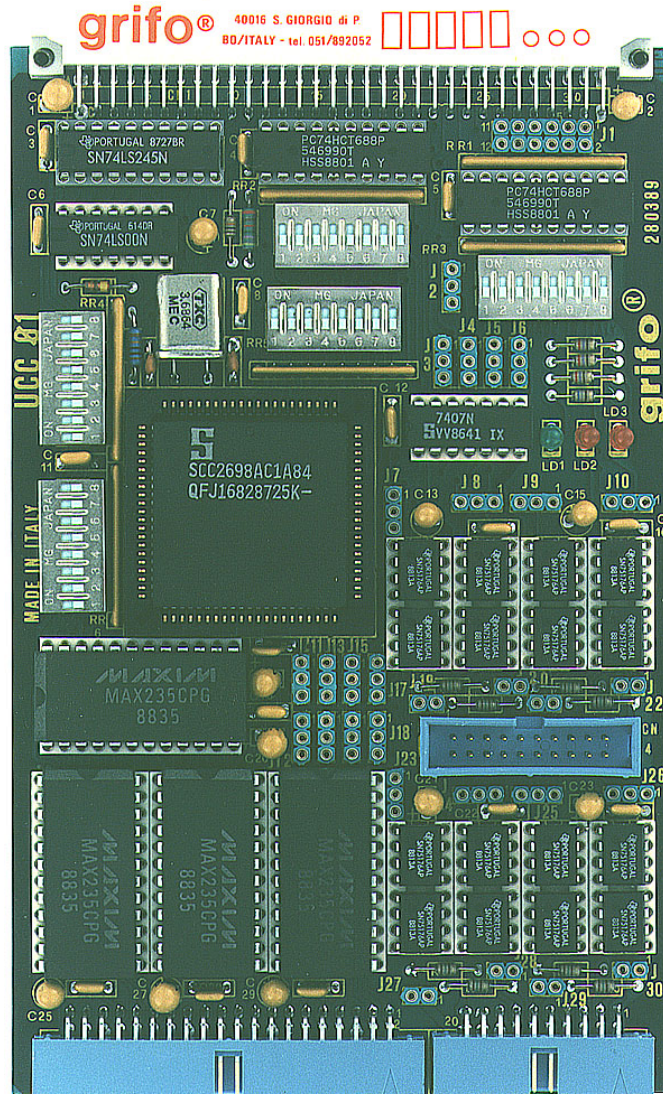


# UCC 01

UART Communication Card

8 Serial Lines

## MANUALE UTENTE



**grifo**<sup>®</sup>

ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6  
40016 San Giorgio di Piano  
(Bologna) ITALY

E-mail: [grifo@grifo.it](mailto:grifo@grifo.it)

<http://www.grifo.it>

<http://www.grifo.com>

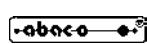
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661



UCC 01

Rel. 2.00

Edizione 10 Luglio 1989

 , GPC<sup>®</sup>, grifo<sup>®</sup>, sono marchi registrati della ditta grifo<sup>®</sup>



# UCC 01

**UART Communication Card**

**8 Serial Lines**

## MANUALE UTENTE

Terminale Video Intelligente con ingombro del tipo Singola Europa. Uscita su 8 indipendenti linee seriali in RS 232 o RS 422-485. Per ogni linea: buffer di 4 caratteri; comunicazione asincrona; Baud rate (da 50 a 38.4K baud), parità, stop bit e lunghezza dato programmabili via software; 3 Dip Switch. BUS a 8 bit; indirizzamento esteso.

**grifo**<sup>®</sup>

ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6  
40016 San Giorgio di Piano  
(Bologna) ITALY

E-mail: [grifo@grifo.it](mailto:grifo@grifo.it)

<http://www.grifo.it>

<http://www.grifo.com>

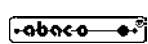
Tel. +39 051 892.052 (r.a.) FAX: +39 051 893.661



UCC 01

Rel. 2.00

Edizione 10 Luglio 1989

 , GPC<sup>®</sup>, grifo<sup>®</sup>, sono marchi registrati della ditta grifo<sup>®</sup>

## Vincoli sulla documentazione **grifo**<sup>®</sup> Tutti i Diritti Riservati

Nessuna parte del presente manuale può essere riprodotta, trasmessa, trascritta, memorizzata in un archivio o tradotta in altre lingue, con qualunque forma o mezzo, sia esso elettronico, meccanico, magnetico ottico, chimico, manuale, senza il permesso scritto della **grifo**<sup>®</sup>.

### IMPORTANTE

Tutte le informazioni contenute sul presente manuale sono state accuratamente verificate, ciononostante **grifo**<sup>®</sup> non si assume nessuna responsabilità per danni, diretti o indiretti, a cose e/o persone derivanti da errori, omissioni o dall'uso del presente manuale, del software o dell' hardware ad esso associato.

**grifo**<sup>®</sup> altresì si riserva il diritto di modificare il contenuto e la veste di questo manuale senza alcun preavviso, con l' intento di offrire un prodotto sempre migliore, senza che questo rappresenti un obbligo per **grifo**<sup>®</sup>.

Per le informazioni specifiche dei componenti utilizzati sui nostri prodotti, l'utente deve fare riferimento agli specifici Data Book delle case costruttrici o delle seconde sorgenti.

### LEGENDA SIMBOLI

Nel presente manuale possono comparire i seguenti simboli:



Attenzione: Pericolo generico

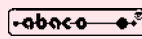


Attenzione: Pericolo di alta tensione



Attenzione: Dispositivo sensibile alle cariche elettrostatiche

### MARCHI REGISTRATI

, GPC<sup>®</sup>, **grifo**<sup>®</sup> : sono marchi registrati della **grifo**<sup>®</sup>.

Altre marche o nomi di prodotti sono marchi registrati dei rispettivi proprietari.

## INTRODUZIONE

L'uso di questi dispositivi è rivolto - **IN VIA ESCLUSIVA** - a personale specializzato.

Questo prodotto non è un **componente di sicurezza** così come definito dalla direttiva **98-37/CE**.



I pin della scheda non sono dotati di protezione contro le cariche elettrostatiche. Visto che esiste un collegamento diretto tra numerosi pin della scheda ed i rispettivi pin dei componenti di bordo e che quest'ultimi sono sensibili ai fenomeni ESD, il personale che maneggia la scheda è invitato a prendere tutte le precauzioni necessarie per evitare i possibili danni che potrebbero derivare dalle cariche elettrostatiche.

Scopo di questo manuale è la trasmissione delle informazioni necessarie all'uso competente e sicuro dei prodotti. Esse sono il frutto di un'elaborazione continua e sistematica di dati e prove tecniche registrate e validate dal Costruttore, in attuazione alle procedure interne di sicurezza e qualità dell'informazione.

I dati di seguito riportati sono destinati - **IN VIA ESCLUSIVA** - ad un'utenza specializzata, in grado di interagire con i prodotti in condizioni di sicurezza per le persone, per la macchina e per l'ambiente, interpretando un'elementare diagnostica dei guasti e delle condizioni di funzionamento anomale e compiendo semplici operazioni di verifica funzionale, nel pieno rispetto delle norme di sicurezza e salute vigenti.

Le informazioni riguardanti installazione, montaggio, smontaggio, manutenzione, aggiustaggio, riparazione ed installazione di eventuali accessori, dispositivi ed attrezzature, sono destinate - e quindi eseguibili - sempre ed in via esclusiva da personale specializzato avvertito ed istruito, o direttamente dall'**ASSISTENZA TECNICA AUTORIZZATA**, nel pieno rispetto delle raccomandazioni trasmesse dal costruttore e delle norme di sicurezza e salute vigenti.

I dispositivi non possono essere utilizzati all'aperto. Si deve sempre provvedere ad inserire i moduli all'interno di un contenitore a norme di sicurezza che rispetti le vigenti normative. La protezione di questo contenitore non si deve limitare ai soli agenti atmosferici, bensì anche a quelli meccanici, elettrici, magnetici, ecc.

Per un corretto rapporto coi prodotti, é necessario garantire leggibilità e conservazione del manuale, anche per futuri riferimenti. In caso di deterioramento o più semplicemente per ragioni di approfondimento tecnico ed operativo, consultare direttamente l'Assistenza Tecnica autorizzata.

Al fine di non incontrare problemi nell'uso di tali dispositivi, é conveniente che l'utente - **PRIMA DI COMINCIARE AD OPERARE** - legga con attenzione tutte le informazioni contenute in questo manuale. In una seconda fase, per rintracciare più facilmente le informazioni necessarie, si può fare riferimento all'indice generale e all'indice analitico, posti rispettivamente all'inizio ed alla fine del manuale.

## 1. INTRODUZIONE

Questo manuale fornisce tutte le informazioni hardware e software per consentire all'utente il miglior utilizzo della scheda Uart Communication Card UCC 01.

## 2. CARATTERISTICHE GENERALI DELLA SCHEDA

La scheda **UCC 01** è un potentissimo modulo dedicato alle comunicazioni seriali per il **BUS Industriale ABACO(R)** ed offre alla CPU notevoli possibilità di interfacciamento ad innumerevoli dispositivi esterni.

La scheda, in formato standard Europa da **100x160 mm**, può gestire separatamente **8** linee seriali con protocollo e velocità di comunicazione settabile da software; ognuna delle **8** linee gestisce automaticamente i **4** segnali canonici per la comunicazione seriale: **TxD, RxD, CTS, RTS**.

Inoltre ogni linea di comunicazione può essere utilizzata in **RS 232** oppure in **RS 422** od **RS485**.

Dal punto di vista del protocollo di comunicazione può essere selezionata una comunicazione asincrona con:

- Baud rate settabile tra 50 e 38,4 Kbaud
- Stop bit selezionabili tra 1; 1,5 e 2 bit
- Lunghezza della parola programmabile da 5 a 8 bit
- Buffer di ricezione di 4 caratteri
- Parità pari, dispari e forzata

Tre dip switches da 8 vie sono leggibili da software e danno, all'utente, la possibilità di poter settare, a livello scheda, delle condizioni particolari quali baud rate, protocollo di comunicazione ed altro, da gestire poi via firmware.

La **UCC 01** occupa uno spazio di indirizzamento di **64** byte ed in tali indirizzi sono mappati tutti i registri che consentono la programmazione e la gestione della scheda via software.

Questi indirizzi possono essere allocati nello spazio di indirizzamento del **BUS ABACO(R)** tramite una coppia di dip switches a **8** vie.

La scheda può generare due segnali di interrupt in corrispondenza del verificarsi di particolari condizioni definibili sempre tramite apposita programmazione software.

Qualora la configurazione completa della scheda risultasse superflua per l'applicazione da realizzare, e' possibile acquistare la scheda con le sole sezioni effettivamente utilizzate, riducendo così i costi ed aumentando la competitività dell'impianto.



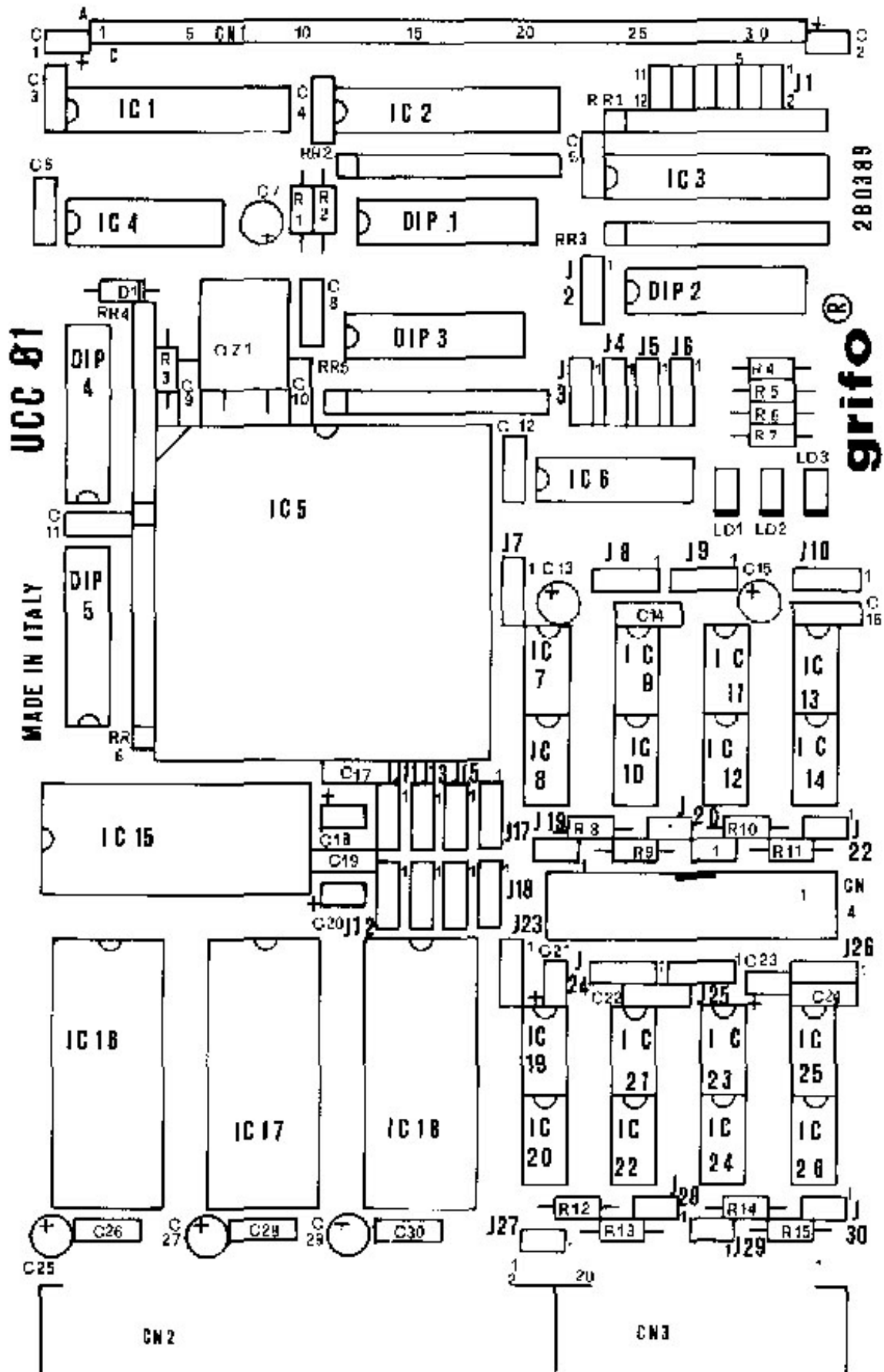


Fig. 2-1: Pianta componenti

## 2.1. Uart di Bordo

La scheda **UCC 01** e' basata sull'**UART SCC2698** che consente di gestire separatamente **8** linee seriali con protocollo di comunicazione, velocita' di comunicazione e generazione interrupts completamente settabile da software.

Fondamentalmente tale **UART** ha le seguenti caratteristiche per ognuna delle **8** linee:

- buffer di **4** caratteri in ricezione
- lunghezza della parola settabile da **5** a **8** bit
- controllo di parita' pari, dispari, senza parita', parita' forzata
- **1; 1.5; 2** bit di stop programmabile con incrementi di **1/16** di bit
- **Baud Rate** selezionabile tra **18** valori variabili da **50** a **38.4 K Baud**
- gestione errori di parity, frame ed overrun
- **8** diverse condizioni di interrupt mascherabili

Inoltre e' provvista di 4 timer programmabili e di **3** port di input, sempre completamente gestibili via software.

Per ulteriori informazioni a riguardo dell'**UART SCC2698**, si faccia riferimento agli appositi dati tecnici della casa costruttrice riportati nell'appendice **B** di questo manuale.

## 2.2. Dispositivo di clock

Sulla scheda e' presente un quarzo da **3.6864 MHz** che ha il compito di fornire all'**UART** tutti i segnali necessari per il suo funzionamento e per la generazione dei **Baud Rate**.

Tutta l'elettronica di interfacciamento della scheda con il Bus industriale **ABACO (R)** e' basata sugli stessi segnali del Bus ed e' per questo indipendente dal quarzo descritto.

### 2.3. Selezione protocollo di comunicazione

Come e' gia stato detto il protocollo logico di comunicazione e' completamente definibile da software tramite la programmazione dell' **UART** di bordo.

Per quanto riguarda il protocollo fisico ( elettrico ) di comunicazione deve invece essere selezionato tramite una serie di comodi jumpers montati a bordo della scheda.

Per ognuna delle **8** linee, tramite questi jumpers, e' possibile selezionare il protocollo **RS 232** oppure il protocollo **RS 422/RS 485**.

Per una linea settata in **RS 422/RS 485** e' inoltre possibile selezionare se utilizzare tale linea in **Full Duplex** o in **Half Duplex**, sempre tramite comodi jumpers; nel caso venga utilizzata la comunicazione in **Half Duplex**, la direzionalita' della linea e' facilmente definibile da software, tramite una apposita linea di controllo.

Per ulteriori informazioni a riguardo della selezione del protocollo fisico di comunicazione, si veda il paragrafo **4.5.4.2** dedicato allo strippaggio della scheda.

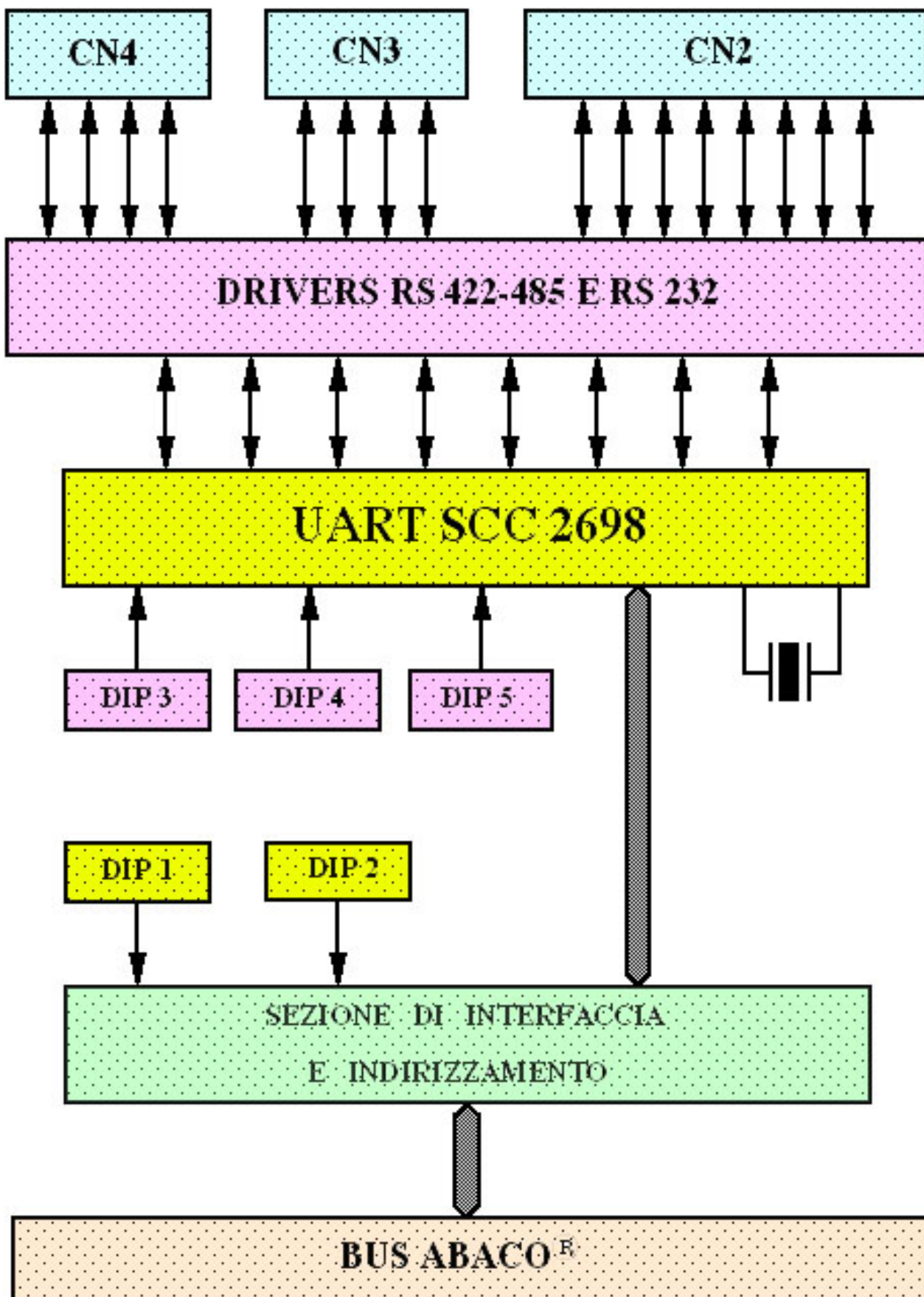


Fig. 2-2: Schema a blocchi

### 3. SPECIFICHE TECNICHE DELLA SCHEDA

#### 3.1. Caratteristiche generali

Tipo di BUS ABACO(R)  
N.ro linee di I/O 8 linee seriali bidirezionali  
con: TxD; RxD; CTS; RTS; in RS 232 o in RS 422-485.  
N.ro byte indirizzamento  
2<sup>8</sup> -> Modo normale  
2<sup>16</sup> -> Modo esteso normale  
2<sup>22</sup> -> Modo esteso potenziato  
N.ro byte occupati 64 byte

#### 3.2. Caratteristiche fisiche

Dimensioni Formato EUROPA ( 100x160 mm.)  
Peso 170 g.  
Connettori BUS 64 pin DIN 41612 corpo C  
CN2: 40 vie scatolino 90 gradi M  
CN3: 20 vie scatolino 90 gradi M  
CN4: 20 vie scatolino verticale M  
Range di temperatura da 10 a 40 gradi Centigradi  
Umidita relativa 20% fino a 90% (senza condensa)

#### 3.3. Caratteristiche elettriche

Tensione di alimentazione +5 Vdc.  
Corrente assorbita 150 mA. (Versione 8 linee RS 232 a riposo).

