour ce faire, le pointeur d'enregistrement du FCB (passé en entrée dans DE) est positionné sur le dernier enregistrement + 1 du fichier.

En sortie, le registre A est nul si la fonction s'est déroulée correctement. Le registre A contient #FF si le fichier spécifié n'a pas été trouvé. Dans ces deux cas, le registre H est positionné à zéro. Si une erreur physique se produit, la fonction COMPUTE FILE SIZE réagit différemment selon le mode d'erreur BDOS courant. Si ce mode est le mode par défaut, un message d'erreur est affiché à l'écran, et le programme appelant est avorté. Dans un autre cas, le registre H peut contenir une des valeurs suivantes :

- 01 si une erreur d'entrée/sortie disque s'est produite ;
- 04 si le lecteur de disquettes est invalide.

Fonction 36**SET RANDOM RECORD**

Numéro du prochain enregistrement accédé

Entrée : C = #24
DE = Adresse FCB

Sortie : r0, r1 et r2 du FCB

Cette fonction indique dans les champs r1, r2 et r3 du FCB la position du prochain enregistrement accédé par une instruction de lecture ou d'écriture séquentielle.

Fonction 37**RESET DRIVE**

Initialisation des lecteurs de disquettes

Entrée : C = #25
DE = Numéro du lecteur de disquette

Sortie : A = #00

Cette fonction positionne les lecteurs de disquettes dans un état initial. Le numéro du lecteur de disquettes est indiqué dans le registre DE en entrée de la fonction. Le bit de poids faible de ce registre correspond au lecteur A:.. Le bit de poids fort correspond au lecteur P:.. Lorsqu'un bit est à un, il désigne le lecteur à réinitialiser.

Fonction 38**ACCESS DRIVE**

Non utilisé sous CP/M +

Fonction 39**FREE DRIVE**

Non utilisé sous CP/M +

Fonction 40**WRITE RANDOM WITH ZERO FILL**

Ecriture en accès direct dans un fichier

Entrée : C = #28
DE = Adresse FCB

Sortie : A = Code d'erreur
H = Code d'erreur physique

Identique à la fonction 34 (WRITE RANDOM) à ceci près qu'une zone mémoire est initialisée à 0 avant l'accès à la disquette. Si cette fonction a été utilisée pour créer un fichier, les enregistrements non écrits accédés par la fonction 33 (READ RANDOM) contiendront des zéros. Si le fichier avait été créé par la fonction 34 (WRITE RANDOM), ces mêmes enregistrements contiendraient des valeurs quelconques non prévisibles.

Fonction 41**TEST AND WRITE RECORD**

Non utilisé sous CP/M +

Fonction 42**LOCK RECORD**

Non utilisé sous CP/M +

Fonction 43**UNLOCK RECORD**

Non utilisé sous CP/M +

Fonction 44**SET MULTI-SECTOR COUNT**

Initialisation du nombre de secteurs lus ou écrits en accès direct.

Entrée : C = #2C

E = Nombre de secteurs

Sortie : A = Code d'erreur

Cette fonction définit le nombre de secteurs lus à chaque accès direct à un fichier en lecture ou en écriture. Le registre E contient le nombre de secteurs entre 1 et 128. Ce nombre reste ainsi défini jusqu'à ce que la fonction UNLOCK RECORD soit appelée.

En sortie, le registre A contient 0 si le registre E a été passé avec une valeur comprise entre 1 et 128. Sinon, le registre A contient #FF.

Fonction 45**SET BDOS ERROR MODE**

Définition du mode d'erreur du BDOS.

Entrée : C = #2D

E = Mode d'erreur BDOS

Sortie : Aucune

Cette fonction définit le mode d'erreur du BDOS en fonction de la valeur qui se trouve dans le registre E au moment de l'appel :

E	Signification
#FF	Renvoi du code d'erreur
#FE	Renvoi et affichage du code d'erreur
Autre	Mode d'erreur par défaut
Valeur	

Lorsque le mode d'erreur par défaut est actif, chaque erreur provoque l'affichage d'un message système (qui identifie l'erreur) et la terminaison prématurée du programme.

Lorsque le mode d'erreur est initialisé à « Renvoi du code d'erreur », le code de l'erreur est placé dans le registre H et retourné au programme sans aucun affichage.

