

GPC[®] 184

General Purpose Controller Z180

MANUALE TECNICO



grifo[®]

ITALIAN TECHNOLOGY

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

E-mail: grifo@grifo.it

<http://www.grifo.it>

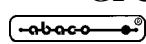
<http://www.grifo.com>

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



GPC[®] 184

Edizione 3.10 Rel. 24 Settembre 1999

, GPC[®], grifo[®], sono marchi registrati della ditta grifo[®]

GPC[®] 184

General Purpose Controller Z180

MANUALE TECNICO

Modulo Intelligente **Abaco[®] BLOCK**, della **Serie 4**, nel formato 100x50 mm; contenitore, opzionale, per guide ad Ω tipo **DIN 46277-1** e **DIN 46277-3**. **CPU Z180**, con quarzo da **18,432 MHz**; **Jumper** di configurazione usato per selezionare il modo **RUN/DEBUG**; fino a **512K** di **EPROM** o **FLASH** e fino a **512K** di **RAM**; **Real Time Clock** con possibilità di generare **INT**; circuiteria di **Back Up** per RAM ed RTC, tramite batteria al **LITIO** interna ed esterna; interfaccia Clocked Serial I/O, a disposizione utente, sul connettore di I/O; **2** canali di Programmable Reload **Timer** interni da 16 bits. **2** linee seriali in **RS232**, di cui una settabile in **RS422**, **RS485** o **Current Loop**; doppio **Baud Rate** generator, settabile da software; circuiteria di **Watch Dog**, disinseribile da hardware; connettore di espansione per **Abaco[®] I/O BUS** da 26 vie. **4** diverse modalita' di riduzione consumi: **Halt**, **STOP**, **Sleep** e **System Stop**; circuiteria di **Power Failure** in grado di generare interrupt; consumo estremamente ridotto: assorbe infatti solo **60 mA** a **5Vdc**; protezione della logica di bordo dai transienti tramite **TransZorbTM**. Vasta disponibilità di software di sviluppo quali **Remote Symbolic Debugger**; **Macro Assembler**; **GET 80**; **compilatori C (HI TECH C 80, DDS MICRO C 85)**; **compilatori PASCAL (PASCAL 80, EMBEDDED PASCAL)**; **FGDOS184**; ecc.

grifo[®]

ITALIAN TECHNOLOGY

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

E-mail: grifo@grifo.it

<http://www.grifo.it>

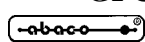
<http://www.grifo.com>

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



GPC[®] 184

Edizione 3.10 Rel. 24 Settembre 1999

, GPC[®], grifo[®], sono marchi registrati della ditta grifo[®]

Vincoli sulla documentazione **grifo**[®] 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**[®].

IMPORTANTE

Tutte le informazioni contenute sul presente manuale sono state accuratamente verificate, ciononostante **grifo**[®] 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[®] 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**[®].

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:

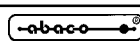


Attenzione: Pericolo generico



Attenzione: Pericolo di alta tensione

Marchi Registrati

 , GPC[®], **grifo**[®] : sono marchi registrati della **grifo**[®].

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

INDICE GENERALE

INTRODUZIONE	1
VERSIONE SCHEDA	1
CARATTERISTICHE GENERALI	2
PROCESSORE DI BORDO	3
MEMORIE	3
COMUNICAZIONE SERIALE	4
CLOCK	4
ALIMENTAZIONE DI BORDO	4
CONFIGURAZIONE SCHEDA	4
ABACO® I/O BUS	6
REAL TIME CLOCK	6
WATCH DOG	6
CONTATTO DI RESET	6
LOGICA DI CONTROLLO	6
SPECIFICHE TECNICHE	7
CARATTERISTICHE GENERALI	7
CARATTERISTICHE FISICHE	7
CARATTERISTICHE ELETTRICHE	8
INSTALLAZIONE	9
CONNESSIONI CON IL MONDO ESTERNO	9
CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP	9
CN1 - CONNETTORE PER ABACO® I/O BUS	10
CN3A - CONNETTORE PER LINEA SERIALE A	12
CN3B - CONNETTORE PER LINEA SERIALE B	14
CN5 - CONNETTORE PER SEGNALI AUSILIARI	20
INTERFACCIAMENTO DEGLI I/O CON IL CAMPO	21
SEGNALAZIONI VISIVE	21
JUMPERS	22
JUMPERS A 2 VIE	24
JUMPERS A 3 VIE	25
JUMPERS A 5 VIE	25
BACK UP	26
SELEZIONE MEMORIE	26
INTERRUPTS	27
INGRESSO DI CONFIGURAZIONE	27
JUMPER A STAGNO	27
COMUNICAZIONE SERIALE	28
RESET E WATCH DOG	30
POWER FAILURE	31
DESCRIZIONE SOFTWARE	32

MAPPAGGI ED INDIRIZZAMENTI	35
INTRODUZIONE	35
MAPPAGGIO DELLE RISORSE DI BORDO	35
MAPPAGGIO ABACO® I/O BUS	35
MAPPAGGIO MEMORIE	36
MAPPAGGIO I/O	37
DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO	39
REAL TIME CLOCK	39
WATCH DOG	41
JUMPER DI CONFIGURAZIONE	41
PERIFERICHE DELLA CPU	41
SCHEDE ESTERNE	42
BIBLIOGRAFIA	44
APPENDICE A: MONTAGGIO MECCANICO DELLA SCHEDA	A-1
APPENDICE B: DESCRIZIONE COMPONENTI DI BORDO	B-1
APPENDICE C: SCHEMI ELETTRICI	C-1
APPENDICE D: INDICE ANALITICO	D-1

INDICE DELLE FIGURE

FIGURA 1: SCHEMA A BLOCCHI	5
FIGURA 2: CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP	9
FIGURA 3: CN1 - CONNETTORE PER ABACO® I/O BUS	10
FIGURA 4: FOTO DELLA SCHEDA	11
FIGURA 5: PIANTA COMPONENTI LATO COMPONENTI E STAGNATURA	11
FIGURA 6: CN3A - CONNETTORE PER LINEA SERIALE A	12
FIGURA 7: SCHEMA DI COMUNICAZIONE SERIALE	13
FIGURA 8: CN3B - CONNETTORE PER LINEA SERIALE B	14
FIGURA 9: DISPOSIZIONE LED, CONNETTORI, MEMORIE,ECC.....	15
FIGURA 10: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 232	16
FIGURA 11: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 422	16
FIGURA 12: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 485	16
FIGURA 13: ESEMPIO COLLEGAMENTO IN RETE IN RS 485	17
FIGURA 14: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 4 FILI	18
FIGURA 15: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 2 FILI	18
FIGURA 16: SCHEMA DI COLLEGAMENTO LINEE DI CN5	19
FIGURA 17: CN5 - CONNETTORE PER SEGNALI AUSILIARI.....	20
FIGURA 18: TABELLA DELLE SEGNALAZIONI VISIVE	21
FIGURA 19: TABELLA RIASSUNTIVA JUMPERS	22
FIGURA 20: DISPOSIZIONE JUMPERS LATO COMPONENTI.....	23
FIGURA 21: DISPOSIZIONE JUMPERS LATO STAGNATURE	23
FIGURA 22: TABELLA JUMPERS A 2 VIE	24
FIGURA 23: TABELLA JUMPERS A 3 VIE	25
FIGURA 24: TABELLA JUMPERS A 5 VIE	25
FIGURA 25: TABELLA DI SELEZIONE MEMORIE	26
FIGURA 26: DISPOSIZIONE DRIVER PER COMUNICAZIONE SERIALE	29
FIGURA 27: MAPPAGGIO DELLE MEMORIE	37
FIGURA 28: TABELLA INDIRIZZAMENTO I/O	38
FIGURA 29: SCHEMA DELLE POSSIBILI CONNESSIONI	43
FIGURA A1: QUOTE PER MONTAGGIO IN PIGGY BACK	A-1
FIGURA A2: MONTAGGIO IN PIGGY BACK	A-2
FIGURA A3: MONTAGGIO SU GUIDA WEIDMULLER	A-2
FIGURA C1: SCHEMA ELETTRICO DI ESPANSIONE PPI.....	C-1
FIGURA C2: SCHEMA ELETTRICO SPA 03	C-2
FIGURA C3: SCHEMA ELETTRICO QTP 16P	C-3
FIGURA C4: SCHEMA ELETTRICO QTP 24P 1/2.....	C-4
FIGURA C5: SCHEMA ELETTRICO QTP 24P 2/2.....	C-5
FIGURA C6: SCHEMA ELETTRICO DI I/O SU ABACO® I/O BUS	C-6
FIGURA C7: SCHEMA ELETTRICO INTERFACCIA BUS	C-7
FIGURA C8: SCHEMA ELETTRICO IAC 01.....	C-8



INTRODUZIONE

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

Scopo di questo manuale é 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.

VERSIONE SCHEDA

Il presente manuale è riferito alla scheda **GPC® 184** versione **100997** e successive. La validità delle informazioni riportate é quindi subordinata al numero di versione della scheda in uso e l'utente deve quindi sempre verificare la giusta corrispondenza tra le due indicazioni. Sulla scheda il numero di versione è riportato in più punti sia a livello di serigrafia che di stampato (ad esempio tra il microprocessore e le memorie sul lato componenti).

CARATTERISTICHE GENERALI

La scheda **GPC® 184**, che fa parte della **serie 4** delle CPU con ingombro di 100x50 mm, e' un potente modulo di controllo, della fascia a **basso prezzo**, in grado di funzionare autonomamente come periferica intelligente e/o remotata in una più vasta rete di telecontrollo e/o di acquisizione.

La **GPC® 184** puo' essere fornita di un supporto in plastica provvisto degli attacchi per le guide ad **Omega** tipo **DIN 46277-1** e **DIN 46277-3**. In questo modo non e' necessario l'uso di un **Rack**, ma la scheda puo' essere montata, in modo piu' economico, direttamente nel quadro elettrico. Viste le ridotte dimensioni della scheda **GPC® 184**, questa puo' essere montata nella stessa guida in plastica che contiene le periferiche di I/O, come ad esempio le **ZBx xxx**, formando in questo modo un unico elemento **BLOCK**. Un'altra tipica applicazione della scheda **GPC® 184**, e' quella di essere adoperata come un modulo di CPU da montare in **Piggy Back** sulle schede periferiche realizzate direttamente dall'utente.

La programmazione e l'uso delle risorse della scheda diventa estremamente semplice grazie all'uso del potente sistema operativo romano **FGDOS**. Esso supporta i linguaggi ad alto livello quali **compilatori C, PASCAL**, ecc. e mette a disposizione le risorse di memoria come se fossero **RAM/ROM disk**, consentendo un immediato utilizzo ad alto livello di questi dispositivi. Aggiungendo un **82C55**, esterno alla scheda, l'**FGDOS** consente la gestione della **MCI 64** per governare le **RAM cards PCMCIA**, di **display LCD** o **Fluorescenti**, di una tastiera a matrice e di una stampante parallela. Per un uso immediato di queste funzionalita', sono disponibili delle schede della serie **KDx xxx** oppure, per chi ha bisogno di un oggetto finito, esistono i pannelli operatore tipo **QTP 24P** e **QTP 16P**. Questi pannelli operatore hanno la stessa estetica della **QTP 24** e **QTP 16** ma, non disponendo di intelligenza locale, vengono comandati direttamente dalla **GPC® 184**, riducendo notevolmente i costi. **FGDOS**, oltre alla nota facilità di debug, consente di programmare, direttamente a bordo scheda, una **FLASH** con il programma utente.

Per una rapida prototipizzazione si può ricorrere alle ottime schede **SPA 03** ed **SPA 04** su cui e' possibile montare, anche in **Piggy Back**, la **GPC® 184**. La presenza del connettore **Abaco® I/O BUS** consente inoltre di poter pilotare direttamente le schede di I/O tipo **ZBR xxx**, **ZBT xxx** e tramite **ABB 03**, **ABB 05**, ecc. e' possibile gestire anche le numerose schede periferiche, disponibili sul **BUS Abaco®**.

- Modulo Intelligente **Abaco® BLOCK**, della **Serie 4**, nel formato 100x50 mm.
- Contenitore, opzionale, per guide ad Ω tipo **DIN 46277-1** e **DIN 46277-3**.
- **CPU Z180**, con quarzo da **18,432 MHz**.
- **Jumper** di configurazione usato per selezionare il modo **RUN/DEBUG**.
- Fino a **512K** di **EPROM** o **FLASH** e fino a **512K** di **RAM**. Tramite **FGDOS** la memoria eccedente i 64K é gestita come **RAMROM disk**. E' possibile cancellare e riprogrammare autonomamente la **FLASH** di bordo per inserire il proprio programma utente.
- **Real Time Clock** con possibilità di generare **INT**.
- Circuiteria di **Back Up** per **RAM** ed **RTC**, tramite batteria al **LITIO** interna ed esterna.
- Interfaccia Clocked Serial I/O, a disposizione utente, sul connettore di I/O.
- **2** canali di Programmable Reload **Timer** interni da 16 bits.
- **2** linee seriali in **RS232**, di cui una settabile in **RS422**, **RS485** o **Current Loop**.
- Doppio **Baud Rate** generator, settabile da software.
- Circuiteria di **Watch Dog**, disinseribile da hardware.
- Connettore di espansione per **Abaco® I/O BUS** da 26 vie.
- **4** diverse modalita' di riduzione consumi: **Halt, STOP, Sleep** e **System Stop**.
- Circuiteria di **Power Failure** in grado di generare interrupt.
- Consumo estremamente ridotto: assorbe infatti solo **60 mA** a **5Vdc**.

- Protezione della logica di bordo dai transienti tramite **TransZorb™**.
- Vasta disponibilità di software di sviluppo quali **Remote Symbolic Debugger; Macro Assembler; GET 80; compilatori C (HITECH C 80, DDS MICRO C 85); compilatori PASCAL (PASCAL 80, EMBEDDED PASCAL); FGDOS184; ecc.**

Viene di seguito riportata una descrizione dei blocchi funzionali della scheda, con indicate le operazioni effettuate da ciascuno di essi. Per una più facile individuazione di tali blocchi e per una verifica delle loro connessioni, fare riferimento alla figura 1.

PROCESSORE DI BORDO

La scheda **GPC® 184** é predisposta per accettare il processore **Z180** prodotto dalla **ZILOG**. Tale processore ad 8 bit é codice compatibile con lo Z80 ed é quindi caratterizzato da un esteso set di istruzioni (170), da un'alta velocità di esecuzione e di manipolazione dati e da un'efficiente gestione vettorizzata degli interrupts. Di fondamentale importanza é la presenza delle seguenti periferiche interne al microprocessore:

- 2 Timer a 16 bit, con funzione di prescaler programmabile (PRT);
- 2 linee seriali asincrone complete di segnali di handshake (ASCI);
- 2 canali di DMA per trasferimenti dati ad alta velocità (DMAC);
- Gestore di memoria estesa (MMU);
- 1 linea seriale sincrona (CSI/O);
- Interrupt controller;
- Generatore di cicli di wait per l'accesso a dispositivi esterni;
- Possibilità di operare in idle e stop mode, per minimizzare i consumi.

Per maggiori informazioni sul componente si faccia riferimento all'apposita documentazione della casa costruttrice, oppure all'appendice B di questo manuale.

MEMORIE

E' possibile dotare la scheda di un massimo di 1024K di memoria variamente suddivisi con un massimo di 512KByte di EPROM o FLASH EPROM, 512KByte di RAM statica. La scelta della configurazione delle memorie presenti sulla scheda può avvenire in relazione all'applicazione da risolvere e quindi alle esigenze dell'utente. Da questo punto di vista si ricorda che la scheda viene normalmente fornita con 128KByte RAM di lavoro e che tutte le rimanenti configurazioni di memoria devono essere quindi opportunamente specificate in fase di ordine della scheda. Sfruttando la circuiteria di back up di bordo più l'eventuale batteria tampone esterna, si ha la possibilità di mantenere i dati anche in assenza di alimentazione. Questa caratteristica fornisce alla scheda la possibilità di ricordare in ogni condizione, una serie di parametri come ad esempio la configurazione o lo stato del sistema, anche per lunghi periodi di inattività. Il mappaggio delle risorse di memoria avviene tramite una opportuna circuiteria di bordo, che provvede ad allocare i dispositivi all'interno dello spazio d'indirizzamento del microprocessore. Per maggiori informazioni fare riferimento al capitolo "DESCRIZIONE HARDWARE" e "DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO". Per una descrizione più approfondita sui dispositivi di memoria, sugli zoccoli da utilizzare e sullo strappaggio della scheda, fare riferimento al paragrafo "SELEZIONE MEMORIE".

COMUNICAZIONE SERIALE

La comunicazione seriale é completamente settabile via software per quanto riguarda sia il protocollo sia la velocità (da un minimo di 75 ad un massimo di 57,6K Baud con frequenza di clock standard) ed in modo completamente autonomo per entrambe le linee di comunicazione. Tali settaggi avvengono tramite la programmazione dell'ASCII interno allo Z180 e della sezione di baud rate generator, di cui la scheda é provvista, quindi per ulteriori informazioni si faccia riferimento alla documentazione tecnica della casa costruttrice o all'appendice B di questo manuale. Dal punto di vista hardware é invece possibile selezionare, tramite una serie di comodi jumpers, il protocollo elettrico di comunicazione. In particolare una linea é sempre bufferata in RS 232, mentre la rimanente può essere bufferata in RS 232, Current Loop, RS 485 ed RS 422; in quest'ultimo caso é definibile anche l'attivazione e/o la direzionalità della linea di comunicazione.

CLOCK

Sulla **GPC® 184** é presente una circuiteria che provvede a generare rispettivamente la frequenza di clock per la CPU (18,432 MHz); da cui vengono ricavate anche le frequenze necessarie per le altre sezioni della scheda (Timer, Seriali, ecc.). In caso di applicazioni particolarmente veloci, la frequenza di clock può essere aumentata fino a l doppio di quella standard, intervenendo via hardware sull'apposita circuiteria (per maggiori informazioni contattare direttamente la **grifo®**). Si ricorda inoltre che la frequenza di clock della CPU risulta essere sempre dimezzata rispetto a quella del quarzo montato a bordo scheda.

ALIMENTAZIONE DI BORDO

L'unica tensione di alimentazione necessaria é di +5 Vdc che deve essere fornita tramite i pin 25 (GND) e 26 (+5 Vdc) di CN1. Sulla scheda sono state adottate tutte le scelte circuitali e componentistiche che tendono a ridurre la sensibilità ai disturbi ed a ridurre i consumi. La **GPC® 184** é il componente ideale in tutte le applicazioni in cui i consumi ridotti sono un requisito fondamentale infatti, con i soli **60 mA** in funzionamento normale, é in grado di essere tranquillamente alimentata da batterie, pannelli solari, piccoli alimentatori, ecc; per ridurre ulteriormente il consumo, si possono facilmente usare le numerose modalità a basso assorbimento del microprocessore. Una interessante circuiteria di power failure consente di riconoscere l'imminente caduta dell'alimentazione e quindi di intervenire opportunamente via software, tramite una procedura di risposta all'interrupt. Si ricorda inoltre che è presente una circuiteria di protezione tramite **TransZorb™** per evitare danneggiamenti causati da tensioni di alimentazione errate.

CONFIGURAZIONE SCHEDA

Allo scopo di rendere configurabile la scheda ed in particolare il programma applicativo sviluppato, é stato previsto il jumper J4. La possibilità di acquisire via software lo stato di questo jumper, fornisce all'utente la possibilità di gestire diverse condizioni tramite un unico programma, senza dover rinunciare ad altre linee d'ingresso (le applicazioni caratteristiche sono: selezione della lingua di rappresentazione, definizione parametri del programma, selezione della modalità operativa, ecc.). Alcuni pacchetti software sviluppati per la **GPC® 184** usano il jumper J4 per selezionare la modalità operativa di RUN o DEBUG, come descritto negli appositi manuali d'uso degli stessi pacchetti.

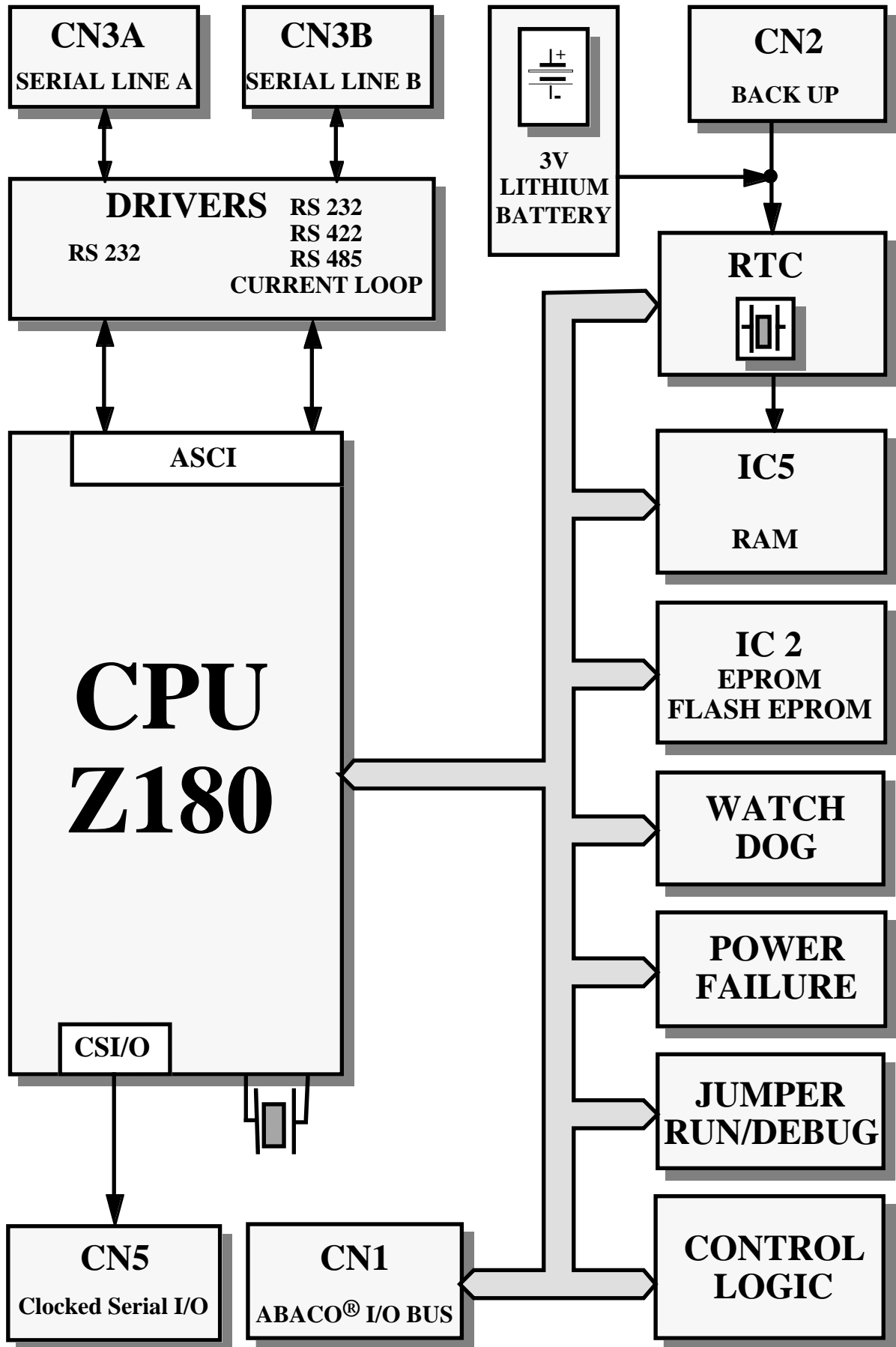


FIGURA 1: SCHEMA A BLOCCHI

ABACO® I/O BUS

Una delle caratteristiche di fondamentale importanza della **GPC® 184** è quella di disporre del cosiddetto **ABACO® I/O BUS**: ovvero un connettore normalizzato **ABACO®** con cui è possibile collegare la scheda ad una serie di moduli esterni intelligenti e non. Tra questi si trovano moduli per acquisizione di segnali analogici (A/D), per la generazione di segnali analogici (D/A), per gestione di linee di I/O logico, per counter, ecc. e ne possono essere realizzati anche su specifiche richieste dell'utente. Utilizzando mother board come l'**ABB 03** o l'**ABB 05** è inoltre possibile gestire tutte le schede periferiche in formato Europa con interfaccia per BUS **ABACO®**. Tale caratteristica rende la scheda espandibile con un ottimo rapporto prezzo/prestazioni e quindi adatta a risolvere molti dei problemi dell'automazione industriale.

REAL TIME CLOCK

La **GPC® 184** dispone di un completo Real Time Clock in grado di gestire ore, minuti, secondi, giorno del mese, mese, anno e giorno della settimana in modo completamente autonomo. L'alimentazione del componente è fornita dalla circuiteria di back up in modo da garantire la validità dei dati in ogni condizione operativa ed è completamente gestito via software, tramite la programmazione di 16 registri situati nello spazio di I/O della CPU da un'apposita logica di controllo. La sezione di RTC può inoltre generare interrupt in corrispondenza di intervalli di tempo programmabili via software, in modo da poter periodicamente distogliere la CPU dalle normali operazioni oppure periodicamente risvegliarla dagli stati di basso assorbimento.

WATCH DOG

La scheda **GPC® 184** è provvista di una circuiteria di watch dog che, se utilizzata, consente di uscire da stati di loop infinito o da condizioni anomale non previste dal programma applicativo. Tale circuiteria è composta da una sezione instabile con un tempo d'intervento medio di 1,5 sec. Tutta la gestione avviene via software (tramite l'accesso ad un opportuno registro situato nello spazio d'indirizzamento della CPU) e conferisce al sistema basato sulla scheda, una sicurezza estrema.

CONTATTO DI RESET

Sulla **GPC® 184** è presente un contatto di reset (P1) che una volta chiuso farà ripartire la scheda da una condizione di azzeramento generale. La funzione principale di questo tasto è quella di uscire da condizioni di loop infinito, soprattutto durante la fase di debug o di garantire uno stato certo di partenza. Per una facile individuazione di tale contatto a bordo scheda, si faccia riferimento alla figura 9.

LOGICA DI CONTROLLO

Il mappaggio di tutti i registri delle periferiche presenti sulla scheda e dei dispositivi di memoria, è affidata ad un'opportuna logica di controllo che si occupa di allocare tali dispositivi nello spazio d'indirizzamento della CPU. Per maggiori informazioni fare riferimento al paragrafo "MAPPAGGIO I/O".

SPECIFICHE TECNICHE

CARATTERISTICHE GENERALI

Risorse della scheda:	2 timer a 16 bit (PRT) 1 linea seriale sincrona (CSI/O) 2 canale DMA (DMAC) 1 linea seriale RS 232 (ASCI1=A) 1 linea seriale RS 232, RS 422, RS 485, current loop (ASCI0=B) 1 contatto locale di reset 1 watch dog hardware astabile 1 real time clock 1 jumper di configurazione 1 interfaccia ABACO® I/O BUS 1 circuiteria di power failure
Memoria indirizzabile:	IC 2: EPROM da 128K x 8 a 512K x 8 FLASH EPROM da 128K x 8 a 512K x 8 IC 5: RAM da 128K x 8 a 512K x 8
CPU di bordo:	ZILOG Z180 o HITACHI 64HD180
Frequenza quarzo (clock):	18,432 (9,216) MHz
Tempo intervento watch dog:	da 1,00 sec a 2,25 sec (tipico 1,50 sec)

CARATTERISTICHE FISICHE

Dimensioni (L x A x P):	100 x 50 x 25 mm (senza contenitore) 110 x 60 x 60 mm (con contenitore per guide DIN)
Peso:	75 g (senza contenitore) 135 g (con contenitore per guide DIN)
Connettori:	CN1: 26 vie scatolino verticale M CN2: 2 vie scatolino verticale M CN3A: Plug a 6 vie F CN3B: Plug a 6 vie F CN5: 5+5 strip verticale M
Range di temperatura:	da 0 a 50 gradi Centigradi
Umidità relativa:	20% fino a 90% (senza condensa)

CARATTERISTICHE ELETTRICHE

Tensione di alimentazione:	+5 Vdc
Corrente assorbita sui 5 Vdc:	60 mA nella configurazione base
Batteria di bordo di back up:	3,0 Vdc; 180 mAh
Batteria esterna di back up:	3,6÷5 Vdc
Corrente di back up:	1,6 μ A (batteria di bordo)
Rete terminazione RS 422, RS 485:	Resistenza terminazione linea= 120 Ω Resistenza di pull up sul positivo= 3,3 K Ω Resistenza di pull down sul negativo=3,3 K Ω
Soglia d'intervento power failure:	1,25 Vdc

INSTALLAZIONE

In questo capitolo saranno illustrate tutte le operazioni da effettuare per il corretto utilizzo della scheda. A questo scopo viene riportata l'ubicazione e la funzione degli strip, dei connettori e dei LEDs, ecc. presenti sulla **GPC® 184**.

CONNESSIONI CON IL MONDO ESTERNO

Il modulo **GPC® 184** è provvisto di 5 connettori con cui vengono effettuati tutti i collegamenti con il campo e con le altre schede del sistema di controllo da realizzare. Di seguito viene riportato il loro pin out ed il significato dei segnali collegati; per una facile individuazione di tali connettori, si faccia riferimento alla figura 9, mentre per ulteriori informazioni a riguardo del tipo di connessioni, fare riferimento alle figure successive che illustrano il tipo di collegamento effettuato a bordo scheda.

CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP

CN2 é un connettore a scatolino, verticale, maschio, con passo 2,54 mm a 2 vie. Tramite CN2 deve essere collegata una batteria esterna che provvede a mantenere i dati della RAM di bordo ed a garantire il funzionamento del real time clock, in assenza di tensione di alimentazione (per maggiori informazioni fare riferimento al paragrafo "BACK UP").

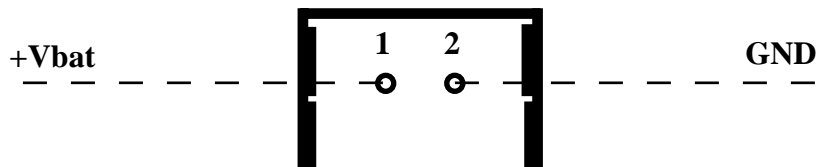


FIGURA 2: CN2 - CONNETTORE PER BATTERIA ESTERNA DI BACK UP

Legenda:

+Vbat	=	I	-	Positivo della batteria esterna di back up
GND	=	-	-	Negativo della batteria esterna di back up

CN1 - CONNETTORE PER ABACO® I/O BUS

CN1 è un connettore a scatolino verticale con passo 2.54 mm a 26 piedini. Tramite CN1 si effettua la connessione tra la scheda e la serie di moduli esterni di espansione, da utilizzare per l'interfacciamento diretto con il campo. Tale collegamento è effettuato tramite l'ABACO® I/O BUS di cui questo connettore riporta tutti i segnali a livello TTL.