grifo® 40016 S. GIORGIO di P.  
BO/ITALY - tel. 051/892052

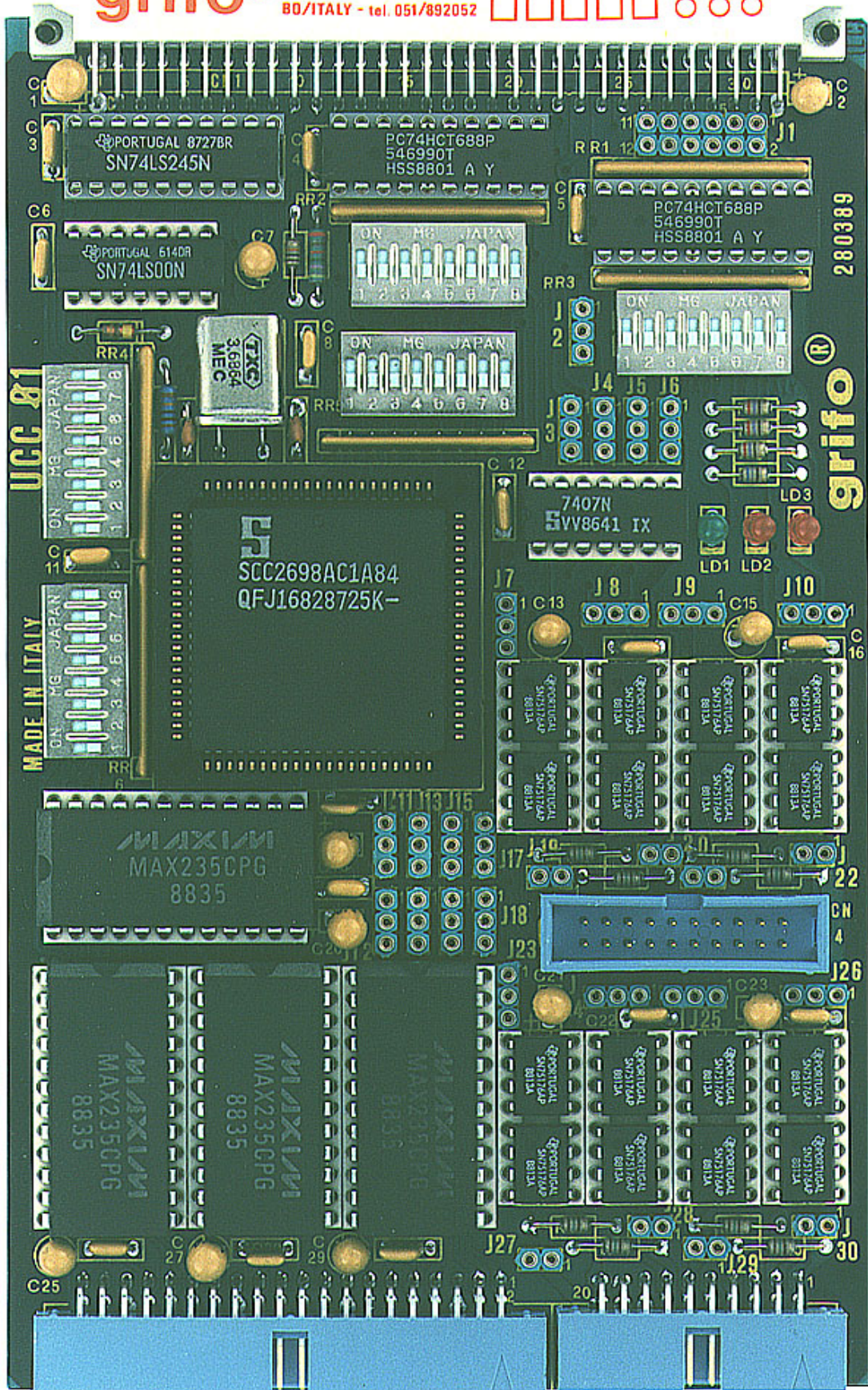


Fig. 3-1: Foto scheda UCC 01



## 4. INSTALLAZIONE DELLA SCHEDA

### 4.1. Introduzione

In questo capitolo saranno illustrate tutte le operazioni da compiere per un corretto utilizzo della scheda.

A questo scopo viene di seguito riportata la descrizione e la disposizione dei dip switches, degli strip, dei connettori e dei **LED**.

## 4.2. Connessioni con il Mondo Esterno

### 4.2.1. CN4 - Connettore 4 Linee Seriali in RS 422 - RS 485

Il connettore **CN4** e' composto da un insieme di **20** pin, con cui si puo' usufruire di **4** delle **8** linee seriali in **RS 422/RS 485** presenti sulla **UCC 01**.

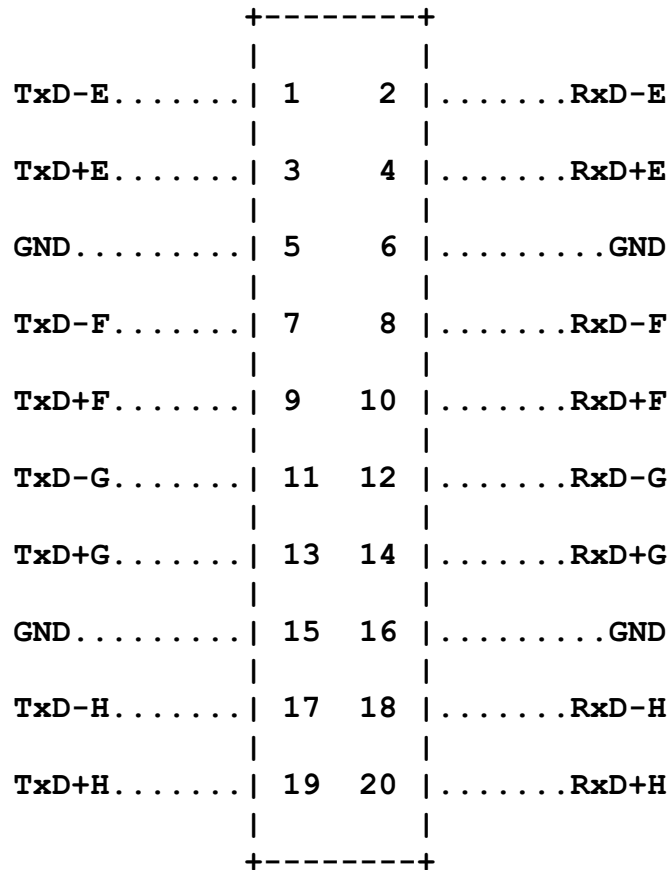
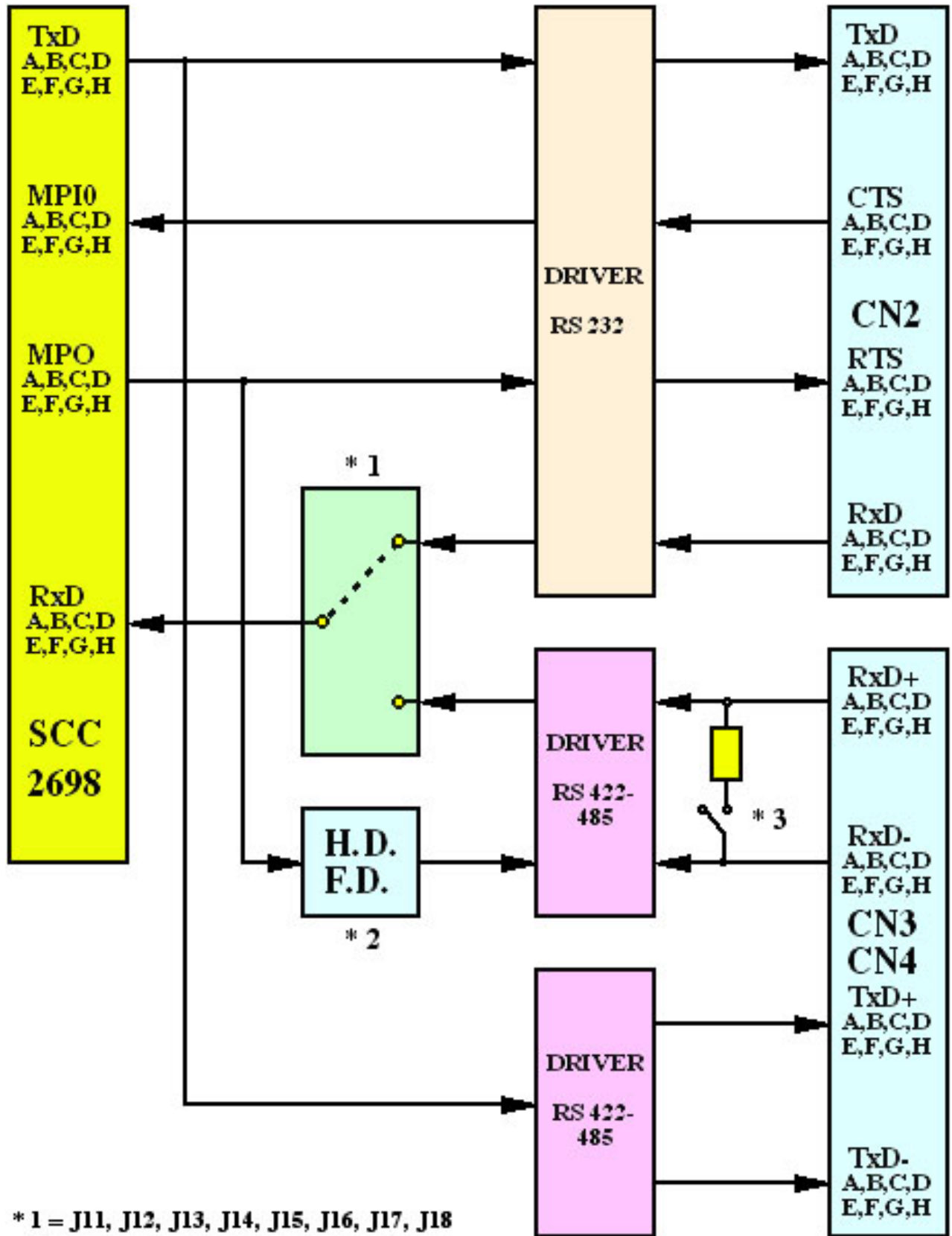


Fig. 4-1: Connettore CN4

#### LEGENDA:

- TxD-y = O** - Trasmit Data negative: linea bipolare negativa per trasmissione seriale RS 422-485 sulla linea y.
- TxD+y = O** - Trasmit Data positive: linea bipolare positiva per trasmissione seriale RS 422-485 sulla linea y.
- GND =** Linea di massa per comunicazione seriale.
- RxD-y = I** - Receive Data negative: linea bipolare negativa per ricezione seriale RS 422-485 sulla linea y.
- RxD+y = I** - Receive Data positive: linea bipolare positiva per ricezione seriale RS 422-485 sulla linea y.





\* 1 = J11, J12, J13, J14, J15, J16, J17, J18

\* 2 = J7, J8, J9, J10, J23, J24, J25, J26

\* 3 = J19, J20, J21, J22, J27, J28, J29, J30

Fig. 4-2: Schema di comunicazione seriale

#### 4.2.2. CN3 - Connettore 4 Linee Seriali in RS 422-485

Il connettore **CN3** e' composto da un insieme di **20** pin, con cui si puo' usufruire di **4** delle **8** linee seriali in **RS 422-485** presenti sulla **UCC 01**.

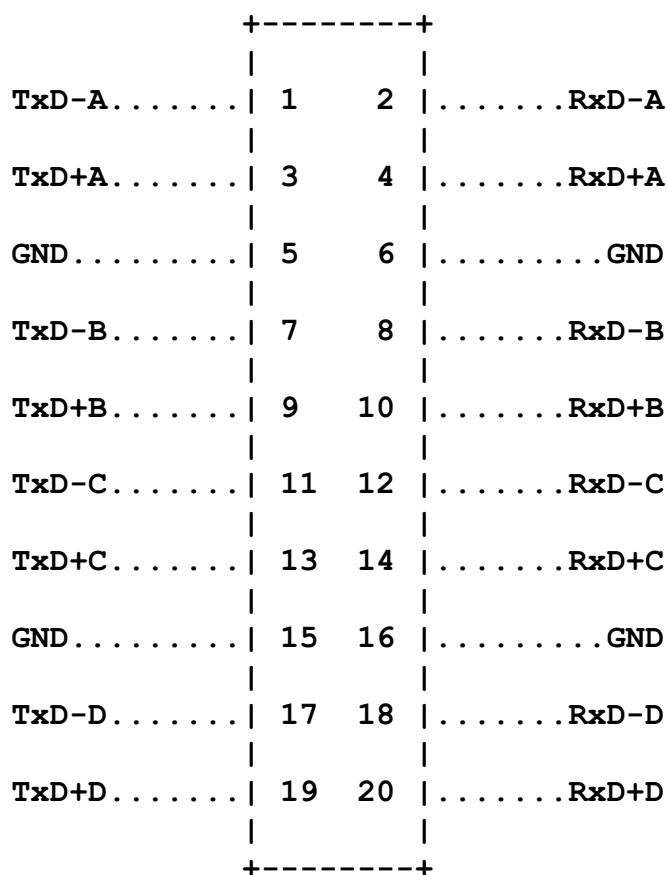


Fig. 4-3: Connettore CN3

#### LEGENDA:

**TxD-y = O** - Trasmit Data negative: linea bipolare negativa per trasmissione seriale RS 422-485 sulla linea y.

**TxD+y = O** - Trasmit Data positive: linea bipolare positiva per trasmissione seriale RS 422-485 sulla linea y.

**GND =** Linea di massa per comunicazione seriale.

**RxD-y = I** - Receive Data negative: linea bipolare negativa per ricezione seriale RS 422-485 sulla linea y.

**RxD+y = I** - Receive Data positive: linea bipolare positiva per ricezione seriale RS 422-485 sulla linea y.

4.2.3. CN1 - Connettore BUS Industriale ABACO(R)

Il connettore CN1 e' formato da un insieme di 64 pin con cui e' possibile effettuare il collegamento della scheda con il BUS Industriale ABACO(R). Nella seguente figura e' rappresentato il pin out del BUS e quindi anche del relativo connettore, con le variazioni per l' utilizzo di CPU a 16 bit rispetto a quelle a 8 bit.

A	A	A	pin	C	C	C
CPU 16 bit	CPU 8 bit	UCC 01		UCC 01	CPU 8 bit	CPU 16 bit
	GND	GND	1	GND	GND	
	+5V	+5V	2	+5V	+5V	
	D0	D0	3			D8
	D1	D1	4			D9
	D2	D2	5			D10
	D3	D3	6	/INT	/INT	
	D4	D4	7	/NMI	/NMI	
	D5	D5	8		/HALT	D11
	D6	D6	9		/MREQ	
	D7	D7	10	/IORQ	/IORQ	
	A0	A0	11	/RD	/RD	/RDLDS
	A1	A1	12	/WR	/WR	/WRLDS
	A2	A2	13		/BUSAK	D12
	A3	A3	14		/WAIT	
	A4	A4	15		/BUSRQ	D13
	A5	A5	16	/RESET	/RESET	
	A6	A6	17	/M1	/M1	/IACK
	A7	A7	18		/RFSH	D14
	A8	A8	19		/MEMDIS	
	A9	A9	20		VDUSEL	A22
	A10	A10	21		/IEI	D15
	A11	A11	22		Ris.	
	A12	A12	23		CLK	
	A13	A13	24			/RDUDS
	A14	A14	25			/WRUDS
	A15	A15	26	A21		A21
A16		A16	27	A20		A20
A17		A17	28	A19		A19
A18		A18	29		R. T.	
	+12V		30		-12V	
	+5V	+5V	31	+5V	+5V	
	GND	GND	32	GND	GND	

Fig. 4-4: Connettore CN1



**LEGENDA:**

CPU A 8 BIT

A0-A15= Address Bus - Bus degli indirizzi;

D0-D7= Data Bus - Bus dei dati;

INT= Interrupt request - Richiesta di interruzione;

NMI= Non Mascherable Interrupt - Richiesta di interruzione non mascherabile;

HALT= Halt State - Stato di Halt;

MREQ= Memory Request - Richiesta di memoria;

IORQ= Input/Output Request - Richiesta di Input/Output;

RD= Read Cycle Status - Richiesta di lettura;

WR= Write Cycle Status - Richiesta di scrittura;

BUSAK= Bus Acknowledge - Riconoscimento del Bus;

WAIT= Wait - Attesa;

BUSRQ= Bus Request - Richiesta del Bus;

RESET= Azzeramento;

M1= Machine Cycle One - Primo Ciclo Macchina;

RFSH= Refresh - Rinfresco;

MEMDIS= Memory Display - Viene emesso dal dispositivo periferico che si sta mappando nell' area di memoria;

VDUSEL= VDU Selection - Abilita il dispositivo periferico ad essere mappato nell' area di memoria;

IEI FIO 02,03= Abilitazione interrupt FIO 02 , 03 CLK= Clock di sistema

R.T.= Tasto di Reset

CPU A 16 BIT

A0-A21= Address Bus - Bus degli indirizzi

D0-D15= Data Bus - Bus dei Dati

RD UDS= Read Upper Data Strobe - Lettura del byte superiore bus dati;

WR UDS= Write Upper Data Strobe - Scrittura del byte superiore bus dati;

IACK= Interrupt Acknowledge - Riconoscimento della richiesta di Interrupt da parte della CPU;

RD LDS= Read Lower Data Strobe - Lettura del byte inferiore su bus dati;

WR LDS= Write Lower Data Strobe - Scrittura del byte inferiore su bus dati;

4.2.4. CN2 - Connettore 8 Linee Seriali in RS 232

Il connettore **CN2** e' composto da un insieme di **40** pin, con cui si puo' usufruire delle **8** linee seriali in **RS 232** presenti sulla scheda **UCC 01**.

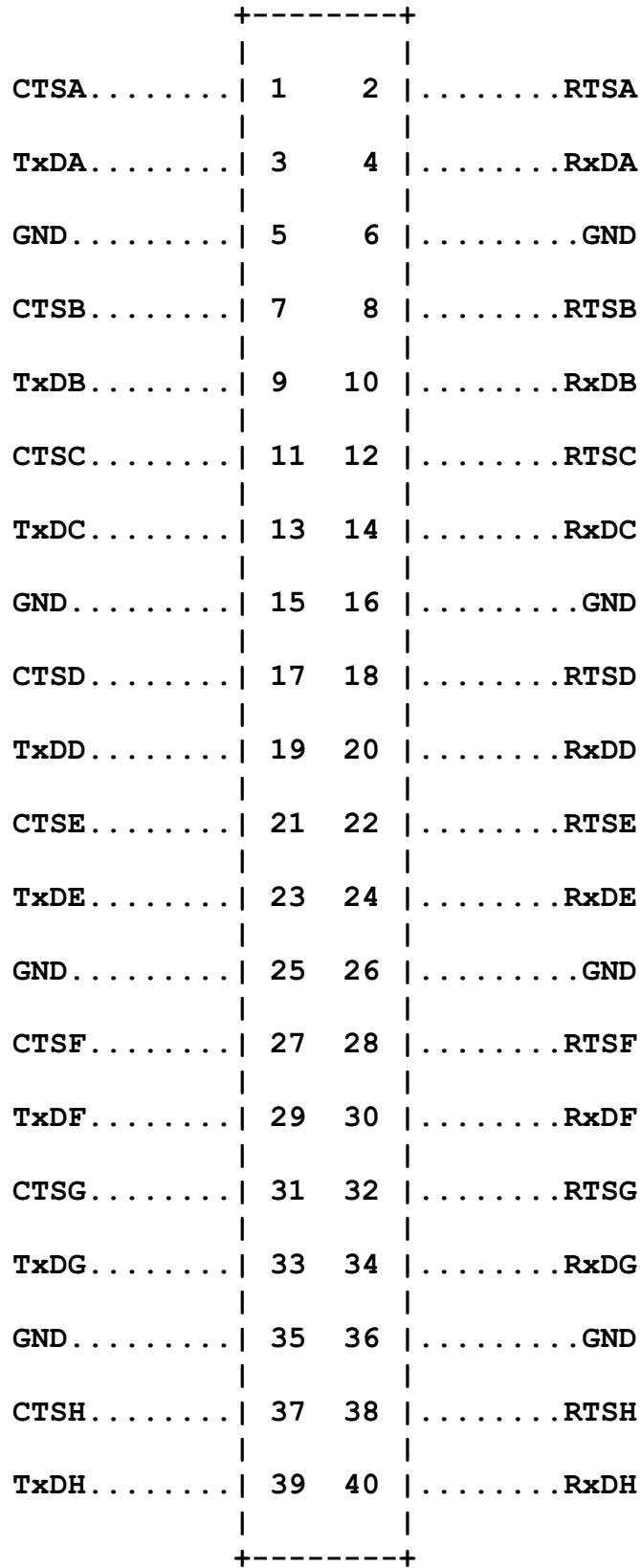


Fig. 4-5: Connettore CN2

**LEGENDA:**

- CTS<sub>y</sub>** = **I** - Clear To Send: linea di abilitazione della trasmissione sulla linea seriale y in RS 232.
- RTS<sub>y</sub>** = **O** - Request To Send: linea di richiesta di trasmissione sulla linea seriale y in RS 232.
- TxD<sub>y</sub>** = **O** - Transmit Data: linea di trasmissione della linea seriale y in RS 232.
- RxD<sub>y</sub>** = **I** - Receive Data: linea di ricezione della linea seriale y in RS 232.

### 4.3. Segnalazioni Visive

La scheda **UCC 01** segnala, tramite una serie di tre **LEDs** alcune particolari condizioni di stato:

**LD1** - Di colore verde segnala quando e' attivo che e' in corso una operazione di accesso (lettura/scrittura) alla scheda.

**LD2** - Di colore rosso segnala quando la scheda attiva la linea di **/INT** del **BUS ABACO (R)**.

**LD3** - Di colore rosso segnala quando la scheda attiva la linea di **/NMI** del **BUS ABACO (R)**.

La funzione di questi **LEDs** e' quella di informare visivamente l'utente sui principali stati di lavoro, senza dover ricorrere ad apposite strutture software.

Questa prerogativa semplifica notevolmente la verifica iniziale ed un' eventuale ricerca guasti di tutto il sistema di cui la scheda fa' parte.

Per quanto riguarda la disposizione dei **LEDs** a bordo della scheda si faccia riferimento alla figura **4-6**.

### 4.4. Timers

Sulla scheda **UCC 01** sono disponibili **4** counter timer indipendenti da **16** bit, completamente gestibili via software.