Lorsque le mode d'erreur est initialisé à « Renvoi et affichage du code d'erreur », chaque erreur provoque l'affichage d'un message système (qui identifie l'erreur) et le retour au programme.

Fonction 46**GET DISK FREE SPACE**

Fonction 46

Espace disponible sur un disque.

Entrée : C = #2E

E = Lecteur (0=A, 1=B, etc., 15=P)

Fonction 46

Sortie : les trois premiers octets du buffer DMA courant

A = Flag d'erreur

H = Erreur physique

Fonction 46

Cette fonction renvoie le nombre de secteurs (128 octets) de disquette non utilisés. Le résultat se trouve dans les trois premiers octets du buffer DMA. Le premier représente le poids faible, le second le poids intermédiaire et le troisième le poids fort.

En sortie, le registre A vaut 0 si la fonction a pu s'exécuter et un code d'erreur dans le cas contraire. Dans ce cas, le registre H contient l'une des deux valeurs suivantes :

#01 = Erreur d'entrée/sortie disque

#04 = Lecteur invalide.

Fonction 47**CHAIN TO PROGRAM**

Fonction 47

Exécution chaînée d'un programme.

Entrée : C = #2F

E = Flag de chaînage

Sortie : Aucune

Cette fonction permet de lancer l'exécution d'un programme à partir du programme courant. Pour ce faire, le buffer DMA doit contenir une ligne de commande terminée par un octet nul.

Lorsque le registre E vaut 0 en entrée de la fonction, le programme est appelé avec les valeurs par défaut du lecteur et du numéro d'utilisateur. Dans le cas contraire, le lecteur et le numéro d'utilisateur sont issus des valeurs par défaut du CCP.

Fonction 48**FLUSH BUFFERS**

Purge des buffers d'écriture disque.

Entrée : C = #30

E = Flag d'effacement

Sortie : A = Flag d'erreur

H = Erreur physique

Cette fonction force l'écriture du ou des buffer(s) de données non vides. Ce (ou ces) buffer(s) est (sont) effacé(s) si le registre E vaut # FF en entrée.

En sortie, le registre A vaut 0 si la fonction a pu s'exécuter. Dans le cas contraire, un code d'erreur physique est éventuellement retourné dans le registre H (en fonction du mode d'erreur courant, voir fonction 45) :

H	Signification
#01	Erreur d'entrée/sortie disque
#02	Disque en lecture seule
#04	Lecteur invalide

Fonction 49

GET/SET SYSTEM CONTROL BLOCK

Lecture ou écriture du SCB (*System Control Block*)

Entrée : C = #31

DE = Adresse du SCBPB (*System Control Block Parameter Block*)

Sortie : A = Octet éventuellement retourné

HL = Mot éventuellement retourné

Cette fonction permet d'accéder (en lecture ou en écriture) aux données qui se trouvent dans le SCB (*System Control Block*). Reportez-vous à la Partie 3, Chap. 4.2.1.2 pour prendre connaissance de la structure du SCB.

Les registres D et E pointent sur une structure qui identifie l'opération effectuée dans le SCB. Cette structure est la suivante :

Nom	Dimension	Signification
Déplacement	Octet	Déplacement du ou des octet(s) à accéder par rapport au début du SCB
Opération	Octet	#FF pour initialiser un octet #FE pour initialiser un mot #00 pour lire une valeur les valeurs #01 à #FD ne doivent pas être utilisées
Valeur	Mot	Octet ou mot à écrire pour les initialisations

La table suivante résume le type d'accès pouvant être effectué à chaque valeur du SCB :

Octet	Type d'accès
#00 à #04	RO
#05	RO
#06 à #09	RW
#0A à #0F	RO
#10 à #11	RW

Octet	Type d'accès
#12 à #19	RO
#1A	RW
#1B	RO
#1C	RW
#1D à #21	RO
#22 à #23	RW
#24 à #25	RW
#26 à #27	RW
#28 à #29	RW
#2A à #2B	RW
#2C	RW
#2D	RO
#2E	RW
#2F	RW
#30 à #32	RO
#33 à #34	RW
#35 à #36	RO
#37	RW
#38	RW
#39 à #3B	RO
#3C à #3D	RO
#3E	RO
#3F à #43	RO
#44	RO
#45 à #49	RO
#4A	RW
#4B	RW
#4C à #4F	RW
#50	RW
#51	RO
#52 à #56	RO
#57	RO
#58 à #59	RW
#5A	RW
#5B	RW
#5C	RW
#5D à #5E	RO
#5F à #63	RO

RO = Read Only (lecture seule)

RW = Read/Write (Lecture et Ecriture).