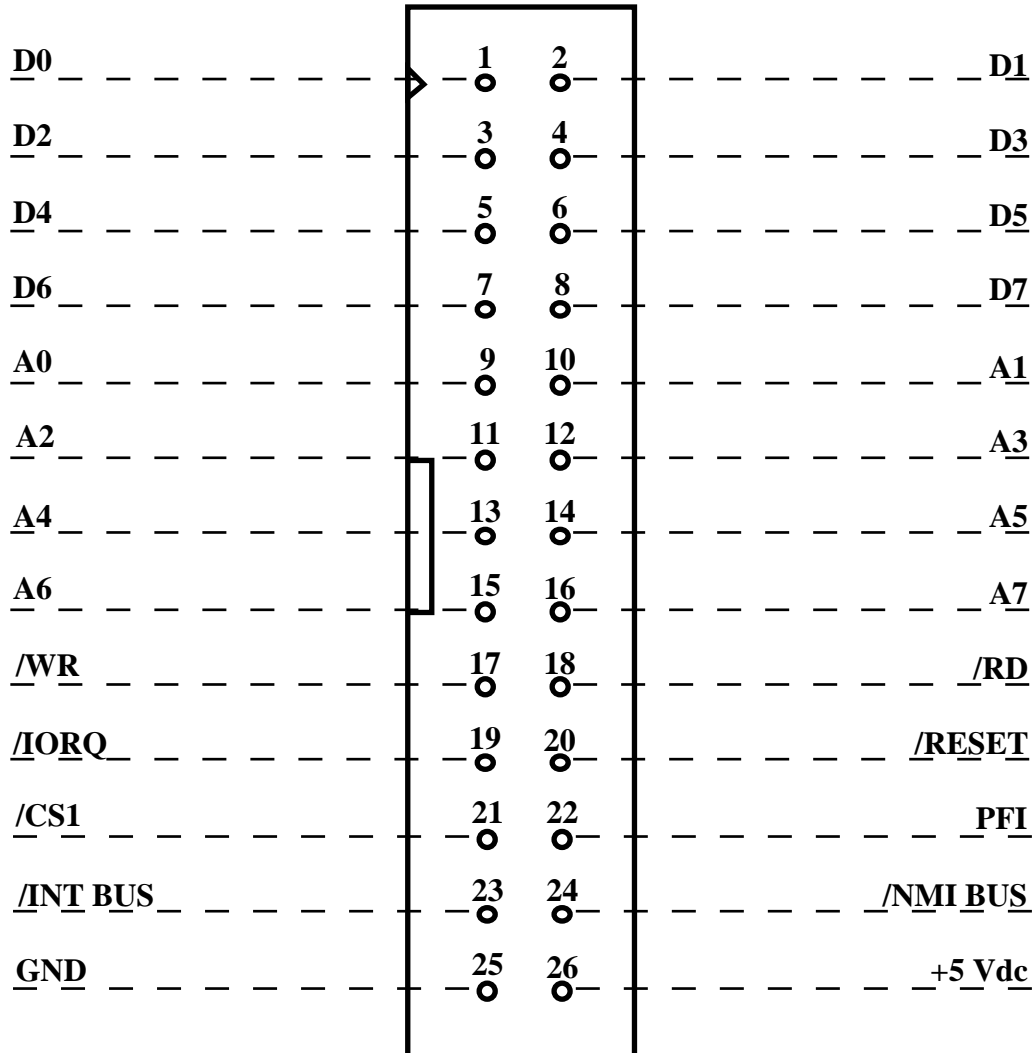


FIGURA 3: CN1 - CONNETTORE PER ABACO® I/O BUS

Legenda:

A0-A7	=	O	- Address BUS: BUS degli indirizzi.
D0-D7	=	I/O	- Data BUS: BUS dei dati.
/INT BUS	=	I	- Interrupt request: richiesta d'interrupt. Deve essere in open collector
/NMI BUS	=	I	- Non Mascarable Interrupt: richiesta d'interrupt non mascherabile.
/IORQ	=	O	- Input Output Request: richiesta operazione Input Output su I/O BUS.
/RD	=	O	- Read cycle status: richiesta di lettura.
/WR	=	O	- Write cycle status: richiesta di scrittura.
/RESET	=	O	- Reset: azzeramento.
/CS1	=	O	- Chip select 1: abilitazione decodificata per periferiche esterne.
PFI	=	I	- Power Failure input: ingresso per riconoscimento caduta alimentazione.
+5 Vdc	=	I/O	- Linea di alimentazione a +5 Vcc.
GND	=		- Linea di massa.

CN3A - CONNETTORE PER LINEA SERIALE A

Il connettore per la comunicazione della linea seriale A, in RS 232, denominato CN3A sulla scheda, é del tipo PLUG a 6 vie. Fisicamente la linea seriale A della **GPC® 184** é collegata alla linea seriale ASCII 1 della CPU. La disposizione di tali segnali, riportata di seguito, é stata studiata in modo da ridurre al minimo le interferenze ed in modo da facilitare la connessione con il campo, mentre i segnali rispettano le normative definite dal CCITT relative allo standard RS 232.

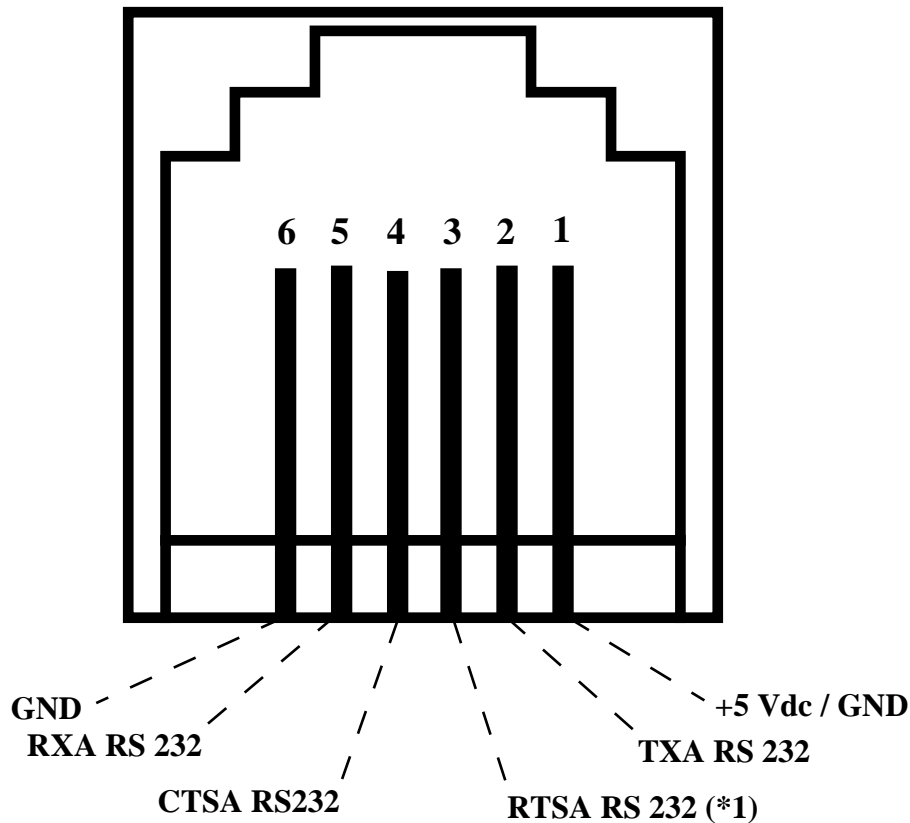


FIGURA 6: CN3A - CONNETTORE PER LINEA SERIALE A

Legenda:

RXA RS 232	= I - Receive Data: linea di ricezione in RS 232 della seriale A=ASCII1.
TXA RS 232	= O - Transmit Data: linea di trasmissione in RS 232 della seriale A=ASCII1.
CTSA RS 232	= I - Clear To Send: linea di abilitazione alla trasmissione in RS 232 della seriale A=ASCII1.
RTSA RS 232	= O - Request To Send: linea di richiesta di trasmissione in RS 232 della seriale A=ASCII1 (*1).
+5 Vdc/GND	= I - Linea di alimentazione a +5 Vcc o linea di massa.
GND	= - Linea di massa.

***1:** Il segnale di handshake di uscita RTSA non é gestibile via software ed é mantenuto sempre disattivo = -10 Vdc. Se questo stato é incompatibile con il sistema a cui si collega, provvedere ad effettuare un collegamento senza questo segnale.

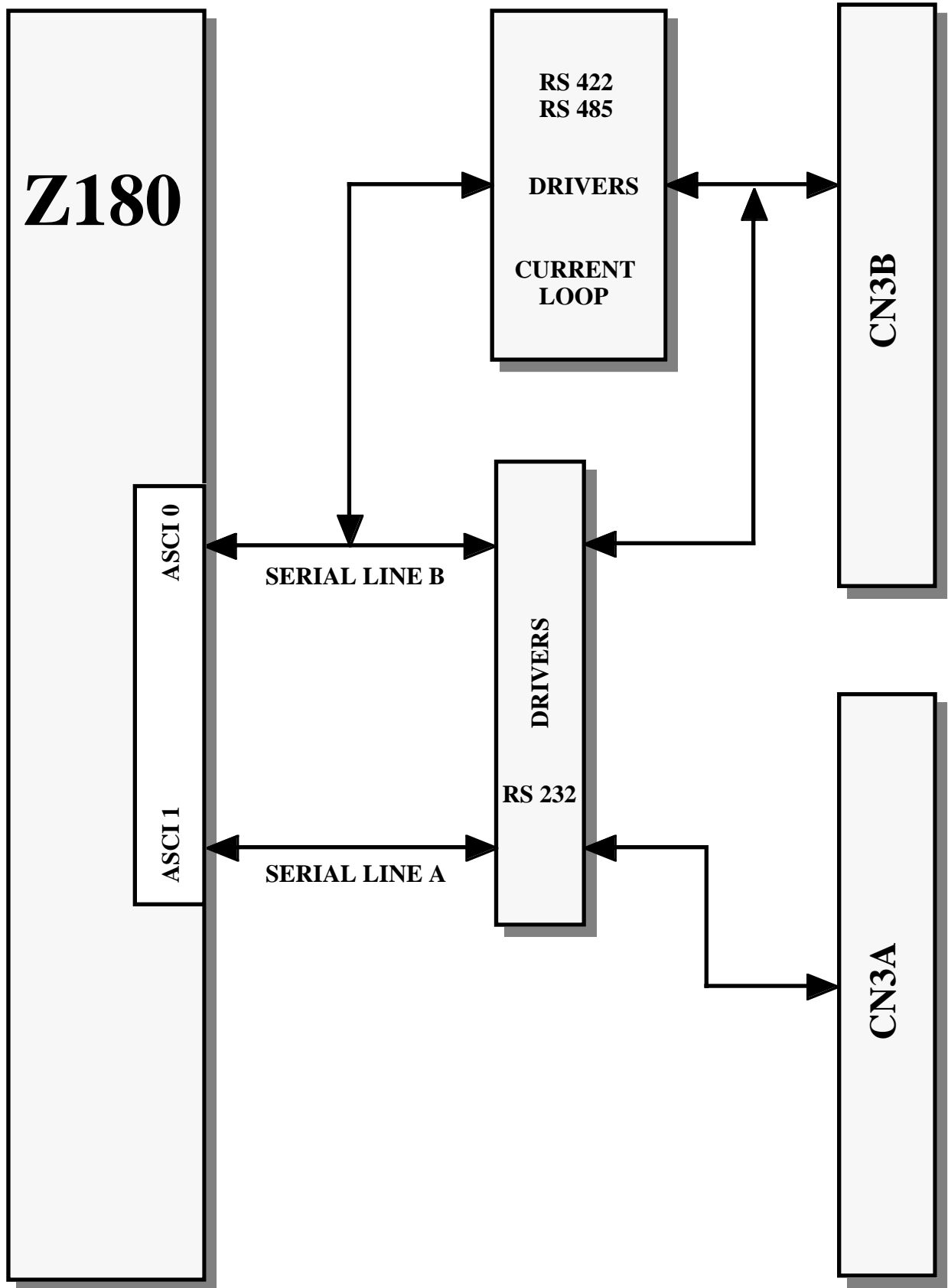


FIGURA 7: SCHEMA DI COMUNICAZIONE SERIALE

CN3B - CONNETTORE PER LINEA SERIALE B

Il connettore per la comunicazione della linea seriale B, in RS 232, RS 422, RS 485 o Current Loop, denominato CN3B sulla scheda, é del tipo PLUG a 6 vie. Fisicamente la linea seriale B della **GPC® 184** é collegata alla linea seriale ASCII 0 della CPU. La disposizione di tali segnali, riportata di seguito, é stata studiata in modo da ridurre al minimo le interferenze ed in modo da facilitare la connessione con il campo, mentre i segnali rispettano le normative definite dal CCITT relative ad ognuno degli standard di comunicazione usati.

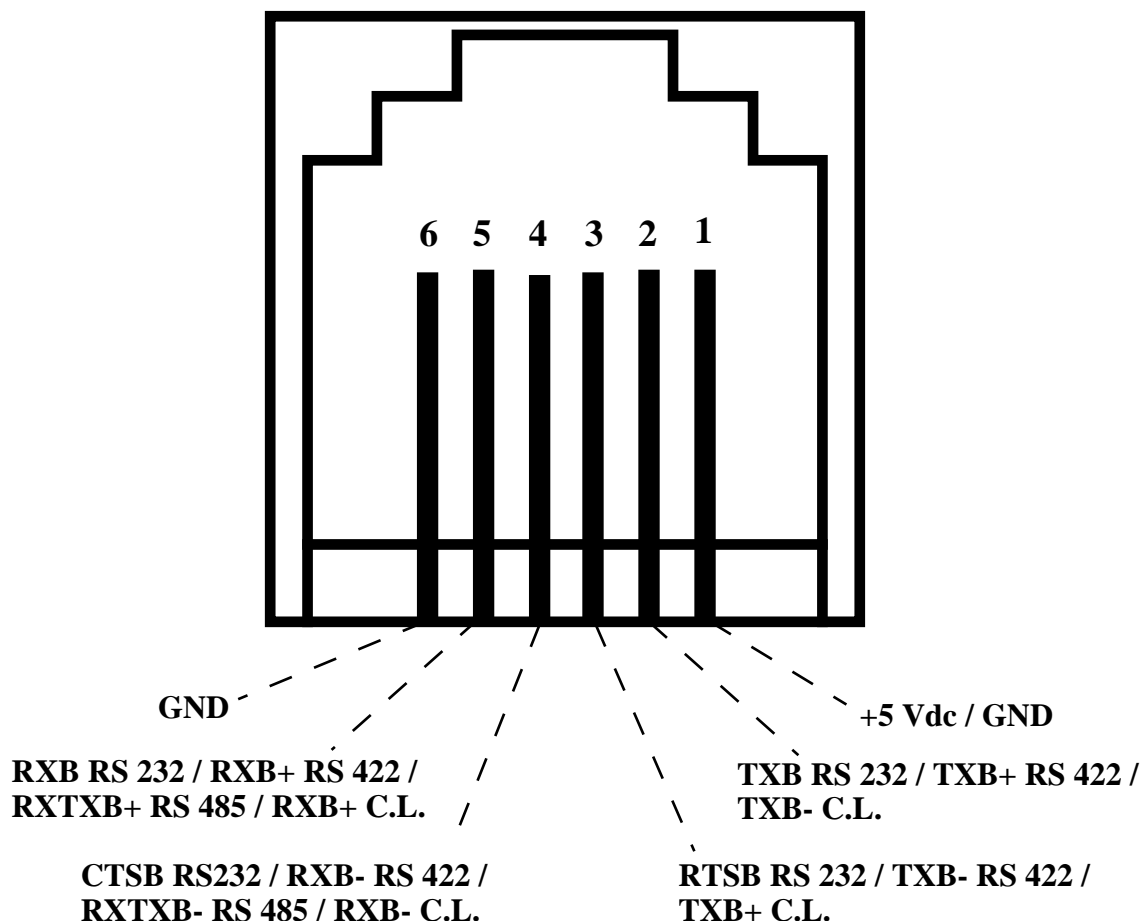


FIGURA 8: CN3B - CONNETTORE PER LINEA SERIALE B

Legenda:

RXB RS 232	= I - Receive Data: linea ricezione in RS 232 della seriale B=ASCII 0.
TXB RS 232	= O - Transmit Data: linea trasmissione in RS 232 della seriale B=ASCII 0.
CTSB RS 232	= I - Clear To Send: linea di abilitazione alla trasmissione in RS 232 della seriale B=ASCII 0.
RTSB RS 232	= O - Request To Send: linea di richiesta di trasmissione in RS 232 della seriale B=ASCII 0.
RXB- RS 422	= I - Receive Data Negative: linea bipolare negativa di ricezione differenziale in RS 422 della seriale B=ASCII 0.
RXB+ RS 422	= I - Receive Data Positive: linea bipolare positiva di ricezione differenziale in RS 422 della seriale B=ASCII 0.
TXB- RS 422	= O - Transmit Data Negative: linea bipolare negativa di trasmissione differenziale in RS 422 della seriale B=ASCII 0.