Questi dispositivi possono essere utilizzati per funzioni generiche richieste dall'applicazione, ma un loro utilizzo classico in modalita' timer e' quella di definire e testare alcune finestre temporali, spesso utilizzate per la comunicazione seriale asincrona di dati, tra piu' dispositivi.

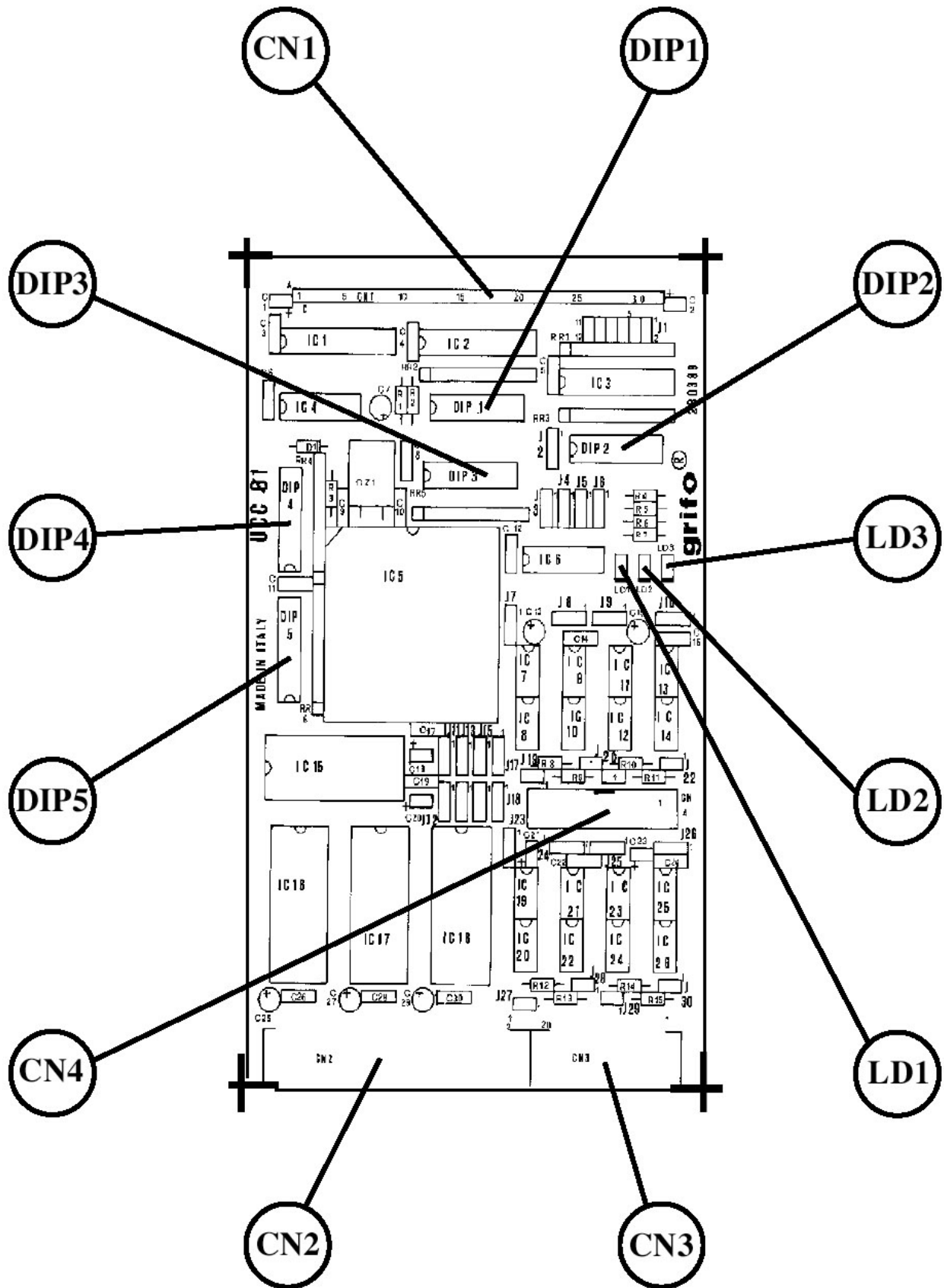


Fig. 4-6: Disposizione LEDs, Dip Switch, Connettori

#### 4.5. Jumpers

Esistono a bordo della scheda **UCC 01 30** jumpers a cavaliere con cui e' possibile effettuare alcune selezioni che riguardano il modo di funzionamento della scheda. In seguito ne e' riportato l'elenco, l'ubicazione e la loro funzione.

L'uso dei jumpers della scheda e' comune a tutte le **8** linee seriali, in modo da facilitarne lo strippaggio e l'uso. Per l'individuazione dei jumpers sulla scheda si faccia riferimento alla figura **4-7** ed all'appendice **A**, di questo manuale.

JUMPERS	N. VIE	UTILIZZO
J1	12	Definisce indirizzo nel caso di indirizzamento esteso potenziato.
J2	3	Seleziona tipo d' indirizzamento della scheda.
J3	3	Seleziona collegamento linea di /INTR delle linee seriali A e B sul Bus Abaco(R).
J4	3	Seleziona collegamento linea di /INTR delle linee seriali C e D sul Bus Abaco(R).
J5	3	Seleziona collegamento linea di /INTR delle linee seriali E e F sul Bus Abaco(R).
J6	3	Seleziona collegamento linea di /INTR delle linee seriali G e H sul Bus Abaco(R).
J7	3	Seleziona comunicazione Full/Half Duplex per la linea seriale H in RS 422-485.
J8	3	Seleziona comunicazione Full/Half Duplex per la linea seriale G in RS 422-485.
J9	3	Seleziona comunicazione Full/Half Duplex per la linea seriale F in RS 422-485.
J10	3	Seleziona comunicazione Full/Half Duplex per la linea seriale E in RS 422-485.
J11	3	Seleziona comunicazione in RS 232 oppure in RS 422-485 per la linea seriale H.
J12	3	Seleziona comunicazione in RS 232 oppure in RS 422-485 per la linea seriale G.
J13	3	Seleziona comunicazione in RS 232 oppure in RS 422-485 per la linea seriale F.
J14	3	Seleziona comunicazione in RS 232 oppure in RS 422-485 per la linea seriale E.
J15	3	Seleziona comunicazione in RS 233 oppure in RS 422-485 per la linea seriale D.
J16	3	Seleziona comunicazione in RS 232 oppure in RS 422-485 per la linea seriale C.

Table 4-1: Tabella riassuntiva jumpers (parte 1)



JUMPERS	N.VIE	UTILIZZO
J17	3	Seleziona comunicazione in RS 232 oppure in RS 422-485 per la linea seriale B.
J18	3	Seleziona comunicazione in RS 232 oppure in RS 422-485 per la linea seriale A.
J19	2	Collega resistenza nominale di terminazione alla linea di ricezione H in RS 422-485.
J20	2	Collega resistenza nominale di terminazione alla linea di ricezione G in RS 422-485.
J21	2	Collega resistenza nominale di terminazione alla linea di ricezione F in RS 422-485.
J22	2	Collega resistenza nominale di terminazione alla linea di ricezione E in RS 422-485.
J23	3	Seleziona comunicazione Full/Half Duplex per la linea seriale D in RS 422-485.
J24	3	Seleziona comunicazione Full/Half Duplex per la linea seriale C in RS 422-485.
J25	3	Seleziona comunicazione Full/Half Duplex per la linea seriale B in RS 422-485.
J26	3	Seleziona comunicazione Full/Half Duplex per la linea seriale A in RS 422-485.
J27	2	Collega resistenza nominale di terminazione alla linea di ricezione D in RS 422-485.
J28	2	Collega resistenza nominale di terminazione alla linea di ricezione C in RS 422-485.
J29	2	Collega resistenza nominale di terminazione alla linea di ricezione B in RS 422-485.
J30	2	Collega resistenza nominale di terminazione alla linea di ricezione A in RS 422-485.

Table 4-2: Tabella riassuntiva jumpers (parte 2)

### 4.5.1. Jumpers a 2 vie

Viene di seguito riportata una tabella che riporta la funzione dei jumpers a 2 vie della scheda **UCC 01** in relazione ai loro possibili modi di connessione.

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J19	Connesso	Connette res. di terminazione alla linea di ricezione della linea seriale H in RS 422-485.	
	non connesso	Non connette res. di terminazione alla linea di ricezione della linea seriale H in RS 422-485.	*
J20	connesso	Connette res. di terminazione alla linea di ricezione della linea seriale G in RS 422-485.	
	non connesso	Non connette res. di terminazione alla linea di ricezione della linea seriale G in RS 422-485.	*
J21	connesso	Connette res. di terminazione alla linea di ricezione della linea seriale F in RS 422-485.	
	non connesso	Non connette res. di terminazione alla linea di ricezione della linea seriale F in RS 422-485.	*
J22	connesso	Connette res. di terminazione alla linea di ricezione della linea seriale E in RS 422-485.	
	non connesso	Non connette res. di terminazione alla linea di ricezione della linea seriale E in RS 422-485.	*

Table 4-3: Tabella jumpers a 2 vie (parte 1)

L' \* indica la connessione di default, ovvero la connessione effettuata durante il montaggio ed il collaudo della scheda.

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J27	connesso	Connette res. di terminazione alla linea di ricezione della linea seriale D in RS 422-485.	
	non connesso	Non connette res. di terminazione allalinea di ricezione della linea seriale D in RS 422-485.	*
J28	connesso	Connette res. di terminazione alla linea di ricezione della linea seriale C in RS 422-485.	
	non connesso	Non connette res. di terminazione all linea di ricezione della linea seriale C in RS 422-485.	*
J29	connesso	Connette res. di terminazione alla linea di ricezione della linea seriale B in RS 422-485.	
	non connesso	Non connette res. di terminazione allalinea di ricezione della linea seriale B in RS 422-485.	*
J30	connesso	Connette res. di terminazione alla linea di ricezione della linea seriale A in RS 422-485.	
	non connesso	Non connette res. di terminazione allalinea di ricezione della linea seriale A in RS 422-485.	*

Table 4-4: Tabella jumpers a 2 vie (parte 2)

L' \* indica la connessione di default, ovvero la connessione effettuata durante il montaggio ed il collaudo della scheda.

#### 4.5.2. Jumpers a 3 vie

Viene di seguito riportata una tabella che riporta la funzione dei jumpers a 3 vie della scheda UCC 01 in relazione ai loro possibili modi di connessione.

Per riconoscere le connessioni sulla scheda si faccia riferimento alla serigrafia della stessa dove viene riportata una numerazione che coincide con quella adottata nella seguente descrizione.

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J2	1-2	Selezione indirizzamento normale a 2 <sup>8</sup> indirizzi.	*
	2-3	Selezione indirizzamento esteso a 2 <sup>16</sup> ind., od esteso potenziato a 2 <sup>22</sup> indirizzi.	
J3	1-2	Collega linea di /INTR delle linee seriali A e B al segnale di /INT del Bus ABACO(R).	
	2-3	Collega linea di /INTR delle linee seriali A e B al segnale di /NMI del Bus ABACO(R).	
J4	1-2	Collega linea di /INTR delle linee seriali C e D al segnale di /INT del Bus ABACO(R).	
	2-3	Collega linea di /INTR delle linee seriali C e D al segnale di /NMI del Bus ABACO(R).	
J5	1-2	Collega linea di /INTR delle linee seriali E e F al segnale di /INT del Bus ABACO(R).	
	2-3	Collega linea di /INTR delle linee seriali E e F al segnale di /NMI del Bus ABACO(R).	
J6	1-2	Collega linea di /INTR delle linee seriali G e H al segnale di /INT del Bus ABACO(R).	
	2-3	Collega linea di /INTR delle linee seriali G e H al segnale di /NMI del Bus ABACO(R).	

Table 4-5: Tabella jumpers a 3 vie (parte 1)

L' \* indica la connessione di default, ovvero la connessione effettuata durante il montaggio ed il collaudo della scheda.

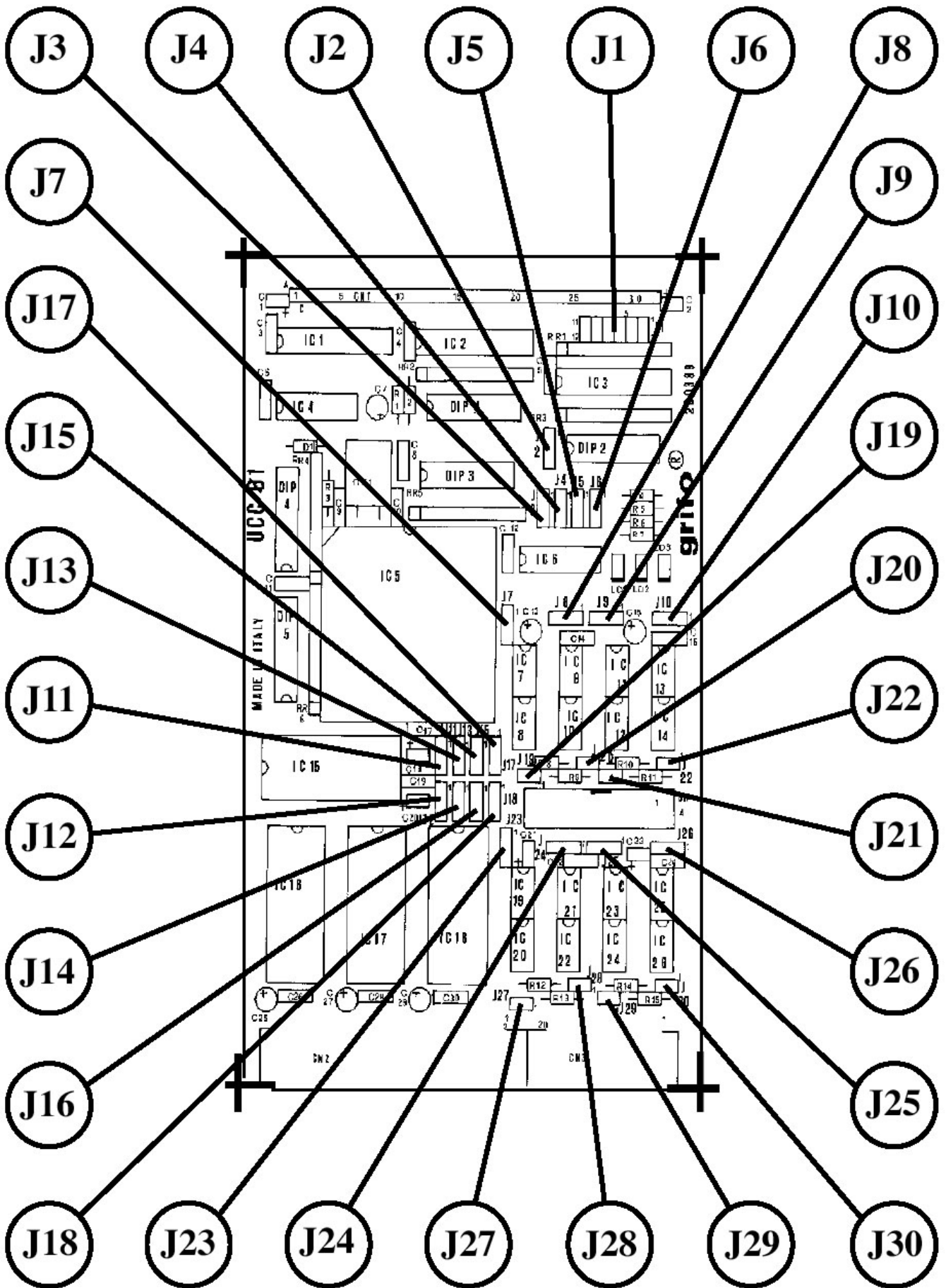


Fig. 4-7: Disposizione Jumpers

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J7	1-2	Seleziona funzionamento in Full Duplex della linea seriale RS 422-485.	*
	2-3	Seleziona funzionamento in Half Duplex della linea seriale RS 422-485.	
J8	1-2	Seleziona funzionamento in Full Duplex della linea seriale RS 422-485.	*
	2-3	Seleziona funzionamento in Half Duplex della linea seriale RS 422-485.	
J9	1-2	Seleziona funzionamento in Full Duplex della linea seriale RS 422-485.	*
	2-3	Seleziona funzionamento in Half Duplex della linea seriale RS 422-485.	
J10	1-2	Seleziona funzionamento in Full Duplex della linea seriale RS 422-485.	*
	2-3	Seleziona funzionamento in Half Duplex della linea seriale RS 422-485.	
J11	1-2	Seleziona comunicazione in RS 422-485 per la linea seriale H.	
	2-3	Seleziona comunicazione in RS 232 per la linea seriale H.	*
J12	1-2	Seleziona comunicazione in RS 422-485 per la linea seriale G.	
	2-3	Seleziona comunicazione in RS 232 per la linea seriale G.	*

Table 4-6: Tabella jumpers a 3 vie (parte 2)

L' \* indica la connessione di default, ovvero la connessione effettuata durante il montaggio ed il collaudo della scheda.

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J13	1-2	Selezione comunicazione in RS 422-485 per la linea seriale F.	
	2-3	Selezione comunicazione in RS 232 per la linea seriale F.	*
J14	1-2	Selezione comunicazione in RS 422-485 per la linea seriale E.	
	2-3	Selezione comunicazione in RS 232 per la linea seriale E.	*
J15	1-2	Selezione comunicazione in RS 422-485 per la linea seriale D.	
	2-3	Selezione comunicazione in RS 232 per la linea seriale D.	*
J16	1-2	Selezione comunicazione in RS 422-485 per la linea seriale C.	
	2-3	Selezione comunicazione in RS 232 per la linea seriale C.	*
J17	1-2	Selezione comunicazione in RS 422-485 per la linea seriale B.	
	2-3	Selezione comunicazione in RS 232 per la linea seriale B.	*
J18	1-2	Selezione comunicazione in RS 422-485 per la linea seriale A.	
	2-3	Selezione comunicazione in RS 232 per la linea seriale A.	*
J23	1-2	Selezione funzionamento in Full Duplex della linea seriale D in RS 422-485.	*
	2-3	Selezione funzionamento in Half Duplex della linea seriale D in RS 422-485.	

Table 4-7: Tabella jumpers a 3 vie (parte 3)

L' \* indica la connessione di default, ovvero la connessione effettuata durante il montaggio ed il collaudo della scheda.

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J24	1-2	Seleziona funzionamento in Duplex della linea seriale C in RS 422-485.	*
	2-3	Seleziona funzionamento in Duplex della linea seriale C in RS 422-485.	
J25	1-2	Seleziona funzionamento in Duplex della linea seriale B in RS 422-485.	*
	2-3	Seleziona funzionamento in Duplex della linea seriale B in RS 422-485.	
J26	1-2	Seleziona funzionamento in Duplex della linea seriale A in RS 422-485.	*
	2-3	Seleziona funzionamento in Duplex della linea seriale A in RS 422-485.	

Table 4-8: Tabella jumpers a 3 vie (parte 4)

L' \* indica la connessione di default, ovvero la connessione effettuata durante il montaggio ed il collaudo della scheda.



### 4.5.3. Jumper a 12 vie.

Sulla scheda **UCC 01** e' presente un jumper a **12** vie da utilizzare nel caso di selezione di indirizzamento esteso, per utilizzare l'indirizzamento esteso potenziato da **2^22** indirizzi.

Viene di seguito riportata una tabella che riporta la funzione di tale jumper in relazione a tutte le sue possibili connessioni.

JUMPER	CONNESSIONE	UTILIZZO	DEF.
J1	1-2	Collega indirizzo A21 alla logica di indirizzamento della scheda.	
J1	3-4	Collega indirizzo A20 alla logica di indirizzamento della scheda.	
J1	5-6	Collega indirizzo A19 alla logica di indirizzamento della scheda.	
J1	7-8	Collega indirizzo A18 alla logica di indirizzamento della scheda.	
J1	9-10	Collega indirizzo A17 alla logica di indirizzamento della scheda.	
J1	11-12	Collega indirizzo A16 alla logica di indirizzamento della scheda.	*

Table 4-9: Tabella jumpers a 12 vie

L' \* indica la connessione di default, ovvero la connessione effettuata durante il montaggio ed il collaudo della scheda.

### 4.5.4. Note

Viene di seguito riportata una serie di indicazioni da utilizzare in fase di strippaggio ed utilizzo della scheda **UCC 01**.

Tali informazioni illustrano in maniera piu' dettagliata e con l'aiuto di esempi, come devono essere posizionati tutti i jumpers presenti sulla scheda.

Per l'individuazione di quest'ultimi, fare sempre riferimento alla figura **4-6**.

#### 4.5.4.1. Selezione tipo di indirizzamento

La scheda **UCC 01** puo' essere utilizzata sfruttando tre diversi tipi di indirizzamento. Per ognuno di queste modalita' la scheda deve essere correttamente configurata. In particolare:

##### a) Indirizzamento Normale

La scheda puo' essere vista in **2<sup>8</sup>** indirizzi diversi e deve essere configurata:

**J1 -> Nessuna connessione.**  
**J2 -> Connessione 1-2.**  
**Dip 2.1 -> OFF**  
**Dip 2.2 -> OFF**  
**Dip 2.3 -> OFF**  
**Dip 2.4 -> OFF**  
**Dip 2.5 -> OFF**  
**Dip 2.6 -> OFF**  
**Dip 2.7 -> Seleziona indirizzo**  
**Dip 2.8 -> Seleziona indirizzo**  
**Dip 1.? -> Indifferente**

##### b) Indirizzamento Esteso

La scheda puo' essere vista in **2<sup>16</sup>** indirizzi diversi e deve essere configurata:

**J1 -> Nessuna connessione.**  
**J2 -> Connessione 2-3.**  
**Dip 2.1 -> OFF**  
**Dip 2.2 -> OFF**  
**Dip 2.3 -> OFF**  
**Dip 2.4 -> OFF**  
**Dip 2.5 -> OFF**  
**Dip 2.6 -> OFF**  
**Dip 2.7 -> Seleziona indirizzo**  
**Dip 2.8 -> Seleziona indirizzo**  
**Dip 1.? -> Seleziona indirizzo**

### c) Indirizzamento Esteso Potenziato

La scheda puo' essere vista in 2<sup>22</sup> indirizzi diversi e deve essere configurata:

- J1 -> Tutti connessi (1-2;3-4;5-6; ecc.)
- J2 -> Connessione 2-3.
- Dip 2.? -> Seleziona indirizzo.
- Dip 1.? -> Seleziona indirizzo.