Type d'accès

Fonction 50

DIRECT BIOS CALLS

Activation d'une fonction du BIOS.

Entrée : C = #32

DE = Adresse du bloc de paramètres du BIOS (BIOS PB)

Sortie : Données renvoyées par le BIOS.

Cette fonction permet d'accéder aux fonctions du BIOS. Le numéro de la fonction à accéder et la valeur des registres en entrée sont spécifiés dans le bloc de paramètres (BIOSPB) dont l'adresse se trouve dans le registre DE en entrée de la fonction. Le bloc de paramètres doit avoir la structure suivante :

Nom	Dimension	Signification
Fonction	Octet	Numéro de la fonction BIOS à accéder
Registre A	Octet	Contenu du registre A
Registres BC	Mot	Contenu des registres B et C (*)
Registres DE	Mot	Contenu des registres D et E (*)
Registres HL	Mot	Contenu des registres H et L (*)

(*) Le premier octet va dans le second registre. Le second octet va dans le premier registre.

Fonction 59

LOAD OVERLAY

Chargement d'un bloc de recouvrement (overlay)

Entrée : C = #3B

DE = Adresse d'un FCB

Sortie : A = Code d'erreur

H = Erreur physique

Cette fonction ne peut être utilisée que par les programmes résidents. Elle charge en mémoire un module relogeable ou absolu. Le FCB spécifié dans DE doit être ouvert avant d'activer la fonction BDOS 59.

Le registre A est nul en sortie de la fonction si elle a pu s'exécuter. Dans le cas contraire, le registre A contient un code d'erreur et le registre H un code d'erreur physique. Reportez-vous à la fonction BDOS 20 pour avoir la liste des erreurs possibles (voir Partie 3, Chap. 4.2.2 p. 10).

Fonction 60

CALL RESIDENT SYSTEM EXTENSION

Appel d'un programme RSX.

Entrée : C = #3C

DE = RSX PB (adresse du bloc de paramètres du RSX)

Sortie : A = Code d'erreur

H = Erreur physique

Cette fonction permet d'activer une des fonctions d'une RSX. Cette fonction ainsi que les paramètres passés sont définis dans un bloc de paramètres dont la structure est la suivante :

Nom	Dimension	Signification
Fonction	Octet	Numéro de la fonction RSX à accéder
NbParam	Octet	Nombre de paramètres mots
P1	Mot	Premier paramètre
...	...	...
Pn	Mot	Paramètre n

Les fonctions RSX 0 à 127 sont à la disposition de l'utilisateur. Les fonctions RSX 128 à 255 sont réservées au système.

Les registres A et H contiennent #FF en sortie si le RSX spécifié n'est pas accessible.

Fonction 98

FREE BLOCKS

Libération des blocs temporaires.

Entrée : C = #62

Sortie : A = Code d'erreur
H = Erreur physique

Cette fonction libère tous les blocs occupés par des fichiers temporaires (c'est-à-dire des fichiers ouverts en écriture mais non fermés) sur tous les lecteurs connectés. Assurez-vous que vos fichiers ont bien été fermés avant d'appeler cette fonction, sans quoi, les données qu'ils contiennent seront perdues.

Le registre A contient la valeur 0 en sortie de la fonction si elle a pu s'exécuter. Dans le cas contraire, le registre A contient la valeur #FF et le registre H (en fonction du mode d'erreur) la valeur #04 (lecteur invalide).

Fonction 99

TRUNCATE FILE

Réduction d'un fichier à accès direct.

Entrée : C = #63
DE = Adresse du FCB

Sortie : A = Code répertoire
H = Erreur étendue ou physique

Cette fonction initialise le dernier enregistrement d'un fichier au numéro d'enregistrement qui se trouve dans le FCB pointé par DE. Le fichier représenté par le FCB ne doit pas être ouvert.