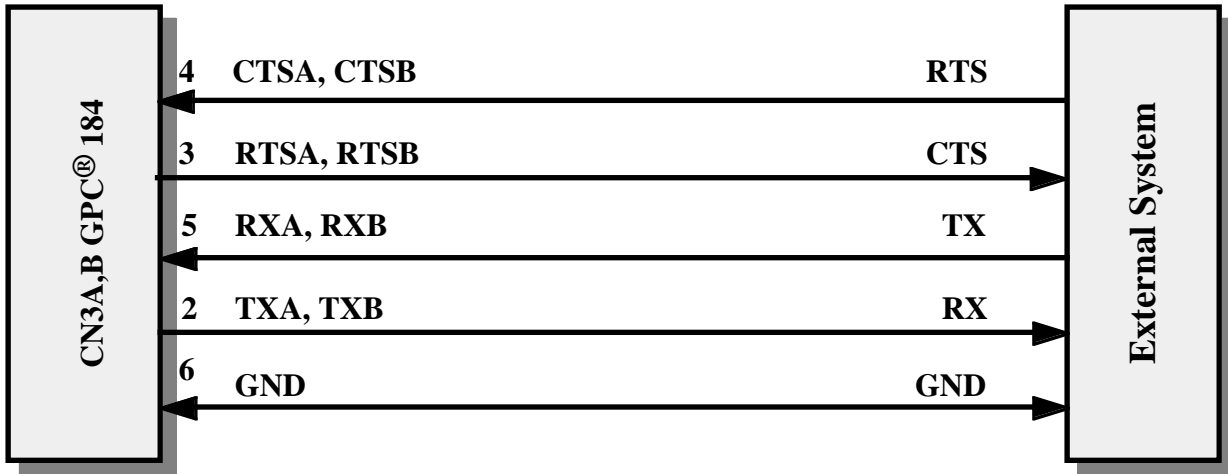


FIGURA 10: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 232

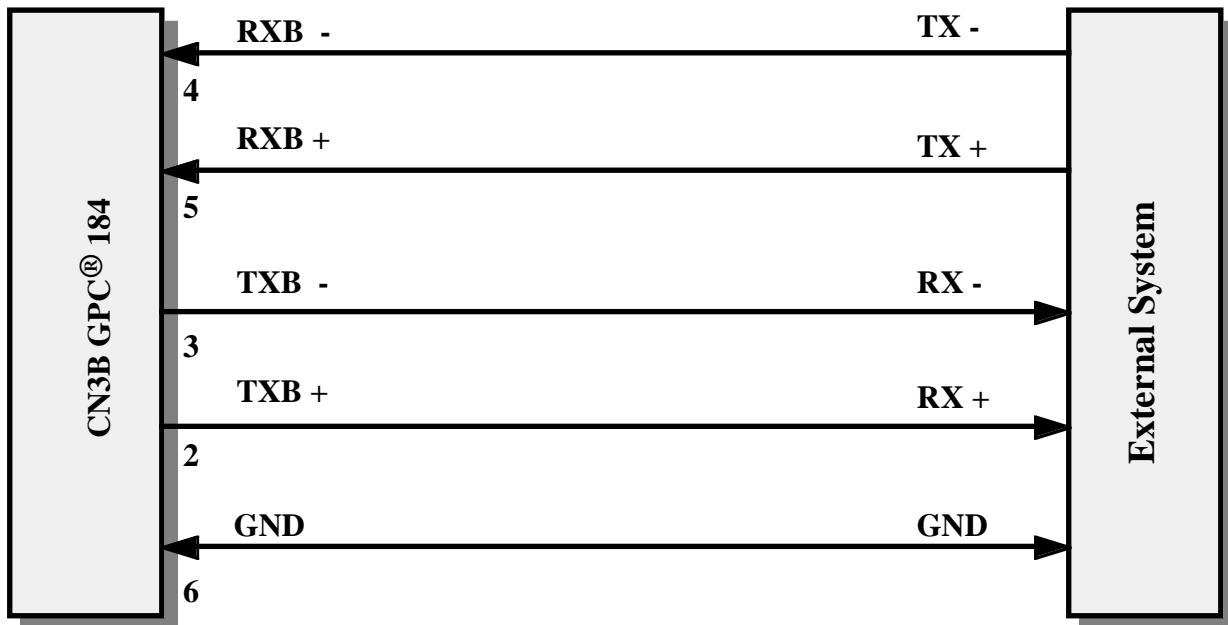


FIGURA 11: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 422

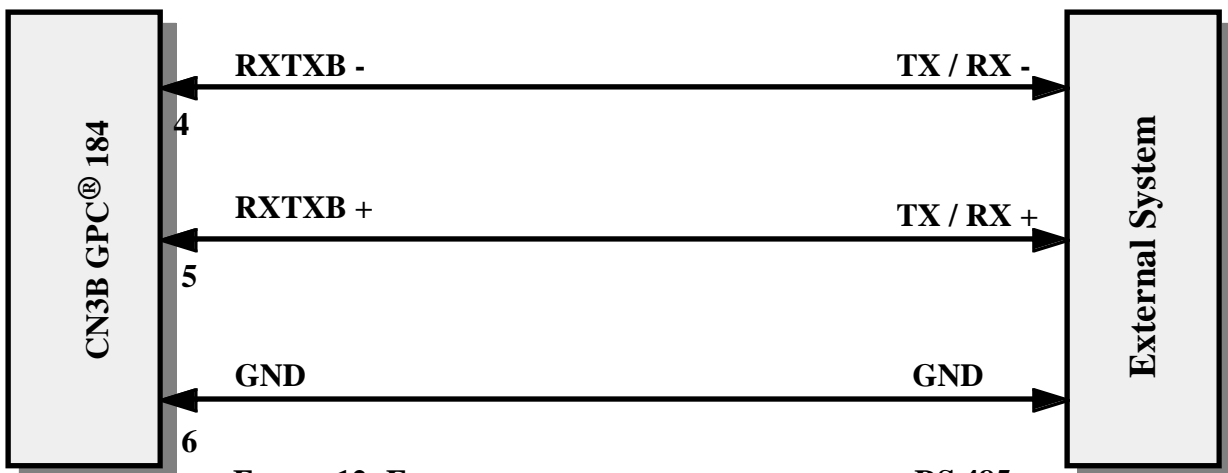


FIGURA 12: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN RS 485

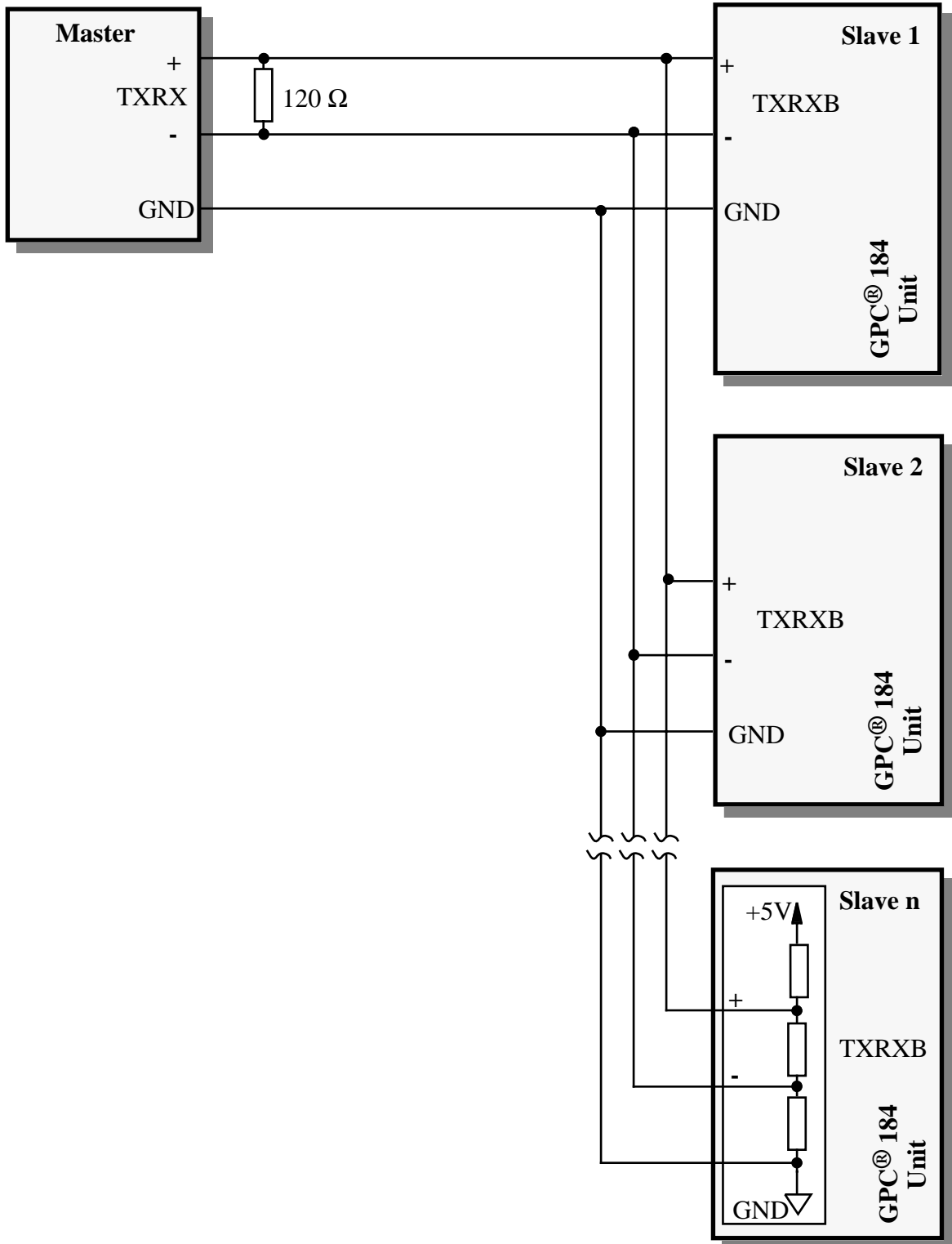


FIGURA 13: ESEMPIO COLLEGAMENTO IN RETE IN RS 485

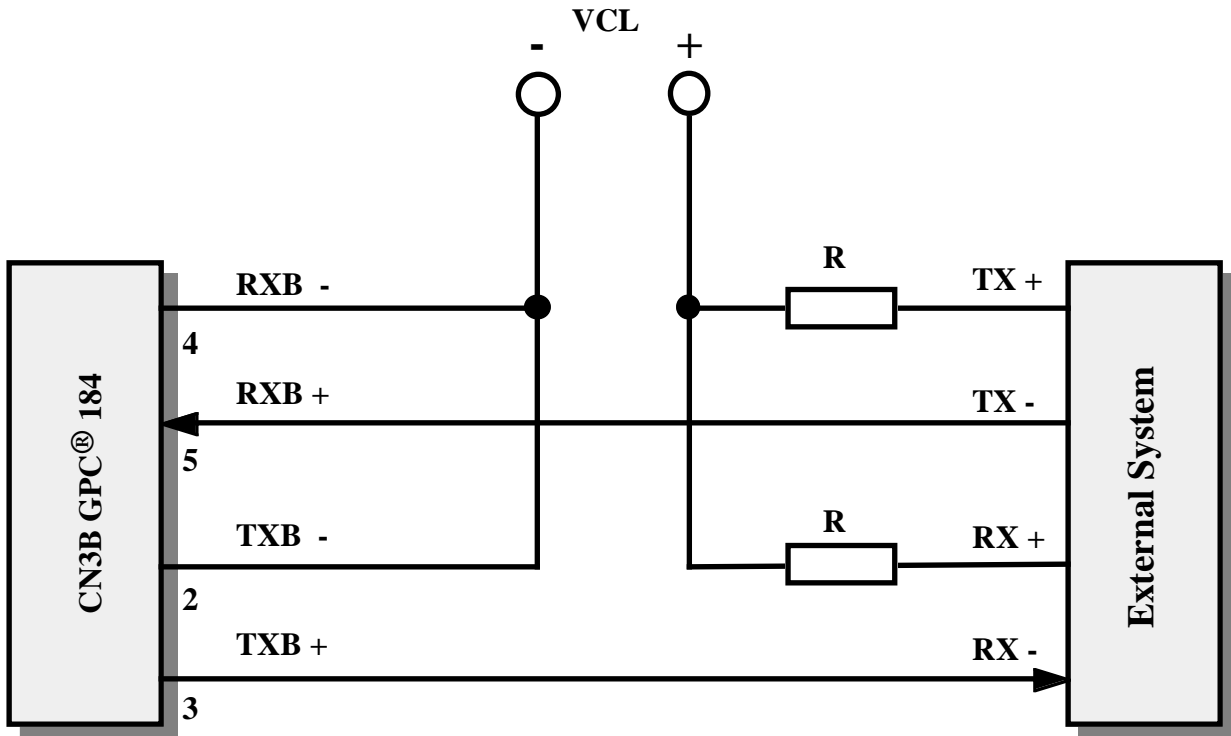


FIGURA 14: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 4 FILI

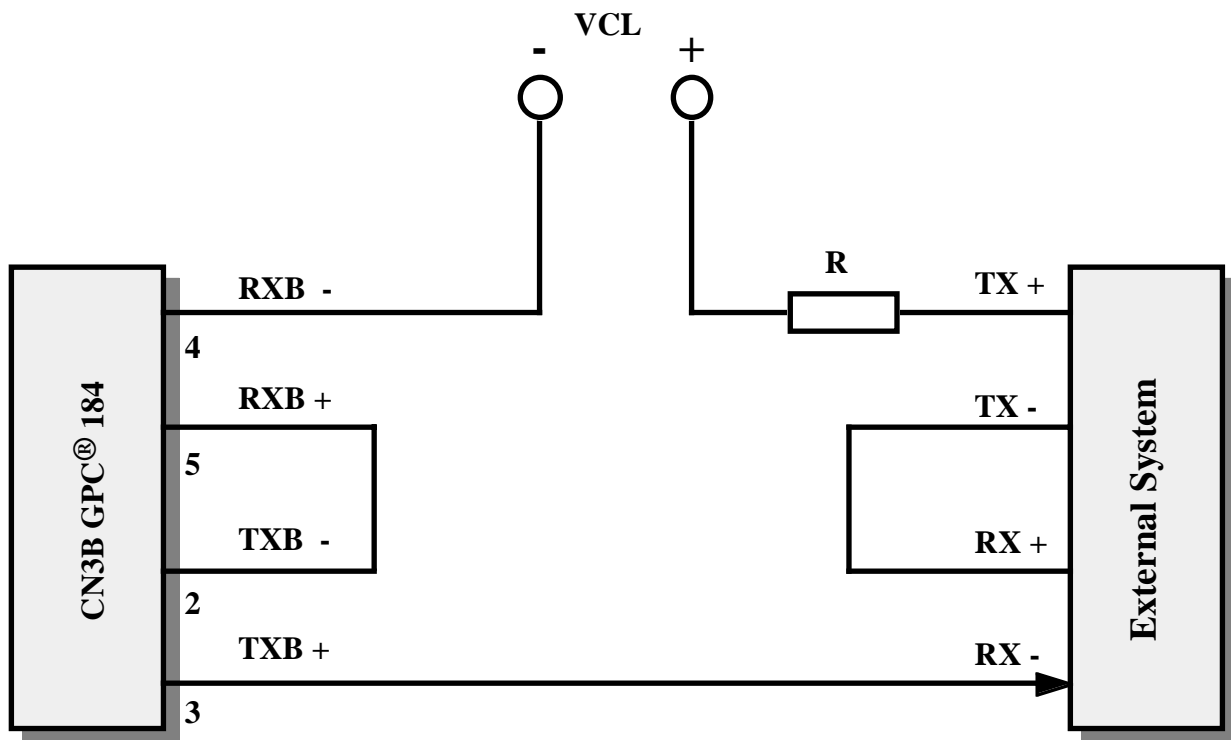


FIGURA 15: ESEMPIO COLLEGAMENTO PUNTO PUNTO IN CURRENT LOOP A 2 FILI

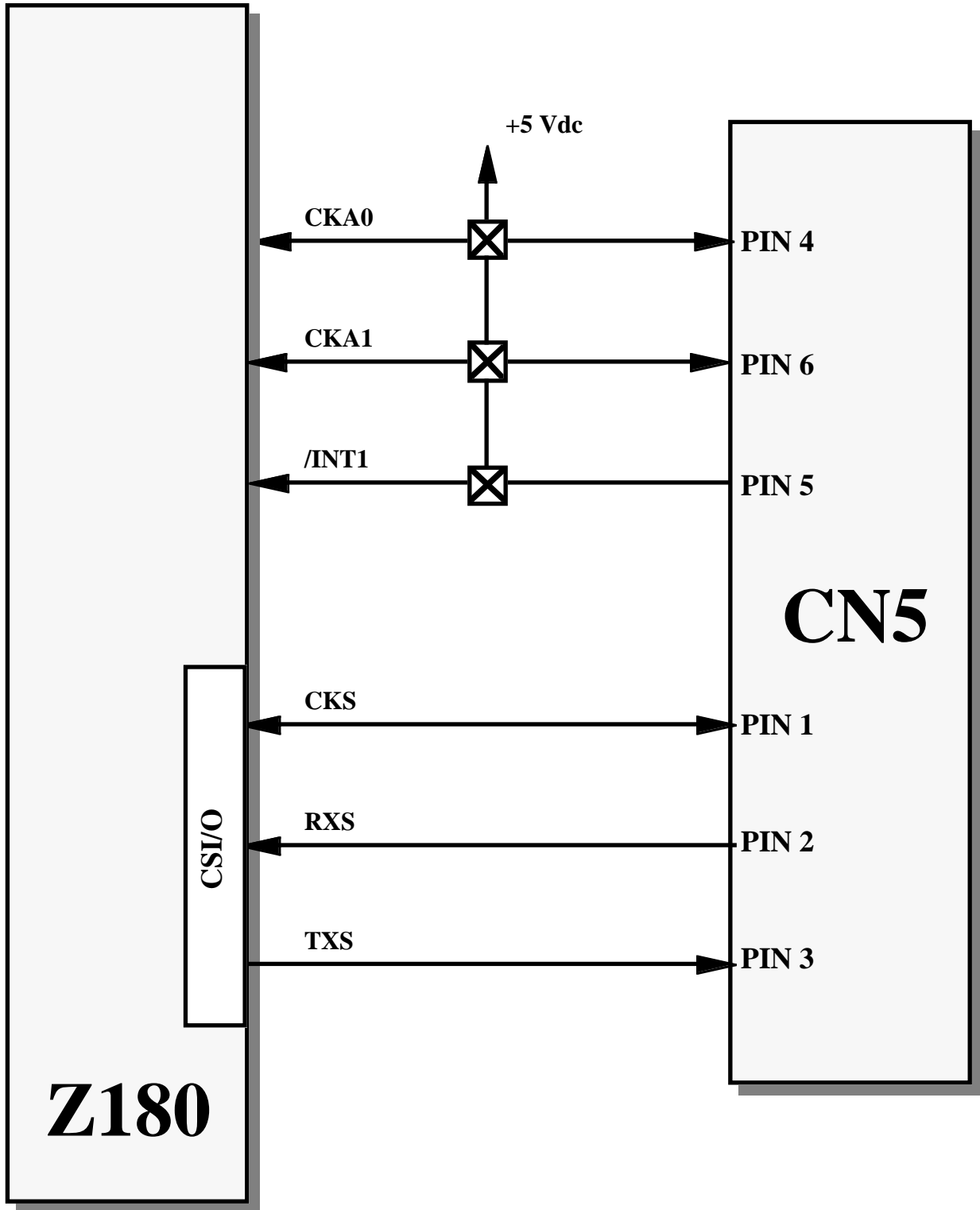


FIGURA 16: SCHEMA DI COLLEGAMENTO LINEE DI CN5

CN5 - CONNETTORE PER SEGNALI AUSILIARI

CN5 è un connettore a strip verticale con passo 2.54 mm a 5+5 piedini. Tramite CN5 si interfacciano le 3 linee per la comunicazione seriale sincrona, un segnale di interrupt /INT1, 2 linee per il baud rate generator ed infine l'alimentazione. Tutti i segnali sono del tipo TTL.

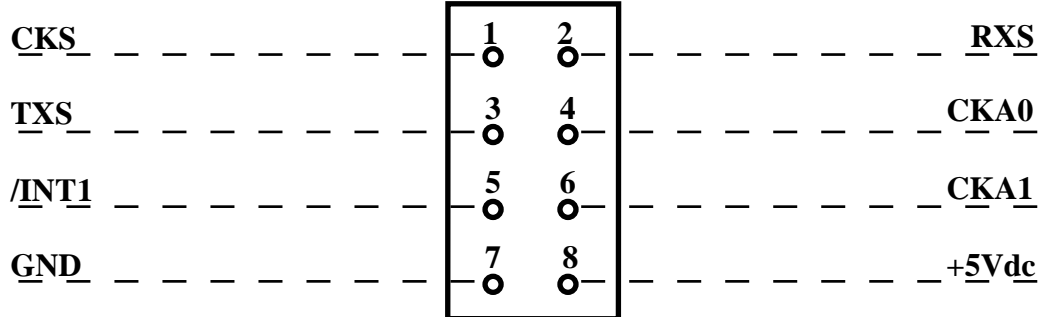


FIGURA 17: CN5 - CONNETTORE PER SEGNALI AUSILIARI

Legenda:

CKS	=	I/O	- Linea di clock per la seriale sincrona.
RXS	=	I	- Linea di ricezione per la seriale sincrona.
TXS	=	O	- Linea di trasmissione per la seriale sincrona.
/INT1	=	I	- Interrupt request: richiesta d'interrupt /INT1.
CKA0	=	I/O	- Linea di clock della sezione di baud rate generator ASCII 0 = seriale B.
CKA1	=	I/O	- Linea di clock della sezione di baud rate generator ASCII 1 = seriale A.
+5 Vdc	=	O	- Linea di alimentazione a +5 Vcc
GND	=		- Linea di massa.

INTERFACCIAMENTO DEGLI I/O CON IL CAMPO

Al fine di evitare eventuali problemi di collegamento della scheda con tutta l'elettronica del campo a cui la **GPC® 184** si deve interfacciare, si devono seguire le informazioni riportate nei precedenti paragrafi e le relative figure che illustrano le modalità interne di connessione.

- Per i segnali che riguardano la comunicazione seriale con i protocolli RS 232, RS 422, RS 485 o Current Loop, fare riferimento alle specifiche standard di ognuno di questi protocolli.
- Tutti i segnali a livello TTL possono essere collegati a linee dello stesso tipo riferite alla massa digitale della scheda. Il livello 0V corrisponde allo stato logico 0, mentre il livello 5V corrisponde allo stato logico 1.

SEGNALAZIONI VISIVE

La scheda **GPC® 184** é dotata di un LED con cui segnala alcune condizioni di stato, come descritto nella seguente tabella:

LED	COLORE	FUNZIONE
LD1	Rosso	Segnala l'attivazione della linea di interrupt (/INT2 della CPU), proveniente dal Real Time Clock.

FIGURA 18: TABELLA DELLE SEGNALAZIONI VISIVE

La funzione principale di questo LED é quella di fornire un'indicazione visiva dello stato della scheda, facilitando quindi le operazioni di verifica di funzionamento di tutto il sistema. Per una più facile individuazione di tali segnalazioni visive, si faccia riferimento alla figura 9, mentre per la descrizione delle modalità di generazione interrupt con il RTC fare riferimento all'apposito paragrafo "REAL TIME CLOCK".

JUMPERS

Esistono a bordo della GPC® 184 14 jumpers, di cui 5 a cavaliere e 9 a stagno, con cui é possibile effettuare alcune selezioni che riguardano il modo di funzionamento della stessa. Di seguito ne é riportato l'elenco, l'ubicazione e la loro funzione nelle varie modalit  di connessione.

JUMPERS	N. VIE	UTILIZZO
J1	3	Seleziona il size del dispositivo di RAM montato su IC5.
J2	3	In abbinamento a J3, seleziona il tipo di dispositivo EPROM o FLASH EPROM, montato su IC2.
J3	3	In abbinamento a J2, seleziona il tipo di dispositivo EPROM o FLASH EPROM, montato su IC2.
J4	2	Definisce stato ingresso di configurazione (RUN o DEBUG).
J5	5	Seleziona tipo di comunicazione per la linea seriale B tra RS 422 e RS 485.
JS1	2	In abbinamento a JS2, collega la circuiteria di terminazione e forzatura alla linea RS 485 o alla linea di ricezione RS 422.
JS2	2	In abbinamento a JS1, collega la circuiteria di terminazione e forzatura alla linea RS 485 o alla linea di ricezione RS 422.
JS3	3	Seleziona il tipo di collegamento per il pin 1 di CN3A.
JS4	3	Seleziona il tipo di collegamento per il pin 1 di CN3B.
JS10	2	Attiva la circuiteria di watch dog.
JS11	2	Mantiene abilitato handshake CTSB della CPU.
JS14	2	Collega la batteria di bordo BT1 alla circuiteria di back up.
JS19	3	Seleziona interrupt a cui collegare la circuiteria di power failure.
JS21	2	Seleziona collegamento dell'handshake CTSA.

FIGURA 19: TABELLA RIASSUNTIVA JUMPERS

Di seguito é riportata una descrizione tabellare delle possibili connessioni dei 14 jumpers con la loro relativa funzione. Per riconoscere tali connessioni sulla scheda si faccia riferimento alla serigrafia della stessa o alla figura 5 di questo manuale, dove viene riportata la numerazione dei pin dei jumpers, che coincide con quella utilizzata nella seguente descrizione. Per l'individuazione dei jumpers a bordo della scheda, si utilizzino invece le figure 20, 21. In tutte le seguenti tabelle l'* indica la connessione di default, ovvero quella impostata in fase di collaudo, con cui la scheda viene fornita.

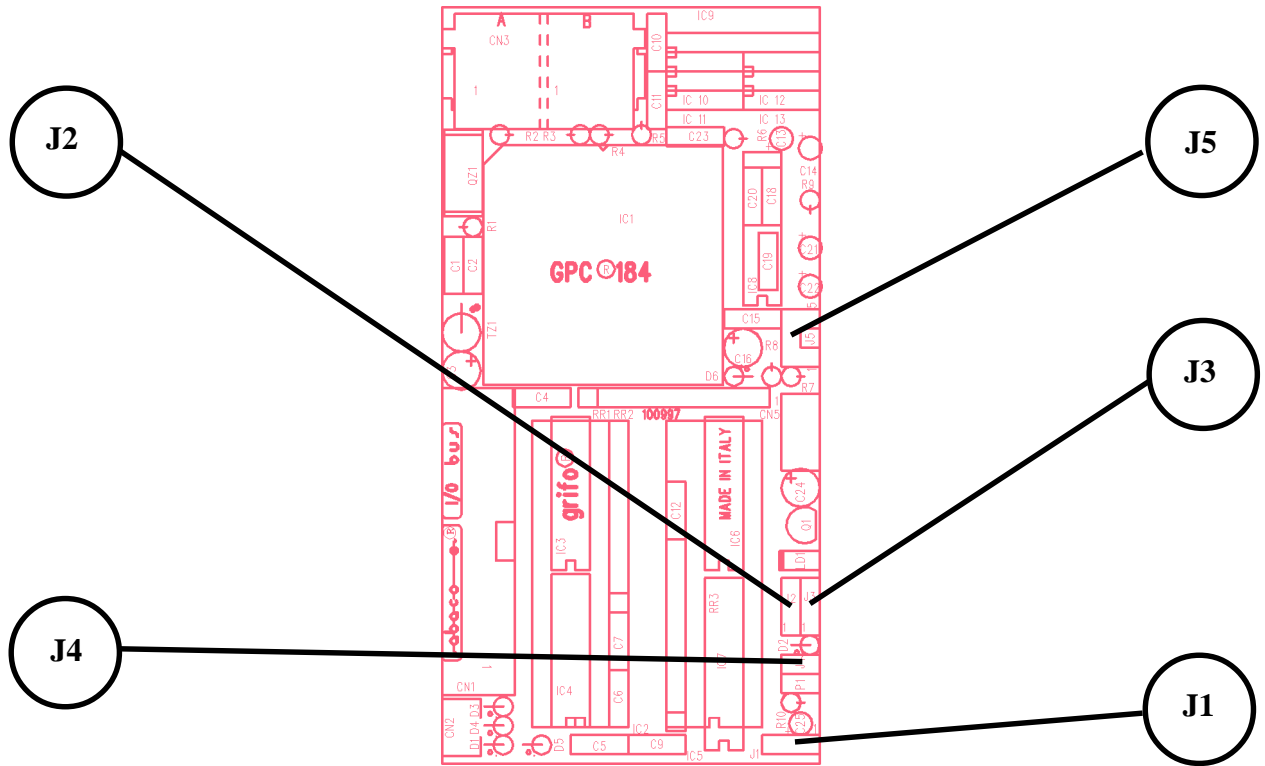


FIGURA 20: DISPOSIZIONE JUMPERS LATO COMPONENTI

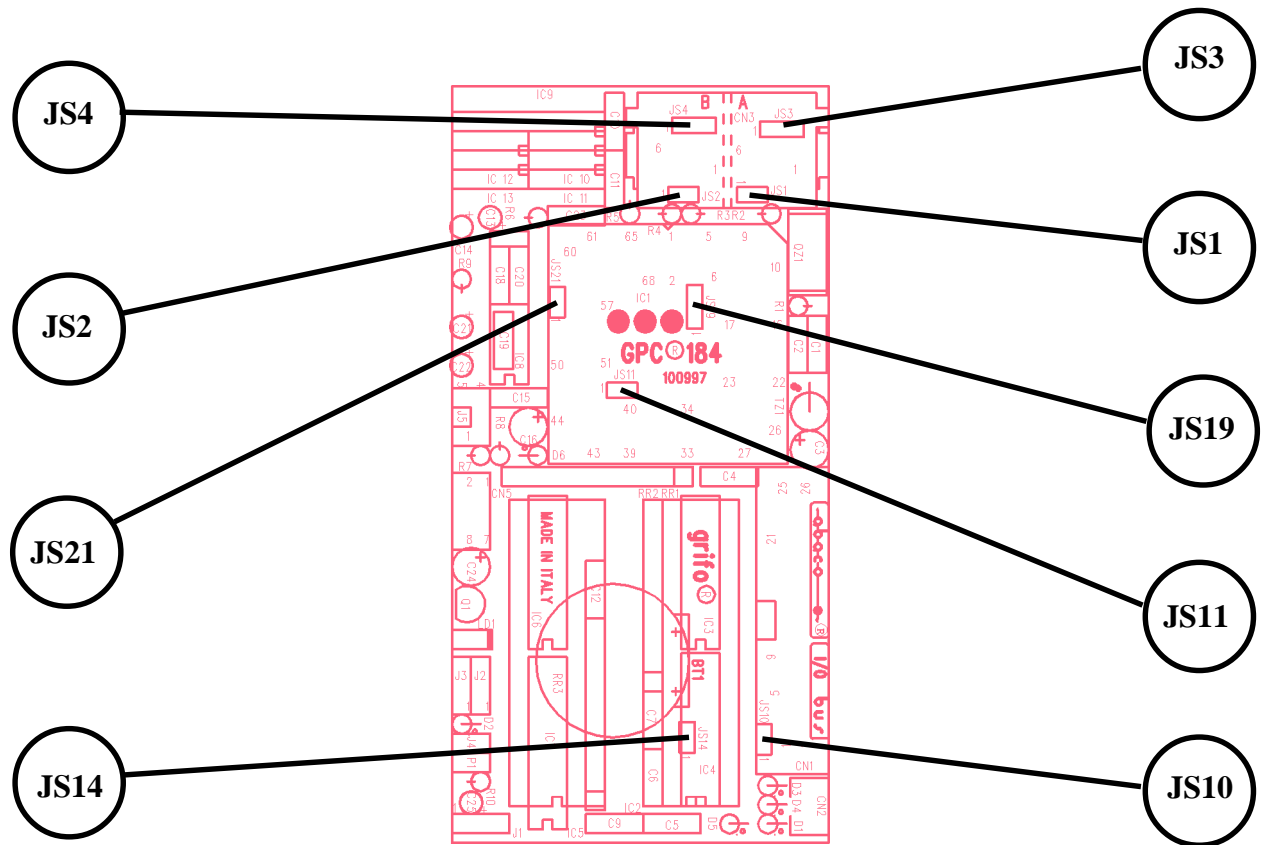


FIGURA 21: DISPOSIZIONE JUMPERS LATO STAGNATURE

JUMPERS A 2 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J4	non connesso	Collega a +5Vcc l'ingresso di configurazione, selezionando la modalità RUN.	*
	connesso	Collega a GND l'ingresso di configurazione, selezionando la modalità DEBUG.	
JS1	non connesso	In abbinamento a JS2, non collega la circuiteria di terminazione e forzatura alla linea seriale RS 485 o alla linea di ricezione RS 422.	*
	connesso	In abbinamento a JS2, collega la circuiteria di terminazione e forzatura alla linea seriale RS 485 o alla linea di ricezione RS 422.	
JS2	non connesso	In abbinamento a JS1, non collega la circuiteria di terminazione e forzatura alla linea seriale RS 485 o alla linea di ricezione RS 422.	*
	connesso	In abbinamento a JS1, collega la circuiteria di terminazione e forzatura alla linea seriale RS 485 o alla linea di ricezione RS 422.	
JS10	non connesso	Disattiva il funzionamento della circuiteria di watch dog.	*
	connesso	Attiva il funzionamento della circuiteria di watch dog.	
JS11	non connesso	Non mantiene abilitato il segnale di handshake CTS0 (=CTSB) della CPU.	*
	connesso	Mantiene abilitato il segnale di handshake CTS0 (=CTSB) della CPU.	
JS14	non connesso	Non collega la batteria di bordo BT1 alla circuiteria di back up.	*
	connesso	Collega la batteria di bordo BT1 alla circuiteria di back up.	
JS21	non connesso	Non collega handshake CTS1 (=CTSA) alla CPU.	*
	connesso	Collega handshake CTS1 (=CTSA) alla CPU.	

FIGURA 22: TABELLA JUMPERS A 2 VIE

JUMPERS A 3 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J1	posizione 1-2	Predisporre IC5 per RAM da 128K.	*
	posizione 2-3	Predisporre IC5 per RAM da 512K.	
J2	posizione 1-2	Predisporre IC2 per FLASH EPROM, in congiunzione con J3.	*
	posizione 2-3	Predisporre IC2 per EPROM, in congiunzione con J3.	
J3	posizione 1-2	Predisporre IC2 per FLASH EPROM, in congiunzione con J2.	*
	posizione 2-3	Predisporre IC2 per EPROM, in congiunzione con J2.	
JS3	posizione 1-2	Collega il pin 1 di CN3A a GND.	*
	posizione 2-3	Collega il pin 1 di CN3A a +5 Vcc.	
JS4	posizione 1-2	Collega il pin 1 di CN3B a GND.	*
	posizione 2-3	Collega il pin 1 di CN3B a +5 Vcc.	
JS19	Non connesso	Non collega la circuiteria di power failure.	*
	posizione 1-2	Collega la circuiteria di power failure al segnale /INT0 della CPU.	
	posizione 2-3	Collega la circuiteria di power failure al segnale /NMI della CPU.	

FIGURA 23: TABELLA JUMPERS A 3 VIE**JUMPERS A 5 VIE**

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J5	non connesso	Seleziona comunicazione in RS 232 su seriale B.	*
	posizione 1-2 e 3-4	Seleziona comunicazione in RS 485 (half duplex a 2 fili) su seriale B.	
	posizione 2-3 e 4-5	Seleziona comunicazione in RS 422 (full duplex o half duplex a 4 fili) su seriale B.	

FIGURA 24: TABELLA JUMPERS A 5 VIE

BACK UP

La **GPC® 184** é provvista di una batteria al litio BT1 che provvede a tamponare la RAM ed il RTC di bordo anche in assenza della tensione di alimentazione. Il jumper JS14 provvede a collegare o meno questa batteria in modo da salvaguardarne la durata prima dell'installazione o in tutti i casi in cui il back up non é necessario. Una seconda batteria esterna può essere collegata alla circuiteria di back up tramite il connettore CN2: quest'ultima non é interessata dalla configurazione del jumper JS14 e sostituisce a tutti gli effetti la BT1.

Per la scelta della batteria esterna di back up seguire le indicazioni del paragrafo "CARATTERISTICHE ELETTRICHE", mentre per la sua individuazione si veda la figura 9.

SELEZIONE MEMORIE

La **GPC® 184** può montare fino ad un massimo di 1024 Kbytes di memoria variamente suddivisa. In particolare per la configurazione seguire le informazioni riportate nella seguente tabella:

IC	DISPOSITIVO	DIMENSIONE	STRIPPAGGIO
2	EPROM	128K Byte	J2, J3 in posizione 2-3
	EPROM	256K Byte	J2, J3 in posizione 2-3
	EPROM	512K Byte	J2, J3 in posizione 2-3
	FLASH EPROM	128K Byte	J2, J3 in posizione 1-2
	FLASH EPROM	512K Byte	J2, J3 in posizione 1-2
5	RAM	128K Byte	J1 in posizione 1-2
	RAM	512K Byte	J1 in posizione 2-3

FIGURA 25: TABELLA DI SELEZIONE MEMORIE

Tutti i dispositivi sopra descritti devono essere con pin out di tipo JEDEC. Per quanto riguarda le sigle dei vari dispositivi che possono essere montati, fare riferimento alla documentazione delle case costruttrici. Per una facile individuazione dei dispositivi di memoria fare riferimento alla figura 9. Normalmente la **GPC® 184** é fornita nella sua configurazione di default con solamente 128K RAM; ogni configurazione diversa può essere autonomamente montata dall'utente oppure richiesta nella fase di ordine. Sotto sono riportate i codici delle opzioni di memoria disponibili:

.512K -> 512K RAM

Per ulteriori informazioni e costi delle opzioni, contattare direttamente la **grifo®**.

INTERRUPTS

Una caratteristica peculiare della **GPC® 184** è la notevole potenza nella gestione delle interruzioni. Di seguito viene riportata una breve descrizione di quali sono i dispositivi che possono generare interrupts e con quale modalità; per quanto riguarda la gestione di tali interrupts si faccia riferimento ai data sheets del microprocessore oppure all'appendice B di questo manuale.

- **ABACO® I/O BUS** -> Genera un /NMI, tramite la linea /NMI BUS di CN1.
Genera un /INT0 non vettorizzato, tramite la linea /INT BUS.
- Real Time Clock -> Genera un /INT2 vettorizzato.
- Segnali ausiliari -> Genera un /INT1 vettorizzato, tramite l'ononimo segnale di CN5.
- Power failure -> Genera un /NMI od un /INT0 non vettorizzato.
- Periferiche della CPU -> Generano un interrupt vettorizzato. In particolare le possibili sorgenti d'interrupt interno sono le sezioni: PRT 0, PRT 1, DMA 0, DMA 1, CSI/O, ASCI 0, ASCI 1.

Sulla scheda é presente una catena di priorità hardware che regola l'attivazione contemporanea di più interrupts. Per gli interrupts vettorizzati gli indirizzi delle procedure di risposta possono essere programmati arbitrariamente via software, tramite i registri interni al microprocessore. In questo modo l'utente ha sempre la possibilità di rispondere in maniera efficace e veloce a qualsiasi evento esterno, stabilendo anche la priorità delle varie sorgenti.

INGRESSO DI CONFIGURAZIONE

La scheda **GPC® 184** è provvista di 1 jumper (J4), tipicamente utilizzato per la configurazione del sistema, acquisibile via software dall'utente. Le applicazioni più immediate possono essere quelle destinate al settaggio delle condizioni di lavoro od alla selezione di parametri relativi al firmware di bordo. La lettura della combinazione fissata sul jumper avviene in logica negata (0 -> jumper connesso ed 1 -> jumper non connesso) effettuando un'operazione di lettura agli indirizzi dedicati dalla logica di controllo della scheda. In abbinamento ad alcuni pacchetti software, il jumper J4 svolge la funzione di selettore della modalità RUN e DEBUG. Per ulteriori informazioni si faccia riferimento ai paragrafi "MAPPAGGIO I/O", mentre per una facile individuazione della sua posizione si veda la figura 20.

JUMPER A STAGNO

La connessione di default dei jumpers a stagno denominati **JSxx**, é effettuata con una sottile pista sul lato stagnature. Quindi, se tale configurazione deve essere variata, si deve prima tagliare la pista con un taglierino affilato e poi effettuare la connessione richiesta con uno stagnatore di bassa potenza utilizzando dello stagno non corrosivo.

COMUNICAZIONE SERIALE

La linea di comunicazione seriale A della scheda **GPC® 184** può essere bufferata solo in RS 232, mentre la linea seriale B può essere bufferata in RS 232, RS 422, RS 485 o current loop. La selezione del tipo d'interfacciamento avviene via hardware e viene effettuata tramite un opportuno strippaggio dei jumpers di bordo, come può essere desunto dalla lettura delle precedenti tabelle. Dal punto di vista software sono invece definibili tutti i parametri del protocollo fisico di comunicazione tramite la programmazione dei registri interni della CPU.

Alcuni componenti necessari per le configurazioni RS 422, RS 485 e current loop non sono montati e collaudati sulla scheda in configurazione di default; per questo la prima configurazione della seriale B non in RS 232 deve essere sempre effettuata dai tecnici **grifo®**. A questo punto l'utente può cambiare autonomamente la configurazione seguendo le informazioni sotto riportate:

- LINEA SERIALE B=ASCI 0 SETTATA IN RS 232 (configurazione default)

J5	=	non connesso	IC9	=	driver MAX 202
JS1, JS2	=	non connessi	IC10	=	nessun componente
JS11	=	non connesso	IC11	=	nessun componente
			IC12	=	nessun componente
			IC13	=	nessun componente

- LINEA SERIALE B=ASCI 0 SETTATA IN CURRENT LOOP (opzione .CLOOP)

J5	=	non connesso	IC9	=	nessun componente
JS1, JS2	=	non connessi	IC10	=	nessun componente
JS11	=	connesso	IC11	=	driver HP 4200
			IC12	=	nessun componente
			IC13	=	driver HP 4100

Da ricordare che l'interfaccia seriale in current loop é di tipo passivo e si deve quindi collegare una linea current loop attiva, ovvero provvista di un proprio alimentatore. L'interfaccia current loop può essere utilizzata per realizzare sia connessioni punto punto che multipunto con un collegamento a 4 o 2 fili.

- LINEA SERIALE B=ASCI 0 SETTATA IN RS 422 (opzione .RS 422)

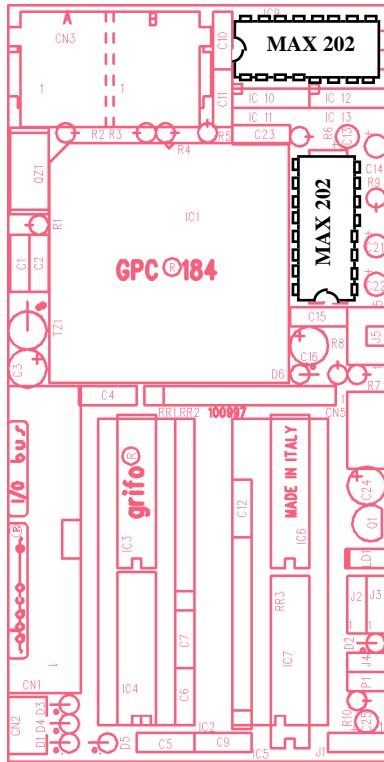
J5	=	posizione 2-3, 4-5	IC9	=	nessun componente
JS1, JS2	=	(*)	IC10	=	driver SN 75176
JS11	=	connesso	IC11	=	nessun componente
			IC12	=	driver SN 75176
			IC13	=	nessun componente

Lo stato del segnale /RTSB=/RTS0, gestito via software, consente di abilitare o disabilitare il trasmettitore come segue:

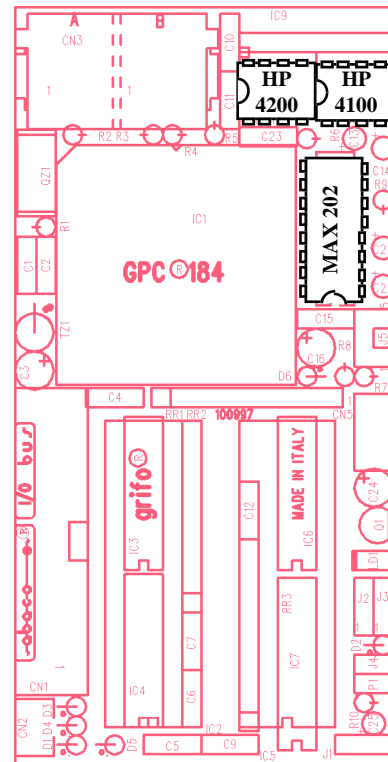
/RTS0 = livello basso = stato logico 0 -> trasmettitore attivo

/RTS0 = livello alto = stato logico 1 -> trasmettitore disattivo

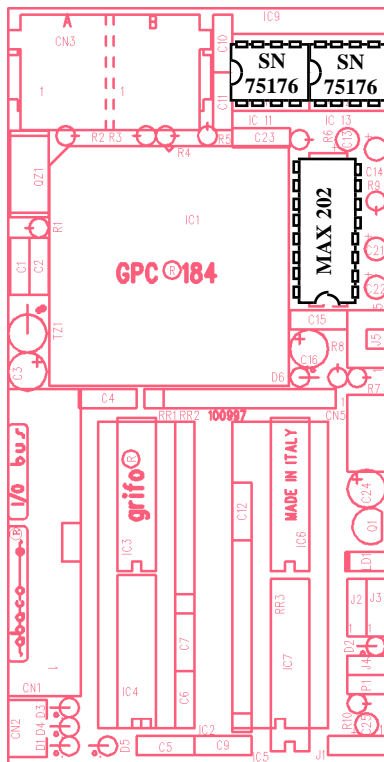
Per sistemi punto punto, la linea /RTS0 può essere mantenuta sempre bassa (trasmettitore sempre attivo), mentre per sistemi multipunto si deve attivare il trasmettitore solo in corrispondenza della trasmissione.