Per quanto riguarda la disposizione dei dip switches di selezione dell'indirizzo, si faccia riferimento al paragrafo 5.2 dedicato al mappaggio della scheda.

#### 4.5.4.2. Selezione tipo di Comunicazione

Premettendo che sulla scheda **UCC 01** possono essere presenti tutti i driver **RS 232** e tutti i driver **RS 422-485**, di seguito si riporta la corrispondenza di questi con le **8** linee seriali:

- IC 18 = Driver RS 232 per le linee seriali A e B.
- IC 17 = Driver RS 232 per le linee seriali C e D.
- IC 16 = Driver RS 232 per le linee seriali E e F.
- IC 15 = Driver RS 232 per le linee seriali G e H.
- IC 26 = Driver RS 422-485 di trasmissione per la linea seriale A.
- IC 25 = Driver RS 422-485 di ricezione per la linea seriale A.
- IC 24 = Driver RS 422-485 di trasmissione per la linea seriale B.
- IC 23 = Driver RS 422-485 di ricezione per la linea seriale B.
- IC 22 = Driver RS 422-485 di trasmissione per la linea seriale C.
- IC 21 = Driver RS 422-485 di ricezione per la linea seriale C.
- IC 20 = Driver RS 422-485 di trasmissione per la linea seriale D.
- IC 19 = Driver RS 422-485 di ricezione per la linea seriale D.
- IC 13 = Driver RS 422-485 di trasmissione per la linea seriale E.
- IC 14 = Driver RS 422-485 di ricezione per la linea seriale E.
- IC 11 = Driver RS 422-485 di trasmissione per la linea seriale F.
- IC 12 = Driver RS 422-485 di ricezione per la linea seriale F.
- IC 9 = Driver RS 422-485 di trasmissione per la linea seriale G.
- IC 10 = Driver RS 422-485 di ricezione per la linea seriale G.
- IC 7 = Driver RS 422-485 di trasmissione per la linea seriale H.
- IC 8 = Driver RS 422-485 di ricezione per la linea seriale H.

Al fine di configurare correttamente la scheda **UCC 01** in relazione al tipo di comunicazione da utilizzare si devono utilizzare le seguenti indicazioni:

### a) Settaggio Linea Seriale in RS 232

Si devono effettuare le seguenti operazioni:

- Montare relativo **Driver RS 232**.
- Connettere relativo jumper di selezione **RS 232** o **RS 422-485** in posizione **2-3**.

### b) Settaggio linea seriale in RS 422-485 Full Duplex

Si devono effettuare le seguenti operazioni:

- Montare relativi Driver **RS 422-485** di trasmissione e ricezione.
- Connettere relativo jumper di selezione **RS 232** o **RS 422-485** in posizione **1-2**.
- Connettere relativo jumper di selezione comunicazione **Full/Half Duplex** in posizione **1-2**.

### c) Settaggio linea seriale in RS 422-485 Half Duplex

Si devono effettuare le seguenti operazioni:

- Montare relativi **Driver RS 422-485** di trasmissione e ricezione.
- Connettere relativo jumper di selezione **RS 232** o **RS 422-485** in posizione **1-2**.
- Connettere relativo jumper di selezione comunicazione **Full/Half Duplex** in posizione **2-3**.
- La direzionalità della linea seriale in **RS 422-485** in **Half Duplex** è stabilita dalla linea di handshake **/RTS** connessa al pin **MPO** gestita direttamente dall'**UART SCC 2698** (vedi figura **4-2**).

**N.B.** Con relativo Driver e relativo jumper s'intende rispettivamente il driver ed il jumper che riguardano la linea seriale in uso, secondo le informazioni riportate all'inizio di questo paragrafo.

#### 4.5.4.3. Esempio

Per configurare la scheda **UCC 01** per la gestione delle linee seriali **A** e **B** in **RS 232**, della linea seriale **C** in **RS 422-485** in **Full Duplex** e della linea seriale **D** in **RS 422-485** in **Half Duplex**, si deve:

- **Montare Driver RS 232** su **IC 18**.
- **Montare Driver RS 422-485** su **IC 19, IC 20, IC 21, IC 22**.
- **Collegare J17 in posizione 2-3**
  - J18 in posizione 2-3**
  - J15 in posizione 1-2**
  - J16 in posizione 1-2**
  - J24 in posizione 1-2**
  - J23 in posizione 2-3**
- La direzionalità della linea seriale **D** è stabilita dalla linea di handshake **/RTS** della stessa linea che corrisponde al segnale **MPOD** dell'**UART**.

#### 4.5.4.4. Gestione interrupts

L'**UART** della scheda **UCC 01** è provvista di quattro linee separate di generazione interrupt, relative alle quattro coppie di linee seriali.

Le condizioni che possono generare interrupt sono tutte mascherabili e coincidono con le normali condizioni di stato che si presentano durante una comunicazione seriale (trasmissione ultimata, carattere ricevuto, errore di comunicazione, ecc.).

Le linee d' interrupt descritte possono essere collegate al **BUS Industriale ABACO(R)** tramite una serie di appositi jumpers, in modo che la scheda di controllo presente sul Bus, possa riconoscere e gestire le condizioni che hanno generato l' **Interrupt**.

Come descritto nella tabella **4-3**, ogni linea di interrupt può essere collegata ad uno dei due segnali del **BUS**; infatti la logica di interfaccia si occupa di evitare eventuali conflitti elettrici che si possono creare collegando più linee d' uscita (**UART**) ad un singolo ingresso (**BUS**).

Dovendo ad esempio collegare l' interrupt delle linee seriali **A** e **B** all' interrupt mascherabile del **BUS ABACO (R)**, l' interrupt delle linee seriali **C** e **D** all' interrupt non mascherabile del **BUS ABACO (R)** e non collegare gli interrupt delle linee seriali **E, F, G, H**, si deve effettuare il seguente strippaggio:

**J3 in connessione 1-2**  
**J4 in connessione 2-3**  
**J5 in nessuna connessione**  
**J6 in nessuna connessione**

Per ulteriori informazioni sulle modalita' di gestione interrupts, si faccia riferimento all' appendice **B**.

#### 4.6. Dip switch

La scheda **UCC 01** e' provvista di una serie di **5** dip switch ad **8** vie.

Per quanto riguarda il specifico utilizzo di questi dip switch si faccia riferimento ai paragrafi **5.2** e **5.3.1**, mentre di seguito viene riportata la loro funzione:

**DIP1, DIP2** = Effettuano il mappaggio della scheda su **BUS ABACO (R)** nel campo di indirizzamento scelto.  
**DIP3, DIP4, DIP5** = Ingressi generici acquisibili da software.

Possono essere utilizzati per variare delle condizioni di lavoro o selezionare una serie di parametri al firmware della scheda di comando.

Da notare che tutti i dip switch sono collegati con logica negata, quindi se posto in **ON** genera uno Zero logico, mentre se posto in **OFF** genera un **Uno** logico.

Per quanto riguarda la disposizione dei dip switch a bordo della scheda si faccia riferimento alla figura **4-6**.

## 5. MAPPAGGIO DELLA SCHEDA

### 5.1. Introduzione

La scheda **UCC 01** viene indirizzata in uno spazio fisico di **64** byte che possono essere allocati a partire da un indirizzo di base diverso a seconda di come viene mappata la scheda.

Questa prerogativa consente di poter montare piu' schede **UCC 01** sullo stesso **BUS Industriale ABACO(R)**, oppure di montare la scheda su di un **BUS** su cui sono gia' montate altre schede periferiche, ottenendo cosi' una struttura espandibile senza difficolta' e senza alcuna modifica del software gia' realizzato.

### 5.2. Indirizzamento della Scheda

L'indirizzo di mappaggio della scheda e' definibile tramite una apposita circuiteria presente sulla stessa scheda; questa circuiteria utilizza due dip switch ad **8** vie da cui preleva l'indirizzo di mappaggio impostato dall'utente.

Di seguito viene riportata la corrispondenza dei dip switch, mentre per la selezione del tipo di indirizzamento si faccia riferimento al paragrafo **4.5.4.1**.

Dip 1.1	->	Bit A8
Dip 1.2	->	Bit A9
Dip 1.3	->	Bit A10
Dip 1.4	->	Bit A11
Dip 1.5	->	Bit A12
Dip 1.6	->	Bit A13
Dip 1.7	->	Bit A14
Dip 1.8	->	Bit A15
Dip 2.1	->	Bit A16
Dip 2.2	->	Bit A17
Dip 2.3	->	Bit A18
Dip 2.4	->	Bit A19
Dip 2.5	->	Bit A20
Dip 2.6	->	Bit A21
Dip 2.7	->	Bit A6
Dip 2.8	->	Bit A7

**N.B.** Tali dip switch sono collegati con logica negata, quindi se posto in ON genera uno **Zero** logico, mentre se posto in **OFF** genera un **Uno** logico.

Se ad esempio si dovesse mappare la scheda **UCC 01** con indirizzamento normale, all' indirizzo di base **040H**, la scheda deve essere configurata come segue:

**J1 -> Nessuna connessione.**  
**J2 -> Connessione 1-2.**

**Dip 1.? -> Indifferente.**  
**Dip 2.1 -> OFF**  
**Dip 2.2 -> OFF**  
**Dip 2.3 -> OFF**  
**Dip 2.4 -> OFF**  
**Dip 2.5 -> OFF**  
**Dip 2.6 -> OFF**  
**Dip 2.7 -> OFF**  
**Dip 2.8 -> ON**

Se invece si dovesse mappare la scheda **UCC 01** con indirizzamento esteso, all' indirizzo di base **0400H**, la scheda deve essere configurata come segue:

**J1 -> Nessuna connessione.**  
**J2 -> Connessione 2-3.**

**Dip 1.1 -> ON**  
**Dip 1.2 -> ON**  
**Dip 1.3 -> OFF**  
**Dip 1.4 -> ON**  
**Dip 1.5 -> ON**  
**Dip 1.6 -> ON**  
**Dip 1.7 -> ON**  
**Dip 1.8 -> ON**  
**Dip 2.1 -> OFF**  
**Dip 2.2 -> OFF**  
**Dip 2.3 -> OFF**  
**Dip 2.4 -> OFF**  
**Dip 2.5 -> OFF**  
**Dip 2.6 -> OFF**  
**Dip 2.7 -> ON**  
**Dip 2.8 -> ON**



### 5.3. Indirizzamento Registri Interni

Indicando con **<indbase>** il primo indirizzo dei **64** in cui e' allocata la scheda, ovvero l'indirizzo di mappaggio descritto nel paragrafo precedente, i registri interni dell'**UART SCC 2698** sono visti agli indirizzi riportati nella seguente tabella:

INDIRIZZO	REGISTRO IN LETTURA	REGISTRO IN SCRITTURA
<indbase>+0	MR1A,MR2A	MR1A,MR2A
<indbase>+1	SRA	CSRA
<indbase>+2	Riservato	CRA
<indbase>+3	RHRA	THRA
<indbase>+4	IPCRA	ACRA
<indbase>+5	ISRA	IMRA
<indbase>+6	CTUA	CTURA
<indbase>+7	CTLA	CTLRA
<indbase>+8	MR1B,MR2B	MR1B,MR2B
<indbase>+9	SRB	CSRB
<indbase>+10	Riservato	CRB
<indbase>+11	RHRB	THRB
<indbase>+12	Riservato	Riservato
<indbase>+13	Input Port A	OPCRA
<indbase>+14	Start C/T A	Riservato
<indbase>+15	Stop C/T A	Riservato
<indbase>+16	MR1C,MR2C	MR1C,MR2C
<indbase>+17	SRC	CSRC
<indbase>+18	Riservato	CRC
<indbase>+19	RHRC	THRC
<indbase>+20	IPCRB	ACRB
<indbase>+21	ISRB	IMRB
<indbase>+22	CTUB	CTURB
<indbase>+23	CTLB	CTLRB
<indbase>+24	MR1D,MR2D	MR1D,MR2D
<indbase>+25	SRD	CSRD
<indbase>+26	Riservato	CRD
<indbase>+27	RHRD	THRD
<indbase>+28	Riservato	Riservato
<indbase>+29	Input Port B	OPCRB
<indbase>+30	Start C/T B	Riservato
<indbase>+31	Stop C/T B	Riservato

Table 5-1: Tabella di Indirizzamento Registri Interni (parte 1)

INDIRIZZO	REGISTRO IN LETTURA	REGISTRO IN SCRITTURA
<indbase>+32	MR1E,MR2E	MR1E,MR2E
<indbase>+33	SRE	CSRE
<indbase>+34	Riservato	CRE
<indbase>+35	RHRE	THRE
<indbase>+36	IPCRC	ACRC
<indbase>+37	ISRC	IMRC
<indbase>+38	CTUC	CTURC
<indbase>+39	CTLC	CTLRC
<indbase>+40	MR1F,MR2F	MR1F,MR2F
<indbase>+41	SRF	CSRF
<indbase>+42	Riservato	CRF
<indbase>+43	RHRF	THRF
<indbase>+44	Riservato	Riservato
<indbase>+45	Input Port C	OPCRC
<indbase>+46	Start C/T C	Riservato
<indbase>+47	Stop C/T C	Riservato
<indbase>+48	MR1G,MR2G	MR1G,MR2G
<indbase>+49	SRG	CSRG
<indbase>+50	Riservato	CRG
<indbase>+51	RHRG	THRG
<indbase>+52	IPCRD	ACRD
<indbase>+53	ISRD	IMRD
<indbase>+54	CTUD	CTURD
<indbase>+55	CTLD	CTLRD
<indbase>+56	MR1H,MR2H	MR1H,MR2H
<indbase>+57	SRH	CSRH
<indbase>+58	Riservato	CRH
<indbase>+59	RHRH	THRH
<indbase>+60	Riservato	Riservato
<indbase>+61	Input Port D	OPCRD
<indbase>+62	Start C/T D	Riservato
<indbase>+63	Stop C/T D	Riservato

Table 5-2: Tabella di indirizzamento registri interni (parte 2)

Per quanto riguarda il significato dei registri sopra riportati si faccia riferimento all'apposita documentazione tecnica dell'UART SCC 2698, presente nell'appendice B di questo manuale.

### 5.3.1. Corrispondenza dip Switch e Registri Interni

Viene di seguito riportata quella che e' la corrispondenza tra i tre dip switch di I/O della scheda **UCC 01** con i registri interni dell' **UART SCC 2698**.

Dip 3.1 -> Bit 5 Input Port A  
Dip 3.2 -> Bit 7 Input Port A  
Dip 3.3 -> Bit 5 Input Port B  
Dip 3.4 -> Bit 7 Input Port B  
Dip 3.5 -> Bit 5 Input Port C  
Dip 3.6 -> Bit 7 Input Port C  
Dip 3.7 -> Bit 5 Input Port D  
Dip 3.8 -> Bit 7 Input Port D  
Dip 4.1 -> Bit 4 Input Port A  
Dip 4.2 -> Bit 6 Input Port A  
Dip 4.3 -> Bit 4 Input Port B  
Dip 4.4 -> Bit 6 Input Port B

Dip 4.5 -> Bit 4 Input Port C  
Dip 4.6 -> Bit 6 Input Port C  
Dip 4.7 -> Bit 4 Input Port D  
Dip 4.8 -> Bit 6 Input Port D  
Dip 5.1 -> Bit 1 Input Port A  
Dip 5.2 -> Bit 3 Input Port A  
Dip 5.3 -> Bit 1 Input Port B  
Dip 5.4 -> Bit 3 Input Port B  
Dip 5.5 -> Bit 1 Input Port C  
Dip 5.6 -> Bit 3 Input Port C  
Dip 5.7 -> Bit 1 Input Port D  
Dip 5.8 -> Bit 3 Input Port D

La funzione di tali dip switch e' quella di effettuare un minimo di colloquio utente programma, direttamente a bordo scheda.

Le applicazioni piu' immediate possono essere quelle di variare delle condizioni di lavoro o selezionare una serie di parametri al firmware del sistema.

Tutti i dip switch sono collegati con logica negata, quindi se posto in **ON** genera uno **Zero** logico, mentre se posto in **OFF** genera un **Uno** logico.

### 5.3.2. Handshake

Ogni linea seriale della **UCC 01** e' caratterizzata da due linee di handshake normalmente indicate con **RTSy** e **CTSy**, dove **y** coincide con il nome della linea seriale (da **A** a **H**).

Di seguito viene riportata la corrispondenza di tali linee di controllo della comunicazione, nei confronti dei pin dell'**UART** e dei suoi registri interni.

HANDSHAKE	PIN SCC 2698	BIT e REGISTRO
CTSA	MPIOA	Bit 0 Input Port A Bit 0 IPCRA
CTSB	MPIOB	Bit 2 Input Port A Bit 2 IPCRA
CTSC	MPIOC	Bit 0 Input Port B Bit 0 IPCRB
CTSD	MPIOD	Bit 2 Input Port B Bit 2 IPCRB
CTSE	MPIOE	Bit 0 Input Port C Bit 0 IPCRC
CTSF	MPIOF	Bit 2 Input Port C Bit 2 IPCRC
CTSG	MPIOG	Bit 0 Input Port D Bit 0 IPCRD
CTSH	MPIOH	Bit 2 Input Port D Bit 2 IPCRD
RTSA	MPOA	Bit 0,1,2 OPCRA
RTSB	MPOB	Bit 4,5,6 OPCRA
RTSC	MPOC	Bit 0,1,2 OPCRB
RTSD	MPOD	Bit 4,5,6 OPCRB
RTSE	MPOE	Bit 0,1,2 OPCRC
RTSF	MPOF	Bit 4,5,6 OPCRC
RTSG	MPOG	Bit 0,1,2 OPCRD
RTSH	MPOH	Bit 4,5,6 OPCRD

Da notare che sia gli handshake in uscita (**RTS**) che quelli d'ingresso (**CTS**) possono essere gestiti autonomamente dall'**UART** anche con diverse modalita' definibili via software.

Per ulteriori informazioni a riguardo dei segnali descritti e dei relativi registri per la gestione software si faccia riferimento all'appendice **B** del manuale.

## 6. SCHEDE ESTERNE A CUI COLLEGARE LA UCC 01

La scheda **UCC 01** ha la possibilita' di accettare come processori tutti quelli presenti sul **BUS ABACO(R)** aumentando, cosi', la sua gia' notevole versatilita'.

A titolo di esempio ne riportiamo un breve elenco:

### **CPU 01 (Central Processor Unit Z80)**

CPU Zilog Z80 B da 6 MHz con 64K RAM dinamica ed uno zoccolo per ospitare un massimo di 32K Bytes di EPROM. Interfaccia al Bus ABACO(R) con una ampia dotazione di Firmware e Software. Disponibilita' di sistemi operativi tipo CP/M, ZCPR3, SCDOS, ecc.

### **CPU 03 (Central Processor Unit Z80)**

CPU Z80 H da 8MHz con due linee seriali in RS 232 C con Baud Rate settabile da software tra 50 e 38K Baud. Una delle linee e' settabile in RS 485 od in RS 422. Zoccoli per un massimo di 448K Bytes di RAM-EPROM di cui 192K RAM e 256K EPROM. Disponibile CP/M.

### **GPC 02 (General Purpose Controller)**

Scheda in grado di supportare la famiglia 51 Intel compreso il tipo mascherato Basic. 16 linee di I/O, 3 Counter, linea RS 232, 4 linee di A/D converter da 10 bit, Buzzer, EPROM programmer su scheda, 32K RAM con Back Up al Litio, Key Display Controller, ecc.

### **GPC F2 (General Purpose Controller Fam. 8052)**

Scheda in grado di supportare la famiglia 51 Intel compreso il tipo mascherato Basic. 16 linee di I/O gestite da 8255, 2 linee RS 232, Buzzer, EPROM programmer, 32K RAM su 64K indirizzabili, Real Time Clock con batteria al Litio, 2 Timer Counter tipo 8253.

### **GPC 51 (General Purpose Controller)**

Scheda in grado di supportare la famiglia 51 Intel compreso il tipo mascherato Basic. 16 linee di I/O, 3 Counter, linea RS 232, 4 linee di A/D converter da 10 bit, Buzzer, EPROM programmer su scheda, 32K RAM con Back Up al Litio, Orologio, Key Display Controller.

### **GPC 65 (General Purpose Controller 6501 Q)**

La compatibilita' software del 6501 Q Rockwell con il noto 6502, consente l' utilizzo immediato di quanto gia' sviluppato. 32K RAM tamponati con batteria al Litio, 4 linee di A/D converter da 10 bit, Key Display Controller, linea RS 422 o 423, unica tensione di alimentazione.

**GPC 68 (General Purpose Controller 68000)**

Due linee RS 232 con Baud Rate settabile da software fino a 38 KBaud piu' una linea RS 485 o RS 422, 3 port paralleli a 8 bit e tre timer gestiti dal 68230, CPU costituita dal 68000 ad 8 MHz, 768 KByte di RAM-EPROM, Watch Dog hardware disinseribile, vari supporti software.

**GPC 80 ( General Purpose Controller 84C00 )**

Potente controllore o SBC basato sullo Z80. Unica tensione di alimentazione con componenti CMOS. Gestisce le FIO e dispone di implementazioni di CP/M, SCDOS, ZCPR3, ecc. 256K RAM tamponata con batteria al Litio; Real Time Clock; 2 linee seriali, 16 linee di I/O TTL, 4 counter.

**GPC 97 (General Purpose Controller 8097)**

Potente controllore a 16 bit basato sullo 8097 Intel con una linea RS 232 ed una in TTL, 32 linee di I/O, 8 linee di A/D converter da 10 bit, Real Time Clock, 16K RAM con batteria al Litio, unica tensione di alimentazione, comodo sviluppo su personal, ecc.

**GPC 451 ( General Purpose Controller 80C451 )**

Scheda in grado di supportare 80C451 compresa la versione con EPROM. Codice compatibile 8031. 32 linee di I/O TTL; Real Time Clock; 96K a bordo di cui 32K RAM tamponati con batteria al Litio. Linea in RS 232 o 422 o 485. BASIC, FORTH, Monitor in ROM, PASCAL, C, ecc.