La valeur retournée dans le registre A est 0 si la fonction s'est exécutée, #FF si le fichier n'existe pas ou si le numéro d'enregistrement est fantaisiste. Le registre H contient (en fonction du mode d'erreur) une des valeurs suivantes :

Valeur	Signification
#01	Erreur d'entrée/sortie disque
#02	Disque à lecture seule
#03	Fichier à lecture seule
#04	Lecteur invalide
#07	Erreur dans le mot de passe du fichier
#09	? dans le champ nom du fichier ou type du fichier

Fonction 100

SET DIRECTORY LABEL

Modification d'une entrée de répertoire.

Entrée : C = #64
DE = Adresse FCB

Sortie : A = Code répertoire
H = Erreur étendue ou physique

Cette fonction crée un label de répertoire ou met à jour un label de répertoire existant. En entrée de la fonction, le registre DE doit pointer sur l'adresse d'une FCB dans lequel se trouvent les champs nom, type et extension du label de répertoire.

L'octet 12 du FCB contient les informations suivantes :

Bit	Signification
7	Nécessité de mot de passe pour les fichiers protégés
6	Accès aux champs date et heure
5	Mise à jour des champs date et heure
4	Création des champs date et heure
0	Affectation d'un nouveau mot de passe

Dans le cas où le fichier est protégé par un mot de passe, ce dernier doit se trouver dans les 8 premiers octets du DMA ou avoir été spécifié dans la fonction BDOS 106.

Si le bit 0 de l'octet 12 du FCB vaut 1, le nouveau mot de passe doit avoir été placé dans le second jeu de 8 octets du DMA.

L'utilitaire INITDIR doit avoir été exécuté avant d'activer cette fonction pour permettre l'utilisation des champs date et heure.

Le registre A vaut 0 en sortie si la fonction s'est exécutée. Il vaut #FF dans le cas contraire, et le registre H contient un des codes suivants (si le mode d'erreur le permet) :

Valeur	Signification
#01	Erreur d'entrée/sortie disque
#02	Disque à lecture seule
#04	Lecteur invalide
#07	Erreur dans le mot de passe du fichier

Fonction 101

RETURN DIRECTORY LABEL DATA

Lecture d'un label de répertoire.

Entrée : C = #65

E = Lecteur (0 = A, 1 = B, ..., 15 = P)

Sortie : A = Octet d'information du répertoire

H = Erreur physique

Cette fonction renvoie dans le registre A le label de répertoire du lecteur spécifié. Les bits de ce registre ont la signification suivante :

Bit	Signification
7	Nécessité de mot de passe pour les fichiers protégés
6	Accès aux champs date et heure
5	Mise à jour des champs date et heure
4	Création des champs date et heure
0	Affectation d'un nouveau mot de passe

Le registre A vaut 0 en sortie si le label de répertoire est inexistant. Il vaut #FF si une erreur s'est produite. Dans ce cas, le registre H contient un des codes suivants (si le mode d'erreur le permet) :

Valeur	Signification
#01	Erreur d'entrée/sortie disque
#04	Lecteur invalide

Fonction 102

READ FILE DATE STAMPS AND PASSWORD MODE

Lecture des champs date et heure et du type de mot de passe.

Entrée : C = #66

DE = Adresse FCB

Sortie : A = Code répertoire

H = Erreur physique

Cette fonction renvoie dans le FCB les informations date et heure et le type de l'éventuel mot de passe utilisé pour le fichier spécifié :

Octet(s)	Signification
12	Mot de passe Bit 7 = en lecture Bit 6 = en écriture Bit 4 = en effacement
24-27	Création ou lecture des champs date et heure
28-31	Modification des champs date et heure

La valeur retournée dans le registre A est 0 si la fonction s'est exécutée, #FF si le fichier n'existe pas. Le registre H contient (en fonction du mode d'erreur) une des valeurs suivantes :

Valeur	Signification
#01	Erreur d'entrée/sortie disque
#04	Lecteur invalide
#09	? dans le champ nom du fichier ou type du fichier

Fonction 103

WRITE FILE XFCB

ou d'un XFCB

Entrée : C = #67

DE = Adresse FCB

Sortie : A = Code répertoire

H = Erreur physique

Cette fonction crée un XFCB ou met à jour un XFCB existant. En entrée, le registre DE pointe sur un FCB pour lequel les champs nom, type et extension ont été initialisés. Le champ extension spécifie le type de mot de passe utilisé :

bit 7 = en lecture

bit 6 = en écriture

bit 4 = en effacement

bit 0 = affectation d'un nouveau mot de passe

Dans le cas où le fichier est protégé par un mot de passe, ce dernier doit se trouver dans les 8 premiers octets du DMA ou avoir été spécifié dans la fonction BDOS 106.