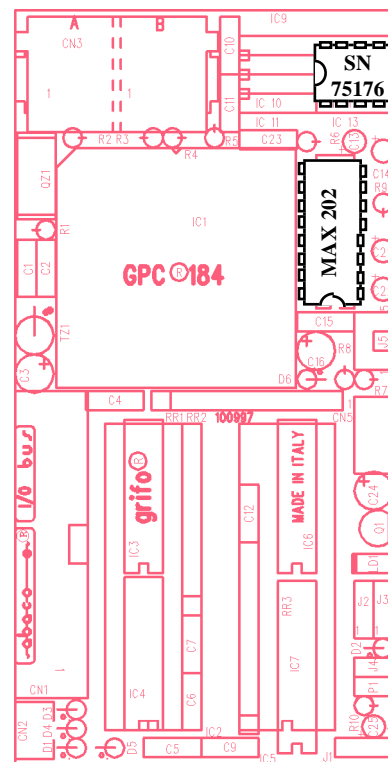
Seriale B = ASCI 0 in RS 232



Seriale B = ASCI 0 in current loop



Seriale B = ASCI 0 in RS 422



Seriale B = ASCI 0 in RS 485

FIGURA 26: DISPOSIZIONE DRIVER PER COMUNICAZIONE SERIALE

- LINEA SERIALE B=ASCI 0 SETTATA IN RS 485 (opzione .RS 485)

J5	=	posizione 1-2, 3-4	IC9	=	nessun componente
JS1, JS2	=	(*)	IC10	=	nessun componente
JS11	=	connesso	IC11	=	nessun componente
			IC12	=	driver SN 75176
			IC13	=	nessun componente

In questa modalità le linee da utilizzare sono i pin 4 e 5 di CN3B, che quindi diventano le linee di trasmissione o ricezione a seconda dello stato del segnale /RTSB=/RTS0, gestito via software, come segue:

/RTS0 = livello basso = stato logico 0 -> linea in trasmissione

/RTS0 = livello alto = stato logico 1 -> linea in ricezione

Questa comunicazione la si utilizza sia per connessioni punto punto che multipunto con un collegamento a 2 fili. Sempre in questa modalità é possibile ricevere quanto trasmesso, in modo da fornire al sistema la possibilità di verificare autonomamente la riuscita della trasmissione; infatti in caso di conflitti sulla linea, quanto trasmesso non viene ricevuto correttamente e viceversa.

- (*) Nel caso si utilizzi la linea seriale in RS 422 o RS 485, con i jumpers JS1 e JS2 é possibile connettere la circuiteria di terminazione e forzatura sulla linea . Tale circuiteria deve essere sempre presente nel caso di sistemi punto punto, mentre nel caso di sistemi multipunto, deve essere collegata solo sulle schede che risultano essere alla maggior distanza, ovvero ai capi della linea di comunicazione.

In fase di reset o power on, il segnale /RTS0 è mantenuto a livello logico baso di conseguenza in seguito ad una di queste fasi il driver RS 485 è in ricezione o il driver di trasmissione RS 422 è disattivo, in modo da eliminare eventuali conflittualità sulla linea di comunicazione.

- LINEA SERIALE A=ASCI 1

Il segnale di handshake di uscita RTSA non é gestibile via software ed é mantenuto sempre disattivo = -10 Vdc. Se questo stato é incompatibile con il sistema a cui si collega, provvedere ad effettuare un collegamento senza questo segnale.

Il jumper JS21 collega il segnale di handshake CTSA proveniente da CN3A tramite l'apposito driver RS 232, al relativo segnale /CTS1 della CPU. Visto che quest'ultimo segnale ha una duplice funzione e può quindi operare anche come RXS, se l'utente intende utilizzare la seriale sincrona riportata su CN5 dovrà aprire tale jumper.

Per ulteriori informazioni relative alla comunicazione seriale fare riferimento agli esempi di collegamento delle figure 10÷13.

RESET E WATCH DOG

La scheda **GPC® 184** è dotata di una circuiteria di watch dog molto efficiente e di facile gestione software. In particolare le caratteristiche di questa circuiteria sono le seguenti:

- funzionamento astabile;
- tempo d'intervento di circa 1,5 sec;
- attivazione via hardware;
- retrigger via software;

Si ricorda che nel funzionamento stabile una volta scaduto il tempo d'intervento la circuiteria si attiva, rimane attiva per il tempo di reset (circa 200 msec) e quindi si disattiva nuovamente. Con il jumper JS10 si seleziona se collegare la circuiteria di watch dog alla circuiteria di reset, ovvero si attiva, via hardware, la sua gestione. Per quanto riguarda l'operazione di retrigger della circuiteria di watch dog esterna, si faccia riferimento al paragrafo "WATCH DOG".

In corrispondenza dell'attivazione e successiva disattivazione del segnale di /RESET la scheda riprende l'esecuzione del programma salvato su IC2 (all'indirizzo 0000H), partendo da una condizione di azzeramento generale.

Si ricorda inoltre che il segnale di /RESET generato dalla scheda è riportato sul connettore CN1 e che tra le sorgenti di /RESET della **GPC® 184**, oltre all'eventuale circuiteria di watch dog, sono sempre presenti la circuiteria di power good ed il contatto P1.

Sui due pin del P1 si può collegare un contatto normalmente aperto (ad esempio un pulsante) ed una volta chiuso questo contatto (cortocircuitando i due pin), la circuiteria di /RESET viene attivata.

POWER FAILURE

In abbinamento alla circuiteria di power management gestita dalla CPU della **GPC® 184** è inoltre disponibile un'interessante circuiteria di power failure. Quest'ultima, con il jumper JS19, può essere collegata a due diversi interrupt (/NMI o /INT0).

La circuiteria si preoccupa di controllare la tensione fornita sul pin PFI di CN1 e quando questa scende al valore di soglia (**1,25 V**), provvede ad attivare l'uscita richiedendo l'attenzione della CPU nel caso in cui JS19 sia collegato in una delle due posizioni.

L'uso classico di questa circuiteria è quello di informare la scheda dell'imminente caduta della tensione di alimentazione, in modo da poter salvare le condizioni di stato necessarie. Il segnale PFI può convenientemente essere collegato alla tensione di alimentazione fornita alla scheda, tramite un partitore che fornisca la tensione di soglia con un sufficiente anticipo rispetto all'intervento del /RESET; tale anticipo deve essere sufficiente per la completa esecuzione della procedura di risposta all'interrupt. Si ricava perciò che la scelta del partitore deve tener conto della tensione di alimentazione usata, della carica residua della sezione alimentatrice e del tempo di risposta all'interrupt.

DESCRIZIONE SOFTWARE

Questa scheda ha la possibilità di usufruire di una ricca serie di strutture software che consentono di utilizzarne al meglio le caratteristiche. In generale la scheda può utilizzare tutte le risorse software disponibili per il processore montato, ovvero i numerosi pacchetti ideati per lo Z80 e Z180. Tra questi ricordiamo:

GET80

Completo programma di EDITOR, Comunicazione e gestione delle Memorie di Massa per le schede della famiglia 80. Questo programma, sviluppato dalla **grifo**[®], consente di operare in condizioni ottimali, tutte le volte che si deve usare il GDOS o la versione per FLASH EPROM FGDOS. Viene fornito in abbinamento all'acquisto di uno dei pacchetti citati e viene personalizzato con il nome ed i dati dell'acquirente. Una serie di comodi menù a tendina facilita l'uso del programma, il quale può funzionare anche in abbinamento ad un mouse. Il programma, oltre che girare in ambiente MS-DOS, gira tranquillamente anche sulle macchine MACINTOSH in abbinamento al programma SOFT-PC. Viene fornito su dischetti MS-DOS da 3"1/2 con relativa documentazione sul manuale GDOS 80.

GDOS 184

Tools di sviluppo completo per le schede della fam. 80. Viene fornito in abbinamento al programma GET80, per consentire un immediato e pieno utilizzo di questo potente strumento di sviluppo. Il GDOS può essere concettualmente diviso in due distinte strutture. Una struttura lavora essenzialmente su PC, mantenendo il collegamento con la seconda tramite la linea seriale. La seconda risiede in EPROM ed opera a bordo scheda. La parte a bordo scheda è essenzialmente un potente Sistema Operativo che si preoccupa di eseguire tutte quelle funzioni a più basso livello e nello stesso tempo consente di poter operare con linguaggi ad Alto Livello direttamente a bordo scheda. L'abbinamento delle due strutture fa sì che la scheda ed il PC si comportino come un'unica macchina. Infatti la scheda usa, come se fossero le proprie, le risorse del PC come la Memoria di Massa quali i Floppy-Disk, l'Hard-Disk; la Stampante ecc. Il tutto avviene in modo completamente trasparente per l'utente il quale usa questo tipo di Macchina Virtuale esattamente come è abituato ad adoperare il suo PC. Molto interessante è la compatibilità del GDOS con tutti i linguaggi ed i programmi CP/M. Questo significa che se l'utente ha dei programmi o dei linguaggi a cui sono legate delle applicazioni o delle sue specifiche conoscenze o altro, può utilizzare tutto quanto ha, virtualmente senza cambiamenti, in modo immediato sotto GDOS.

Il GDOS, oltre ai tipici drivers del PC, gestisce come RAM-Disk e ROM-Disk tutte le risorse di memoria della scheda, eccedenti i 64KBytes, là dove queste siano presenti. Questo significa che i dispositivi di RAM a bordo scheda, che spesso sono tamponati tramite batterie, possono essere gestite in modo diretto dai linguaggi ad alto livello, trattando comodamente come Files, le informazioni da archiviare o ricercare.

Il Tools viene fornito in EPROM, in abbinamento al disco in formato MS-DOS del GET80, alcuni esempi di uso e la relativa manualistica sul Sistema Operativo.

FGDOS 184

Caratteristiche analoghe al GDOS, con la differenza che è in grado di programmare e cancellare le FLASH-EPROM a bordo scheda, con i programmi generati dall'utente. In questo modo non è necessario un programmatore di EPROM esterno per congelare il programma. E' inoltre possibile, tramite un PC Portatile, intervenire direttamente a bordo macchina per cambiare il programma di gestione.

Il Tools viene fornito in FLASH-EPROM, in abbinamento al disco in formato MS-DOS del GET80, alcuni esempi di uso e la relativa manualistica sul Sistema Operativo.

xGDOS MCI 184

Versione del GDOS, o del FGDOS, in grado di gestire, ad alto livello, le schede di Memory-Card tipo PCMCIA. In abbinamento alla scheda MCI 64, il Sistema Operativo di bordo gestisce come RAM-Disk o ROM-Disk le Memory-Card. Questo consente di risolvere rapidamente, e senza crearsi problemi di gestione software, tutte quelle problematiche di raccolta dati che spesso si incontrano nella realizzazione di strutture di Data-Logging, gestendo questi dispositivi sempre con linguaggi ad alto livello.

Il Tools viene fornito in EPROM od in FLASH-EPROM, a secondo le necessità dell'utente, in abbinamento al disco in formato MS-DOS del GET80, alcuni esempi di uso e la relativa manualistica sul Sistema Operativo.

PASCAL 80

Completo e molto efficiente Compilatore PASCAL per la famiglia 80 di CPU. Ha delle caratteristiche operative analoghe a quelle del Turbo PASCAL Ver.3 della Borland, a cui si fa riferimento per quanto riguarda sia le caratteristiche che la manualistica. Il PASCAL-80 lavora in abbinamento ad una delle varie versioni di Sistema Operativo GDOS. Le modalità di Emulazione Terminale offerta dal programma GET80, supportano pienamente il tipico Editor a pieno schermo del PASCAL, compresa la gestione degli attributi. Sfruttando la possibilità di gestione di RAM-Disk e ROM-Disk, offerta dal GDOS, si possono sfruttare appieno le possibilità di OVERLAY del PASCAL per superare il limite dei 64KBytes di indirizzamento delle CPU della famiglia 80.

Il programma viene fornito in EPROM, assieme al sistema operativo della serie GDOS, e su dischetto MS-DOS in abbinamento alle note tecniche e ad una serie di esempi.

HI TECH C 80

Cross Compilatore C professionale della Hi-Tech Software. Questo compilatore é estremamente veloce e genera pochissimo codice. Questo risultato é ottenuto grazie a delle avanzate tecniche di ottimizzazione del codice generato, basato su tecniche di Intelligenza Artificiale che gli consentono di ottenere un codice compatto ed estremamente veloce. Il pacchetto comprende IDE, Compiler, Ottimizzatore del codice, Assembler, Linker, Remote debugger, ecc. Questo tools é Full ANSI/ISO Standard C ed é Full Library Source Code. Una volta fatto il porting del modulo di Remote-Debugger, consente di debuggare il software direttamente nell'hardware in sperimentazione. Questo tipo di specializzazione del Remote-Debugger é già disponibile, e viene fornito, per tutte le schede di CPU della grifo®. Il pacchetto software viene fornito su dischetti da 3"1/2 nel formato MS-DOS, completo di un esauriente manuale.

Questa versione supporta le CPU Z80, Z180, 84C011, 84C11, 84C013, 80C13, 80C015, 84C15, 64180, NCS800, Z181, Z182.

DDS MICRO C 85

E' un comodo pacchetto software, a basso costo, che tramite un completo I.D.E. permette di utilizzare un editor, un compilatore "C" (integer), un assembler, un linker e un remote debugger abbinato ad un monitor. Sono inclusi i sorgenti delle librerie, una serie di utility ed una ricca documentazione su dischetto da 3"1/2 nel formato MS-DOS.

NOICE: Potente struttura di debugger composta da un monitor debugger residente sulla scheda e da un apposito programma MS-DOS. I due programmi comunicano tramite una linea seriale in RS 232. Il NOICE include: debug a livello sorgente, disassemblatore, visualizzatore di file, editor e visualizzazione della memoria, numero di breakpoint illimitato, esecuzione di singole istruzioni indipendente dall'hardware, definizione di simboli, possibilità di eseguire file di comandi, gestione del back trace, help in linea, ecc.

RSD 184

Questo Tools é un Remote Symbolic Debugger che ha due modalit  operative. La prima é una modalit  di debugger in simulazione. La seconda é una modalit  di debugger in remoto. In questo ultimo caso si riesce ad effettuare il debugger del codice direttamente sulla scheda target. Tramite la linea seriale, si effettuato il Down Load del programma in HEX e della relativa tabella dei simboli. Fatto il caricamento, é possibile debuggare il codice in modo simbolico, in modalit  Step To Step, mettere break-point, ecc. con delle caratteristiche di comodit  simili a quelle di un In Circuit Emulator. Il programma RSD é in grado di supportare sia il codice Z80 che i codici aggiuntivi dello Z180. Le possibilit  di debugger del Tools RSD possono espletarsi sia in abbinamento ad un Macro Assembler come lo ZASM 80, che in abbinamento al C Compiler CC 80. Molto importante é la possibilit  di gestire dei Break-Point software, legati ad una molteplicit  di possibilit  ed un Break-Point hardware che fa capo al segnale di NMI.

Il Tools viene fornito in EPROM e su un dischetto MS-DOS con il relativo manuale tecnico.

ZASM 80

Macro Cross-Assembler in grado di lavorare su un qualsiasi PC in ambiente MS-DOS. E' in grado di supportare sia il mnemonico dello Z80 che i codici aggiuntivi presenti nello Z180. Il codice generato pu  essere debuggato sia in simulazione sul PC che direttamente sul target, in modalit  remota, utilizzando il comodo tools RSD. Lo ZASM é compatibile con il C Compiler CC-80, di cui assembla il risultato della compilazione.

Il programma viene fornito su dischetto MS-DOS e con il relativo manuale tecnico.

CC 80

Compilatore C, ANSI/ISO Standard, completo di Floating-Point, in grado di generare codice per le CPU della famiglia Z80 e Z180. Si abbina al Cross-Assembler ZASM-80 ed al Tools di Remote Symbolic Debugger, RSD.

Il programma viene fornito su dischetto MS-DOS e con il relativo manuale tecnico.

EMBEDDED PASCAL Z80 - Z180: Cross compilatore a basso costo per files sorgenti scritti in PASCAL, disponibile in ambiente WINDOWS 95 ed NT con un comodo IDE che mette a disposizione un editor, il compilatore, l'assemblatore, l'ottimizzatore ed il gestore di librerie. Comprende molti modelli di memoria, svariati tipi di dati ed i sorgenti delle librerie.

MAPPAGGI ED INDIRIZZAMENTI

INTRODUZIONE

In questo capitolo ci occuperemo di fornire tutte le informazioni relative all'utilizzo della scheda, dal punto di vista della programmazione via software. Tra queste si trovano le informazioni riguardanti il mappaggio delle memorie, delle periferiche, ecc.

MAPPAGGIO DELLE RISORSE DI BORDO

La gestione delle risorse della scheda è affidata ad una logica di controllo completamente realizzata con logica combinatoria. Essa si occupa del mappaggio delle memorie e di tutte le periferiche di bordo con una semplice gestione software e con un bassissimo assorbimento.

La logica di controllo è realizzata in modo da gestire separatamente il mappaggio delle memorie di bordo ed il mappaggio delle periferiche viste in Input/Output. Complessivamente la CPU Z180 indirizza direttamente 64K Byte di memoria e 256 indirizzi di I/O, quindi alla logica di controllo è assegnato il compito di allocare lo spazio logico d'indirizzamento delle memorie nello spazio fisico massimo di 1024K Byte. Questa gestione è effettuata via software tramite la programmazione della circuiteria di MMU con cui si può definire quali memorie utilizzare con una suddivisione in segmenti di dimensioni programmabili. Per quanto riguarda il mappaggio dell'I/O si deve invece ricordare che la logica di controllo provvede naturalmente a non utilizzare le locazioni riservate per le periferiche interne della CPU, in modo da evitare ogni problema di conflittualità.

Riassumendo i dispositivi mappati sulla scheda sono essenzialmente:

- **ABACO®** I/O BUS
- Fino a 512K Byte di EPROM o FLASH EPROM su IC 2
- Fino a 512K Byte di RAM su IC 5
- Jumper di configurazione J4
- Real Time Clock
- Circuiteria di Watch Dog

Questi occupano gli indirizzi riportati nei paragrafi seguenti e non possono essere riallocati in nessun altro indirizzo. In caso di specifiche esigenze in termini di mappaggio, contattare direttamente la **grifo®**.

MAPPAGGIO ABACO® I/O BUS

La logica di controllo della **GPC® 184** provvede anche alla gestione dell'**ABACO®** I/O BUS, definendo gli indirizzi in cui tale BUS viene allocato. In particolare, come si può notare dalla successiva tabella indirizzamento I/O, tale BUS occupa gli indirizzi 80H÷FFH. Un accesso in I/O in un qualsiasi indirizzo compreso in questo range abilita il segnale /IORQ e tutti gli altri segnali di controllo di CN1. Nel sottogruppo d'indirizzi 80H÷9FH viene inoltre abilitato anche il segnale /CS1, dedicato all'abilitazione decodificata di periferiche esterne.

MAPPAGGIO MEMORIE

Sulla scheda i 1024K Byte di memoria che possono essere montati sono così allocati:

Fino a 512K Byte di EPROM o 512K Byte di FLASH EPROM allocati nello spazio di memoria
Fino a 512K Byte di RAM allocati nello spazio di memoria

La **GPC® 184** può indirizzare direttamente un massimo di 64K Byte di memoria che coincide con lo spazio d'indirizzamento logico del microprocessore. Questa capacità di memoria, sulla scheda, può essere suddivisa in tre segmenti distinti ognuno dei quali può avere indirizzi d'inizio e dimensioni programmabili via software. La circuiteria di MMU, interna al microprocessore, si occupa appunto di dividere lo spazio direttamente indirizzato dalla CPU in questi tre segmenti e di allocarli in memoria nello spazio dei dispositivi fisici. Programmando la circuiteria di MMU tramite gli appositi registri, è quindi possibile indirizzare indirettamente, un'area notevolmente superiore a quella supportata direttamente dal microprocessore. Viene di seguito riportata una figura che illustra le possibili configurazioni dei dispositivi allocati nello spazio di memoria: per maggiori informazioni sulle modalità di gestione della MMU e sul significato esatto dei tre segmenti gestiti (Common Area 0, Common Area 1 e Bank Area) fare riferimento all'appendice B, mentre per una facile individuazione e configurazione dei dispositivi di memoria fare riferimento alle figure 9 e 25.

All'atto del power on o del reset l'MMU è programmata in modo da allocare i 64K della CPU all'inizio dello spazio fisico d'indirizzamento, quindi la scheda parte con l'esecuzione del codice posto all'indirizzo logico 00000H della EPROM o FLASH EPROM di IC 2.

Per quanto riguarda la configurazione della scheda per la selezione del tipo e delle dimensioni dei dispositivi di memoria installati sugli zoccoli IC2 ed IC5, utilizzare gli appositi jumpers come descritto nel paragrafo "SELEZIONE MEMORIE".

Alcuni pacchetti software, come il GDOS, si occupano autonomamente della gestione della circuiteria di MMU per allocare tutta la memoria fisicamente presente a bordo scheda nello spazio d'indirizzamento del microprocessore, senza interessare direttamente l'utente.

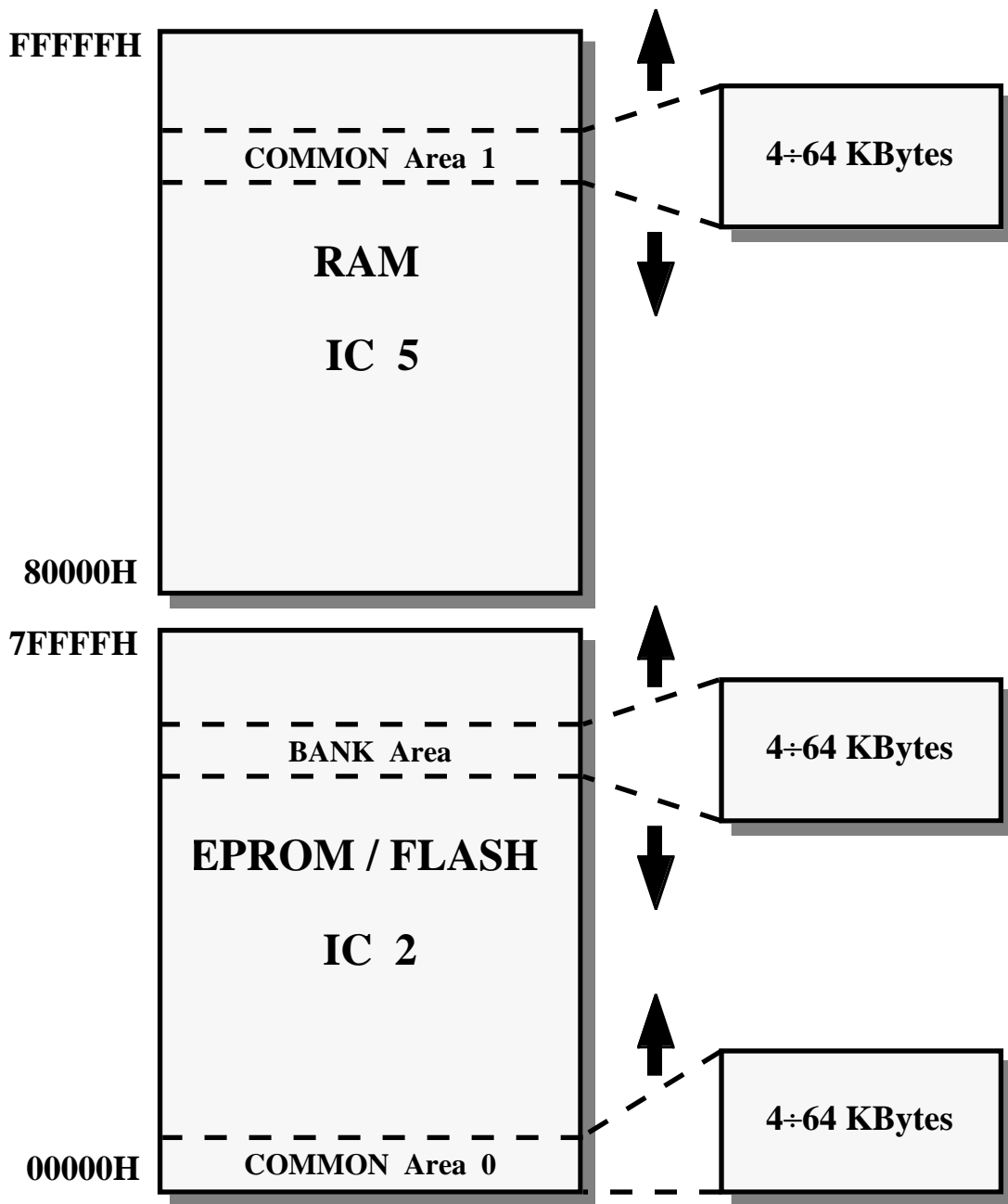


FIGURA 27: MAPPAGGIO DELLE MEMORIE

MAPPAGGIO I/O

Il mappaggio delle periferiche di bordo è gestito dalla logica di controllo che provvede ad indirizzare tali dispositivi all'interno dello spazio di I/O del microprocessore. Nella seguente tabella sono riportati i nomi, gli indirizzi, il tipo di accesso ed una breve descrizione dei registri dei dispositivi; per quanto riguarda la descrizione del loro significato, si faccia riferimento al capitolo successivo "DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO".

DISP.	REG.	IND.	R/W	SIGNIFICATO
ASCI	ASCI	00H÷09H	R/W	Registri , interni al microprocessore, di gestione delle linee seriali asincrone
CSI/O	CSIO	0AH÷0BH	R/W	Registri , interni al microprocessore, di gestione del Clocked Serial I/O Port
TIMER	TMR	0CH÷1FH	R/W	Registri, interni al microprocessore, di gestione dei Timer/Counter
DMA	DMA	20H÷32H	R/W	Registri, interni al microprocessore, di gestione delle linee di DMA
INTERRUPT	INT	33H÷35H	R/W	Registri, interni al microprocessore, di gestione degli interrupt
REFRESH	RCR	36H÷37H	R/W	Registri, interni al microprocessore, di gestione della circuiteria di Refresh
MMU	MMU	38H÷3AH	R/W	Registri, interni al microprocessore, di gestione del Memory Management Unit
I/O	ICR	3BH÷3FH	R/W	Registri di gestione delle periferiche interne al microprocessore
REAL TIME CLOCK	SEC1	40H	R/W	Registro dati per unità secondi
	SEC10	41H	R/W	Registro dati per decine secondi
	MIN1	42H	R/W	Registro dati per unità minuti
	MIN10	43H	R/W	Registro dati per decine minuti
	HOU1	44H	R/W	Registro dati per unità ore
	HOU10	45H	R/W	Registro dati per decine ore e AM/PM
	DAY1	46H	R/W	Registro dati per unità giorno
	DAY10	47H	R/W	Registro dati per decine giorno
	MON1	48H	R/W	Registro dati per unità mese
	MON10	49H	R/W	Registro dati per decine mese
	YEA1	4AH	R/W	Registro dati per unità anno
	YEA10	4BH	R/W	Registro dati per decine anno
	WEE	4CH	R/W	Registro dati per giorno della settimana
	REGD	4DH	R/W	Registro di controllo D
	REGE	4EH	R/W	Registro di controllo E
REGF	4FH	R/W	Registro di controllo F	
JUMPER C.	RUNDEB	60÷7FH	R	Registro acquisizione jumper configurazione
W. DOG	RWD	60÷7FH	R	Registro retrigger watch dog
ABACO®	/CS1	80H÷9FH	R/W	Indirizzi ABACO® I/O BUS con abilitazione del segnale /CS1
I/O BUS	I/O BUS	50H÷EFH	R/W	Indirizzi ABACO® I/O BUS

FIGURA 28: TABELLA INDIRIZZAMENTO I/O

DESCRIZIONE SOFTWARE DELLE PERIFERICHE DI BORDO

Nel paragrafo precedente sono stati riportati gli indirizzi di allocazione di tutte le periferiche e di seguito viene riportata una descrizione dettagliata della funzione e del significato dei relativi registri (al fine di comprendere le successive informazioni, fare sempre riferimento alla tabella di indirizzamento I/O). Qualora la documentazione riportata fosse insufficiente fare riferimento direttamente alla documentazione tecnica della casa costruttrice del componente. In questo paragrafo inoltre non vengono descritte le sezioni che fanno parte del microprocessore; per quanto riguarda la programmazione di quest'ultime si faccia riferimento all'appendice B di questo manuale. Nei paragrafi successivi si usano le indicazioni D0÷D7 per fare riferimento ai bits della combinazione utilizzata nelle operazioni di I/O.