**GPC 188 ( General Purpose Controller 80C188 )**

Potentissimo controllore con 256K EPROM e 256K RAM con batteria al Litio; Real Time Clock; 2 linee seriali in RS 232 di cui 1 settabile in RS 422-485; EEPROM; 24 linee di I/O TTL; 4 linee di A/D converter da 12 bit; Watch Dog. Efficiente Monitor Debugger, Compilatore PASCAL.

7. DISPOSIZIONE JUMPERS

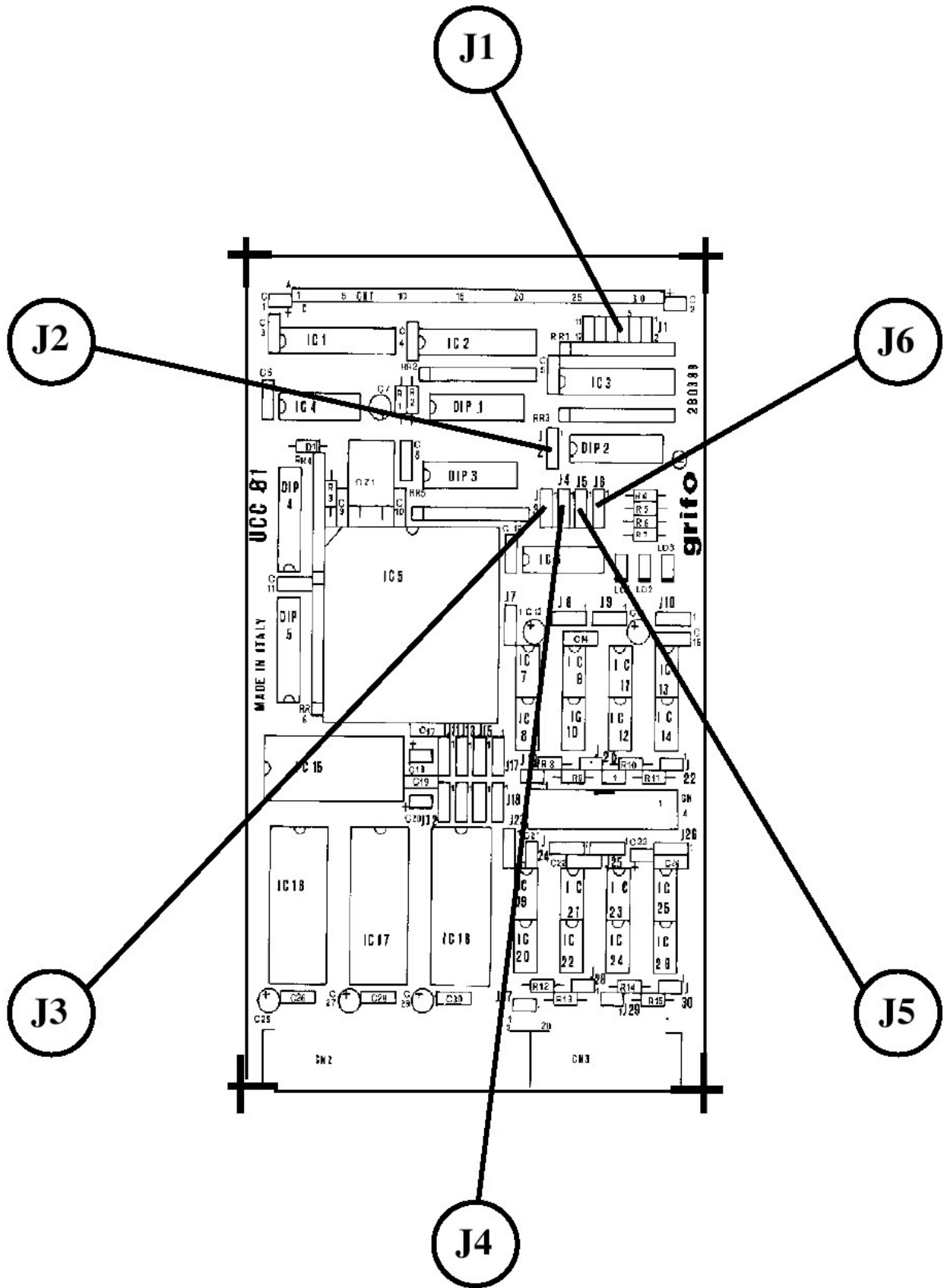


Fig. 7-1: Jumpers per Indirizzamento ed Interrupts

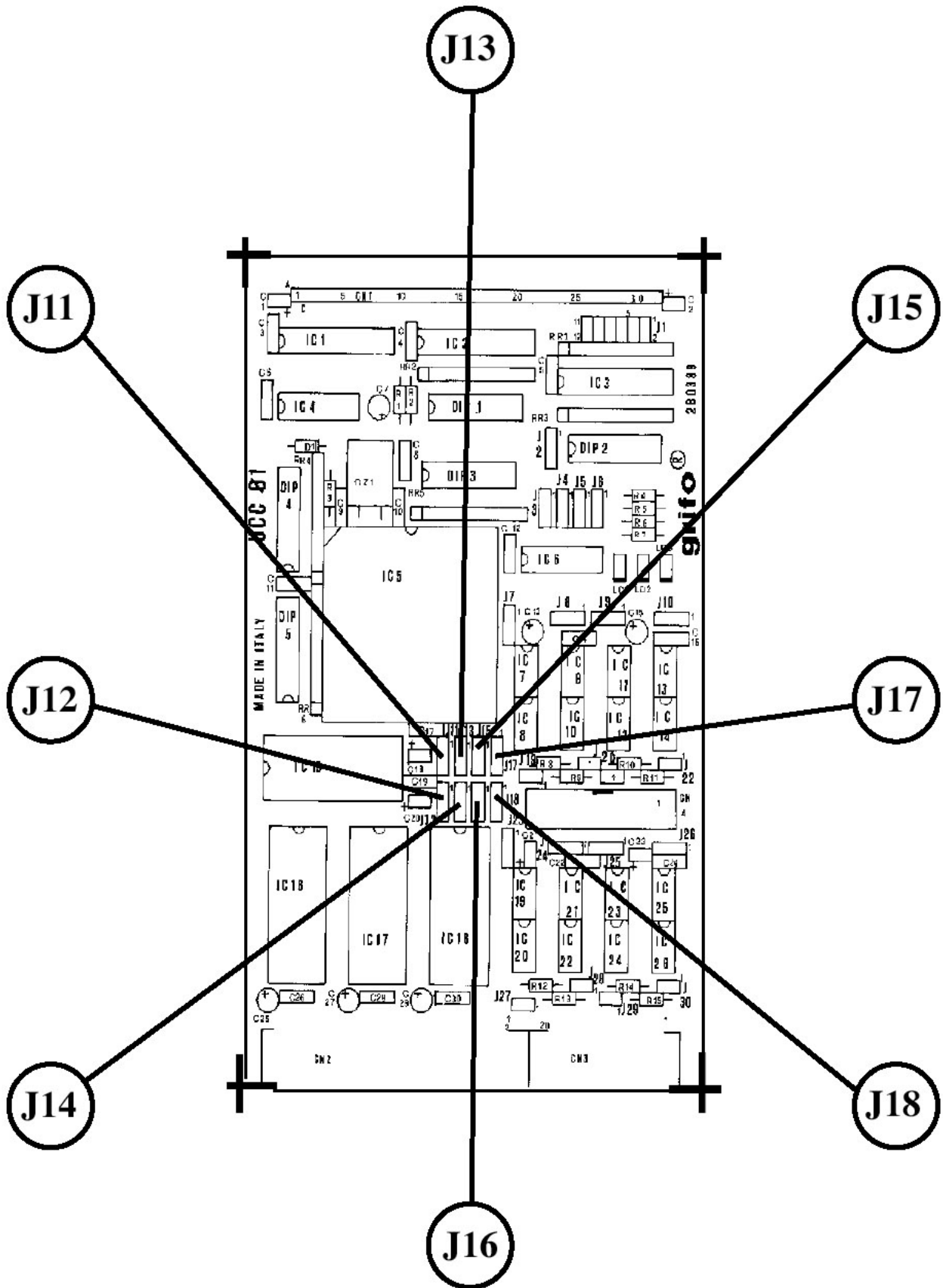


Fig. 7-2: Jumpers per Selezione RS 232 e RS 422-485



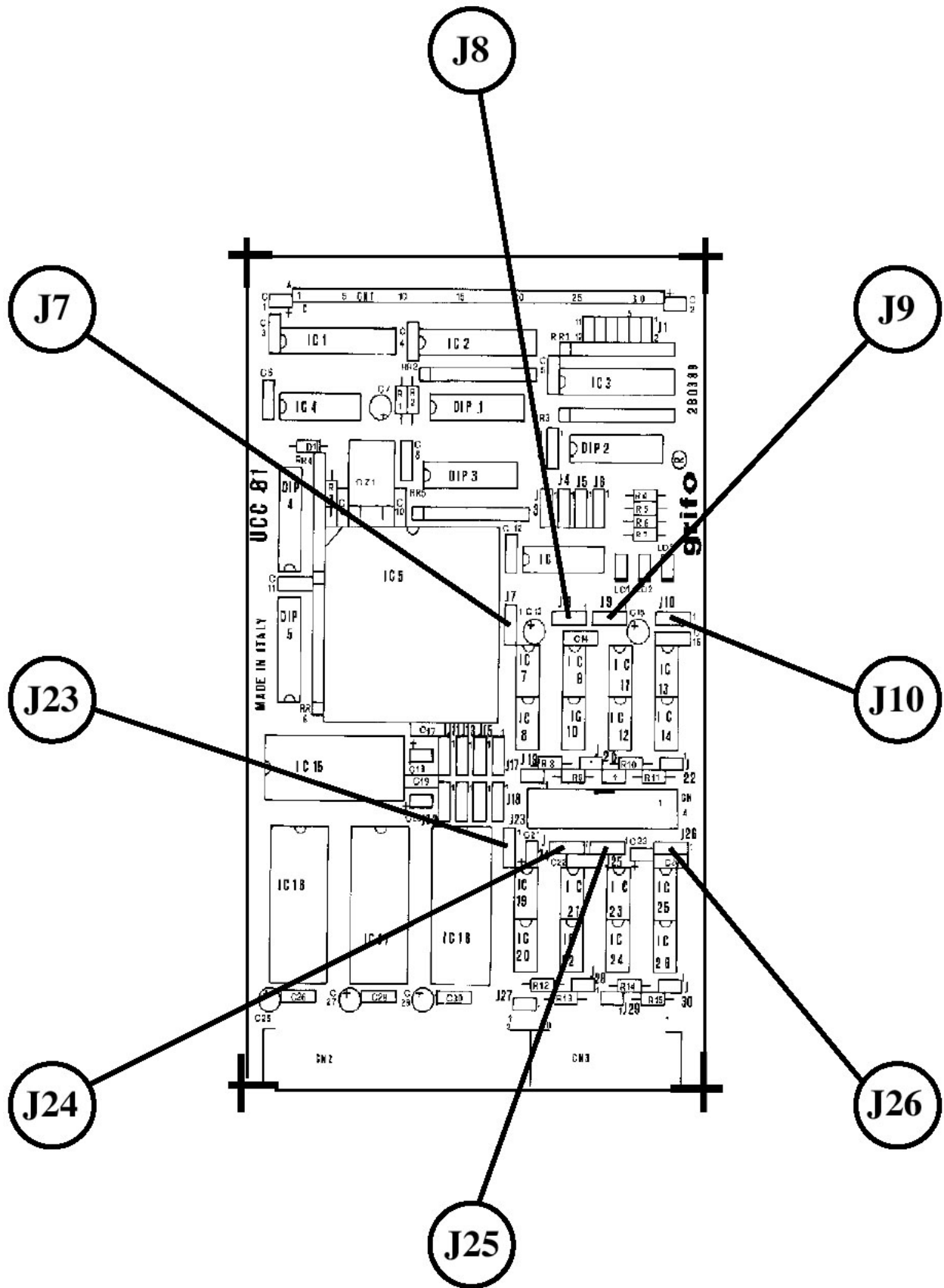


Fig. 7-3: Jumpers per Selezione Comunicazione Full/Half Duplex

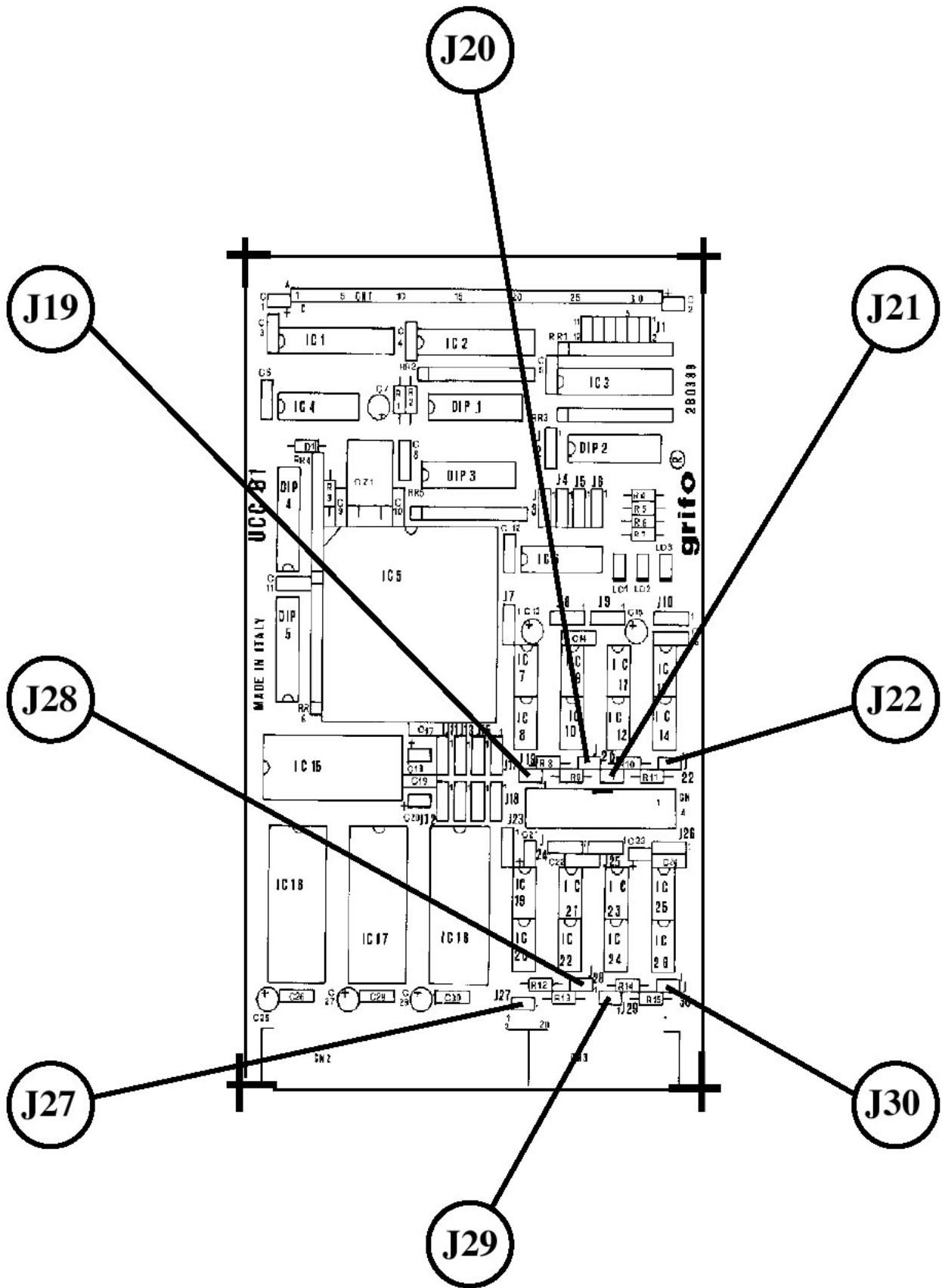


Fig. 7-4: Jumpers per Resistenza di Terminazione in RS 422-485

## 8. DATI TECNICI UART SCC 2698

Philips Semiconductors

Product specification

### Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

#### DESCRIPTION

The SCC2698B Enhanced Octal Universal Asynchronous Receiver/Transmitter (Octal UART) is a single chip MOS-LSI communications device that provides eight full-duplex asynchronous receiver/transmitter channels in a single package. It is fabricated with CMOS technology which combines the benefits of high density and low power consumption.

The operating speed of each receiver and transmitter can be selected independently as one of 26 fixed baud rates, a 16X clock derived from a programmable counter/timer, or an external 1X or 16X clock. The baud rate generator and counter/timer can operate directly from a crystal or from external clock inputs. The ability to independently program the operating speed of the receiver and transmitter make the Octal UART particularly attractive for dual-speed channel applications such as clustered terminal systems.

The receiver is quadruple buffered to minimize the potential of receiver overrun or to reduce interrupt overhead in interrupt driven systems. In addition, a handshaking (RTS/CTS) capability is provided to disable a remote UART transmitter when the receiver buffer is full.

The UART provides a power-down mode in which the oscillator is frozen but the register contents are stored. This results in reduced power consumption on the order of several magnitudes. The Octal UART is fully TTL compatible and operates from a single +5V power supply.

The SCC2698B is an upwardly compatible version of the 2698A Octal UART. In PLCC packaging, it is enhanced by the addition of receiver ready or FIFO full status outputs, and transmitter empty status outputs for each channel on 16 multipurpose I/O pins. The multipurpose pins of the 2698B RIO pins, thus DMA and modem control is provided.

#### FEATURES

- Eight full-duplex independent asynchronous receiver/transmitters
- Quadruple buffered receiver data register
- Programmable data format:
  - 5 to 8 data bits plus parity
  - Odd, even, no parity or force parity
  - 1, 1.5 or 2 stop bits programmable in 1/16-bit increments
- Baud rate for the receiver and transmitter selectable from:
  - 26 fixed rates: 50 to 38.4K baud
  - Non-standard rates to 115.2K baud
  - User-defined rates from the programmable counter/timer associated with each of four blocks
  - External 1x or 16x clock
- Parity, framing, and overrun error detection
- False start bit detection
- Line break detection and generation
- Programmable channel mode
  - Normal (full-duplex), automatic echo, local loop back, remote loopback
- Four multi-function programmable 16-bit counter/timers
- Four interrupt outputs with eight maskable interrupting conditions for each output
- Receiver ready/FIFO full and transmitter ready status available on 16 multi-function pins in PLCC package
- On-chip crystal oscillator
- TTL compatible
- Single +5V power supply with low power mode
- Eight multi-purpose output pins
- Sixteen multi-purpose I/O pins
- Sixteen multi-purpose Input pins with pull-up resistors

#### ORDERING INFORMATION

PACKAGES	COMMERCIAL	INDUSTRIAL	DWG #
	$V_{CC} = +5V \pm 5\%$ , $T_A = 0^\circ C$ to $+70^\circ C$	$V_{CC} = +5V \pm 5\%$ , $T_A = -40^\circ C$ to $+85^\circ C$	
84-Pin Plastic Leaded Chip Carrier (PLCC)	SCC2698BC1A84	SCC2698BE1A84	SOT189-3

NOTE: Pin Grid Array (PGA) package version is available from Philips Components Military Division.

#### ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING	UNIT
$T_A$	Operating ambient temperature range <sup>2</sup>	Note 4	°C
$T_{STG}$	Storage temperature range	-65 to +150	°C
$V_{CC}$	Voltage from $V_{DD}$ to GND <sup>3</sup>	-0.5 to +7.0	V
$V_S$	Voltage from any pin to ground <sup>3</sup>	-0.5 to $V_{CC} + 0.5$	V
$P_D$	Power dissipation	1	W

#### NOTES:

1. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operation section of this specification is not implied.
2. For operating at elevated temperatures, the device must be derated based on +150°C maximum junction temperature.
3. This product includes circuitry specifically designed for the protection of its internal devices from damaging effects of excessive static charge. Nonetheless, it is suggested that conventional precautions be taken to avoid applying any voltages larger than the rated maxima.
4. Parameters are valid over specified temperature range. See ordering information table for applicable temperature range and operating supply range.

Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

PIN CONFIGURATIONS

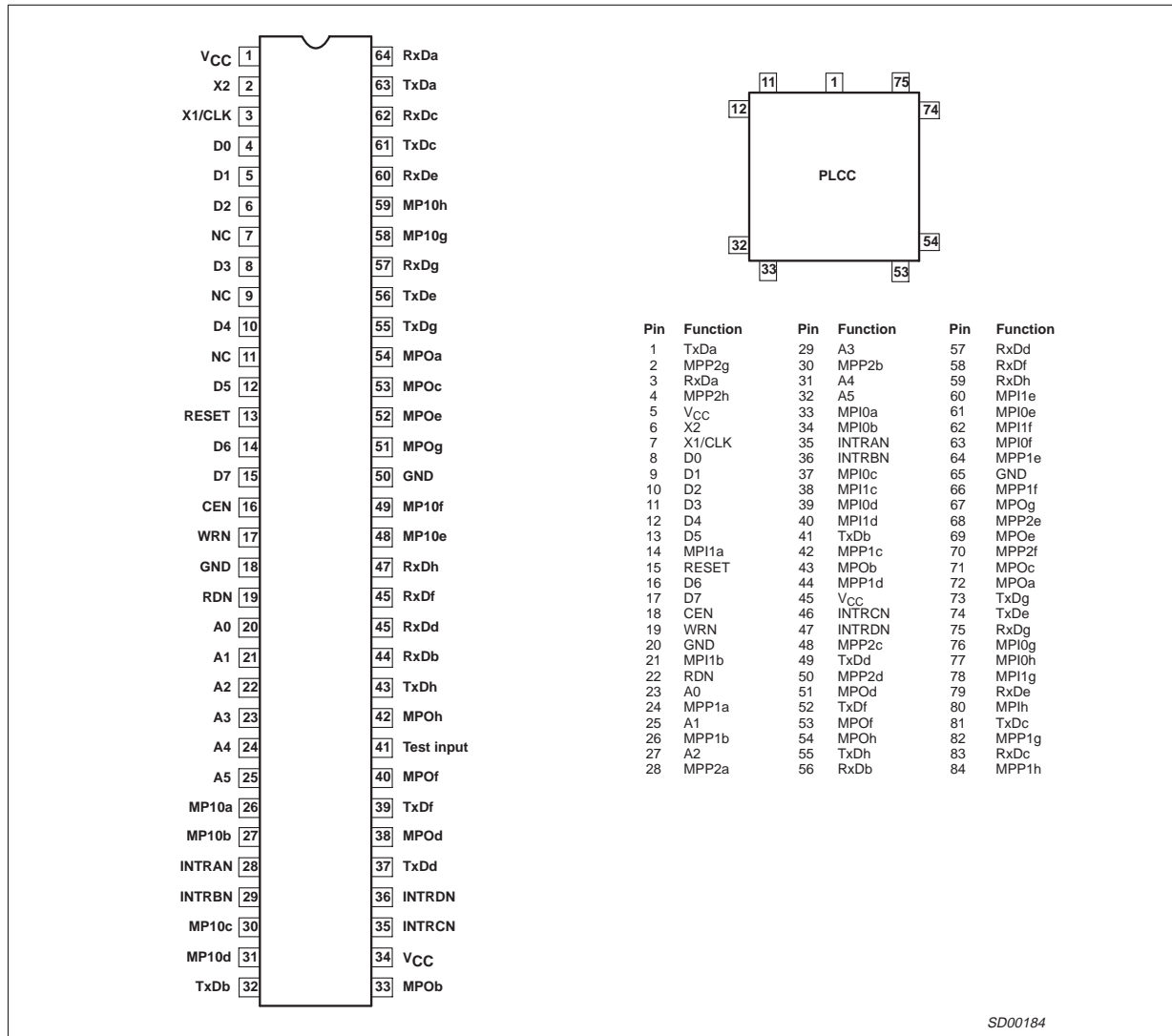


Figure 1. Pin Configurations



Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

BLOCK DIAGRAM

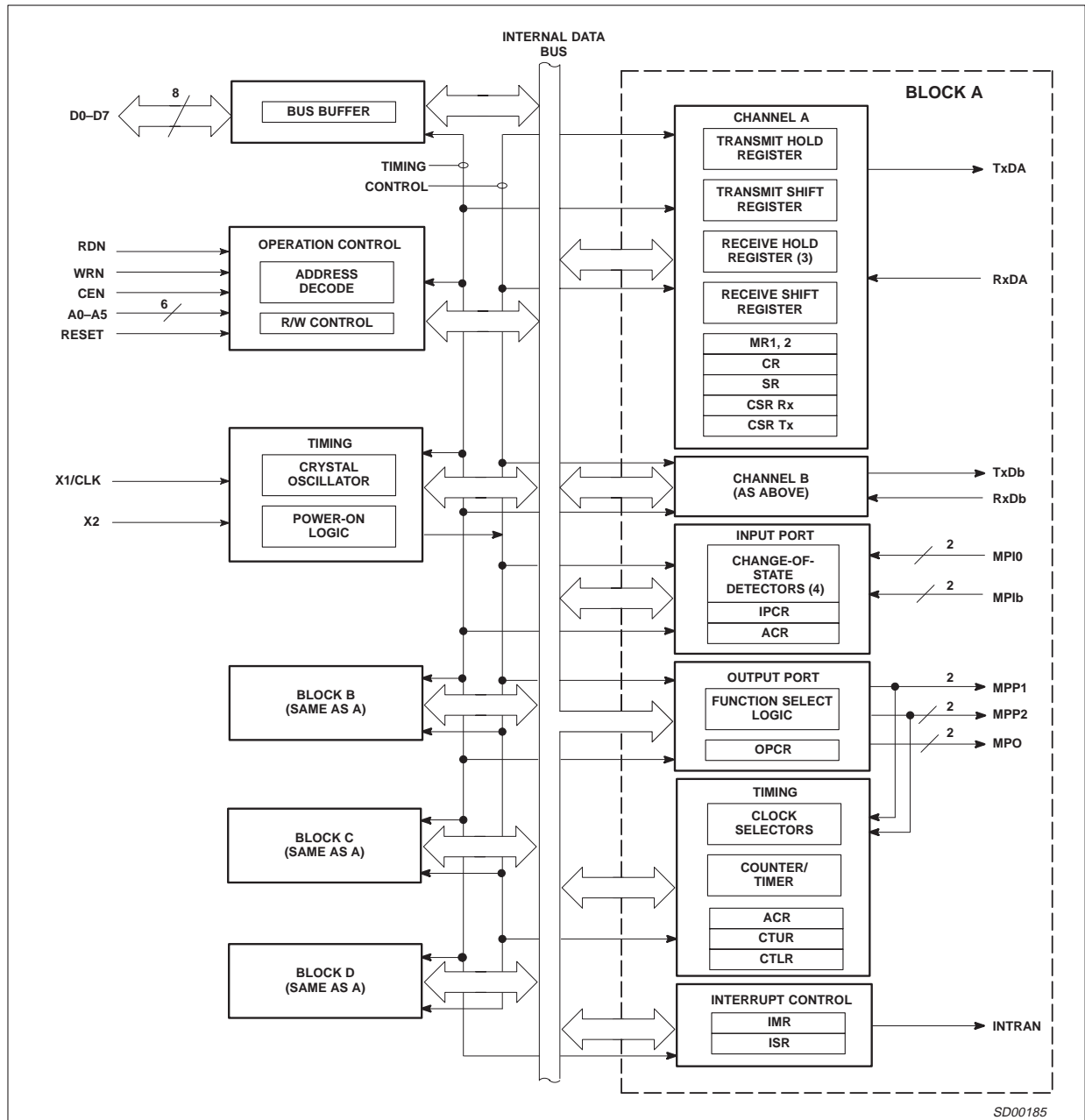


Figure 2. Block Diagram



## Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

**PIN DESCRIPTION**

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
D0–D7	8–13, 16, 17	I/O	<b>Data Bus:</b> Active–High 8-bit bidirectional 3-State data bus. Bit 0 is the LSB and bit 7 is the MSB. All data, command, and status transfers between the CPU and the Octal UART take place over this bus. The direction of the transfer is controlled by the WRN and RDN inputs when the CEN input is low. When the CEN input is High, the data bus is in the 3-State condition.
CEN	18	I	<b>Chip Enable:</b> Active-Low input. When Low, data transfers between the CPU and the Octal UART are enabled on D0–D7 as controlled by the WRN, RDN and A0–A5 inputs. When CEN is High, the Octal UART is effectively isolated from the data bus and D0–D7 are placed in the 3-State condition.
WRN	19	I	<b>Write Strobe:</b> Active-Low input. A Low on this pin while CEN is Low causes the contents of the data bus to be transferred to the register selected by A0–A5. The transfer occurs on the trailing (rising) edge of the signal.
RDN	22	I	<b>Read Strobe:</b> Active-Low input. A Low on this pin while CEN is Low causes the contents of the register selected by A0–A5 to be placed on the data bus. The read cycle begins on the leading (falling) edge of RDN.
A0–A5	23, 25, 27, 29, 31, 32	I	<b>Address Inputs:</b> Active-High address inputs to select the Octal UART registers for read/write operations.
RESET	15	I	<b>Reset:</b> Master reset. A High on this pin clears the status register (SR), clears the interrupt mask register (IMR), clears the interrupt status register (ISR), clears the output port configuration register (OPCR), places the receiver and transmitter in the inactive state causing the TxD output to go to the marking (High) state, and stops the counter/timer. Clears power-down mode and interrupts. Clears Test Modes, sets MR pointer to MR1.
INTRAN– INTRDN	35, 36, 46, 47	O	<b>Interrupt Request:</b> This active-Low open drain output is asserted on occurrence of one or more of eight maskable interrupting conditions. The CPU can read the interrupt status register to determine the interrupting condition(s). These pins require a pullup device and may be wire ORed.
X1/CLK	7	I	<b>Crystal 1:</b> Crystal or external clock input. When using the crystal oscillator, this pin serves as the connection for one side of the crystal. If a crystal is not used, an external clock is supplied at this input. An external clock (or crystal) is required even if the internal baud rate generator is not utilized. This clock is used to drive the internal baud rate generator, as an optional input to the timer/counter, and to provide other clocking signals required by the chip.
X2	6	I	<b>Crystal 2:</b> Connection for other side of crystal. If an external source is used instead of a crystal, this connection should be left open (see Figure 9).
RxDa–RxDh	3, 56, 83, 57, 79, 58, 75, 59	I	<b>Receiver Serial Data Input:</b> The least significant bit is received first. If external receiver clock is specified, this input is sampled on the rising edge of the clock. If internal clock is used, the RxD input is sampled on the rising edge of the RxC1x signal as seen on the MPO pin.
TxDa–TxDh	1, 41, 81, 49, 74, 52, 73, 55	O	<b>Transmitter Serial Data Output:</b> The least significant bit is transmitted first. This output is held in the marking (High) condition when the transmitter is idle or disabled and when the Octal UART is operating in local loopback mode. If external transmitter is specified, the data is shifted on the falling edge of the transmitter clock. If internal clock is used, the TxD output changes on the falling edge of the TxC1x signal as seen on the MPO pin.
MPOa–MPOh	72, 43, 71, 51, 69, 53, 67, 54	O	<b>Multi-Purpose Output:</b> Each of the four DUARTS has two MPO pins (one per UART). One of the following eight functions can be selected for this output pin by programming the OPCR (output port configuration register). Note that reset conditions MPO pins to RTSN. <b>RTSN</b> – Request to send active-Low output. This output is asserted and negated via the command register. By appropriate programming of the mode registers, (MR1[7])=1 RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted or when the receiver FIFO and shift register are full. RTSN is an internal signal which normally represents the condition of the receiver FIFO not full, i.e., the receiver can request more data to be sent. However, it can also be controlled by the transmitter empty and the commands 8h and 9h written to the CR (command register). <b>C/TO</b> – The counter/timer output. <b>TxC1X</b> – The 1X clock for the transmitter. <b>TxC16X</b> – The 16X clock for the transmitter. <b>RxC1X</b> – The 1X clock for the receiver. <b>RxC16X</b> – The 16X clock for the receiver. <b>TxRDY</b> – Transmitter holding register empty signal. <b>RxRDY/FFULL</b> – Receiver FIFO not empty/full signal.
MPI0a–MPI0h	33, 34, 37, 39, 61, 63, 76, 77	I	<b>Multi-Purpose Input 0:</b> This pin (one in each UART) is programmable. Its state can always be read through the IPCR bit 0, or the IPR bit 0. <b>CTSN:</b> By programming MR2[4] to a 1, this input controls the clear-to-send function for the transmitter. It is active low. This pin is provided with a change-of-state detector.

# Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

## PIN DESCRIPTION (Continued)

MNEMONIC	PIN NO.	TYPE	NAME AND FUNCTION
MPI1a–MPI1h	14, 21, 38, 40, 60, 62, 78, 80	I	<b>Multi-Purpose Input 1:</b> This pin (one for each UART) is programmable. Its state can always be determined by reading the IPCR bit 1 or IPR bit 1. <b>C/CLK</b> – This input will serve as the external clock for the counter/timer when ACR[5] is set to 0. This occurs only for channels a, c, e, and g since there is one counter/timer for each DUART block. This pin is provided with a change-of-state detector.
MPP1a–MPP1h	24, 26, 42, 44, 64, 66, 82, 84	I/O	<b>Multi-Purpose Pin 1:</b> This pin (one for each UART) is programmed to be an input or an output according to the state of OPCR[7]. (0 = input, 1 = output). The state of the multi-purpose pin can always be determined by reading the IPR. When programmed as an input, it will be the transmitter clock (TxCLK). It will be 1x or 16x according to the clock select registers (CSR[3.0]). When programmed as an output, it will be the status register TxRDY bit. These pins have a small pull-up device.
MPP2a–MPP2h	28, 30, 48, 50, 68, 70, 2, 4	I/O	<b>Multi-Purpose Pin 2:</b> This pin (one for each UART) is programmed to be an input or an output according to the state of OPCR[7]. (0 = input, 1 = output). The state of the multi-purpose pin can always be determined by reading the IPR. When programmed as an input, it will be the receiver clock (RxCLK). It will be 1x or 16x according to the clock select registers (CSR[7:4]). When programmed as an output, it will be the ISR status register RxRDY/FIFO full bit. These pins have a small pull-up device.
Test Input	–	I	<b>Test Input:</b> This pin is used as an input for test purposes at the factory while in test mode. This pin can be treated as 'N/C' by the user. It can be tied high, or left open.
V <sub>CC</sub>	5, 45	I	<b>Power Supply:</b> +5V supply input.
GND	20, 65	I	<b>Ground</b>

## BLOCK DIAGRAM

As shown in the block diagram, the Octal UART consists of: data bus buffer, interrupt control, operation control, timing, and eight receiver and transmitter channels. The eight channels are divided into four different blocks, each block independent of each other (see Figure 3). Figure 2 represents the DUART block.

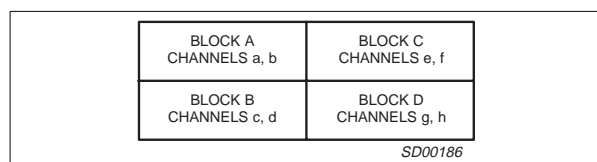


Figure 3. Channel Architecture

### Channel Blocks

There are four blocks (Figure 3), each containing two sets of receiver/transmitters. In the following discussion, the description applies to Block A which contains channels a and b. However, the same information applies to all channel blocks.

### Data Bus Buffer

The data bus buffer provides the interface between the external and internal data buses. It is controlled by the operation control block to allow read and write operations to take place between the controlling CPU and the Octal UART.

### Interrupt Control

A single interrupt output per DUART (INTRN) is provided which is asserted on occurrence of any of the following internal events:

- Transmit holding register ready for each channel
- Receive holding register ready or FIFO full for each channel
- Change in break received status for each channel
- Counter reached terminal count
- Change in MPI input

Associated with the interrupt system are the interrupt mask register (IMR) and the interrupt status register (ISR). The IMR can be programmed to select only certain conditions, of the above, to cause INTRN to be asserted. The ISR can be read by the CPU to determine all currently active interrupting conditions. However, the bits of the ISR are not masked by the IMR. The transmitter ready status and the receiver ready or FIFO full status can be provided on MPP1a, MPP1b, MPP2a, and MPP2b by setting OPCR[7]. these outputs are not masked by IMR.

### Operation Control

The operation control logic receives operation commands from the CPU and generates appropriate signals to internal sections to control device operation. It contains address decoding and read and write circuits to permit communications with the microprocessor via the data bus buffer. The functions performed by the CPU read and write operations are shown in Table 1.

Mode registers 1 and 2 are accessed via an auxiliary pointer. The pointer is set to MR1 by RESET or by issuing a reset pointer command via the command register. Any read or write of the mode register while the pointer is at MR1 switches the pointer to MR2 after the read or write. The pointer then remains at MR2 so that subsequent accesses are to MR2. To access MR1, the command 0001 of the command register must be executed.

### Timing Circuits

The timing block consists of a crystal oscillator, a baud rate generator, a programmable 16-bit counter/timer for each block, and two clock selectors.

### Crystal Clock

The crystal oscillator operates directly from a 3.6864MHz crystal connected across the X1/CLK and X2 inputs with a minimum of external components. If an external clock of the appropriate frequency is available, it may be connected to X1/CLK. If an external



# Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

## SCC2698B

clock is used instead of a crystal, X1 must be driven and X2 left floating as shown in Figure 9. The clock serves as the basic timing reference for the baud rate generator (BRG), the counter/timer, and

other internal circuits. A clock frequency, within the limits specified in the electrical specifications, must be supplied even if the internal BRG is not used.

**Table 1. Register Addressing**

Units A and B									Units E and F								
A5	A4	A3	A2	A1	A0	READ (RDN=0)	WRITE (WRN=0)	A5	A4	A3	A2	A1	A0	READ (RDN=0)	WRITE (WRN=0)		
0	0	0	0	0	0	MR1a, MR2a	MR1a, MR2a	1	0	0	0	0	0	MR1e, MR2e	MR1e, MR2e		
0	0	0	0	0	1	SRa	CSRa	1	0	0	0	0	1	SRe	CSRe		
0	0	0	0	1	0	BRG Test <sup>2</sup>	CRa	1	0	0	0	1	0	Reserved <sup>1</sup>	CRe		
0	0	0	0	1	1	RHRa	THRa	1	0	0	0	1	1	RHRe	THRe		
0	0	0	1	0	0	IPCRA	ACRA	1	0	0	1	0	0	IPCRC	ACRC		
0	0	0	1	0	1	ISRA	IMRA	1	0	0	1	0	1	ISRC	IMRC		
0	0	0	1	1	0	CTUA	CTPUA	1	0	0	1	1	0	CTUC	CTPUC		
0	0	0	1	1	1	CTLA	CTPLA	1	0	0	1	1	1	CTLC	CTPLC		
0	0	1	0	0	0	MR1b, MR2b	MR1b, MR2b	1	0	1	0	0	0	MR1f, MR2f	MR1f, MR2f		
0	0	1	0	0	1	SRb	CSRb	1	0	1	0	0	1	SRf	CSRf		
0	0	1	0	1	0	1X/16X Test <sup>2</sup>	CRb	1	0	1	0	1	0	Reserved <sup>1</sup>	CRf		
0	0	1	0	1	1	RHRb	THRb	1	0	1	0	1	1	RHRf	THRf		
0	0	1	1	0	0	Reserved <sup>1</sup>	Reserved <sup>1</sup>	1	0	1	1	0	0	Reserved <sup>1</sup>	Reserved <sup>1</sup>		
0	0	1	1	0	1	Input port A	OPCRA	1	0	1	1	0	1	Input port C	OPCRC		
0	0	1	1	1	0	Start C/T A	Reserved <sup>1</sup>	1	0	1	1	1	0	Start C/T C	Reserved <sup>1</sup>		
0	0	1	1	1	1	Stop C/T A	Reserved <sup>1</sup>	1	0	1	1	1	1	Stop C/T C	Reserved <sup>1</sup>		
Units C and D									Units G and H								
0	1	0	0	0	0	MR1c, MR2c	MR1c, MR2c	1	1	0	0	0	0	MR1g, MR2g	MR1g, MR2g		
0	1	0	0	0	1	SRc	CSRc	1	1	0	0	0	1	SRg	CSRg		
0	1	0	0	1	0	Reserved <sup>1</sup>	CRc	1	1	0	0	1	0	Reserved <sup>1</sup>	CRg		
0	1	0	0	1	1	RHRc	THRc	1	1	0	0	1	1	RHRg	THRg		
0	1	0	1	0	0	IPCRB	ACRB	1	1	0	1	0	0	IPCRC	ACRC		
0	1	0	1	0	1	ISRB	IMRB	1	1	0	1	0	1	ISR	IMR		
0	1	0	1	1	0	CTUB	CTPUB	1	1	0	1	1	0	CTUD	CTPUD		
0	1	0	1	1	1	CTLB	CTPLB	1	1	0	1	1	1	CTLD	CTPLD		
0	1	1	0	0	0	MR1d, MR2d	MR1d, MR2d	1	1	1	0	0	0	MR1h, MR2h	MR1h, MR2h		
0	1	1	0	0	1	SRd	CSRd	1	1	1	0	0	1	SRh	CSRh		
0	1	1	0	1	0	Reserved <sup>1</sup>	CRd	1	1	1	0	1	0	Reserved <sup>1</sup>	CRh		
0	1	1	0	1	1	RHRd	THRd	1	1	1	0	1	1	RHRh	THRh		
0	1	1	1	0	0	Reserved <sup>1</sup>	Reserved <sup>1</sup>	1	1	1	1	0	0	Reserved <sup>1</sup>	Reserved <sup>1</sup>		
0	1	1	1	0	1	Input port B	OPCRB	1	1	1	1	0	1	Input port D	OPCRD		
0	1	1	1	1	0	Start C/T B	Reserved <sup>1</sup>	1	1	1	1	1	0	Start C/T D	Reserved <sup>1</sup>		
0	1	1	1	1	1	Stop C/T B	Reserved <sup>1</sup>	1	1	1	1	1	1	Stop C/T D	Reserved <sup>1</sup>		

**NOTE:**

1. Reserved registers should never be read during normal operation since they are reserved for internal diagnostics.

ACR = Auxiliary control register  
 CR = Command register  
 CSR = Clock select register  
 CTL = Counter/timer lower  
 CTPL = Counter/timer preset lower register  
 CTU = Counter/timer upper  
 CTPU = Counter/timer preset upper register  
 MR = Mode register

SR = Status Register  
 THR = Tx holding register  
 RHR = Rx holding register  
 IPCR = Input port change register  
 ISR = Interrupt status register  
 IMR = Interrupt mask register  
 OPCR = Output port configuration register

2. See Table 5 for BRG Test frequencies in this data sheet, and "Extended baud rates for SCN2681, SCN68681, SCC2691, SCC2692, SCC68681 and SCC2698B" Philips Semiconductors ICs for Data Communications, IC-19, 1994.

**BRG**

The baud rate generator operates from the oscillator or external clock input and is capable of generating 26 commonly used data communications baud rates ranging from 50 to 115.2K baud. Thirteen of these are available simultaneously for use by the receiver and transmitter. Eight are fixed, and one of two sets of five can be selected by programming ACR[7]. The clock outputs from the BRG are at 16X the actual baud rate. The counter/timer can be

used as a timer to produce a 16X clock for any other baud rate by counting down the crystal clock or an external clock. The clock selectors allow the independent selection, by the receiver and transmitter, of any of these baud rates or an external timing signal.

**Counter-Timer**

The four Counter/Timers are programmable 16 bit dividers that are used for generating miscellaneous clocks or generating timeout





## Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

periods. These clocks may be used by any or all of the receivers and transmitters in the OCTART or may be directed to an I/O pin for miscellaneous use.

### Counter/Timer programming

The counter timer is a 16-bit programmable divider that operates in one of three modes: counter, timer, and time out.

- Timer mode generates a square wave.
- Counter mode generates a time delay.
- Time out mode counts time between received characters.

The C/T uses the numbers loaded into the Counter/Timer Lower Register (CTPL) and the Counter/Timer Upper Register (CTPU) as its divisor. The counter timer is controlled with six commands: Start/Stop C/T, Read/Write Counter/Timer lower register and Read/Write Counter/Timer upper register. These commands have slight differences depending on the mode of operation. Please see the detail of the commands under the CTPL/CTPU register descriptions.