Si le bit 0 de l'octet 12 du FCB vaut 1, le nouveau mot de passe doit avoir été placé dans le second jeu de 8 octets du DMA.

La valeur retournée dans le registre A est 0 si la fonction s'est exécutée, #FF si le fichier n'existe pas ou si aucun label n'existe pour le répertoire

spécifié. Le registre H contient (en fonction du mode d'erreur) une des valeurs suivantes :

Valeur	Signification
#01	Erreur d'entrée/sortie disque
#02	Disque à lecture seule
#04	Lecteur invalide
#07	Erreur dans le mot de passe du fichier
#09	? dans le champ nom du fichier ou type du fichier

Fonction 104

SET DATE AND TIME

Initialisation de la date et de l'heure système.

Entrée : C = #68

DE = Adresse d'un buffer

Sortie : Aucune

Cette fonction initialise les date et heure courantes. Le registre DE pointe sur un buffer qui contient les informations suivantes :

Octet(s)	Information
0 et 1	Date (1 = 1/1/1978)
2	Heure (en BCD)
3	Minute (en BCD)

Fonction 105

GET DATE AND TIME

Lecture de la date et de l'heure système.

Entrée : C = #69

DE = Adresse d'un buffer

Sortie : A = Secondes

Cette fonction renvoie les date et heure systèmes dans le buffer pointé par le registre DE. En sortie de la fonction, ce buffer contient les informations suivantes :

Octet(s)	Information
0 et 1	Date (1 = 1/1/1978)
2	Heure (en BCD)
3	Minute (en BCD)

Fonction 106

SET DEFAULT PASSWORD

Définition du mot de passe par défaut.

Entrée : C = #6A
DE = Adresse du mot de passe

Sortie : Aucune

Cette fonction définit le mot de passe par défaut. En entrée de la fonction, le registre DE doit pointer sur une zone mémoire de 8 octets dans laquelle se trouve le mot de passe.

Fonction 107

RETURN SERIAL NUMBER

Lecture du numéro de série de CP/M Plus.

Entrée : C = #6B
DE = Pointeur sur le buffer numéro de série

Sortie : Numéro de série initialisé.

Cette fonction renvoie le numéro de série de CP/M Plus dans un buffer de 6 octets pointé par le registre DE en entrée de la fonction.

Fonction 108

GET/SET PROGRAM RETURN CODE

Lecture ou initialisation du code de retour.

Entrée : C = #6CH
DE = #FFFF (Lecture)
ou Code de retour (Ecriture)

Sortie : HL = Code retourné par le programme (le cas échéant)

Cette fonction lit ou définit un code de retour. Si le registre DE vaut #FFFF en entrée de la fonction, le code de retour du programme est renvoyé dans le registre HL. Si le registre DE est différent de #FFFF en entrée de la fonction, il est interprété comme le code de retour du programme.

Fonction 109

GET/SET CONSOLE MODE

Lecture ou initialisation du mode d'écran.

Entrée : C = #6D
DE = #FFFF (lecture)
ou Mode (écriture)

Sortie : HL = Mode d'écran (le cas échéant)

Cette fonction lit ou initialise le mode d'écran. Si le registre DE vaut #FFFF en entrée de la fonction, le mode d'écran est retourné dans le registre HL selon les conventions suivantes :

Bit	Signification
0	si 0 : La fonction BDOS 11 renseigne la frappe d'une touche quelconque

Bit	Signification
1	si 1 : La fonction BDOS 11 renseigne uniquement la frappe de la séquence Ctrl C si 0 : Les touches Ctrl S et Ctrl Q permettent (respectivement) de stopper et de reprendre l'affichage
2	si 1 : Les touches Ctrl S et Ctrl Q sont dévalidées si 0 : Fonctions de sortie normales
3	si 1 : Dévalidation des touches Tab pour les fonctions BDOS 2, 9 et 11 et de l'appui sur Ctrl P si 0 : Validation du Ctrl C
8 et 9	si 1 : Dévalidation du Ctrl C Etat de la console pour les RSX qui utilisent la redirection : 0 0 : Etat conditionnel 0 1 : Etat False 1 0 : Etat True 1 1 : Redirection

Fonction 110**GET/SET OUTPUT DELIMITER**

Fonction 108

Lecture ou initialisation de délimiteur de sortie.