REAL TIME CLOCK

Questa periferica è vista in 16 locazioni di I/O consecutive di cui 3 di stato e le rimanenti 13 per i dati. I registri dati sono utilizzati sia per operazioni di lettura (dell'orario attuale) che di scrittura (per l'inizializzazione dell'orologio) così come i registri di stato i quali sono utilizzati in scrittura (per la programmazione del modo di funzionamento dell'orologio) ed in lettura (per determinare lo stato dell'orologio). Per quanto riguarda il significato dei registri dati vale la corrispondenza:

SEC1	- Unita' dei secondi	- 4 bit meno significativi:	SEC1.3÷SEC.0
SEC10	- Decine dei secondi	- 3 bit meno significativi:	SEC10.2÷SEC10.0
MIN1	- Unita' dei minuti	- 4 bit meno significativi:	MIN1.3÷MIN1.0
MIN10	- Decine dei minuti	- 3 bit meno significativi:	MIN10.2÷MIN10.0
HOU1	- Unita' delle ore	- 4 bit meno significativi:	HOU1.3÷HOU1.0
HOU10	- Decine delle ore	- 2 bit meno significativi:	HOU10.1÷HOU10.0
Il terzo bit di tale registro, H10.2, indica l'AM/PM			
DAY1	- Unita' del giorno	- 4 bit meno significativi:	DAY1.3÷DAY1.0
DAY10	- Decine del giorno	- 2 bit meno significativi:	DAY10.1÷DAY10.0
MON1	- Unita' del mese	- 4 bit meno significativi:	MON1.3÷MON1.0
MON10	- Decine del mese	- 1 bit meno significativo:	MON10.0
YEA1	- Unita' dell' anno	- 4 bit meno significativi:	YEA1.3÷YEA1.0
YEA10	- Decine dell' anno	- 4 bit meno significativi:	YEA10.3÷YEA10.0
WEE	- Giorno della settimana	- 3 bit meno significativi:	WEE.2÷WEE.0

Per quest'ultimo registro vale la corrispondenza:

WEE.2	WEE.1	WEE.0	Giorno della settimana
0	0	0	Domenica
0	0	1	Lunedì
0	1	0	Martedì
0	1	1	Mercoledì
1	0	0	Giovedì
1	0	1	Venerdì
1	1	0	Sabato

I tre registri di controllo sono invece utilizzati come segue:

D7 D6 D5 D4 D3 D2 D1 D0

REGD = NU NU NU NU 30S IF B H

dove:

NU = Non usato

30S = Se attivo (1) permette di effettuare una correzione di 30 secondi dell'orario. Una volta settato i secondi del RTC vengono azzerati ed i minuti incrementati se il precedente valore dei secondi era superiore o uguale a 30.

IF = Gestisce lo stato d'interrupt del RTC. In lettura riporta lo stato attuale d'interrupt (1=attivo e viceversa), mentre se resettato con una scrittura determina la fine interrupt, quando il RTC lavora in interrupt mode.

B = Indica se possono essere effettuate operazioni di lettura/scrittura dei registri:
1 -> operazioni impossibili e viceversa.

H = Se attivo (1) effettua la memorizzazione dell'orario fissato.

D7 D6 D5 D4 D3 D2 D1 D0

REG E = NU NU NU NU T1 T0 I M

dove:

NU = Non usato.

T1 T0 = Determinano la durata del periodo di interrupt

0 0 -> 1/64 secondo

0 1 -> 1 secondo

1 0 -> 1 minuto

1 1 -> 1 ora

I = Determina modalità di gestione interrupt: se settato seleziona l'interrupt mode in cui l'interrupt si attiva allo scadere del periodo programmato e si disattiva con un reset del bit IF del registro D; se resettato seleziona lo standard mode in cui l'interrupt si attiva allo scadere del periodo programmato e si disattiva autonomamente dopo 7,8 msec.

M = Se settato disabilita il pin di interrupt del RTC e viceversa.

D7 D6 D5 D4 D3 D2 D1 D0

REG F = NU NU NU NU T 24/12 S R

dove:

NU = Non usato.

T = Stabilisce da quale contatore interno prelevare il segnale di conteggio:

1 -> contatore principale (conteggio veloce per test);

0 -> 15° contatore (conteggio normale).

24/12 = Stabilisce il modo di conteggio delle ore:

1 -> 0÷23;

0 -> 0÷11 con AM/PM.

S = Se settato provoca l'arresto dell'avanzamento dell'orologio fino alla successiva abilitazione.

R = Se settato (1) provoca il reset di tutti i contatori interni.

WATCH DOG

Il retrigger della circuiteria di watch dog presente sulla **GPC® 184**, avviene tramite una semplice operazione di input al registro RWD. Affinché la circuiteria di watch dog non intervenga, è indispensabile retriggerarla ad intervalli regolari di durata inferiore al tempo d'intervento. Se ciò non avviene e tramite il jumper JS14 la circuiteria è connessa alla sezione di reset, una volta scaduto il tempo d'intervento la scheda viene resettata. Il tempo d'intervento nella condizione di default è mediamente di circa 1,5 sec.

JUMPER DI CONFIGURAZIONE

Lo stato del jumper di configurazione J4 può essere acquisito via software, effettuando una semplice operazione di input all'indirizzo di allocazione del registro RUNDEB ed esaminando il bit D7. L'acquisizione è in logica negata, ovvero il jumper connesso fornisce lo stato logico 0 al corrispondente bit, mentre il jumper non connesso fornisce lo stato logico 1.

Tale jumper svolge la funzione di selettore delle modalità RUN (non connesso) o DEBUG (connesso), caratteristica di alcuni pacchetti software della **grifo®**.

PERIFERICHE DELLA CPU

La descrizione dei registri e del relativo significato di tutte le periferiche interne della CPU (ASCI, CSI/O, TIMER, DMA, INTERRUPT, REFRESH, MMU, I/O) è disponibile nell'appendice B. Qualora queste informazioni fossero ancora insufficienti, fare riferimento alla documentazione tecnica della casa costruttrice.

SCHEDE ESTERNE

La scheda **GPC® 184** si interfaccia a buona parte dei moduli della serie BLOCK e di interfaccia utente. Le risorse di bordo possono essere facilmente aumentate collegando la **GPC® 184** alle numerose schede periferiche del carteggio **grifo®** tramite l'**ABACO®** I/O BUS. Anche schede in formato Europa con BUS **ABACO®** possono essere collegate, sfruttando gli appositi mother boards. A titolo di esempio ne riportiamo un elenco con una breve descrizione delle caratteristiche di massima; per maggiori informazioni richiedere la documentazione specifica:

QTP G26

Quick Terminal Panel 26 tasti con LCD grafico

Interfaccia operatore provvista di display grafico da 240x128 pixel retroilluminato a LEDs; tastiera a membrana da 26 tasti di cui 6 configurabili dall'utente; 16 LEDs di stato; alimentatore a bordo scheda; interdaccia seriale in RS 232, RS 422-485 o current loop; linea seriale ausiliaria in RS 232. Tasti ed etichette personalizzabili dall'utente tramite serigrafie da inserire in apposite tasche; contenitore metallico e plastico; EEPROM di set up; 256K EPROM o FLASH; Real Time Clock; 128K RAM; buzzer. Firmware di gestione che svolge funzione di terminale con primitive grafiche.

ADC 812

Analog to Digital Converter, 8 channels, 12 bits multi-range

Modulo periferico della serie 4 (100x50 mm); A/D converter DAS (Data Acquisition System) multi-range a 8 canali da 12 bit; Track-Hold; tempo di conversione 6 μ s; range dei segnali d'ingresso ± 10 , ± 5 , $+10$, $+5$ Vdc oppure $0\div 20$, $4\div 20$ mA; interfaccia per **ABACO®** I/O BUS; possibilità di montaggio diretto su guide Ω di tipo DIN 46277-1 e 3.

DAC 212

Digital to Analog Converter 12 bits, multi-range

Modulo periferico della serie 4 (100x50 mm); D/A converter multi-range a 2 canali da 12 bit; range dei segnali d'uscita ± 10 o $0/+10$ Vdc; interfaccia per **ABACO®** I/O BUS; possibilità di montaggio diretto su guide Ω di tipo DIN 46277-1 e 3.

CAN 14

Control Area Network, 1 channel, galvanically insulated

Modulo periferico della serie 4 (100x50 mm); UART CAN SJA1000; 1 canale seriale galvanicamente isolato; interfaccia per **ABACO®** I/O BUS; possibilità di montaggio diretto su guide Ω di tipo DIN 46277-1 e 3.

IBC 01

Interface Block Communication

Scheda di conversioni per comunicazioni seriali. 2 linee RS 232; 1 linea RS 422-485; 1 linea in fibra ottica; interfaccia DTE/DCE selezionabile; attacco rapido per guide tipo DIN 46277-1 e 3.

ZBR xxx

Zippered BLOCK Relays xx Input + xx Output

Periferica per xx Input optoisolati e visualizzati tipo NPN; xx relé da 3A con MOV; connettori a morsettiera per ingressi optoisolati e uscite; connettore normalizzato **ABACO®** I/O BUS; LEDs di visualizzazione; sezione alimentatrice a bordo; attacco rapido per guide Ω . Le possibili configurazioni in termini di numero di I/O sono: xxx=324 con 32 In e 24 Out; xxx=246 con 24 In e 16 Out; xxx=168 con 16 In e 8 Out; xxx=84 con 8 In e 4 Out.

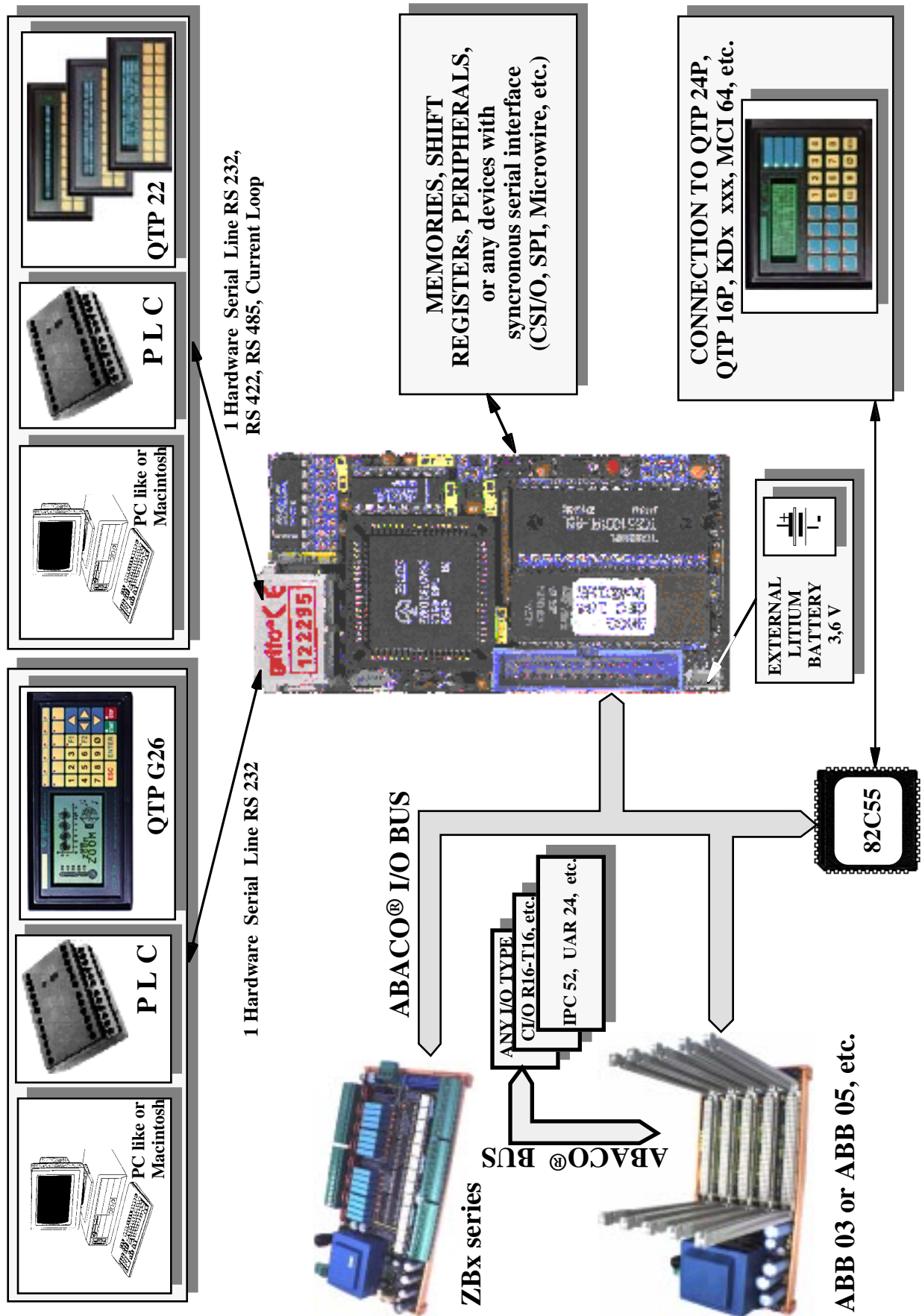


FIGURA 29: SCHEMA DELLE POSSIBILI CONNESSIONI

ZBT xxx

Zipped BLOCK Transistors xx Input + xx Output

Periferica per xy Input optoisolati e visualizzati tipo NPN; yz darlington da 3A con diodo di ricircolo; connettori a morsettiera per ingressi optoisolati e uscite; connettore normalizzato **ABACO**® I/O BUS; 61 LEDs di visualizzazione; sezione alimentatrice a bordo; attacco rapido per guide Ω . Le possibili configurazioni in termini di numero di I/O sono: xxx=324 con 32 In e 24 Out; xxx=246 con 24 In e 16 Out; xxx=168 con 16 In e 8 Out; xxx=84 con 8 In e 4 Out.

ABB 05

Abaco® Block BUS 5 slots

Mother board **ABACO**® da 5 slots; passo 4 TE; guidaschede; connettori normalizzati di alimentazione; tasto di reset; LEDs per alimentazioni; interfaccia **ABACO**® I/O BUS; sezione alimentatrice per +5 Vdc; sezione alimentatrice per +V Opto; sezioni alimentatrici galvanicamente isolate; tre tipi di alimentazione: da rete, bassa tensione o stabilizzata. Attacco rapido per guide Ω .

ABB 03

Abaco® Block BUS 3 slots

Mother board **ABACO**® da 3 slots; passo 4 TE; guidaschede; connettori normalizzati di alimentazione; tasto di reset; LEDs per alimentazioni; interfaccia **ABACO**® I/O BUS. Attacco rapido per guide Ω .

FBC 110

Flat Block Contact 10 vie

Interfaccia tra 1 connettore a perforazione di isolante (scatolino da 10 vie maschio) e la filatura da campo (morsettiera a rapida estrazione). Attacco rapido per guide tipo DIN 46277-1 e 3.

BIBLIOGRAFIA

E' riportato di seguito, un elenco di manuali e note tecniche, a cui l'utente può fare riferimento per avere maggiori chiarimenti, sui vari componenti montati a bordo della scheda **GPC**® 184.

Manuale TEXAS INSTRUMENTS:	<i>The TTL Data Book - SN54/74 Families</i>
Manuale TEXAS INSTRUMENTS:	<i>RS-422 and RS-485 Interface Circuits</i>
Manuale NEC:	<i>Memory Products</i>
Manuale HEWLETT PACKARD:	<i>Optoelectronics Designer's Catalog</i>
Manuale MAXIM:	<i>New Releases Data Book - Volume IV</i>
Manuale MAXIM:	<i>New Releases Data Book - Volume V</i>
Manuale SEIKO EPSON:	<i>REAL TIME CLOCK MODULE RTC-72421</i> <i>Applicatio manual</i>
Manuale ZILOG:	<i>Z80180 Z180 MPU Technical Manual</i>

Per ulteriori informazioni ed aggiornamenti si possono visitare anche i siti internet delle case costruttrici sopra riportate.

APPENDICE A: MONTAGGIO MECCANICO DELLA SCHEDA

La **GPC® 184** può essere interfacciata al mondo esterno in due modalità; il primo é il cosiddetto montaggio in piggy back, con cui si monta la scheda al di sopra del proprio hardware, sfruttando il prolungamento dei pin dei connettori CN1 e CN5. Questi infatti si estendono nel lato saldature per circa 7 mm, permettendo quindi un comodo inserimento su connettori femmina, del tipo strip a passo 2.54 mm.

La seconda modalità di connessione, invece, consiste nell'inserire la scheda, eventualmente abbinata ad una scheda periferica (ad esempio un modulo tipo **ZBR** o **ZBT**), su una guida Weidmuller tipo RS/100 (codice 414487), per il montaggio su barre Ω del tipo DIN 46277-1 e 3; questo contenitore plastico può essere ordinato alla **grifo®** come opzione **WM.xxx**, dove xxx indica, in mm, la lunghezza desiderata.

In questo caso il collegamento elettrico tra la **GPC® 884** e la scheda periferica avviene tramite un flat cable, che deve essere più corto possibile.

Nelle figure seguenti sono riportate le quote meccaniche, relative alla posizioni dei connettori ed alcune immagini riguardanti queste due modalità di connessione.

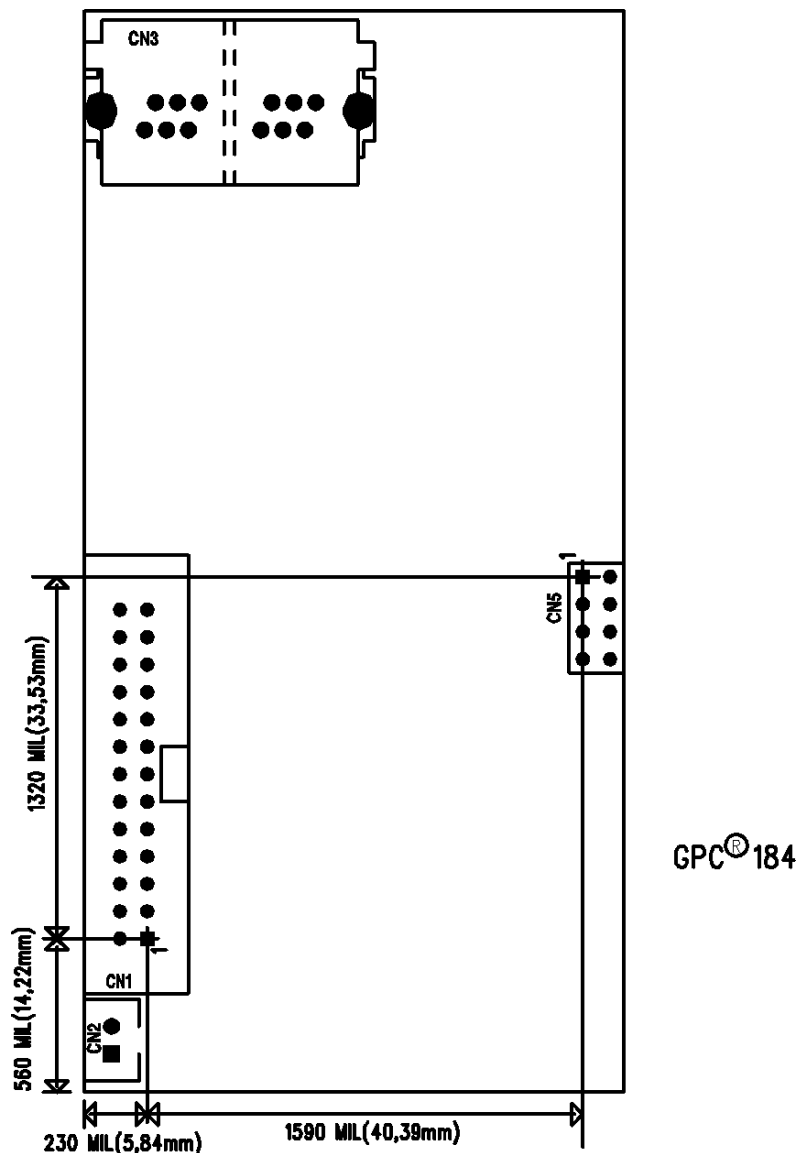


FIGURA A1: QUOTE PER MONTAGGIO IN PIGGY BACK

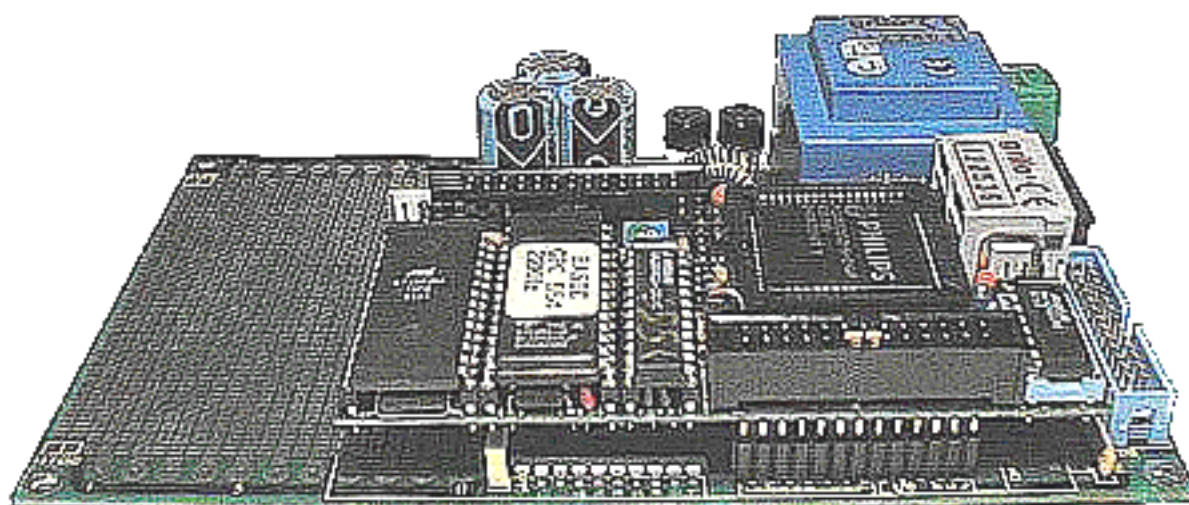
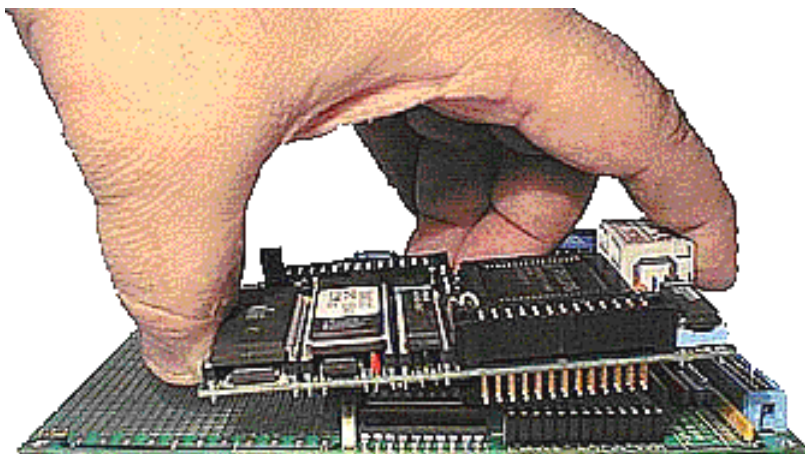


FIGURA A2: MONTAGGIO IN PIGGY BACK

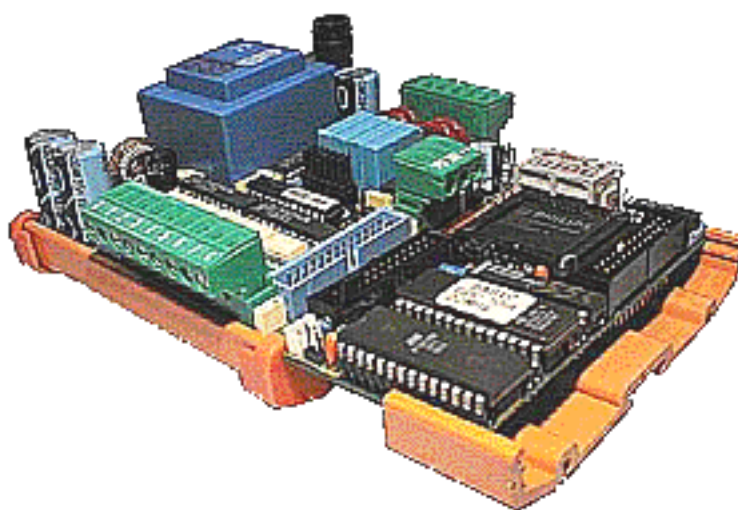


FIGURA A3: MONTAGGIO SU GUIDA WEIDMULLER

APPENDICE B: DESCRIZIONE COMPONENTI DI BORDO

Z80180/Z8S180/Z8L180
Enhanced Z180 Microprocessor

Zilog

Notes: All signals with a preceding front slash, "/" are active Low, for example, B/W (WORD is active Low); /B/W (BYTE is active Low, only). Alternatively, an over slash may be used to signify active Low, for example WR

Power connections follow conventional descriptions below:

Connection	Circuit	Device
Power	V _{CC}	V _{DD}
Ground	GND	V _{SS}

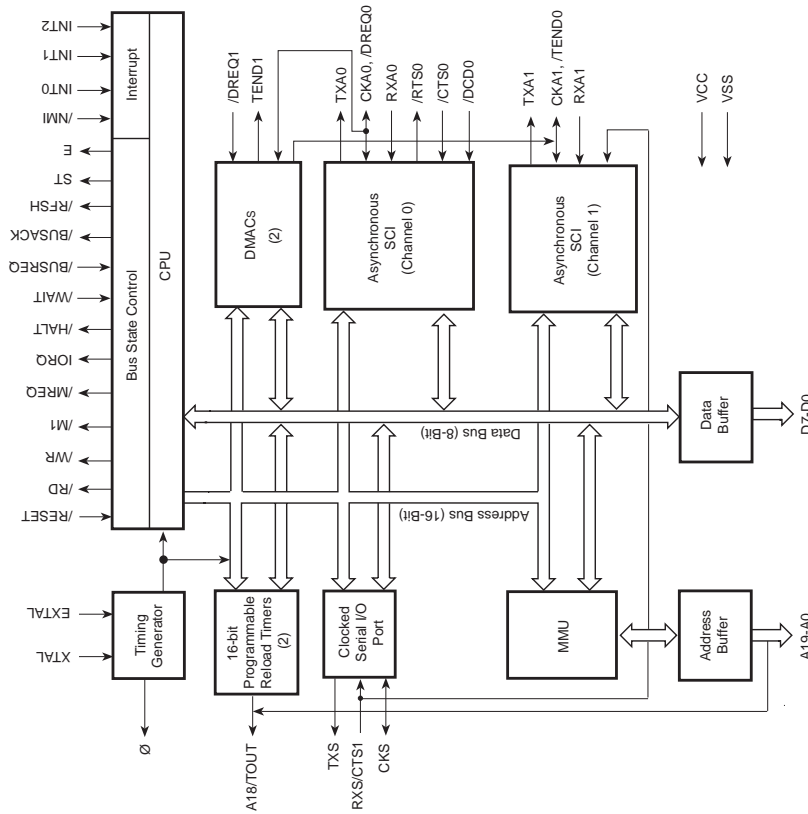


Figure 1. Z80180/Z8S180/Z8L180 Functional Block Diagram

PRELIMINARY PRODUCT SPECIFICATION

Z80180/Z8S180/
Z8L180 SL1919
ENHANCED Z180 MICROPROCESSOR

- FEATURES**
- Code Compatible with Zilog Z80® CPU
 - Extended Instructions
 - Two Chain-Linked DMA Channels
 - Low Power-Down Modes
 - On-Chip Interrupt Controllers
 - Three On-Chip Wait-State Generators
 - On-Chip Oscillator/Generator
 - Expanded MMU Addressing (up to 1 MB)
 - Clocked Serial I/O Port
 - Two 16-Bit Counter/Timers
 - Two Enhanced UARTs (up to 512 Kbps)
 - Clock Speeds: 6, 8, 10, 20, 33 MHz
 - Operating Range: 5V (3.3V @ 20 MHz)
 - Operating Temperature Range: 0°C to +70°C
 - 40°C to +85°C Extended Temperature Range
 - Three Packaging Styles
 - 68-Pin PLCC
 - 64-Pin DIP
 - 80-Pin QFP

GENERAL DESCRIPTION

The enhanced Z80180/Z8S180/Z8L180™ significantly improves on the previous Z80180 models while still providing full backward compatibility with existing Zilog Z80 devices. The Z80180/Z8S180/Z8L180 now offers faster execution speeds, power saving modes, and EMI noise reduction.

This enhanced Z180 design also incorporates additional feature enhancements to the ASCIs, DMAs, and L_{CC} STANDBY Mode power consumption. With the addition of "ESCC-like" Baud Rate Generators (BRGs), the two ASCIs now have the flexibility and capability to transfer data asynchronously at rates of up to 512 Kbps. In addition, the ASCII receiver has added a 4-byte First In First Out (FIFO), which can be used to buffer incoming data to reduce the incidence of overrun errors. The DMAs have been modified to allow for a "chain-linking" of the two DMA channels when set to take their DMA requests from the same peripherals device. This feature allows for non-stop DMA operation between the two DMA channels, reducing the amount of CPU intervention (Figure 1).

Not only does the Z80180/Z8S180/Z8L180 consume less power during normal operations than the previous mode it has also been designed with three modes intended to further reduce the power consumption. Zilog reduced L_{CC} power consumption during STANDBY Mode to a minimum of 10 µA by stopping the external oscillators and internal clock. The SLEEP mode reduces power by placing the CPU into a "stopped" state, thereby consuming less current while the on-chip I/O device is still operating. The SYSTEM STOP mode places both the CPU and the on-chip peripherals into a "stopped" mode, thereby reducing power consumption even further.

A new clock doubler feature has been implemented in the Z80180/Z8S180/Z8L180 device that allows the programmer to double the internal clock from that of the external clock. This provides a systems cost savings by allowing the use of lower cost, lower frequency crystals instead of the higher cost, and higher speed oscillators.

The Enhanced Z180 is housed in 80-pin QFP, 68-pin PLCC, and 64-pin DIP packages.



HALT and Low-Power Operating Modes. The Z80180/Z8S180/Z8L180 can operate in seven modes with respect to activity and power consumption:

- Normal Operation
- HALT Mode
- IOSTOP Mode
- SLEEP Mode
- SYSTEM STOP Mode
- IDLE Mode
- STANDBY Mode (with or without QUICK RECOVERY)

Normal Operation. The Z80180/Z8S180/Z8L180 processor is fetching and running a program. All enabled functions and portions of the device are active, and the HALT pin is High.

The Z80180/Z8S180/Z8L180 leaves HALT mode in response to a Low on RESET, on to an interrupt from an enabled on-chip source, an external request on NMI, or an enabled external request on INTO, INT1, or INT2. In case of an interrupt, the return address will be the instruction following the HALT instruction; at that point the program can either branch back to the HALT instruction to wait for another interrupt, or can examine the new state of the system/application and respond appropriately.