### Baud Rate Generation

When these timers are selected as baud rates for receiver or transmitter via the Clock Select register their output will be configured as a 16x clock. Therefore one needs to program the timers to generate a clock 16 times faster than the data rate. The formula for calculating 'n', the number loaded to the CTPU and CTPL registers, based on a particular input clock frequency is shown below.

For the timer mode the formula is as follows:

$$n = \frac{\text{Clockinputfrequency}}{2^{16} \cdot \text{Baudratedesired}}$$

NOTE: 'n' may not assume values of 0 and 1.

The frequency generated from the above formula will be at a rate 16 times faster than the desired baud rate. The transmitter and receiver state machines include divide by 16 circuits, which provide the final frequency and provide various timing edges used in the qualifying the serial data bit stream. Often this division will result in a non-integer value: 26.3 for example. One may only program integer numbers to a digital divider. There for 26 would be chosen. If 26.7 were the result of the division then 27 would be chosen. This gives a baud rate error of 0.3/26.3 or 0.3/26.7 that yields a percentage error of 1.14% or 1.12% respectively, well within the ability of the asynchronous mode of operation. Higher input frequency to the counter reduces the error effect of the fractional division

One should be cautious about the assumed benign effects of small errors since the other receiver or transmitter with which one is communicating may also have a small error in the precise baud rate. In a "clean" communications environment using one start bit, eight data bits and one stop bit the total difference allowed between the transmitter and receiver frequency is approximately 4.6%. Less than eight data bits will increase this percentage.

### Receiver and Transmitter

The Octal UART has eight full-duplex asynchronous receiver/transmitters. The operating frequency for the receiver and transmitter can be selected independently from the baud rate generator, the counter/timer, or from an external input.

Registers associated with the communications channel are the mode registers (MR1 and MR2), the clock select register (CSR), the command register (CR), the status register (SR), the transmit holding register (THR), and the receive holding register (RHR).

### Transmitter

The SCC2698 is conditioned to transmit data when the transmitter is enabled through the command register. The SCC2698 indicates to the CPU that it is ready to accept a character by setting the TxRDY bit in the status register. This condition can be programmed to generate an interrupt request at MPO or MPP1 and INTRN. When the transmitter is initially enabled the TxRDY and TxEMT bits will be set in the status register. When a character is loaded to the transmit FIFO the TxEMT bit will be reset. The TxEMT will not set until: 1) the transmit FIFO is empty and the transmit shift register has finished transmitting the stop bit of the last character written to the transmit FIFO, or 2) the transmitter is disabled and then re-enabled. The TxRDY bit is set whenever the transmitter is enabled and the TxFIFO is not full. Data is transferred from the holding register to transmit shift register when it is idle or has completed transmission of the previous character. Characters cannot be loaded into the TxFIFO while the transmitter is disabled.

The transmitter converts the parallel data from the CPU to a serial bit stream on the TxD output pin. It automatically sends a start bit followed by the programmed number of data bits, an optional parity bit, and the programmed number of stop bits. The least significant bit is sent first. Following the transmission of the stop bits, if a new character is not available in the TxFIFO, the TxD output remains High and the TxEMT bit in the Status Register (SR) will be set to 1. Transmission resumes and the TxEMT bit is cleared when the CPU loads a new character into the TxFIFO.

If the transmitter is disabled, it continues operating until the character currently being transmitted and any characters in the TxFIFO including parity and stop bit(s) have been completed.

The transmitter can be forced to send a continuous Low condition by issuing a send break command from the command register. The transmitter output is returned to the normal high with a stop break command.

The transmitter can be reset through a software command. If it is reset, operation ceases immediately and the transmitter must be enabled through the command register before resuming operation.

If CTS option is enabled (MR2[4] = 1), the CTSN input at MPI0 must be Low in order for the character to be transmitted. The transmitter will check the state of the CTS input at the beginning of each character transmitted. If it is found to be High, the transmitter will delay the transmission of any following characters until the CTS has returned to the low state. CTS going high during the serialization of a character will not affect that character.

### Transmitter "RS485 turnaround"

The transmitter can also control the RTSN outputs, MPO via MR2[5]. When this mode of operation is set, the meaning of the MPO signal will usually be 'end of message'. See description of the MR2[5] bit for more detail.

### Transmitter Flow control

The transmitter may be controlled by the CTSN input when enabled by MR2(4). The CTSN input would be connected to RTSN output of the receiver to which it is communicating. See further description in the MR 1 and MR2 register descriptions.

### Receiver

The SCC2698 is conditioned to receive data when enabled through the command register. The receiver looks for a High-to-Low (mark-to-space) transition of the start bit on the RxD input pin. If a transition is detected, the state of the RxD pin is sampled each 16X clock for 7-1/2 clocks (16X clock mode) or at the next rising edge of

## Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

**SCC2698B**

the bit time clock (1X clock mode). If RxD is sampled high, the start bit is invalid and the search for a valid start bit begins again. If RxD is still low, a valid start bit is assumed. The receiver then continues to sample the input at one-bit time intervals at the theoretical center of the bit. When the proper number of data bits and parity bit (if any) have been assembled, with one half-stop bit the character will be considered complete. The least significant bit is received first. The data is then transferred to the Receive FIFO and the RxRDY bit in the SR is set to a 1. This condition can be programmed to generate an interrupt at MPO or MPP2 and INTRN. If the character length is less than 8 bits, the most significant unused bits in the RxFIFO are set to zero.

### Receiver FIFO

The RxFIFO consists of a First-In-First-Out (FIFO) stack with a capacity of 3 characters. Data is loaded from the receive shift register into the topmost empty position of the FIFO. The RxRDY bit in the status register is set whenever one or more characters are available to be read, and a FFULL status bit is set if all three (3) stack positions are filled with data. Either of these bits can be selected to cause an interrupt. A read of the RxFIFO outputs the data at the top of the FIFO. After the read cycle, the data FIFO and its associated status bits (see below) are 'popped' thus emptying a FIFO position for new data.

### Receiver Status Bits

There are five (5) status bits that are evaluated with each byte (or character) received: received break, framing error, parity error, overrun error, and change of break. The first three are appended to each byte and stored in the RxFIFO. The last two are not necessarily related to the byte being received or a byte that is in the RxFIFO. They are however developed by the receiver state machine.

The received break, framing error, parity error and overrun error (if any) are strobed into the RxFIFO at the received character boundary, before the RxRDY status bit is set. For character mode (see below) status reporting the SR (Status Register) indicates the condition of these bits for the character that is the next to be read from the FIFO

The "received break" will always be associated with a zero byte in the RxFIFO. It means that zero character was a break character and not a zero data byte. The reception of a break condition will always set the "change of break" (see below) status bit in the Interrupt Status Register (ISR). The Change of break condition is reset by a reset error status command in the command register

### Break Detection

If a break condition is detected (RxD is Low for the entire character including the stop bit), a character consisting of all zeros will be loaded into the RxFIFO and the received break bit in the SR is set to 1. The change of break bit also sets in the ISR. The RxD input must return to high for two (2) clock edges of the X1 crystal clock for the receiver to recognize the end of the break condition and begin the search for a start bit.

This will usually require a high time of one X1 clock period or 3 X1 edges since the clock of the controller is not synchronous to the X1 clock.

### Framing Error

A framing error occurs when a non-zero character whose parity bit (if used) and stop; bit are zero. If RxD remains low for one half of the bit period after the stop bit was sampled, then the receiver operates as if the start bit of the next character had been detected.

The parity error indicates that the receiver-generated parity was not the same as that sent by the transmitter.

The framing, parity and received break status bits are reset when the associated data byte is read from the RxFIFO since these "error" conditions are attached to the byte that has the error

### Overrun Error

The overrun error occurs when the RxFIFO is full, the receiver shift register is full, and another start bit is detected. At this moment the receiver has 4 valid characters and the start bit of the 5<sup>th</sup> has been seen. At this point the host has approximately 6/16-bit time to read a byte from the RxFIFO or the overrun condition will be set. The 5<sup>th</sup> character then overruns the 4<sup>th</sup> and the 6<sup>th</sup> the 5<sup>th</sup> and so on until an open position in the RxFIFO is seen. ("seen" meaning at least one byte was read from the RxFIFO.)

Overrun is cleared by a use of the "error reset" command in the command register.

The fundamental meaning of the **overrun** is that data has been lost. Data in the RxFIFO remains valid. The receiver will begin placing characters in the RxFIFO as soon as a position becomes vacant.

Note: Precaution must be taken when reading an overrun FIFO. There will be 3 valid characters in the receiver FIFO. There will be one character in the receiver shift register. However it will NOT be known if more than one "over-running" character has been received since the overrun bit was set. The 4<sup>th</sup> character is received and read as valid but it will not be known how many characters were lost between the two characters of the 3<sup>rd</sup> and 4<sup>th</sup> reads of the RxFIFO

The "Change of break" means that either a break has been detected or that the break condition has been cleared. This bit is available in the ISR. The break change bit being set in the ISR and the received break bit being set in the SR will signal the beginning of a break. At the termination of the break condition only the change of break in the ISR will be set. After the break condition is detected the termination of the break will only be recognized when the RxD input has returned to the high state for **two** successive edges of the 1x clock; 1/2 to 1 bit time (see above).

The receiver is disabled by reset or via CR commands. A disabled receiver will not interrupt the host CPU under any circumstance in the **normal** mode of operation. If the receiver is in the multi-drop or special mode, it will be partially enabled and thus may cause an interrupt. Refer to section on Wake-Up and the register description for MR1 for more information.

### Receiver Status Modes (block and character)

In addition to the data word, three status bits (parity error, framing error, and received break) are also appended to each data character in the FIFO (overrun is not). Status can be provided in two ways, as programmed by the error mode control bit in the mode register. In the 'character' mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the 'block' mode, the status provided in the SR for these three bits is the logical-OR of the status for all characters coming to the top of the FIFO since the last 'reset error' command was issued. In either mode reading the SR does not affect the FIFO. The FIFO is 'popped' only when the RxFIFO is read. Therefore the status register should be read prior to reading the FIFO.

### Receiver Flow Control

The receiver can control the deactivation of RTS. If programmed to operate in this mode, the RTSN output will be negated when a valid start bit was received and the FIFO is full. When a FIFO position becomes available, the RTSN output will be re-asserted automatically. This feature can be used to prevent an overrun, in the receiver, by connecting the RTSN output to the CTSN input of the transmitting device.

## Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

receiver indicating that the receiver is ready to receive data. It is also active low and is, thus, called RTSN. RTSN is on pin MPO. A receiver's RTS output will usually be connected to the CTS input of the associated transmitter. Therefore, one could say that RTS and CTS are different ends of the same wire!

MR2(4) is the bit that allows the transmitter to be controlled by the CTS pin (MPI0). When this bit is set to one AND the CTS input is driven high, the transmitter will stop sending data at the end of the present character being serialized. It is usually the RTS output of the receiver that will be connected to the transmitter's CTS input. The receiver will set RTS high when the receiver FIFO is full AND the start bit of the fourth character is sensed. Transmission then stops with four valid characters in the receiver. When MR2(4) is set to one, CTSN must be at zero for the transmitter to operate. If MR2(4) is set to zero, the MPI0 pin will have no effect on the operation of the transmitter.

MR1(7) is the bit that allows the receiver to control MPO. When MPO is controlled by the receiver, the meaning of that pin will be RTS. However, a point of confusion arises in that MPO may also be controlled by the transmitter. When the transmitter is controlling this pin, its meaning is not RTS at all. It is, rather, that the transmitter has finished sending its last data byte. Programming the MPO pin to be controlled by the receiver and the transmitter at the same time is allowed, but would usually be incompatible.

RTS can also be controlled by the commands 1000 and 1001 in the command register. RTS is expressed at the MP0 pin which is still an output port. Therefore, the state of MP0 should be set low (either by commands of the CR register or by writing to the Output Port Configuration Register) for the receiver to generate the proper RTS signal. The logic at the output is basically a NAND of the MP0 bit register and the RTS signal as generated by the receiver. When the RTS flow control is selected via the MR1(7) bit the state of the MP0 register is not changed. Terminating the use of "Flow Control" (via the MR registers) will return the MP0 pin to the control of the MP0 register.

### Transmitter Disable Note

When the TxEMT bit is set the sequence of instructions: enable transmitter — load transmit holding register — disable transmitter will often result in nothing being sent. In the condition of the TxEMT being set do not issue the disable until the TxRDY bit goes active again after the character is loaded to the TxFIFO. The data is not sent if the time between the end of loading the transmit holding register and the disable command is less than 3/16 bit time in the 16x mode. One bit time in the 1x mode.

This is sometimes the condition when the RS485 automatic "turn-around" is enabled. It will also occur when only one character is to be sent and it is desired to disable the transmitter immediately after the character is loaded.

In general, when it is desired to disable the transmitter before the last character is sent AND the TxEMT bit is set in the status register

be sure the TxRDY bit is active immediately before issuing the transmitter disable instruction. (TxEMT is always set if the transmitter has underrun or has just been enabled), TxRDY sets at the end of the "start bit" time. It is during the start bit that the data in the transmit holding register is transferred to the transmit shift register.

### MULTI-PURPOSE INPUT PIN

The inputs to this unlatched 8-bit port for each block can be read by the CPU, by performing a read operation as shown in Table 1. A High input results in a logic one, while a Low input results in a logic zero. When the input port pins are read on the 84-pin LLCC, they will appear on the data bus in alternating pairs (i.e., DB0 = MP10a, DB1 = MP11a, DB2 = MPI0b, DB3 = MP11b, DB4 = MPP1a, DB5 = MPP2a, DB6 = MPP1b, DB7 = MPP2b). Although this example is shown for input port 'A', all ports will have a similar order).

The MPI pin can be programmed as an input to one of several Octal UART circuits. The function of the pin is selected by programming the appropriate control register. Change-of-state detectors are provided for MPI0 and MPI1 for each channel in each block. A High-to-Low or Low-to-High transition of the inputs lasting longer than 25 to 50µs sets the MPI change-of-state bit in the interrupt status register. The bit is cleared via a command. The change-of-state can be programmed to generate an interrupt to the CPU by setting the corresponding bit in the interrupt mask register.

The input port pulse detection circuitry uses a 38.4KHz sampling clock, derived from one of the baud rate generator taps. This produces a sampling period of slightly more than 25µs (assuming a 3.6864MHz oscillator input). The detection circuitry, in order to guarantee that a true change in level has occurred, requires two successive samples be observed at the new logic level. As a consequence, the minimum duration of the signal change is 25µs if the transition occurs coincident with the first sample pulse. (The 50µs time refers to the condition where the change-of-state is just missed and the first change of state is not detected until after an additional 25µs.)

### MULTI-PURPOSE I/O PINS

The multi-purpose pins (MPP) can be programmed as inputs or outputs using OPCR[7]. When programmed as inputs, the functions of the pins are selected by programming the appropriate control registers. When programmed as outputs, the two MPP1 pins (per block) will provide the transmitter ready (TxRDY) status for each channel and the MPP2 pins will provide the receiver ready or FIFO full (RxRDY/FFULL) status for each channel.

### MULTI-PURPOSE OUTPUT PIN

This pin can be programmed to serve as a request-to-send output, the counter/timer output, the output for the 1X or 16X transmitter or receiver clocks, the TxRDY output or the RxRDY/FFULL output (see OPCR [2:0] and OPCR [6:4] – MPO Output Select).

## Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

**SCC2698B**

### REGISTERS

The operation of the Octal UART is programmed by writing control words into the appropriate registers. Operational feedback is provided via status registers which can be read by the CPU. Addressing of the registers is described in Table 1.

The bit formats of the Octal UART registers are depicted in Table 2. These are shown for block A. The bit format for the other blocks is the same.

#### MR1 – Mode Register 1

MR1 is accessed when the MR pointer points to MR1. The pointer is set to MR1 by RESET or by a set pointer command applied via the CR. After reading or writing MR1, the pointers are set at MR2.

##### MR1[7] – Receiver Request-to-Send Control

This bit controls the deactivation of the RTSN output (MPO) by the receiver. This output is manually asserted and negated by commands applied via the command register. MR1[7] = 1 causes RTSN to be automatically negated upon receipt of a valid start bit if the receiver FIFO is full. RTSN is reasserted when an empty FIFO position is available. This feature can be used to prevent overrun in the receiver by using the RTSN output signal to control the CTS input of the transmitting device.

##### MR1[6] – Receiver Interrupt Select

This bit selects either the receiver ready status (RxRDY) or the FIFO full status (FFULL) to be used for CPU interrupts.

##### MR1[5] – Error Mode Select

This bit selects the operating mode of the three FIFOed status bits (FE, PE, received break). In the character mode, status is provided on a character-by-character basis; the status applies only to the character at the top of the FIFO. In the block mode, the status provided in the SR for these bits is the accumulation (logical-OR) of the status for all characters coming to the top of the FIFO since the last reset error command was issued.

##### MR1[4:3] – Parity Mode Select

If 'with parity' or 'force parity' is selected, a parity bit is added to the transmitted character and the receiver performs a parity check on incoming data. MR1[4:3] = 11 selects the channel to operate in the special wake-up mode.

##### MR1[2] – Parity Type Select

This bit selects the parity type (odd or even) if the 'with parity' mode is programmed by MR1[4:3], and the polarity of the forced parity bit if the 'force parity' mode is programmed. It has no effect if the 'no parity' mode is programmed. In the special 'wake-up' mode, it selects the polarity of the transmitted A/D bit.

##### MR1[1:0] – Bits Per Character Select

This field selects the number of data bits per character to be transmitted and received. The character length does not include the start, parity, and stop bits.

#### MR2 – Mode Register 2

MR2 is accessed when the channel MR pointer points to MR2, which occurs after any access to MR1. Accesses to MR2 do not change the pointer.

#### MR2[7:6] – Mode Select

The Octal UART can operate in one of four modes. MR2[7:6] = 00 is the normal mode, with the transmitter and receiver operating independently. MR2[7:6] = 01 places the channel in the automatic echo mode, which automatically re-transmits the received data. The following conditions are true while in automatic echo mode:

1. Received data is re-clocked and retransmitted on the TxD output.
2. The receive clock is used for the transmitter.
3. The receiver must be enabled, but the transmitter need not be enabled.
4. The TxRDY and TxEMT status bits are inactive.
5. The received parity is checked, but is not regenerated for transmission, i.e., transmitted parity bit is as received.
6. Character framing is checked, but the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.
8. CPU-to-receiver communication continues normally, but the CPU-to-transmitter link is disabled.

Two diagnostic modes can also be selected. MR2[7:6] = 10 selects local loopback mode. In this mode:

1. The transmitter output is internally connected to the receiver input.
2. The transmit clock is used for the receiver.
3. The TxD output is held high.
4. The RxD input is ignored.
5. The transmitter must be enabled, but the receiver need not be enabled.
6. CPU to transmitter and receiver communications continue normally.

The second diagnostic mode is the remote loopback mode, selected by MR2[7:6] = 11. In this mode:

1. Received data is re-clocked and retransmitted on the TXD output.
2. The receive clock is used for the transmitter.
3. Received data is not sent to the local CPU, and the error status conditions are inactive.
4. The received parity is not checked and is not regenerated for transmission, i.e., the transmitted parity bit is as received.
5. The receiver must be enabled, but the transmitter need not be enabled.
6. Character framing is not checked, and the stop bits are retransmitted as received.
7. A received break is echoed as received until the next valid start bit is detected.

The user must exercise care when switching into and out of the various modes. The selected mode will be activated immediately upon mode selection, even if this occurs in the middle of a received or transmitted character. Likewise, if a mode is deselected, the device will switch out of the mode immediately. An exception to this is switching out of autoecho or remote loopback modes; if the deselection occurs just after the receiver has sampled the stop bit (indicated in autoecho by assertion of RxRDY), and the transmitter is enabled, the transmitter will remain in autoecho mode until the entire stop bit has been retransmitted.

Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

Table 2. Register Bit Formats

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-------	-------	-------	-------	-------	-------	-------	-------

MR1 (Mode Register 1)

RxRTS Control	RxINT Select	Error Mode*	Parity Mode	Parity Type	Bits per Character
0 = No	0 = RxRDY	0 = Char	00 = With parity	0 = Even	00 = 5
1 = Yes	1 = FFULL	1 = Block	01 = Force parity	1 = Odd	01 = 6
			10 = No parity		10 = 7
			11 = Special mode		11 = 8

NOTE: \*In block error mode, block error conditions must be cleared by using the error reset command (command 4x) or a receiver reset.

MR2 (Mode Register 2)

Channel Mode	TxRTS Control	CTS Enable Tx	Stop Bit Length*
00 = Normal			0 = 0.563 4 = 0.813 8 = 1.563 C = 1.813
01 = Auto-echo	0 = No	0 = No	1 = 0.625 5 = 0.875 9 = 1.625 C = 1.875
10 = Local loop	1 = Yes	1 = Yes	2 = 0.688 6 = 0.938 A = 1.688 E = 1.938
11 = Remote loop			3 = 0.750 7 = 1.000 B = 1.750 F = 2.000

NOTE: \*Add 0.5 to values shown above for 0–7, if channel is programmed for 5 bits/char.

CR (Command Register)

Miscellaneous Commands	Disable Tx	Enable Tx	Disable Rx	Enable Rx
See text	0 = No	0 = No	0 = No	0 = No
	1 = Yes	1 = Yes	1 = Yes	1 = Yes

NOTE: Access to the upper four bits of the command register should be separated by three (3) edges of the X1 clock. A disabled transmitter cannot be loaded

SR (Status Register)

Rec'd Break*	Framing Error*	Parity Error*	Overrun Error	TxEML	TxRDY	FFULL	RxRDY
0 = No	0 = No	0 = No	0 = No	0 = No	0 = No	0 = No	0 = No
1 = Yes	1 = Yes	1 = Yes	1 = Yes	1 = Yes	1 = Yes	1 = Yes	1 = Yes

NOTE: \*These status bits are appended to the corresponding data character in the receive FIFO. A read of the status register provides these bits [7:5] from the top of the FIFO together with bits [4:0]. These bits are cleared by a reset error status command. In character mode, they must be reset when the corresponding data character is read from the FIFO. In block error mode, block error conditions must be cleared by using the error reset command (command 4x) or a receiver reset.



## Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

**Table 2. Register Bit Formats (Continued)**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-------	-------	-------	-------	-------	-------	-------	-------

**CSR (Clock Select Register)**

Receiver Clock Select	Transmitter Clock Select
See text	See text

\* See Table 5 for BRG Test frequencies in this data sheet, and "Extended baud rates for SCN2681, SCN68681, SCC2691, SCC2692, SCC68681 and SCC2698B" Philips Semiconductors ICs for Data Communications, IC-19, 1994.