Entrée : C = #6E
DE = #FFFF (lecture)
ou E = Délimiteur de sortie (écriture)

Sortie : A = Délimiteur de sortie (le cas échéant)

Cette fonction lit ou initialise le délimiteur de sortie. Si le registre DE contient #FFFF en entrée de la fonction, le délimiteur courant est retourné dans le registre A. Si le registre E ne contient pas la valeur #FF, il représente le code ASCII du nouveau délimiteur.

Le délimiteur par défaut est le signe dollar (\$).

Fonction 111**PRINT BLOCK**

Fonction 109

Envoi d'une chaîne vers CONOUT:.

Entrée : C = #6F
DE = Adresse CCB (*Character Control Block*)

Sortie : Aucune

Cette fonction envoie la chaîne pointée par le CCB dont l'adresse se trouve dans le registre DE vers le périphérique CONOUT:.

Le format du CCB doit être le suivant :

Octet	Contenu
0 et 1	Adresse de la chaîne
2 et 3	Longueur de la chaîne

Fonction 112**LIST BLOCK**

Envoi d'une chaîne vers LST:.

Entrée : C = #70

DE = Adresse CCB

Sortie : Aucune

Cette fonction envoie la chaîne pointée par le CCB dont l'adresse se trouve dans le registre DE vers le périphérique LST:.

Le format du CCB doit être le suivant :

Octet	Contenu
0 et 1	Adresse de la chaîne
2 et 3	Longueur de la chaîne

Fonction 152**PARSE FILENAME**

Extraction d'un nom de fichier.

Entrée : C = #98

DE = Adresse d'un buffer

Sortie : HL = Code de retour

Cette fonction crée un FCB à partir d'une chaîne ASCII de 128 caractères au maximum qui doit contenir une spécification de fichier sous la forme suivante :

[<Lecteur:>]<Nom>[.<Type>][;<Passe>]

où <Lecteur> est un nom d'unité,

<Nom> un nom de fichier,

<Type> un type de fichier,

et <Passe> un mot de passe.

Le registre DE pointe sur un buffer qui doit contenir les informations suivantes :

Taille	Information
2	Adresse de la chaîne
2	Adresse du FCB

Les caractères blancs et tabulations sont éliminés de la chaîne, et le FCB est rempli en extrayant le premier nom de fichier valide dans la chaîne. Ce nom doit être suivi d'un des délimiteurs suivants :

Espace, Tab, Carriage Return, Nul, Point Virgule (;), Egal à (=), Inférieur à (<), Supérieur à (>), Deux points (:), Point (.), Virgule (,), Barre verticale (|), Crochet ouvrant ([), Crochet fermant (]).

La rencontre d'un autre caractère de code ASCII inférieur à 31 produit une erreur.

En sortie de la fonction, le format du FCB est le suivant :

Octet(s)	Signification
0	Lecteur (0 = par défaut, 1 = A:, 2 = B:, ...)
1 à 8	Nom du fichier. Les caractères sont convertis en majuscules. Le nom est éventuellement complété avec des blancs.
9 à 11	Extension, éventuellement complétée avec des blancs. Chaque caractère est converti en majuscule.
12 à 15	Octets nuls.
16 à 23	Mot de passe ou suite de blancs si aucun mot de passe n'est utilisé. Chaque caractère est converti en majuscule.
24 à 31	Réservé.

Une erreur se produit si le nom du fichier dépasse 8 caractères, si l'extension dépasse 3 caractères, ou si le mot de passe dépasse 8 caractères. Le registre HL est alors initialisé à #FFFF.

Si aucune erreur ne s'est produite, HL contient 0 si la chaîne a été entièrement parcourue et la position du prochain délimiteur, blanc, ou tabulation dans le cas contraire.

[<Nom>] [<Pass>]

Table

Information	Table
	2
	2

Les caractères blancs et tabulation sont éliminés de la chaîne et la position du prochain délimiteur, blanc, ou tabulation est renvoyée en extrant le contenu du registre HL.

Exemple :

Si un caractère de code ASCII inférieur