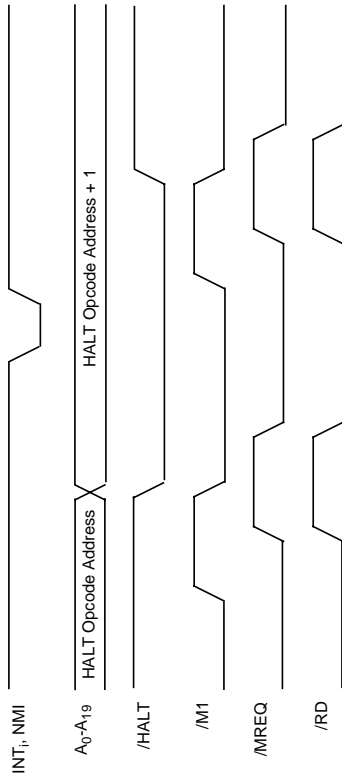


Figure 13. HALT Timing

SLEEP Mode. This mode is entered by keeping the IOSTOP bit (ICR5) bits 3 and 6 of the CPU Control Register (CCR3, CCR6) all zero and executing the SLP instruction. The oscillator and PHI output continue operating, but are blocked from the CPU core and DMA channels to reduce power consumption. DRAM refresh stops but interrupts and granting to external master can occur. Except when the bus is granted to an external master, A19-0 and all control signals except /HALT are maintained High. /HALT is Low. I/O operations continue as before the SLP instruction, except for the DMA channels.

The Z80180/Z8S180/Z8L180 leaves SLEEP mode in response to a low on /RESET, an interrupt request from an on-chip source, an external request on /NMI, or an external request on /INT0, 1, or 2.

If an interrupt source is individually disabled, it cannot bring the Z80180/Z8S180/Z8L180 out of SLEEP mode. If an interrupt source is globally enabled, and the IEF bit is 1 so that interrupts are globally enabled (by an EI instruction), the highest priority active interrupt will occur, with the return address being the instruction after the SLP instruction. If an interrupt source is individually enabled, but the IEF bit is 0 so that interrupts are globally disabled (by a DI instruction), the Z80180/Z8S180/Z8L180 leaves

SLEEP mode by simply executing the following instruction(s).

This provides a technique for synchronization with high-speed external events without incurring the latency imposed by an interrupt response sequence. Figure 14 shows the timing for exiting SLEEP mode due to an interrupt request. Note that the Z80180/Z8S180/Z8L180 takes about 1.5 clocks to restart.

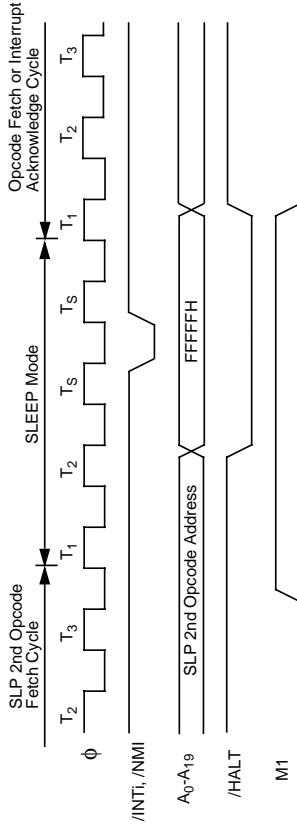


Figure 14. SLEEP Timing

IOSTOP Mode. IOSTOP mode is entered by setting the IOSTOP bit of the I/O Control Register (ICR) to 1. In this case, on-chip I/O (ASCI, CSIO, PRT) stops operating. However, the CPU continues to operate. Recovery from IOSTOP mode is by resetting the IOSTOP bit in ICR to 0.

SYSTEM STOP Mode. SYSTEM STOP mode is the combination of SLEEP and IOSTOP modes. SYSTEM STOP mode is entered by setting the IOSTOP bit in ICR to 1 followed by execution of the SLP instruction. In this mode, on-chip I/O and CPU stop operating, reducing power consumption, but the PHI output continues to operate. Recovery from SYSTEM STOP mode is the same as recovery from SLEEP mode except that internal I/O sources (disabled by IOSTOP) cannot generate a recovery interrupt.

IDLE Mode. Software can put the Z80180/Z8S180/Z8L180 into this mode by setting the IOSTOP bit (ICR5) to 1, CCR6 to 0, CCR3 to 1 and executing the SLP instruction. The oscillator keeps operating but its output is blocked to all circuitry including the PHI pin. DRAM refresh and all internal devices stop, but external interrupts can occur. Bus granting to external masters can occur if the BREST bit in the CPU control Register (CCR5) was set to 1 before IDLE mode was entered.

The Z80180/Z8S180/Z8L180 leaves IDLE mode in response to a Low on RESET, an external interrupt request on NMI, or an external interrupt request on /INT0, /INT1 or /INT2 that is enabled in the INT/TRAP Control Register. As previously described for SLEEP mode, when the Z80180/Z8S180/Z8L180 leaves IDLE mode due to an NMI, or due to an enabled external interrupt request when the IEF flag is 1 due to an EI instruction, it starts by performing the interrupt with the return address being that of the instruction after the SLP instruction.

If an external interrupt enables the INT/TRAP control register while the IEF1 bit is 0, Z80180/Z8S180/Z8L180 leaves IDLE mode; specifically, the processor restarts by executing the instructions following the SLP instruction.



Figure 15 shows the timing for exiting IDLE mode due to an interrupt request. Note that the OPCODE Fetch or Interrupt Acknowledge Cycle takes about 9.5 clocks to restart.

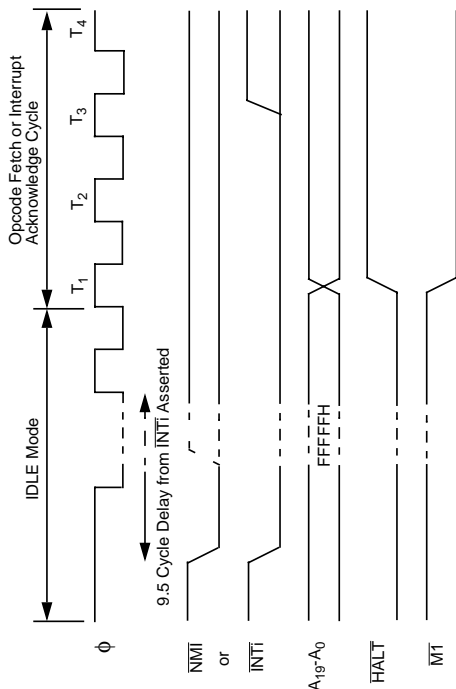


Figure 15. Z80180/Z8S180/Z8L180 IDLE Mode Exit due to External Interrupt

While the Z80180/Z8S180/Z8L180 is in IDLE mode, it will grant the bus to an external master if the BREXT bit (CCR5) is 1. Figure 16 shows the timing for this sequence. Note that the part takes 8 clock cycles longer to respond to the Bus Request than in normal operation.

After the external master negates the Bus Request, the Z80180/Z8S180/Z8L180 disables the PHI clock and remains in IDLE mode.

Figure 32. ASCII Block Diagram

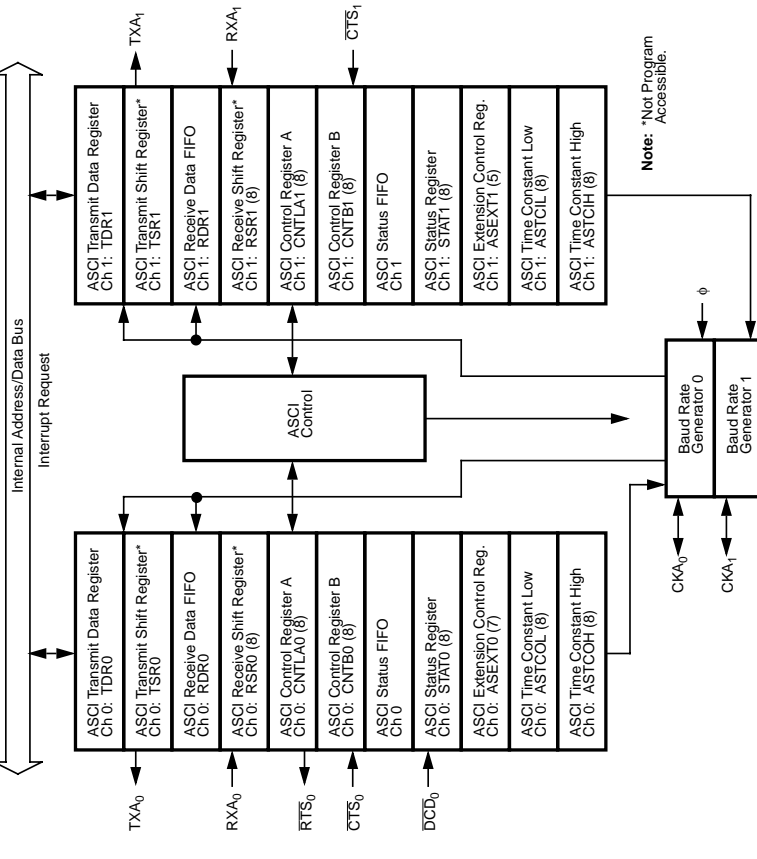


Figure 32. ASCII Block Diagram



The following paragraphs explain the various functions of the ASCII registers.

ASCII Transmit Register 0. When the ASCII Transmit Register receives data from the ASCII Transmit Data Register (TDR), the data is shifted out to the TxA pin. When transmission is completed, the next byte (if available) is automatically loaded from TDR into TSR and the next transmission starts. If no data is available for transmission, TSR IDLEs by outputting a continuous high level. This register is not program accessible.

ASCII Transmit Data Register 0,1 (TDR0, 1; IO address = 06H, 07H). Data written to the ASCII Transmit Data Register is transferred to the TSR as soon as TSR is empty. Data can be written while TSR is shifting out the previous byte of data. Thus, the ASCII transmitter is double buffered.

Data can be written into and read from the ASCII Transmit Data Register. If data is read from the ASCII Transmit Data

Zilo;

Register, the ASCII data transmit operation will not be affected by this read operation.

ASCII Receive Shift Register 0,1 (RSR0,1). This register receives data shifted in on the RxA pin. When full, data is automatically transferred to the ASCII Receive Data Register (RDR) if it is empty. If RSR is not empty when the next incoming data byte is shifted in, an overrun error occurs. This register is not program accessible.

ASCII Receive Data FIFO 0,1 (RDR0, 1; IO Address = 08H, 09H). The ASCII Receive Data Register is a read-only register. When a complete incoming data byte is assembled in RSR, it is automatically transferred to the 4-character Receive Data First-In First-Out (FIFO) memory. The oldest character in the FIFO (if any) can be read from the Receive Data Register (RDR). The next incoming data byte can be shifted into RSR while the FIFO is full. Thus, the ASCII receiver is well buffered.

ASCII STATUS FIFO

This 4 entry FIFO contains Parity Error, Framing Error, Rx Overrun, and Break status bits associated with each character in the receive data FIFO. The status of the oldest character (if any) can be read from the ASCII status registers as described below.

ASCII CHANNEL CONTROL REGISTER A

ASCII Control Register A 0 (CNTLA0; I/O Address = 00H)									
Bit	7	6	5	4	3	2	1	0	
	MPE	RE	TE	RTS0	MPBR/EFER	MOD2	MOD1	MOD0	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ASCII Control Register A 1 (CNTLA1; I/O Address = 01H)									
Bit	7	6	5	4	3	2	1	0	
	MPE	RE	TE	CKA1D	MPBR/EFER	MOD2	MOD1	MOD0	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Figure 33. ASCII Channel Control Register A

MPE: Multi-Processor Mode Enable (bit 7). The ASCII has a multiprocessor communication mode that utilizes an extra data bit for selective communication when a number of processors share a common serial bus. Multiprocessor data format is selected when the MP bit in CNTLB is set to 1. If multiprocessor mode is not selected (MP bit in CNTLB = 0), MPE has no effect. If multiprocessor mode is selected, MPE enables or disables the "wake-up" feature as follows. If MBE is set to 1, only received bytes in which the MPB (multiprocessor bit) = 1 can affect the RDRF and error flags. Effectively, other bytes (with MPB = 0) are "ignored" by the ASCII. If MPE is reset to 0, all bytes, regardless of the state of the MPB data bit, affect the RDR and error flags. MPE is cleared to 0 during RESET.

RE: Receiver Enable (bit 6). When RE is set to 1, the ASCII transmitter is enabled. When RE is reset to 0, the transmitter is disabled and any transmit operation in progress is interrupted. However, the TDRE flag is not reset and the previous contents of TDRE are held. TE is cleared to 0 in IOSTOP mode during RESET.

TE: Transmitter Enable (bit 5). When TE is set to 1, the ASCII receiver is enabled. When TE is reset to 0, the transmitter is disabled and any transmit operation in progress is interrupted. However, the TDRE flag is not reset and the previous contents of TDRE are held. TE is cleared to 0 in IOSTOP mode during RESET.

RTS0: Request to Send Channel 0 (bit 4 in CNTLA1 only). If bit 4 of the System Configuration Register is 0, the RTS0/TXS pin has the RTS0 function. RTS0 allows the ASCII to control (start/stop) another communication device's transmission (for example, by connecting to that device's CTS input). RTS0 is essentially a 1 bit output port having no side effects on other ASCII registers or flags.

Bit 4 in CNTLA1 is used.
CKA1D = 1, CKA1/TEND₀ pin = TEND₀
CKA1D = 0, CKA1/TEND₀ pin = CKA1
Cleared to 0 on reset.

MPBR/EFER: Multiprocessor Bit Receive/Error Flag Register (bit 3). When multiprocessor mode is enabled (MP bit in CNTLB = 1), MPBR, when read, contains the value of the MPB bit for the last receive operation. When written to the EFER function is selected to reset all error flags (OVRNFE, PE and BRK in the ASEXT Register) to 0. MPBREFE is undefined during RESET.



MOD2, 1, 0: ASCII Data Format Mode 2, 1, 0 (bits 2-0).
These bits program the ASCII data format as follows.

MOD2
= 0 → 7 bit data
= 1 → 8 bit data

MOD1
= 0 → No parity
= 1 → Parity enabled

MOD0
= 0 → 1 stop bit
= 1 → 2 stop bits

Table 5. Data Formats

MOD2	MOD1	MOD0	Data Format
0	0	0	Start + 7 bit data + 1 stop
0	0	1	Start + 7 bit data + 2 stop
0	1	0	Start + 7 bit data + parity + 1 stop
0	1	1	Start + 7 bit data + parity + 2 stop
1	0	0	Start + 8 bit data + 1 stop
1	0	1	Start + 8 bit data + 2 stop
1	1	0	Start + 8 bit data + parity + 1 stop
1	1	1	Start + 8 bit data + parity + 2 stop

Zilog

The data formats available based on all combinations of MOD2, MOD1, and MOD0 are shown in Table 5-6.

ASCII CHANNEL CONTROL REGISTER B

ASCII Control Register B 0 (CNTLB0: I/O Address = 02H)
ASCII Control Register B 1 (CNTLB1: I/O Address = 03H)

Bit 7	6	5	4	3	2	1	0
MPBT	MP	CTS/PS	PEO	DR	SS2	SS1	SS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Figure 34. ASCII Channel Control Register B

MPBT: Multiprocessor Bit Transmit (bit 7). When multiprocessor communication format is selected (MP bit = 1), MPBT is used to specify the MPB data bit for transmission. If MPBT = 1, then MPB = 1 is transmitted. If MPBT = 0, then MPB = 0 is transmitted. MPBT state is undefined during and after RESET.

MP: Multiprocessor Mode (bit 6). When MP is set to 1, the data format is configured for multiprocessor mode based on the MOD2 (number of data bits) and MOD0 (number of stop bits) bits in CNTLA. The format is as follows.

Start bit + 7 or 8 data bits + MPB bit + 1 or 2 stop bits

Note that multiprocessor (MP=1) format has no provision for parity. If MP = 0, the data format is based on MOD0, MOD1, MOD2, and may include parity. The MP bit is cleared to 0 during RESET.

CTS/PS: Clear to Send/Prescale (bit 5). When read, /CTS/PS reflects the state of the external /CTS input. If the /CTS input pin is HIGH, /CTS/PS will be read as 1. Note that when the /CTS input pin is HIGH, the TDRE bit is inhibited (i.e. held at 0). For channel 1, the /CTS input is multiplexed with RXS pin (Clocked Serial Receive Data).



Zilog

If these bits are not 111 and the BRG mode bit is ASEXT is 0, then these bits specify a power-of-two divider for the PHI clock as shown in Table 9.

Setting or leaving these bits as 111 makes sense for a channel only when its CKA pin is selected for the CKA function. CKA0/CKS has the CKA0 function when bit 4 of the System Configuration Register is 0. DCD0/CKA1 has

the System Configuration Register is 0. DCD0/CKA1 has

the System Configuration Register is 0. DCD0/CKA1 has

Table 6. Divide Ratio

SS2	SS1	SS0	Divide Ratio
0	0	0	+1
0	0	1	+2
0	1	0	+4
0	1	1	+8
1	0	0	+16
1	0	1	+32
1	1	0	+64
1	1	1	External Clock

ASCII STATUS REGISTER 0, 1 (STAT0, 1)

Each channel status register allows interrogation of ASCII communication, error and modem control signal status, and enabling or disabling of ASCII interrupts.

ASCII Status Register 0 (STAT0: I/O Address = 04H)

Bit 7	6	5	4	3	2	1	0
RDRF	OVRN	PE	FE	RE	DCD ₀	TDRE	TIE
R	R	R	R	R/W	R	R	R/W

ASCII Status Register 1 (STAT1: I/O Address = 05H)

Bit 7	6	5	4	3	2	1	0
RDRF	OVRN	PE	FE	RE	CTSIE	TDRE	TIE
R	R	R	R	R/W	R/W	R	R/W

Figure 35. ASCII Status Registers

RDRF: Receive Data Register Full (bit 7). RDRF is set to 1 when an incoming data byte is loaded into an empty Rx FIFO. Note that if a framing or parity error occurs, RDRF is still set and the receive data (which generated the error) is still loaded into the FIFO. RDRF is cleared to 0 by reading RDR and last character in the FIFO from IOSTOP mode, during RESET and for ASCII0 if the /DCD0 input is auto-enabled and is negated (High).

OVRN: Overrun Error (bit 6). An overrun condition occurs if the receiver has finished assembling a character but the Rx FIFO is full so there is no room for the character. However, this status bit is not set until the last character received before the overrun becomes the oldest byte in the FIFO. This bit is cleared when software writes a 1 to the

EFR bit in the CNTLA register, and also by Reset, IOSTOP mode, and for ASCII0 if the /DCD0 pin is auto-enabled and is negated (High).

Note that when an overrun occurs, the receiver does not place the character in the shift register into the FIFO, no any subsequent characters, until the last good character has come to the top of the FIFO so that OVRN is set, and software then writes a 1 to EFR to clear it.

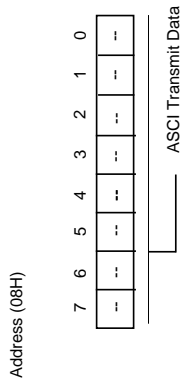


Figure 38. ASCII Receive Register Channel 0

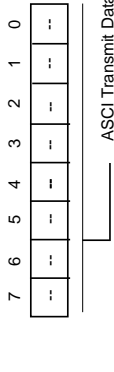


Figure 39. ASCII Receive Register Channel 1R

CS/O CONTROL/STATUS REGISTER

(CNTR: I/O Address = 0AH). CNTR is used to monitor CS/O status, enable and disable the CS/O, enable and disable interrupt generation, and select the data clock speed and source.

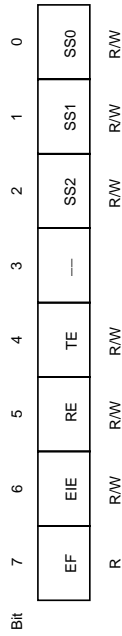


Figure 40. CS/O Control Register

EF: End Flag (bit 7). EF is set to 1 by the CS/O to indicate completion of an 8-bit data transmit or receive operation. If EIE (End Interrupt Enable) bit = 1 when EF is set to 1, a CPU interrupt request is generated. Program access of TRDR only occurs if EF = 1. The CS/O clears EF to 0 when TRDR is read or written. EF is cleared to 0 during RESET and IOSTOP mode.

EIE: End Interrupt Enable (bit 6). EIE is set to 1 to generate a CPU interrupt request. The interrupt request is inhibited if EIE is reset to 0. EIE is cleared to 0 during RESET.

RE: Receive Enable (bit 5). A CS/O receive operation is started by setting RE to 1. When RE is set to 1, the data clock is enabled. In internal clock mode, the data clock is output from the CKS pin. In external clock mode, the clock is input on the CKS pin. In either case, data is shifted in on the RXS pin in synchronization with the (internal or external) data clock. After receiving 8 bits of data, the CS/O automatically clears RE to 0, EF is set to 1, and an interrupt

ASCI0 requests an interrupt when /DCD0 goes High. RIT is cleared to 0 by Reset.

DCDD: Data Carrier Detect (bit 2 STAT0). This bit is set to 1 when the pin is High. It is cleared to 0 on the first read of STAT0 following the pin's transition from High to Low and during RESET. Bit 6 of the ASEXTO register is 0 to select auto-enabling, and the pin is negated (High). Channel 1 has an external CTST input which is multiplexed with the receive data pin RSX for the CS/O.

Bit 2 = 0; Select RXS function.

Bit 2 = 1; Select CTST function.

TDRE: Transmit Data Register Empty (bit 1). TDRE indicates that the TDR is empty and the next transmit data byte is written to TDR. After the byte is written to TDR TDRE is cleared to 0 until the ASCII transfers the byte from TDR to the TSR and then TDRE is again set to 1. TDRE is set to 1 in IOSTOP mode and during RESET. On ASCII if the CTS0 pin is auto-enabled in the ASEXTO register and the pin is High, TDRE is reset to 0.

TIE: Transmit Interrupt Enable (bit 0). TIE should be set to 1 to enable ASCII transmit interrupt requests. If TIE = 1 an interrupt will be requested when TDRE = 1. TIE is cleared to 0 during RESET.

Channel 1

Mnemonics TDR1

Address (07H)

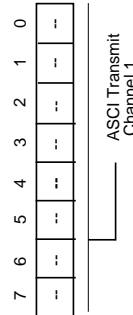


Figure 37. ASCII Register

Mnemonics TSR0 --

PE: Parity Error (bit 5). A parity error is detected when parity checking is enabled by the MOD1 bit in the CNT1LA register being 1, and a character has been assembled in which the parity does not match the PEO bit in the CNTLB register. However, this status bit is not set until/unless the error character becomes the oldest one in the RXFIFO. PE is cleared when software writes a 1 to the EFR bit in the CNTLA register, and also by Reset, in IOSTOP mode, and for ASCII0 if the /DCD0 pin is auto-enabled and is negated (High).

FE: Framing Error (bit 4). A framing error is detected when the stop bit of a character is sampled as 0/Space. However, this status bit is not set until/unless the error character becomes the oldest one in the RXFIFO. FE is cleared when software writes a 1 to the EFR bit in the CNTLA register, and also by Reset, in IOSTOP mode, and for ASCII0 if the /DCD0 pin is auto-enabled and is negated (High).

REI: Receive Interrupt Enable (bit 3). REI should be set to 1 to enable ASCII receive interrupt requests. When REI is 1, the Receiver requests an interrupt when a character is received and RDRF is set, but only if either DMA channel has its Request-routing field set to receive data from this ASCII. That is, if SM1-0 are 11 and SAR17-16 are 10, or DIM1 is 1 and IAR17-16 are 10, then ASCII0 doesn't request an interrupt for RDRF. If REI is 1, either ASCII requests an interrupt when OVRN, PE or FE is set, and

ASCI TRANSMIT DATA REGISTERS

Register addresses 06H and 07H hold the ASCII transmit data for channel 0 and channel 1, respectively.

Channel 0

Mnemonics TDR0

Address (06H)

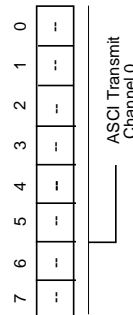


Figure 36. ASCII Register

ASCI Receive Register

Register addresses 08H and 09H hold the ASCII receive data for channel 0 and channel 1, respectively.

Channel 0



Timer Data Register Channel 0L

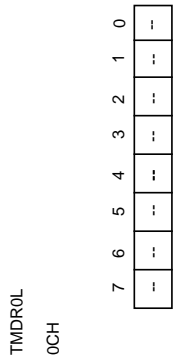


Figure 42. Timer Register Channel 0L

Timer Reload Register 0H

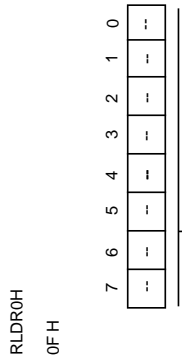


Figure 45. Timer Reload Register Channel

Transmit Enable (bit 4). A CS/I/O transmit operation is started by setting TE to 1. When TE is set to 1, the data clock is enabled. When in internal clock mode, the data clock is output from the CKS pin. In external clock mode, the clock is input on the CKS pin. In either case, data is shifted out on the TXS pin synchronous with the (internal or external) data clock. After transmitting 8 bits of data, the CS/I/O automatically clears TE to 0. EF is set to 1, and an interrupt (if enabled by EIE = 1) is generated. TE and RE are never both set to 1 at the same time. TE is cleared to 0 during RESET and IOSTOP mode.

SS2, 1, 0: Speed Select 2, 1, 0 (bits 2-0). SS2, SS1 and SS0 select the CS/I/O transmit/receive clock source and speed. SS2, SS1 and SS0 are all set to 1 during RESET. Table 10 shows CS/I/O Baud Rate Selection.

Table 7. CS/I/O Baud Rate Selection

SS2	SS1	SS0	Divide Ratio
0	0	0	+20
0	0	1	+40
0	1	0	+80
0	1	1	+160
1	0	0	+320
1	0	1	+640
1	1	0	+1280
1	1	1	External Clock Input (less than +20.)

After RESET, the CKS pin is configured as an external clock input (SS2, SS1, SS0 = 1). Changing these values causes CKS to become an output pin and the selected clock is output when transmit or receive operations are enabled.

CS/I/O Transmit/Receive Data Register (TRDR: I/O Address = 0BH).

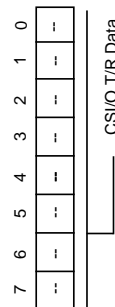


Figure 41. CS/I/O Transmit/Receive Data Register 1R

Timer Reload Register 0L

RLDR0L

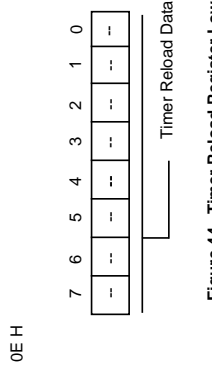


Figure 44. Timer Reload Register Low

TIMER CONTROL REGISTER (TCR)

TCR monitors both channels (PRT0, PRT1), TMDR status. It also controls enabling and disabling of down counting

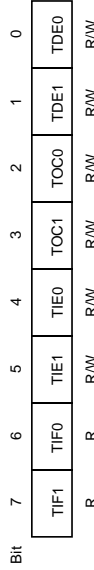


Figure 46. Timer Control Register (TCR: I/O Address = 10H)

TIF1: Timer Interrupt Flag 1 (bit 7). When TMDR1 decrements to 0, TIF1 is set to 1. This generates an interrupt request if enabled by TIE1 = 1. TIF1 is reset to 0 when TCR is read and the higher or lower byte of TMDR1 is read. During RESET, TIF1 is cleared to 0.

TIF0: Timer Interrupt Flag 0 (bit 6). When TMDR0 decrements to 0, TIF0 is set to 1. This generates an interrupt request if enabled by TIE0 = 1. TIF0 is reset to 0 when TCR is read and the higher or lower byte of TMDR0 is read. During RESET, TIF0 is cleared to 0.

TIE1: Timer Interrupt Enable 1 (bit 5). When TIE0 is set to 1, TIF1 = 1 generates a CPU interrupt request. When TIE0 is reset to 0, the interrupt request is inhibited. During RESET, TIE0 is cleared to 0.

TOC1, 0: Timer Output Control (bits 3, 2). TOC1 and TOC0 control the output of PRT1 using the multiplexer TOUT/DREQ pin as shown in Table 11. During RESET TOC1 and TOC0 are cleared to 0. If bit 3 of the IAR1B register is 1, the TOUT function is selected. By programming TOC1 and TOC0, the TOUT/DREQ pin can be forced High, Low, or toggled when TMDR1 decrements to 0.

Table 8. Timer Output Control

TOC1	TOC0	Output
0	0	Inhibited The TOUT/DREQ pin is not affected by the PRT.
0	1	Toggled If bit 3 of IAR1B is 1, the TOUT/DREQ pin is toggles or
1	0	0 TOUT/DREQ pin is toggles or
1	1	1 set Low or High as indicated.



TDE1, 0: Timer Down Count Enable (bits 1, 0). TDE1 and TDE0 enable and disable down counting for TMDR1 and TMDR0, respectively. When TDEn (n = 0, 1) is set to 1, down counting is stopped and TMDRn is freely read or written. TDE1 and TDE0 are cleared to 0 during RESE and TMDRn will not decrement until TDEn is set to 1.

Timer Data Register Channel 1L
Mnemonic TMDR1L
Address 14

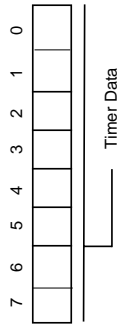


Figure 48. Timer Data Register 1L

Timer Data Register Channel 1H
Mnemonic TMDR1H
Address 15

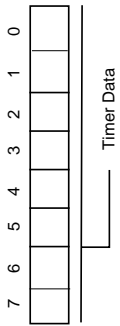


Figure 49. Timer Data Register 1H

Timer Reload Register Channel 1L
Mnemonic RLDR1L
Address 16

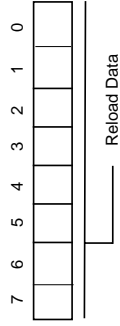


Figure 50. Timer Reload Channel 1L

Timer Reload Register Channel 1L
Mnemonic RLDR1H
Address 17



Figure 51. Timer Reload Register Channel 1L

Free Running Counter (Read Only)
Mnemonic FRC
Address 18

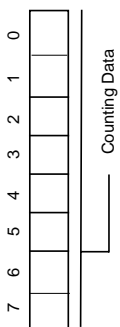


Figure 52. Free Running Counter



DMA SOURCE ADDRESS REGISTER CHANNEL 0

(SAR0: I/O Address = 20H to 22H) specifies the physical source address for channel 0 transfers. The register contains 20 bits and can specify up to 1024 KB memory addresses or up to 64 KB I/O addresses. Channel 0 source can be memory, I/O, or memory mapped I/O. For I/O, the MS bits of this register identify the Request Handshake signal.

DMA Source Address Register, Channel 0L

Mnemonic: SAR0L
Address 20

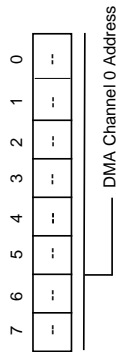


Figure 55. DMA Source Address Register 0L

DMA Source Address Register, Channel 0H

Mnemonic: SAR0H
Address 21



Figure 56. DMA Source Address Register 0H

DMA DESTINATION ADDRESS REGISTER CHANNEL 0

(DAR0: I/O Address = 23H to 25H) specifies the physical destination address for channel 0 transfers. The register contains 20 bits and can specify up to 1024 KB memory addresses or up to 64 KB I/O addresses. Channel 0 destination can be memory, I/O, or memory mapped I/O. For I/O, the MS bits of this register identify the Request Handshake signal for channel 0.