**OPCR (Output Port Configuration Register)** This register controls the MPP I/O pins and the MPO multi-purpose output pins.

MPP Function Select	MPOb Pin Function Select	Power-Down Mode*	MPOa Pin Function Select
0 = input	000 = RTSN	0 = Off	000 = RTSN
1 = output	001 = C/TO	1 = On	001 = C/TO
	010 = TxC (1X)		010 = TxC (1X)
	011 = TxC (16X)		011 = TxC (16X)
	100 = RxC (1X)		100 = RxC (1X)
	101 = RxC (16X)		101 = RxC (16X)
	110 = TxRDY		110 = TxRDY
	111 = RxRDY/FF		111 = RxRDY/FF

**NOTE:** \*Only OPCR[3] in block A controls the power-down mode.

**ACR (Auxiliary Control Register)**

BRG Select	Counter/Timer Mode and Source	Delta MPI1bINT	Delta MPI0bINT	Delta MPI1aINT	Delta MPI0aINT
0 = set 1 1 = set 2	See Text	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

**IPCR (Input Port Change Register)**

Delta MPI1b	Delta MPI0b	Delta MPI1a	Delta MPI0a	MPI1b	MPI0b	MPI1a	MPI0a
0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High

**ISR (Interrupt Status Register)**

MPI Port Change	Delta BREAKb	RxRDY/FFULLb	TxRDYb	Counter Ready	Delta BREAKa	RxRDY/FFULLa	TxRDYa
0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes	0 = No 1 = Yes

**IMR (Interrupt Mask Register)**

MPI Port Change INT	Delta BREAKb INT	RxRDY/FFULLb INT	TxRDYb INT	Counter Ready INT	Delta BREAKa INT	RxRDY/FFULLa INT	TxRDYa INT
0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

**CTPU (Counter/Timer Upper Register)**

C/T[15]	C/T[14]	C/T[13]	C/T[12]	C/T[11]	C/T[10]	C/T[9]	C/T[8]

**CTPU (Counter/Timer Lower Register)**

C/T[7]	C/T[6]	C/T[5]	C/T[4]	C/T[3]	C/T[2]	C/T[1]	C/T[0]

**IPR (Input Port Register) MPP and MPI Pins**

MPP2b	MPP1b	MPP2a	MPP1a	MPI1b	MPI0b	MPI1a	MPI0a
0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High	0 = Low 1 = High

**NOTE:** When TxEMT and TxRDY bits are at one just before a write to the Transmit Holding register, a command to disable the transmitter should be delayed until the TxRDY is at one again. TxRDY will set to one at the end of the start bit time.

# Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

### MR2[5] – Transmitter Request-to-Send Control

CAUTION: When the transmitter controls the OP pin (usually used for the RTSN signal) the meaning of the pin is not RTSN at all! Rather, it signals that the transmitter has finished the transmission (i.e., end of block).

This bit allows deactivation of the RTSN output by the transmitter. This output is manually asserted and negated by the appropriate commands issued via the command register. MR2[5] set to 1 caused the RTSN to be reset automatically one bit time after the character(s) in the transmit shift register and in the THR (if any) are completely transmitted (including the programmed number of stop bits) if a previously issued transmitter disable is pending. This feature can be used to automatically terminate the transmission as follows:

1. Program the auto-reset mode: MR2[5]=1
2. Enable transmitter, if not already enabled
3. Assert RTSN via command
4. Send message
5. Disable the transmitter after the last byte of the message is loaded to the TxFIFO. At the time the disable command is issued, be sure that the transmitter ready bit is on and the transmitter empty bit is off. If the transmitter empty bit is on (indicating the transmitter is underrun) when the disable is issued, the last byte will not be sent.
6. The last character will be transmitted and the RTSN will be reset one bit time after the last stop bit is sent.

NOTE: The transmitter is in an underrun condition when both the TxRDY and the TxEMT bits are set. This condition also exists immediately after the transmitter is enabled from the disabled or reset state. When using the above procedure with the transmitter in the underrun condition, the issuing of the transmitter disable must be delayed from the loading of a single, or last, character until the TxRDY becomes active again after the character is loaded.

### MR2[4] – Clear-to-Send Control

The state of this bit determines if the CTSN input (MPI) controls the operation of the transmitter. If this bit is 0, CTSN has no effect on the transmitter. If this bit is a 1, the transmitter checks the state of CTSN each time it is ready to send a character. If it is asserted (Low), the character is transmitted. If it is negated (High), the TxD output remains in the marking state and the transmission is delayed until CTSN goes Low. Changes in CTSN, while a character is being transmitted do not affect the transmission of that character. This feature can be used to prevent overrun of a remote receiver.

### MR2[3:0] – Stop Bit Length Select

This field programs the length of the stop bit appended to the transmitted character. Stop bit lengths of 9/16 to 1 and 1–9/16 to 2 bits, in increments of 1/16 bit, can be programmed for character lengths of 6, 7, and 8 bits. For a character length of 5 bits, 1–1/16 to 2 stop bits can be programmed in increments of 1/16 bit. In all cases, the receiver only checks for a mark condition at the center of the first stop bit position (one bit time after the last data bit, or after the parity bit if parity is enabled). If an external 1X clock is used for the transmitter, MR2[3] = 0 selects one stop bit and MR2[3] = 1 selects two stop bits to be transmitted.

### CSR – Clock Select Register

Table 3. Baud Rate

CSR[7:4]	ACR[7] = 0	ACR[7] = 1
0 0 0 0	50	75
0 0 0 1	110	110
0 0 1 0	134.5	38.4k
0 0 1 1	200	150
0 1 0 0	300	300
0 1 0 1	600	600
0 1 1 0	1,200	1,200
0 1 1 1	1,050	2,000
1 0 0 0	2,400	2,400
1 0 0 1	4,800	4,800
1 0 1 0	7,200	1,800
1 0 1 1	9,600	9,600
1 1 0 0	38.4k	19.2k
1 1 0 1	Timer	Timer
1 1 1 0	MP2 – 16X	MP2 – 16X
1 1 1 1	MP2 – 1X	MP2 – 1X

The receiver clock is always a 16X clock, except for CSR[7:4] = 1111. When MPP2 is selected as the input, MPP2a is for channel a and MPP2b is for channel b. See Table 5.

### CSR[7:4] – Receiver Clock Select

When using a 3.6864MHz crystal or external clock input, this field selects the baud rate clock for the receiver as shown in Table 3.

### CSR[3:0] – Transmitter Clock Select

This field selects the baud rate clock for the transmitter. The field definition is as shown in Table 3, except as follows:

CSR[3:0]	ACR[7] = 0	ACR[7] = 1
1 1 1 0	MPP1 – 16X	MPP1 – 16X
1 1 1 1	MPP1 – 1X	MPP1 – 1X

When MPP1 is selected as the input, MPP1a is for channel a and MPP1b is for channel b.

### CR – Command Register

CR is used to write commands to the Octal UART.

### CR[7:4] – Miscellaneous Commands

The encoded value of this field can be used to specify a single command as follows:

NOTE: Access to the upper four bits of the command register should be separated by three (3) edges of the X1 clock.

- 0000 No command.
- 0001 Reset MR pointer. Causes the MR pointer to point to MR1.
- 0010 Reset receiver. Resets the receiver as if a hardware reset had been applied. The receiver is disabled and the FIFO pointer is reset to the first location.
- 0011 Reset transmitter. Resets the transmitter as if a hardware reset had been applied.
- 0100 Reset error status. Clears the received break, parity error, framing error, and overrun error bits in the status register (SR[7:4]). Used in character mode to clear OE status (although RB, PE, and FE bits will also be cleared), and in block mode to clear all error status after a block of data has been received.



## Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

- 0101 Reset break change interrupt. Causes the break detect change bit in the interrupt status register (ISR[2 or 6]) to be cleared to zero.
- 0110 Start break. Forces the TxD output low (spacing). If the transmitter is empty, the start of the break condition will be delayed up to two bit times. If the transmitter is active, the break begins when transmission of the character is completed. If a character is in the THR, the start of break is delayed until that character or any others loaded after it have been transmitted (TxEMT must be true before break begins). The transmitter must be enabled to start a break
- 0111 Stop break. The TxD line will go high (marking) within two bit times. TxD will remain high for one bit time before the next character, if any, is transmitted.
- 1000 Assert RTSN. Causes the RTSN output to be asserted (Low).
- 1001 Negate RTSN. Causes the RTSN output to be negated (High).
- 1010 Set Timeout Mode On. The register in this channel will restart the C/T as each receive character is transferred from the shift register to the RHR. The C/T is placed in the counter mode, the START/STOP counter commands are disabled, the counter is stopped, and the Counter Ready Bit, ISR[3], is reset.
- 1011 Reserved.
- 1100 Disable Timeout Mode. This command returns control of the C/T to the regular START/STOP counter commands. It does not stop the counter, or clear any pending interrupts. After disabling the timeout mode, a 'Stop Counter' command should be issued.
- 1101 Reserved.
- 111x Reserved for testing.

### CR[3] – Disable Transmitter

This command terminates transmitter operation and resets the TxRDY and TxEMT status bits. However, if a character is being transmitted or if a character is in the THR when the transmitter is disabled, the transmission of the character(s) is completed before assuming the inactive state.

### CR[2] – Enable Transmitter

Enables operation of the transmitter. The TxRDY status bit will be asserted.

### CR[1] – Disable Receiver

This command terminates operation of the receiver immediately – a character being received will be lost. The command has no effect on the receiver status bits or any other control registers. If the special wake-up mode is programmed, the receiver operates even if it is disabled (see Wake-up Mode).

### CR[0] – Enable Receiver

Enables operation of the receiver. If not in the special wake-up mode, this also forces the receiver into the search for start bit state.

## SR – Channel Status Register

### SR[7] – Received Break

This bit indicates that an all zero character of the programmed length has been received without a stop bit. Only a single FIFO position is occupied when a break is received; further entries to the FIFO are inhibited until the RxDA line returns to the marking state for at least one-half bit time two successive edges of the internal or

external 1x clock. **This will usually require a high time of one X1 clock period or 3 X1 edges since the clock of the controller is not synchronous to the X1 clock.**

When this bit is set, the change in break bit in the ISR (ISR[6 or 2]) is set. ISR[6 or 2] is also set when the end of the break condition, as defined above, is detected. The break detect circuitry is capable of detecting breaks that originate in the middle of a received character. However, if a break begins in the middle of a character, it must last until the end of the next character in order for it to be detected.

### SR[6] – Framing Error (FE)

This bit, when set, indicates that a stop bit was not detected when the corresponding data character in the FIFO was received. The stop bit check is made in the middle of the first stop bit position.

### SR[5]– Parity Error (PE)

This bit is set when the 'with parity' or 'force parity' mode is programmed and the corresponding character in the FIFO was received with incorrect parity. In special 'wake-up mode', the parity error bit stores the received A/D bit.

### SR[4] – Overrun Error (OE)

This bit, when set, indicates that one or more characters in the received data stream have been lost. It is set upon receipt of a new character when the FIFO is full and a character is already in the receive shift register waiting for an empty FIFO position. When this occurs, the character in the receive shift register (and its break detect, parity error and framing error status, if any) is lost. This bit is cleared by a reset error status command.

### SR[3] – Transmitter Empty (TxEMT)

This bit will be set when the transmitter underruns, i.e., both the transmit holding register and the transmit shift register are empty. It is set after transmission of the last stop bit of a character. If no character is in the THR awaiting transmission. It is reset when the THR is loaded by the CPU, or when the transmitter is disabled.

### SR[2] – Transmitter Ready (TxRDY)

This bit, when set, indicates that the THR is empty and ready to be loaded with a character. This bit is cleared when the THR is loaded by the CPU and is set when the character is transferred to the transmit shift register. TxRDY is reset when the transmitter is disabled and is set when the transmitter is first enabled, e.g., characters loaded in the THR while the transmitter is disabled will not be transmitted.

### SR[1] – FIFO Full (FFULL)

This bit is set when a character is transferred from the receive shift register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when the CPU reads the FIFO and there is no character in the receive shift register. If a character is waiting in the receive shift register because the FIFO is full, FFULL is not reset after reading the FIFO once.

### SR[0] – Receiver Ready (RxRDY)

This bit indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the RHR, and no more characters are in the FIFO.



# Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

## OPCR – Output Port Configuration Register

### OPCR[7] – MPP Function Select

When this bit is a zero, the MPP pins function as inputs, to be used as general purpose inputs or as receiver or transmitter external clock inputs. When this bit is set, the MPP pins function as outputs. MPP1 will be a TxRDY indicator, and MPP2 will be an RxRDY/FFULL indicator.

### OPCR[6:4] – MPOb Output Select

This field programs the MPOb output pin to provide one of the following:

- 000 Request-to-send active-Low output (RTSN). This output is asserted and negated via the command register. Mode RTSN can be programmed to be automatically reset after the character in the transmitter is completely shifted out or when the receiver FIFO and receiver shift register are full using MR2[5] and MR1[7], respectively.
- 001 The counter/timer output. In the timer mode, this output is a square wave with a period of twice the value (in clock periods) of the contents of the CTPU and CTPL. In the counter mode, the output remains high until the terminal count is reached, at which time it goes low. The output returns to the High state when the counter is stopped by a stop counter command.
- 010 The 1X clock for the transmitter, which is the clock that shifts the transmitted data. If data is not being transmitted, a non-synchronized 1X clock is output.
- 011 The 16X clock for the transmitter. This is the clock selected by CSR[3:0], and is a 1X clock if CSR[3:0] = 1111.
- 100 The 1X clock for the receiver, which is the clock that samples the received data. If data is not being received, a non-synchronized 1X clock is output.
- 101 The 16X clock for the receiver. This is the clock selected by CSR[7:4], and is a 1X clock if CSR[7:4] = 1111.
- 110 The transmitter register ready signal, which is the same as SR[2].
- 111 The receiver ready or FIFO full signal.

### OPCR[3] – Power Down Mode Select

This bit, when set, selects the power-down mode. In this mode, the 2698B oscillator is stopped and all functions requiring this clock are suspended. The contents of all registers are saved. It is recommended that the transmitter and receiver be disabled prior to placing the 2698B in this mode. This bit is reset with RESET asserted. Note that this bit must be set to a logic 1 after power up. Only OPCR[3] in block A controls the power-down mode.

### OPCR[2:0] – MPOa Output Select

This field programs the MPOa output pin to provide one of the same functions as described in OPCR[6:4].

## ACR – Auxiliary Control Register

### ACR[7] – Baud Rate Generator Set Select

This bit selects one of two sets of baud rates generated by the BRG.

- Set 1: 50, 110, 134.5, 200, 300, 600, 1.05k, 1.2k, 2.4k, 4.8k, 7.2k, 9.6k, and 38.4k baud.
- Set 2: 75, 110, 150, 300, 600, 1.2k, 1.8k, 2.0k, 2.4k, 4.8k, 9.6k, 19.2k, and 38.4k baud.

The selected set of rates is available for use by the receiver and transmitter.

### ACR[6:4] – Counter/Timer Mode and Clock Source Select

This field selects the operating mode of the counter/timer and its clock source (see Table 4).

The MPI1 pin available as the Counter/Timer clock source is MPI1 a,c,e, and g only.

**Table 4. ACR[6:4] Operating Mode**

[6:4]	Mode	Clock Source
0 0 0	Counter	MPI1a pin
0 0 1	Counter	MPI1a pin divided by 16
0 1 0	Counter	TxC-1XA clock of the transmitter
0 1 1	Counter	Crystal or MPI pin (X1/CLK) divided by 16
1 0 0	Timer	MPI1a pin
1 0 1	Timer	MPI1a pin divided by 16
1 1 0	Timer	Crystal or external clock (X1/CLK)
1 1 1	Timer	Crystal or MPI pin (X1/CLK) divided by 16

NOTE: The timer mode generates a squarewave.

### ACR[3:0] – MPI1b, MPI0b, MPI1a, MPI0a Change-of-State Interrupt Enable

This field selects which bits of the input port change register (IPCR) cause the input change bit in the interrupt status register, ISR[7], to be set. If a bit is in the 'on' state, the setting of the corresponding bit in the IPCR will also result in the setting of ISR[7], which results in the generation of an interrupt output if IMR[7] = 1. If a bit is in the 'off' state, the setting of that bit in the IPCR has no effect on ISR[7].

## IPCR – Input Port Change Register

### IPCR[7:4] – MPI1b, MPI0b, MPI1a, MPI0a Change-of-State

These bits are set when a change of state, as defined in the Input Port section of this data sheet, occurs at the respective pins. They are cleared when the IPCR is read by the CPU. A read of the IPCR also clears ISR[7], the input change bit in the interrupt status register. The setting of these bits can be programmed to generate an interrupt to the CPU.

### IPCR[3:0] – MPI1b, MPI0b, MPI1a, MPI0a Change-of-State

These bits provide the current state of the respective inputs. The information is unlatched and reflects the state of the inputs pins during the time the IPCR is read.

## ISR – Interrupt Status Register

This register provides the status of all potential interrupt sources. The contents of this register are masked by the interrupt mask register (IMR). If a bit in the ISR is a '1' and the corresponding bit in the IMR is also a '1', the INTRN output is asserted (Low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask the reading of the ISR; the true status is provided regardless of the contents of the IMR.



## Enhanced octal universal asynchronous receiver/transmitter (Octal UART)

SCC2698B

### ISR[7] – MPI Change-of-State

This bit is set when a change-of-state occurs at the MPI1b, MPI0b, MPI1a, MPI0a input pins. It is reset when the CPU reads the IPCR.

### ISR[6] – Channel b Change in Break

This bit, when set, indicates that the receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a reset break change interrupt command.

### ISR[5] – Receiver Ready or FIFO Full Channel b

The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the receiver FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read.

If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when FIFO is read and there is no character in the receiver shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

### ISR[4] – Transmitter Ready Channel b

This bit is a duplicate of TxRDY (SR[2]).

### ISR[3] – Counter Ready

In the counter mode of operation, this bit is set when the counter reaches terminal count and is reset when the counter is stopped by a stop counter command. It is initialized to '0' when the chip is reset.

In the timer mode, this bit is set once each cycle of the generated square wave (every other time the C/T reaches zero count). The bit is reset by a stop counter command. The command, however, does not stop the C/T.

### ISR[2] – Channel a Change in Break

This bit, when set, indicates that the receiver has detected the beginning or the end of a received break. It is reset when the CPU issues a reset break change interrupt command.

### ISR[1] – Receiver Ready or FIFO Full Channel a

The function of this bit is programmed by MR1[6]. If programmed as receiver ready, it indicates that a character has been received and is waiting in the FIFO to be read by the CPU. It is set when the character is transferred from the receive shift register to the FIFO and reset when the CPU reads the receiver FIFO. If the FIFO contains more characters, the bit will be set again after the FIFO is read. If programmed as FIFO full, it is set when a character is transferred from the receive holding register to the receive FIFO and the transfer causes the FIFO to become full, i.e., all three FIFO positions are occupied. It is reset when FIFO is read and there is no character in the receiver shift register. If there is a character waiting in the receive shift register because the FIFO is full, the bit is set again when the waiting character is transferred into the FIFO.

### ISR[0] – Transmitter Ready Channel a

This bit is a duplicate of TxRDY (SR[2]).

### IMR – Interrupt Mask Register

The programming of this register selects which bits in the ISR cause an interrupt output. If a bit in the ISR is a '1' and the corresponding

bit in the IMR is a '1', the INTRN output is asserted (Low). If the corresponding bit in the IMR is a zero, the state of the bit in the ISR has no effect on the INTRN output. Note that the IMR does not mask reading of the ISR.

### CTPU and CTPL – Counter/Timer Registers

The CTPU and CTPL hold the eight MSBs and eight LSBs, respectively, of the value to be used by the counter/timer in either the counter or timer modes of operation. The minimum value which may be loaded into the CTPU/CTPL registers is H'0002'. Note that these registers are write-only and cannot be read by the CPU.

In the timer (programmable divider) mode, the C/T generates a square wave with a period of twice the value (in clock periods) of the CTPU and CTPL. The waveform so generated is often used for a data clock. The formula for calculating the divisor  $n$  to load to the CTPU and CTPL for a particular 1X data clock is shown below:

$$n = \frac{C/T \text{ Clock Frequency}}{2 \times 16 \text{ Baud rate desired}}$$

Often this division will result in a non-integer number; 26.3, for example. One can only program integer numbers in a digital divider. Therefore, 26 would be chosen. This gives a baud rate error of 0.3/26.3 which is 1.14%; well within the ability asynchronous mode of operation.

If the value in CTPU or CTPL is changed, the current half-period will not be affected, but subsequent half-periods will be. The C/T will not be running until it receives an initial 'Start Counter' command (read at address A3–A0 = 1110). After this, while in timer mode, the C/T will run continuously. Receipt of a subsequent start counter command causes the C/T to terminate the current timing cycle and to begin a new cycle using the values in the CTPU and CTPL.

The counter ready status bit (ISR[3]) is set once each cycle of the square wave. The bit is reset by a stop counter command read with A3–A0 = H'F'). The command, however, does not stop the C/T. The generated square wave is output on MPO if it is programmed to be the C/T output.

In the counter mode, the C/T counts down the number of pulses loaded in CTPU and CTPL by the CPU. Counting begins upon receipt of a start counter command. Upon reaching the terminal count H'0000', the counter ready interrupt bit (ISR[3]) is set. The counter continues counting past the terminal count until stopped by the CPU. If MPO is programmed to be the output of the C/T, the output remains High until the terminal count is reached, at which time it goes Low. The output returns to the High state and ISR[3] is cleared when the counter is stopped by a stop counter command. The CPU may change the values of CTPU and CTPL at any time, but the new count becomes effective only on the next start counter command. If new values have not been loaded, the previous values are preserved and used for the next count cycle.

In the counter mode, the current value of the upper and lower eight bits of the counter (CTU, CTL) may be read by the CPU. It is recommended that the counter be stopped when reading to prevent potential problems which may occur if a carry from the lower eight bits to the upper eight bits occurs between the times that both halves of the counter is read. However, note that a subsequent start counter command will cause the counter to begin a new count cycle using the values in CTPU and CTPL.