DMA Destination Address Register Channel 0L

Mnemonic: DAR0L
Address 23



Figure 58. DMA Destination Address Register Channel 0L

DMA Destination Address Register Channel 0B

Mnemonic: DAR0B
Address 25



Figure 60. DMA Destination Address Register Channel 0B

DMA Destination Address Register Channel 0H

Mnemonic: DAR0H
Address 24



Figure 59. DMA Destination Address Register Channel 0H

Note: In the R1 and Z Mask, these DMA registers are expanded from 4 bit to 3 bits in the package version of CP68

A19*	A18	A17	A16	DMA Transfer Request
X	X	0	0	DREQ0
X	X	0	1	TDR0 (ASCI0)
X	X	1	0	TDR1 (ASCI1)
X	X	1	1	Not Used



DMA BYTE COUNT REGISTER CHANNEL 0

(BCRO: I/O Address = 26H to 27H) specifies the number of bytes to be transferred. This register contains 16 bits and may specify up to 64 KB transfers. When one byte is transferred, the register is decremented by one. If "n" bytes should be transferred, "n" must be stored before the DMA operation.

Note: All DMA Count Register channels are undefined during reset.

DMA Byte Count Register Channel 0L
Mnemonic: BCR0L
Address: 26



Figure 61. DMA Byte Count Register 0L

DMA Byte Count Register Channel 0H
Mnemonic: BCR0H
Address: 27



Figure 62. DMA Byte Count Register 0H

DMA Byte Count Register Channel 1L
Mnemonic: BCR1L
Address: 2E



Figure 63. DMA Byte Count Register 1L

DMA Byte Count Register Channel 0H
Mnemonic: BCR1H
Address: 2F



Figure 64. DMA Byte Count Register 0H

DMA MEMORY ADDRESS REGISTER CHANNEL 1

(MAR1: I/O Address = 28H to 2AH) specifies the physical memory address for channel 1 transfers. This may be destination or source memory address. The register contains 20 bits and may specify up to 1024-KB memory address.

DMA Memory Address Register, Channel 1L
Mnemonic: MAR1L
Address: 28



Figure 65. DMA Memory Address Register, Channel 1L

DMA Memory Address Register, Channel 1H
Mnemonic: MAR1H
Address: 29

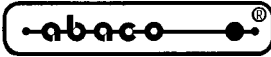


Figure 66. DMA Memory Address Register, Channel 1H

DMA Memory Address Register, Channel 1B
Mnemonic: MAR1B
Address: 2A



Figure 67. DMA Memory Address Register, Channel 1B



DMA I/O ADDRESS REGISTER CHANNEL 1

(IAR1: I/O Address = 2BH to 2DH) specifies the I/O address for channel 1 transfers. This may be destination or source I/O address. The register contains 16 bits of I/O address; its most significant byte identifies the Request

Handshake signal and controls the Alternating Channel feature.

All bits in IAR1B reset to 0.

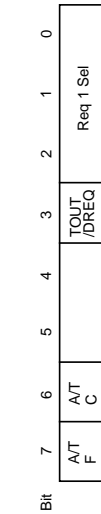


Figure 68. IAR MS Byte Register (IAR1B: I/O Address 2DH)

DMA I/O Address Register Channel 1L

Mnemonic: IAR1L

Address 2B



Figure 69. DMA I/O Address Register Channel 1L

DMA I/O Address Register Channel 1H

Mnemonic: IAR1H

Address 2C

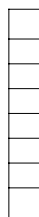


Figure 70. DMA I/O Address Register Channel 1H

DMA I/O Address Register Channel 1B

Mnemonic: IAR1B

Address 2D



Figure 71. DMA I/O Address Register Channel 1B

DMA STATUS REGISTER (DSTAT)

DSTAT is used to enable and disable DMA transfer and DMA termination interrupts. DSTAT also indicates DMA transfer status, in other words, completed or in progress.

Mnemonic: DSTAT

Address 30

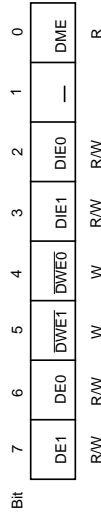


Figure 72. DMA Status Register (DSTAT: I/O Address = 30H)

DE1: DMA Enable Channel 1 (bit 7). When DE1 = 1 and DME = 1, channel 1 DMA is enabled. When a DMA transfer terminates (BCR1 = 0), DE1 is reset to 0 by the DMAC. When DE1 = 0 and the DMA interrupt is enabled (DIE1 = 1), a DMA interrupt request is made to the CPU.

To perform a software write to DE1, DWE1 should be written with 0 during the same register write access. Writing DE1 to 0 disables channel 1 DMA, but DMA is restartable. Writing DE1 to 1 enables channel 1 DMA and automatically sets DME (DMA Main Enable) to 1. DE1 is cleared to 0 during RESET.

DE0: DMA Enable Channel 0 (bit 6). When DE0 = 1 and DME = 1, channel 0 DMA is enabled. When a DMA transfer terminates (BCR0 = 0), DE0 is reset to 0 by the DMAC. When DE0 = 0 and the DMA interrupt is enabled (DIE0 = 1), a DMA interrupt request is made to the CPU.

To perform a software write to DE0, DWE0 should be written with 0 during the same register write access. Writing DE0 to 0 disables channel 0 DMA. Writing DE0 to 1 enables channel 0 DMA and automatically sets DME (DMA Main Enable) to 1. DE0 is cleared to 0 during RESET.

DWE1: DE1 Bit Write Enable (bit 5). When performing any software write to DE1, DWE1 should be written with 0 during the same access. DWE1 always reads as 1.

DWE0: DE0 Bit Write Enable (bit 4). When performing any software write to DE0, DWE0 should be written with 0 during the same access. DWE0 always reads as 1.

DIE1: DMA Interrupt Enable Channel 1 (bit 3). When DIE0 is set to 1, the termination channel 1 DMA transfer (indicated when DE1 = 0) causes a CPU interrupt request to be generated. When DIE0 = 0, the channel 0 DMA termination interrupt is disabled. DIE0 is cleared to 0 during RESET.

DIE0: DMA Interrupt Enable Channel 0 (bit 2). When DIE0 is set to 1, the termination channel 0 or DMA transfer (indicated when DE0=0) causes a CPU interrupt request to be generated. When DIE0=0, the channel 0 DMA termination interrupt is disabled. DIE0 is cleared to 0 during RESET.

DME: DMA Main Enable (bit 0). A DMA operation is enabled when its DE bit (DE0 for channel 0, DE1 for channel 1) and the DME bit is set to 1.

When \overline{NMI} occurs, DME is reset to 0, thus disabling DMA activity during the NMI interrupt service routine. To restart DMA, DE- and/or DE1 should be written with 1 (even if the contents are already 1). This automatically sets DME to 1 allowing DMA operations to continue. Note that DME can not be directly written. It is cleared to 0 by NMI or indirectly set to 1 by setting DE0 and/or DE1 to 1. DME is cleared to 0 during RESET.



DMA MODE REGISTER (DMODE).

DMODE is used to set the addressing and transfer mode for channel 0.

Mnemonic DMODE

Address 31H

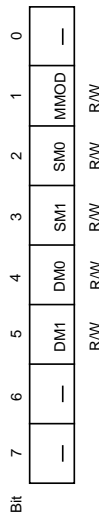


Figure 73. DMA Mode Register (DMODE: I/O Address = 31H)

DM1, DM0: Destination Mode Channel 0 (bits 5,4) specifies whether the destination for channel 0 transfers is memory or I/O, and whether the address should be incremented or decremented for each byte transferred. DM1 and DM0 are cleared to 0 during RESET.

Table 10. Channel 0 Destination

DM1	DM0	Memory I/O	Increment/Decrement
0	0	Memory	+1
0	1	Memory	-1
1	0	Memory	fixed
1	1	I/O	fixed

SM1, SM0: Source Mode Channel 0 (bits 3, 2) specifies whether the source for channel 0 transfers is memory or I/O, and whether the address should be incremented or decremented for each byte transferred.

Table 11. Channel 0 Source

SM1	SM0	Memory I/O	Increment/Decrement
0	0	Memory	+1
0	1	Memory	-1
1	0	Memory	fixed
1	1	I/O	fixed

Table 12 shows all DMA transfer mode combinations of DM0, DM1, SM0, and SM1. Since I/O to/from I/O transfers are not implemented, 12 combinations are available.

Table 12. Transfer Mode Combinations

DM1	DM0	SM1	SM0	Transfer Mode	Address Increment/Decrement
0	0	0	0	Memory→Memory	SAR0+1, DAR0+1
0	0	0	1	Memory→Memory	SAR0-1, DAR0+1
0	0	1	0	Memory*→Memory	SAR0 fixed, DAR0+1
0	0	1	1	I/O→Memory	SAR0 fixed, DAR0+1
0	1	0	0	Memory→Memory	SAR0+1, DAR0-1
0	1	0	1	Memory→Memory	SAR0-1, DAR0-1
0	1	1	0	Memory*→Memory	SAR0 fixed, DAR0-1
0	1	1	1	I/O→Memory	SAR0 fixed, DAR0-1
1	0	0	0	Memory→Memory*	SAR0+1, DAR0 fixed
1	0	0	1	Memory→Memory*	SAR0-1, DAR0 fixed
1	0	1	0	Reserved	
1	0	1	1	Reserved	
1	1	0	0	Memory→I/O	SAR0+1, DAR0 fixed
1	1	0	1	Memory I/O	SAR0-1, DAR0 fixed
1	1	1	0	Reserved	
1	1	1	0	Reserved	

Note: * Includes memory mapped I/O.

MMOD: Memory Mode Channel 0 (bit). When channel 0 is configured for memory to/from memory transfers there is no Request Handshake signal to control the transfer timing. Instead, two automatic transfer timing modes are selectable: burst (MMOD = 1) and cycle steal (MMOD = 0). For burst memory to/from memory transfers, the DMAC takes control of the bus continuously until the DMA transfer completes (as shown by the byte count register = 0). In cycle steal mode, the CPU is given a cycle for each DMA byte transfer cycle until the transfer is completed.

For channel 0 DMA with I/O source or destination, the selected Request signal times the transfer and thus MMOD is ignored. MMOD is cleared to 0 during RESET.



DMA/WAIT CONTROL REGISTER (DCNTL)

DCNTL controls the insertion of wait states into DMAC (and CPU) accesses of memory or I/O. Also, it defines the Request signal for each channel as level or edge sense.

DCNTL also sets the DMA transfer mode for channel 1 which is limited to memory to/from I/O transfers.

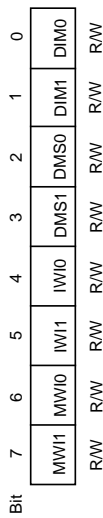


Figure 74. DMA/WAIT Control Register (DCNTL: I/O Address = 32H)

MW1, MW0: Memory Wait Insertion (bits 7-6). Specifies the number of wait states introduced into CPU or DMAC memory access cycles. MW1 and MW0 are set to 1 during RESET.

MW1	MW0	Wait State
0	0	0
0	1	1
1	0	2
1	1	3

IW1, IW0: I/O Wait Insertion (bits 5-4). Specifies the number of wait states introduced into CPU or DMAC I/O access cycles. IW1 and IW0 are set to 1 during RESET. See the section on Wait-State Generation for details.

IW1	IW0	Wait State
0	0	0
0	1	2
1	0	3
1	1	4

DMS1, DMS0: DMA Request Sense (bits 3-2). DMS1 and DMS0 specify the DMA request sense for channel 1 and channel 0 respectively. When reset to 0, the input is level sense. When set to 1, the input is edge sense. DMS1 and DMS0 are cleared to 0 during RESET.

DMS1	Sense
1	Edge Sense
0	Level Sense

Typically, for an input/source device, the associated DM bit should be programmed as 0 for level sense because the device has a relatively long time to update its Request signal after the DMA channel reads data from it in the first of the two machine cycles involved in transferring a byte.

An output/destination device has much less time to update its Request signal, after the DMA channel starts a write operation to it, as the second machine cycle of the two cycle involved in transferring a byte. With zero-wait state I/O cycles, which apply only to the ASCIs, it is impossible for a device to update its Request signal in time, and edge sensing must be used.

With one-wait-state I/O cycles (the fastest possible except for the ASCIs), it is unlikely that an output device will be able to update its Request in time, and edge sense is required.

DIM1, DIM0: DMA Channel 1 I/O and Memory Mode (bits 1-0). Specifies the source/destination and address

modifier for channel 1 memory to/from I/O transfer modes. DIM1 and DIM0 are cleared to 0 during RESET.

Table 13. Channel 1 Transfer Mode

DIM1	DIM0	Transfer Mode	Increment/Decrement	Address
0	0	Memory→I/O	MAR1+1, IAR1 fixed	
0	1	Memory→I/O	MAR1-1, IAR1 fixed	
1	0	I/O→Memory	IAR1 fixed, MAR1+1	
1	1	I/O→Memory	IAR1 fixed, MAR1-1	

INTERRUPT VECTOR LOW REGISTER

Mnemonic: IL

Address 33

Bits 7-5 of IL are used as bits 7-5 of the synthesized interrupt vector during interrupts for the INT1 and INT2 pins; and for the DMAs, ASCIs, PRTs, and CSI/O. These three bits are cleared to 0 during Reset (Figure 75).

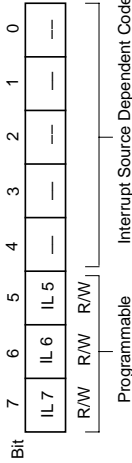


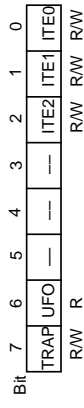
Figure 75. Interrupt Vector Low Register (IL: I/O Address = 33H)

INT/TRAP CONTROL REGISTER

Mnemonic: ITC

Address 34

INT/TRAP Control Register (ITC, I/O Address 34H). This register is used in handling TRAP interrupts, and to enable or disable Maskable Interrupt Level 0 and the INT1 and INT2 pins.



TRAP (bit 7). This bit is set to 1 when an undefined Opcode is fetched. TRAP can be reset under program control by writing it with a 0, however, it cannot be written with 1 under program control. TRAP is reset to 0 during RESET.

UFO: Undefined Fetch Object (bit 6). When a TRAP interrupt occurs, the contents of UFO allow determination of

the starting address of the undefined instruction. This is necessary since the TRAP may occur on either the second or third byte of the Opcode. UFO allows the stacked PC value to be correctly adjusted. If UFO = 0, the first Opcode should be interpreted as the stacked PC-1. If UFO = 1, the first Opcode address is stacked PC-2. UFO is Read-Only

ITE2, 1, 0: Interrupt Enable 2, 1, 0 (bits 2-0). ITE2 and ITE1 enable and disable the external interrupt inputs /INT2 and /INT1, respectively. ITE0 enables and disables interrupts from the on-chip ESCC, CTCs and Bidirectional Centronics controller as well as the external interrupt input /INT0. A 1 in a bit enables the corresponding interrupt level while a 0 disables it. A Reset sets ITE0 to 1 and clear ITE1 and ITE2 to 0.

TRAP Maskable. The Z80180/Z8S180/Z8L180 generate a non-maskable (not affected by the state of IEF1) TRAP interrupt when an undefined Opcode fetch occurs. This feature can be used to increase software reliability, implement an "extended" instruction set, or both. TRAP may occur during Opcode fetch cycles and also if an undefined



Opcode is fetched during the interrupt acknowledge cycle for INT₀ when Mode 0 is used.

When a TRAP interrupt occurs, the Z80180/Z8S180/Z8L180 operates as follows:

1. The TRAP bit in the Interrupt TRAP/Control (ITC) register is set to 1.
2. The current PC (Program Counter) value, reflecting the location of the undefined Opcode, is saved on the stack.
3. The Z80180/Z8S180/Z8L180 vectors to logical address 0. Note that if logical address 0000H is mapped to physical address 00000H, the vector is the same as for RESET. In this case, testing the TRAP bit in ITC will reveal whether the restart at physical address 00000H was caused by RESET or TRAP.

All TRAP interrupts occur after fetching an undefined second Opcode byte following one of the "prefix" Opcode CBH, DDH, EDH, or FDH, or after fetching an undefined third Opcode byte following one of the "double prefix" Op codes DDCBH or FDCBH.

The state of the Undefined Fetch Object (UFO) bit in ITC allows TRAP software to correctly "adjust" the stacked PC depending on whether the second or third byte of the Op code generated the TRAP. If UFO=0, the starting address of the invalid instruction is equal to the stacked PC-1. If UFO=1, the starting address of the invalid instruction is equal to the stacked PC-2.

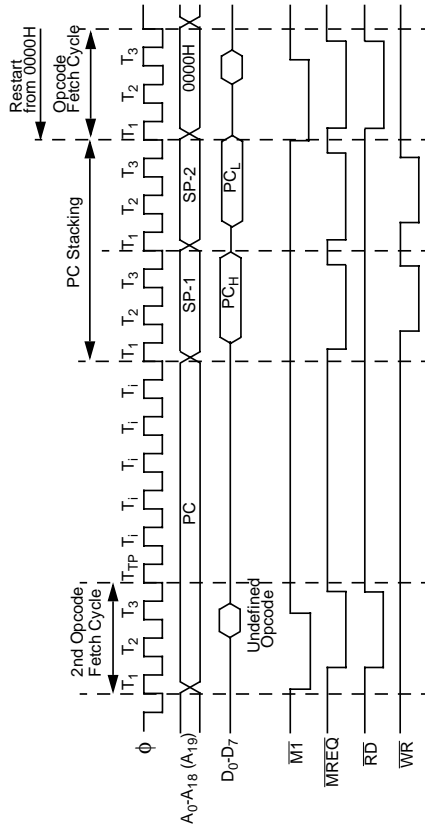


Figure 76. TRAP Timing-2nd Opcode Undefined

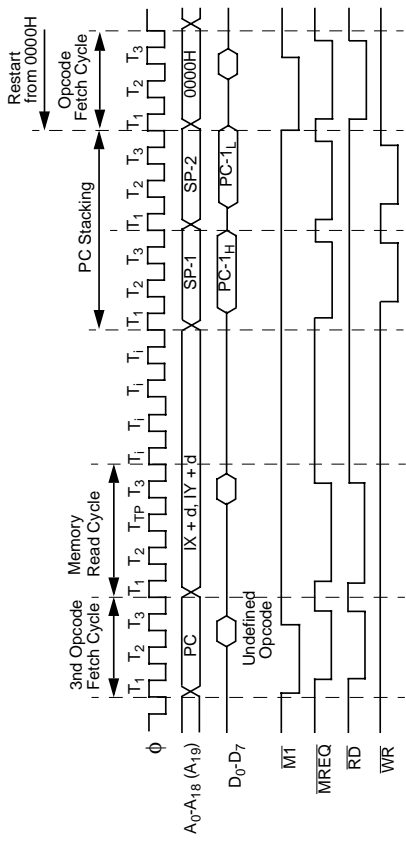
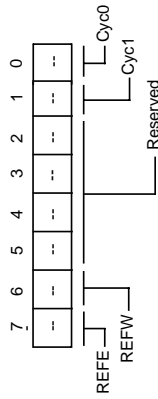


Figure 77. TRAP Timing-3rd Opcode Undefined

REFRESH CONTROL REGISTER

Mnemonic RCR

Address 36



REFE: Refresh Enable (bit 7). REFE = disables the refresh controller while REFE = 1 enables refresh cycle in section. REFE is set to 1 during RESET.

REFW: Refresh Wait (bit 6). REFW = 0 causes the refresh cycle to be two clocks in duration. REFW = 1 causes the refresh cycle to be three clocks in duration by adding refresh wait cycle (TRW). REFW is set to 1 during RESET.

CYC1, 0: Cycle Interval (bit 1, 0). CYC1 and CYC0 specify the interval (in clock cycles) between refresh cycles. In the case of dynamic RAMs requiring 128 refresh cycles every 2 ms (or 256 cycles in every 4 ms), the required refresh interval is less than or equal to 15.625 μs. Thus, the underlined values indicate the best refresh interval depending on CPU clock frequency. CYC0 and CYC1 are cleared to 0 during RESET (see Table 14).

Figure 78. Refresh Control Register (RCA: I/O Address = 36H)

The RCR specifies the interval and length of refresh cycles, while enabling or disabling the refresh function.



Table 14. DRAM Refresh Intervals

CYC1	CYC0	Insertion Interval	∅: 10 MHz	8 MHz	6 MHz	4 MHz	2.5 MHz
0	0	10 states	(1.0 µs)*	(1.25 µs)*	1.66 µs	2.5 µs	4.0 µs
0	1	20 states	(2.0 µs)*	(2.5 µs)*	3.3 µs	5.0 µs	8.0 µs
1	0	40 states	(4.0 µs)*	(5.0 µs)*	6.6 µs	10.0 µs	16.0 µs
1	1	80 states	(8.0 µs)*	(10.0 µs)*	13.3 µs	20.0 µs	32.0 µs

Note: *calculated interval

Refresh Control and Reset. After RESET, based on the initialized value of RCR, refresh cycles will occur with an interval of 10 clock cycles and be 3 clock cycles in duration.

which the first refresh cycle occurs after the Z80180/Z8S180/Z8L180 re-acquires the bus depends on the refresh timer and has no timing relationship with the bus exchange.

Dynamic RAM Refresh Operation

1. Refresh Cycle insertion is stopped when the CPU is in the following states:
 - a. During RESET
 - b. When the bus is released in response to BUSREQ.
 - c. During SLEEP mode.
 - d. During WAIT states.
2. Refresh cycles are suppressed when the bus is released in response to BUSREQ. However, the refresh timer continues to operate. Thus, the time at which the first refresh cycle occurs after the bus is released is incremented by one from that of the previous refresh bus cycles.
3. Refresh cycles are suppressed during SLEEP mode if a refresh cycle is requested during SLEEP mode. The refresh cycle request is internally "latched" (not replaced with the next refresh request). The "latched" refresh cycle is inserted at the end of the first machine cycle after SLEEP mode is exited. After this initial cycle, the time at which the next refresh cycle occurs depends on the refresh time and has no relationship with the exit from SLEEP mode.
4. The refresh address is incremented by one for each successful refresh cycle, not for each refresh. Thus independent of the number of "missed" refresh requests, each refresh bus cycle will use a refresh address incremented by one from that of the previous refresh bus cycles.

MMU COMMON BASE REGISTER

Mnemonic CBR
Address 38

MMU Common Base Register (CBR). CBR specifies the base address (on 4-KB boundaries) used to generate a 20-bit physical address for Common Area 1 accesses. All bits of CBR are reset to 0 during RESET.

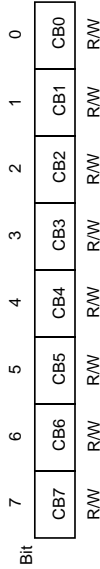


Figure 79. MMU Common Base Register (BBR: I/O Address = 38H)

MMU BANK BASE REGISTER (BBR).

Mnemonic BBR
Address 39

BBR specifies the base address (on 4-KB boundaries) used to generate a 19-bit physical address for Bank Area accesses. All bits of BBR are reset to 0 during RESET.

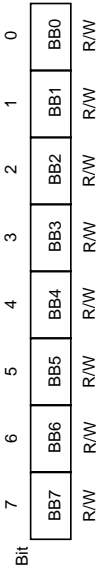


Figure 80. MMU Bank Base Register (BBR: I/O Address = 39H)

MMU COMMON/BANK AREA REGISTER (CBAR).

Mnemonic CBAR
Address 3A

CBAR specifies boundaries within the Z80180/Z8S180/Z8L180 64-KB logical address space for up to three areas: Common Area, Bank Area and Common Area 1.

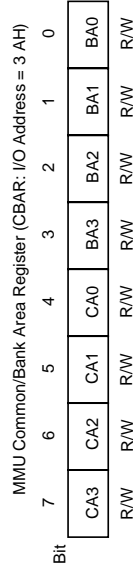


Figure 81. MMU Common/Bank Area Register (CBAR: I/O Address = 3AH)

CA3-CA0:CA (bits 7-4). CA specifies the start (Low) address (on 4-KB boundaries) for the Common Area 1. This also determines the last address of the Common Area. All bits of CA are set to 1 during RESET.

BA-BA0 (bits 3-0). BA specifies the start (Low) address (on 4-KB boundaries) for the Bank Area. This also determines the last address of the Common Area 0. All bits of BA are set to 1 during RESET.



IOA7, 6: I/O Address Relocation (bits 7,6). IOA7 and IOA6 relocate internal I/O as shown in Figure 85. Note that the high-order 8 bits of 16-bit internal I/O address are always 0. IOA7 and IOA6 are cleared to 0 during Reset.

IOA7-IOA6 = 1 1	00FFH
IOA7-IOA6 = 1 0	00COH 00BFH
IOA7-IOA6 = 0 1	0080H 0070H
IOA7-IOA6 = 0 0	0040H 003FH 0000H

Figure 85. I/O Address Relocation

IOSTOP, IOSTOP Mode (bit 5). IOSTOP mode is enabled when IOSTOP is set to 1. Normal I/O operation resumes when IOSTOP is reprogrammed or Reset to 0

OPERATION MODE CONTROL REGISTER

Mnemonic: OMCR

Address: 3E

The Z80180/Z8S180/Z8L180 is descended from two different "ancestor" processors, Zilog's original Z80 and the Hitachi 64180. The Operating Mode Control Register (OMCR) can be programmed to select between certain differences between the Z80 and the 64180.

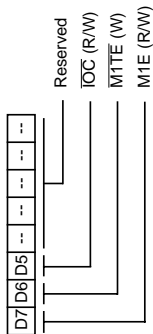


Figure 82. Operating Control Register (OMCR: I/O Address = 3EH)

MIE (M1 Enable). This bit controls the M1 output and is set to a 1 during reset.

When MIE=1, the $\overline{M1}$ output is asserted Low during the opcode fetch cycle, the INT0 acknowledge cycle, and the first machine cycle of the NMI acknowledge.

On the Z80180/Z8S180/Z8L180, this choice makes the processor fetch an RETI instruction once, and when fetching an RETI from zero-wait-state memory, will use three clock machine cycles which are not fully Z80-timing compatible but are compatible with the on-chip CTCs.

When MIE=0, the processor does not drive $\overline{M1}$ Low during instruction fetch cycles, and after fetching an RETI instruction once with normal timing, it goes back and re-fetches the instruction using fully Z80-compatible cycles that include driving M1 Low. This may be needed by some external Z80 peripherals to properly decode the RETI instruction. I/O Control Register (ICR).

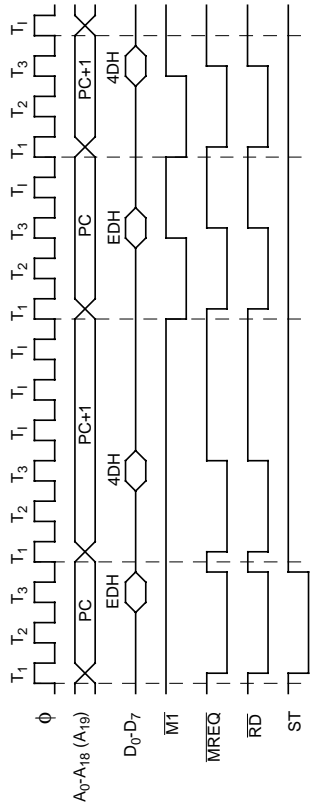


Figure 83. RETI Instruction Sequence with MIE=0

ICR allows relocating of the internal I/O addresses. ICR also controls enabling/disabling of the IOSTOP mode (Figure 84).

Bit	7	6	5	4	3	2	1	0
	IOA7	IOA6	IOSTOP	R/W	R/W	R/W	R/W	R/W

Figure 84. I/O Control Register (ICR: I/O Address = 3FH)



APPENDICE C: SCHEMI ELETTRICI

In questa appendice sono disponibili gli schemi elettrici delle interfacce per la GPC® 184 più frequentemente utilizzate. Tutte queste interfacce possono essere prodotte autonomamente dall'utente mentre solo alcune di esse sono schede grifo® standard e possono quindi essere ordinate.

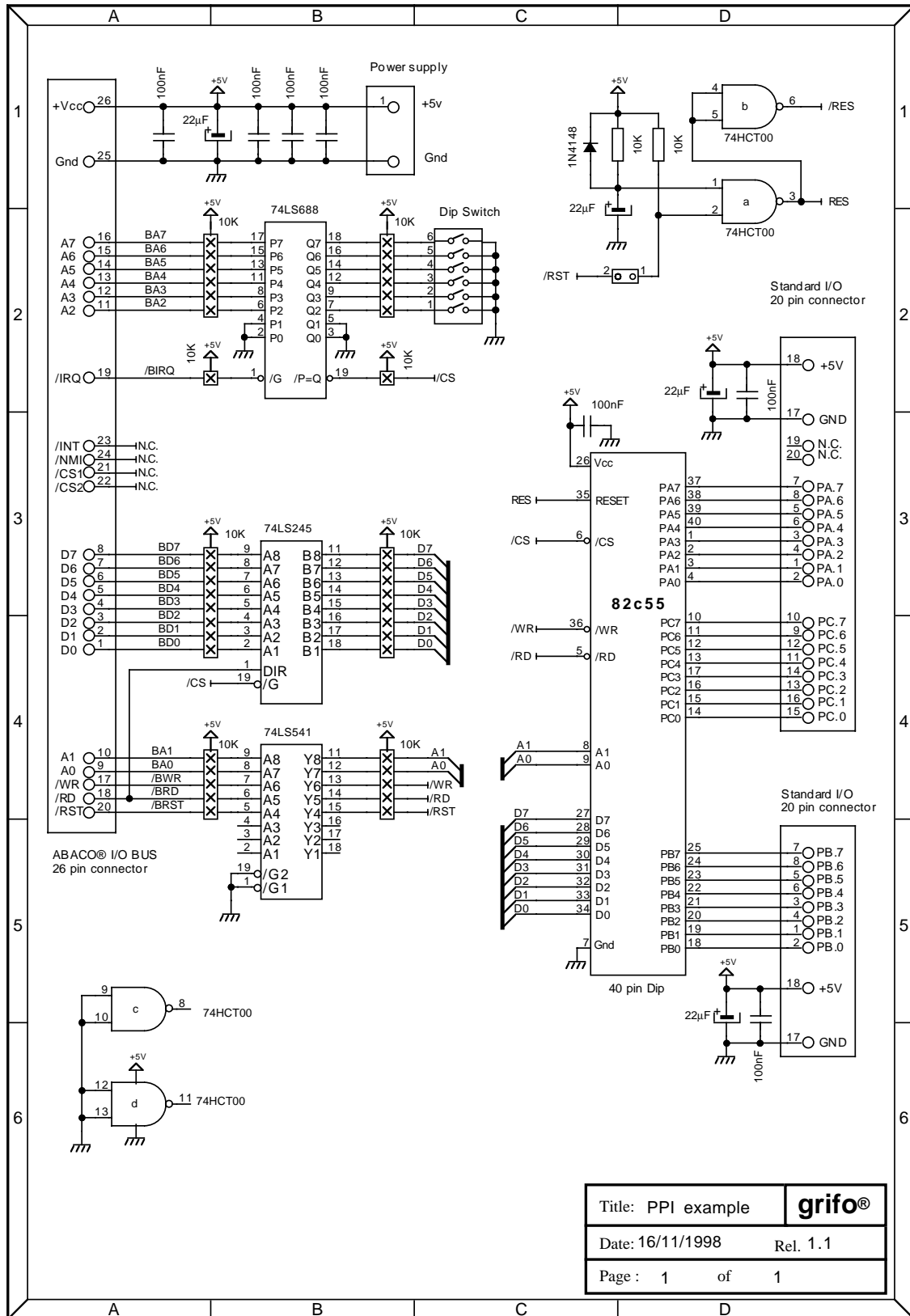


FIGURA C1: SCHEMA ELETTRICO DI ESPANSIONE PPI



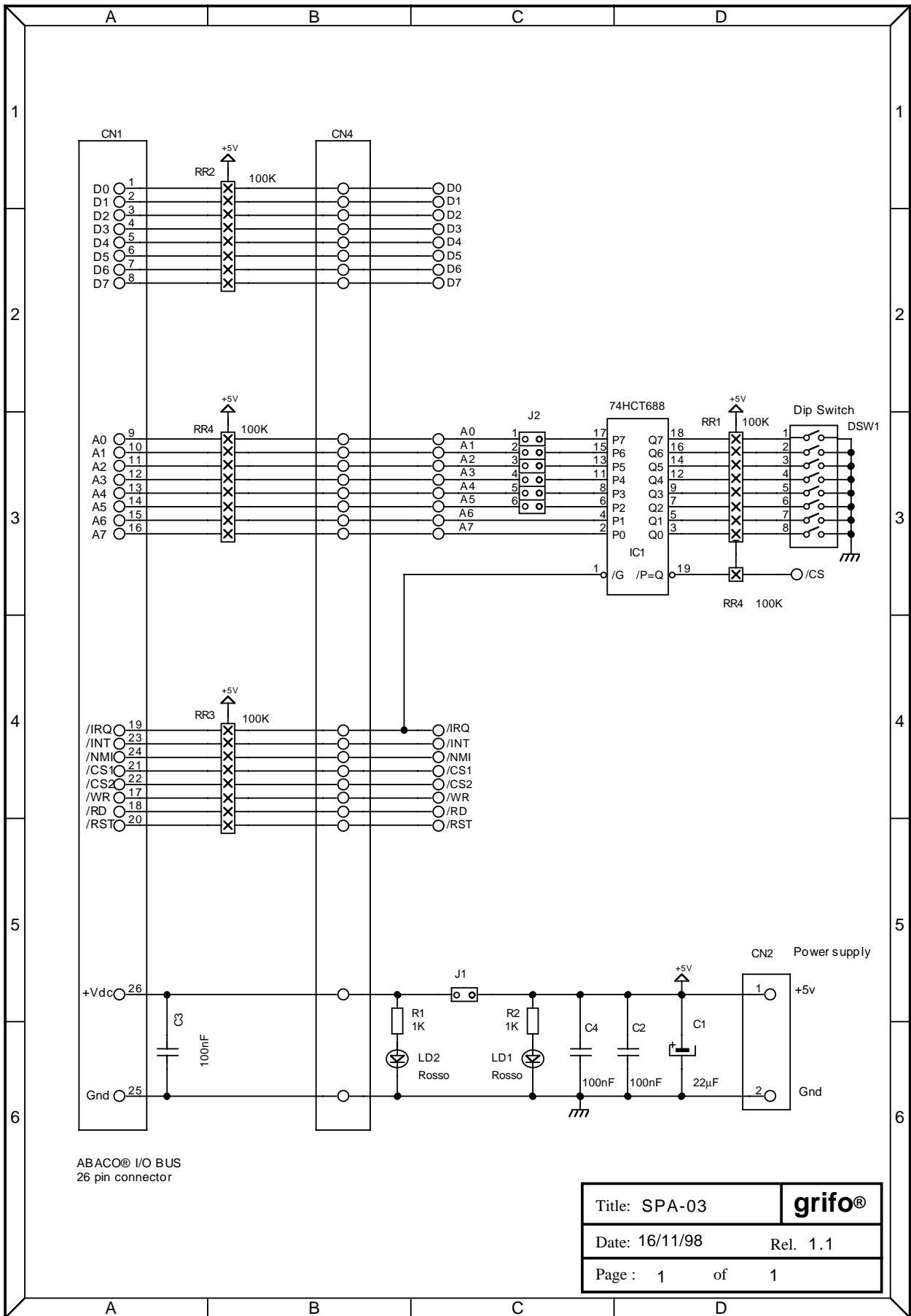
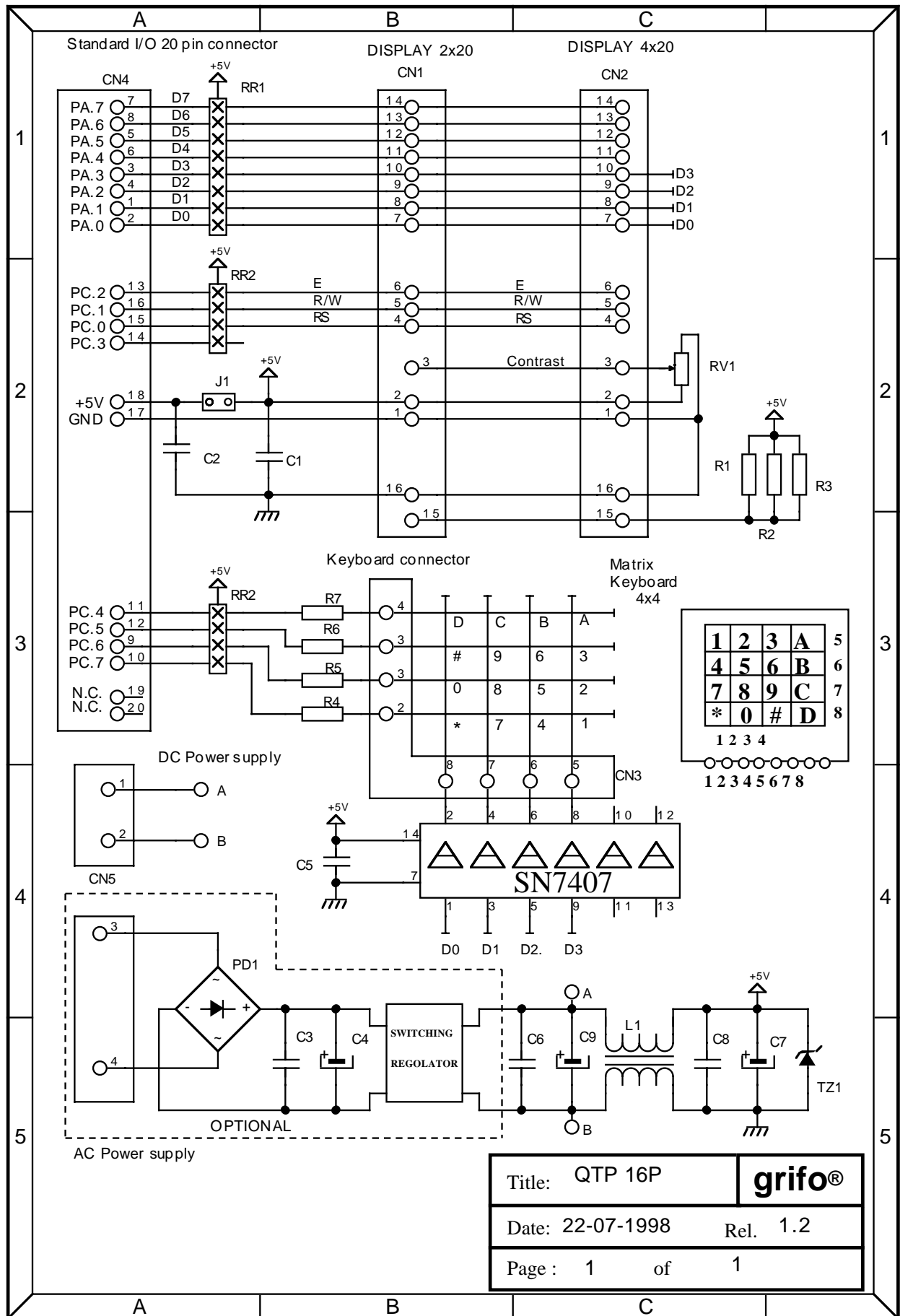


FIGURA C2: SCHEMA ELETTRICO SPA 03





Title: QTP 16P	grifo®
Date: 22-07-1998	Rel. 1.2
Page : 1	of 1

FIGURA C3: SCHEMA ELETTRICO QTP 16P



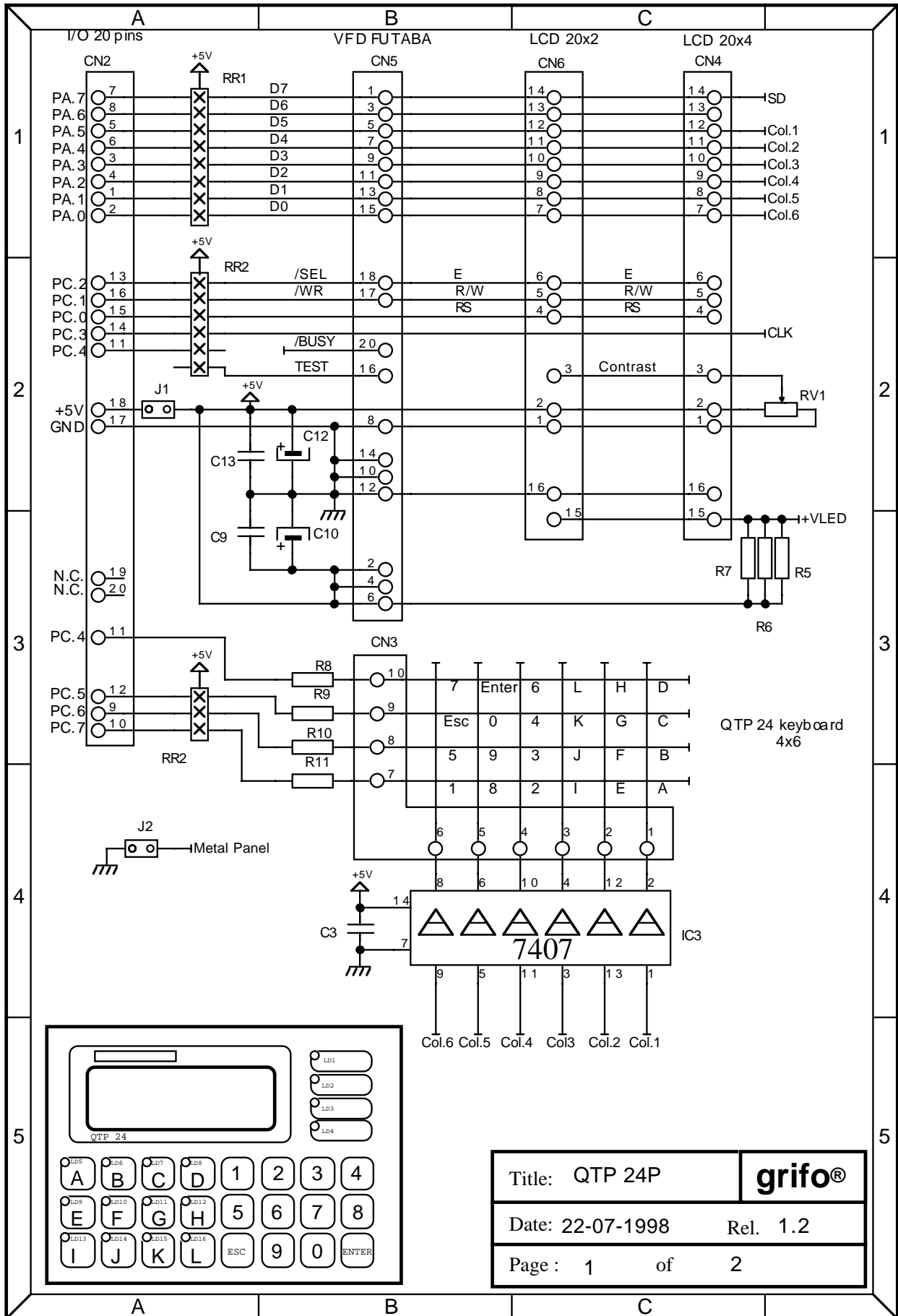


FIGURA C4: SCHEMA ELETTRICO QTP 24P 1/2

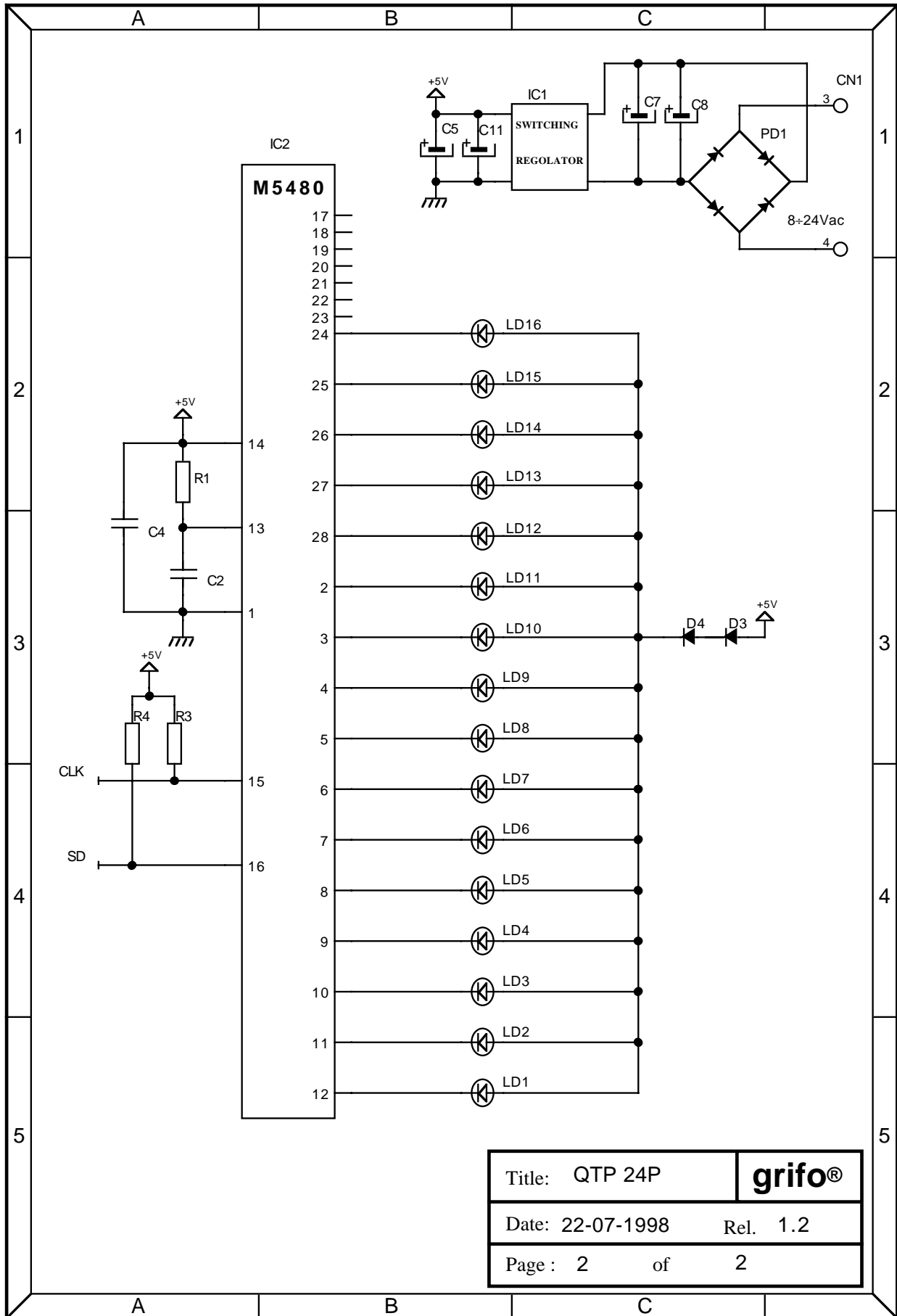
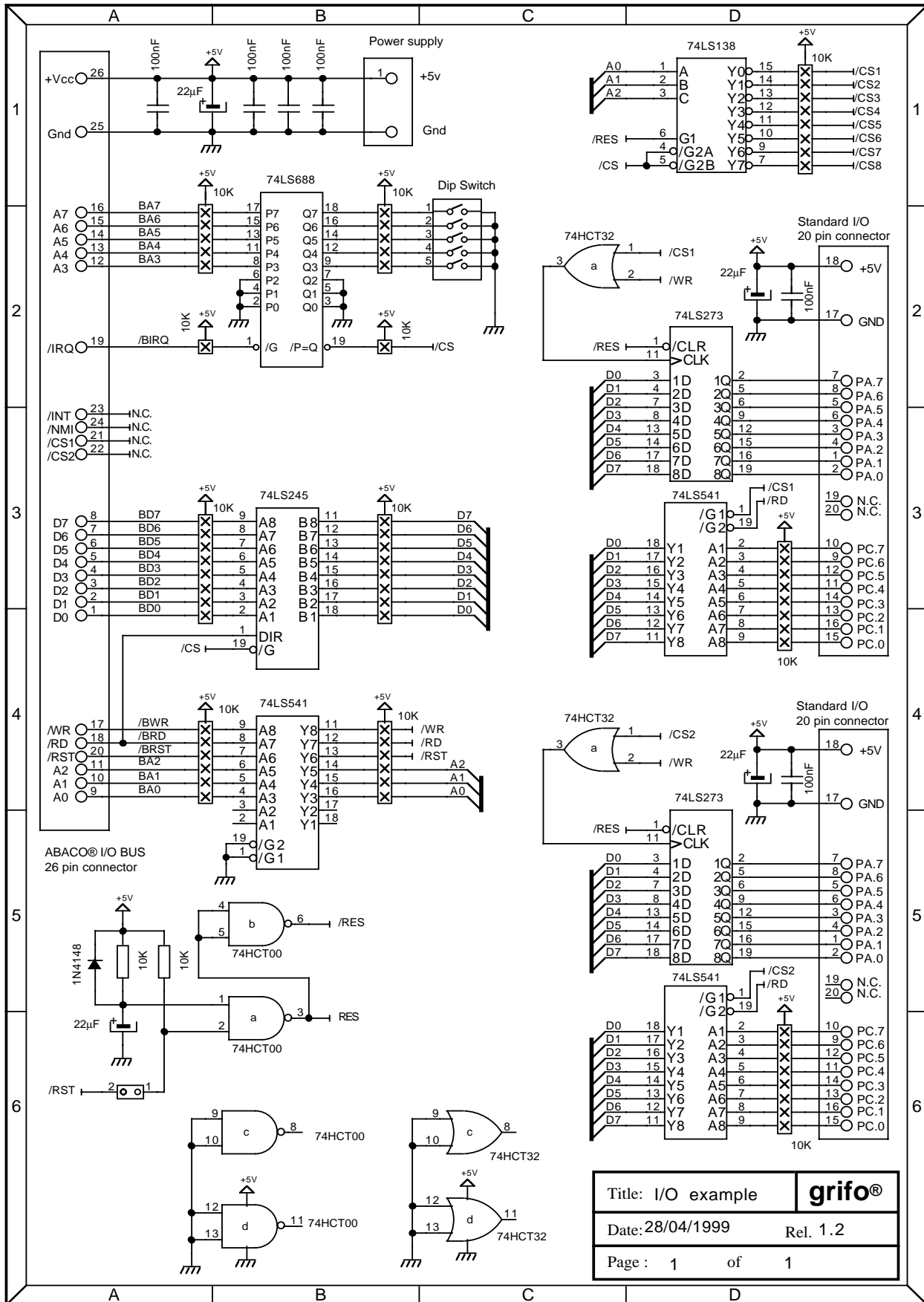


FIGURA C5: SCHEMA ELETTRICO QTP 24P 2/2





Title: I/O example	grifo®
Date: 28/04/1999	Rel. 1.2
Page : 1	of 1

FIGURA C6: SCHEMA ELETTRICO DI I/O SU ABACO® I/O BUS



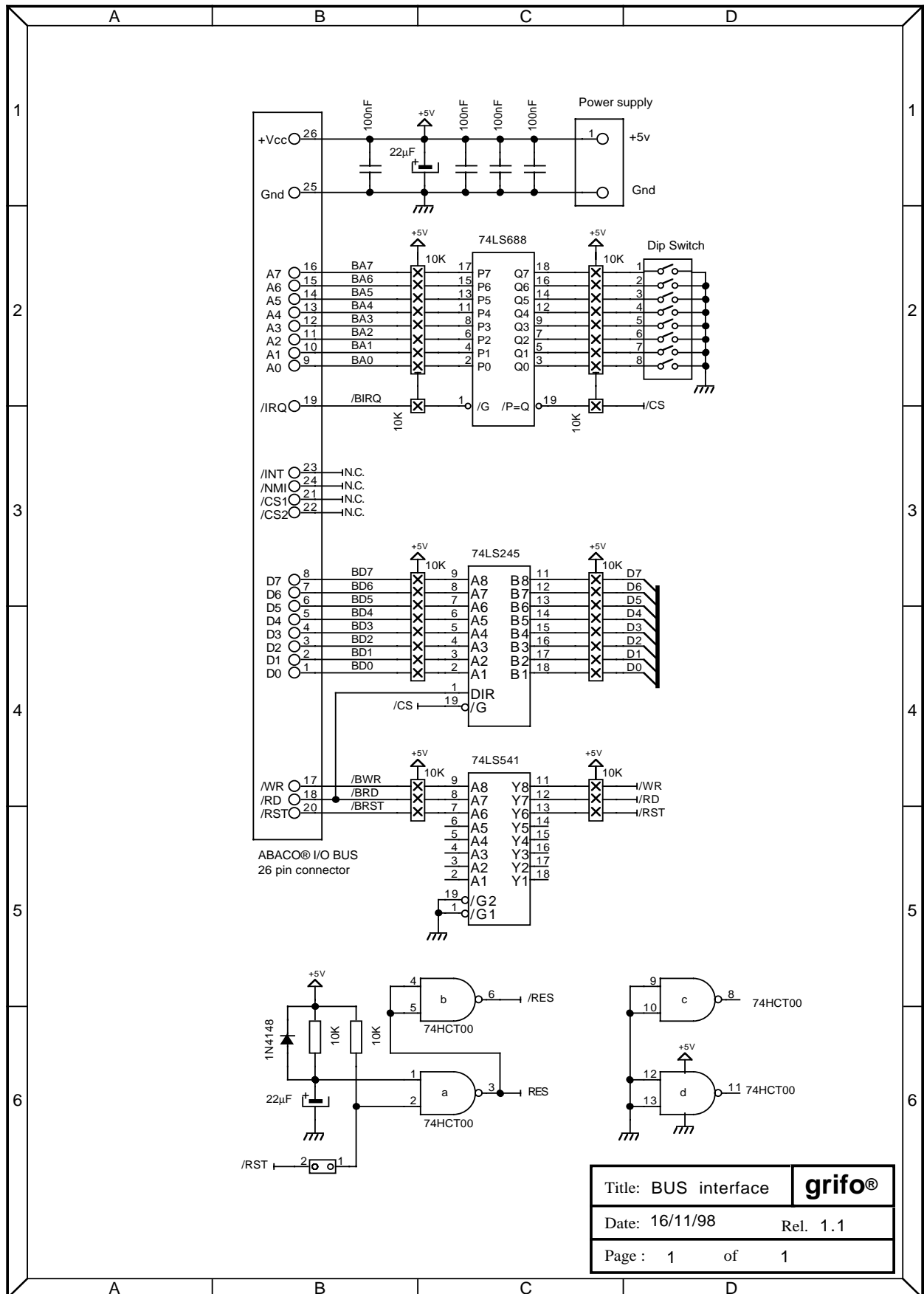
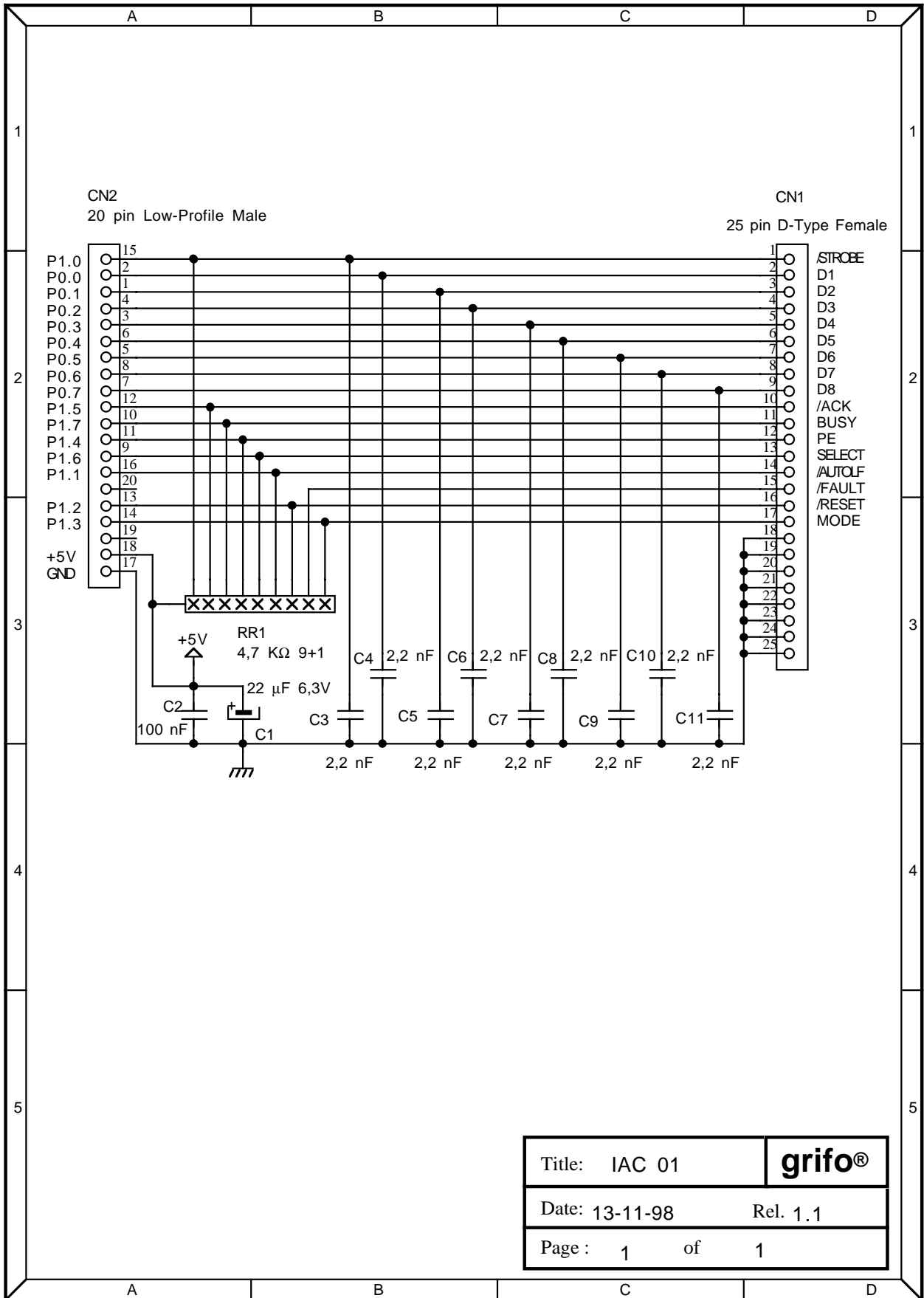


FIGURA C7: SCHEMA ELETTRICO INTERFACCIA BUS



Title: IAC 01	grifo®
Date: 13-11-98	Rel. 1.1
Page : 1	of 1

FIGURA C8: SCHEMA ELETTRICO IAC 01



APPENDICE D: INDICE ANALITICO

82c55 43, C-1

A

ABACO® I/O BUS 6, 10, 35, 42, C-5

Alimentazione 4, 8

Assistenza 1

B

Back up 8, 9, 26

Barre Ω A-1

Batteria 8, 9, 15, 24, 26

Bibliografia 44

C

Caratteristiche elettriche 8

Caratteristiche fisiche 7

Caratteristiche generali 2, 7

Clock 4, 7

Configurazione di default 22, 26, 28

Conessioni 43

Connettori 7, 14

CN1 10, 31, 35

CN2 9

CN3A 12

CN3B 14, 30

CN5 19, 20

Consumo 8

Contatto di reset 6, 31

Contenitore 1

Current loop 4, 14, 18, 28

D

DEBUG 4, 27

Dimensioni 7

Driver 29

E

EPROM 3, 7, 26, 36

Espandibilità 6, 42, C-1

F

FLASH EPROM 3, 7, 26, 36

Foto 11

G

Garanzia 1

H

Handshake 12, 30

I

Indirizzamenti 35

Ingresso configurazione 4, 24, 27, 41

Installazione 9

Interfacciamento 21

Interfacce C-1

Interfacce operatore C-3

Interrupt 10, 20, 31, 40

Interrupts 27

J

J4 4, 27, 41

Jumpers 22

2 vie 24

3 vie 25

5 vie 25

Jumpers a stagno 27

L

LED 15, 21

Linee di I/O C-1

Logica di controllo 6

M

Manutenzione 1

Mappaggio 35

ABACO® I/O BUS 35

I/O 37

memorie 36

Memorie 3, 7, 15, 25, 26, 36

Montaggio A-1

O

Opzioni 26, 28, A-1

P

P1 6, 31

Periferiche esterne 42

Periferiche interne 41, B-1

Peso 7

Pianta componenti 11

Piggy back A-1

Power failure 8, 10, 25, 31

Processore 3, 7, 41, B-1

Q

Quote A-1

R

RAM 3, 7, 26, 36

Real Time Clock 6, 39

Real time clock 21

Registri 38

Reset 6, 30, 31, 36

Risorse della scheda 7

RS 232 4, 12, 14, 16, 28

RS 422 4, 8, 14, 16, 25, 28

RS 485 4, 8, 14, 16, 17, 25, 30

RUN 4, 27

S

Schema a blocchi 5

Schemi elettrici C-1

Segnalazioni visive 21

Seriale A 12, 30

Seriale B 14, 28

Seriale sincrona 20

Seriali 4, 13, 21, 24, 28

Software 32

Specifiche tecniche 7

Stampante C-8

T

Temperatura 7

TransZorb™ 4

TTL 21

U

Umidità 7

V

Versione scheda 1

W

Watch dog 6, 7, 24, 30, 41

