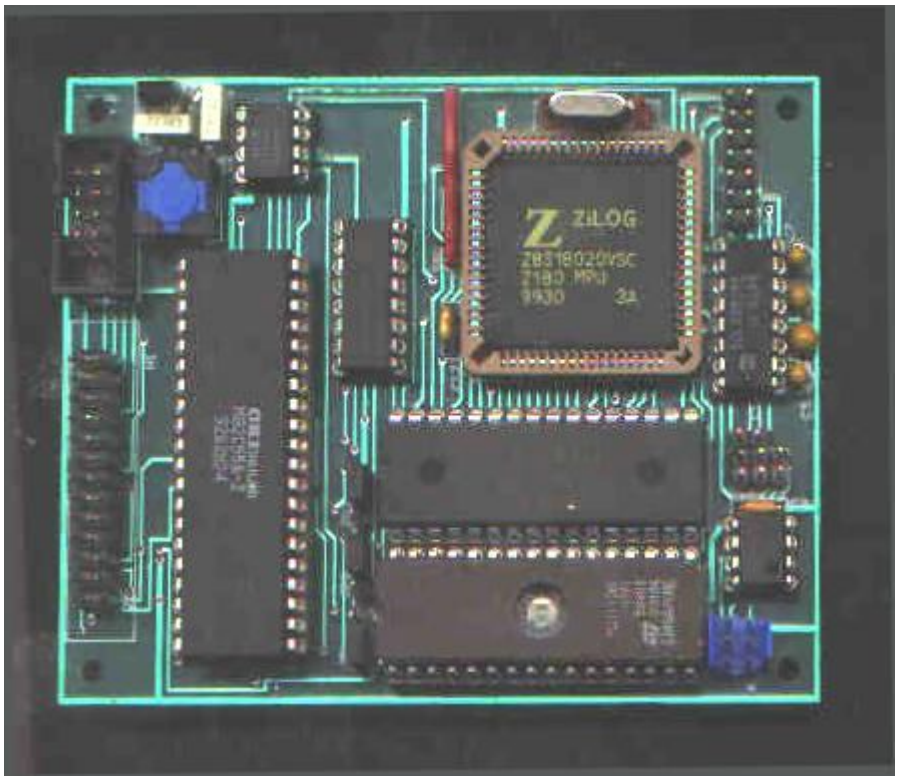
SINGLE BOARD COMPUTER ELC-180



INDICE

Questo manuale fornisce le istruzioni per il Single Board Computer ELC-180 della ELCOSYS. La scheda è basata sul microprocessore Z8S180 Zilog (versione statica dello Z180) con frequenza di clock 18.432 MHz e costituisce un ottimo hardware sia per lo sviluppo ed il test del software sia per l'impiego come scheda target nella fase applicativa. Viene fornito un programma Monitor / debugger che consente il caricamento e l'esecuzione di programmi assemblati in formato Intel hex da un PC.

Il programma di Debug monitor viene fornito su una EPROM da 16k x 8 (32k x 8).

1. Installazione

La scheda deve essere alimentata con una tensione continua compresa tra 7 e 12 VDC ed assorbe una corrente di circa 85 mA; deve essere collegata alla porta seriale di un PC o ad un terminale ASCII. Non sono richieste altre operazioni per l'impiego dell'ELC-180.

1.1 Connessione a terminale ASCII

Il connettore H3 della scheda deve essere connesso tramite un connettore standard DB-25 alla porta seriale RS-232C del terminale. La configurazione di default della scheda è 9600 baud, 8 bit dati, 1 bit di stop, no parity. Alimentando la scheda il terminale risponderà con il messaggio di apertura:

ELCOSYS

ELC-180 Single board computer

Zilog Z8S180 Debug Monitor Ver. 1.2

DEBUG MONITOR >>

1.2 Connessione a PC

Viene fornito con la scheda un programma di emulazione terminale IBM. Il programma TERM.EXE realizza una semplice emulazione di terminale e permette il caricamento di codice applicativo in formato Intel Hex. La scheda ELC-180 può essere collegata sia alla COM1 che alla COM2 di un PC. Digitando TERM si avvia il programma di emulazione (dopo aver connesso la tensione di alimentazione e il cavo seriale come descritto al par. 1.1) Apparirà un prompt '>' sul video del PC. Se non esiste ancora il file di configurazione 'TERM.CFG' il programma richiederà i dati di configurazione che saranno salvati in 'TERM.CFG'.

In alternativa, è possibile specificare i parametri della porta con la seguente linea di comando:

TERM p bbbb

dove p (=1 o 2) specifica la porta (COM1 o COM2) e bbbb è il baud rate. Come menzionato in precedenza, la scheda è configurata di default come 9600,N,8,1.

2 Hardware

2.1 Connettori

La scheda ELC-180 ha 4 connettori per l'alimentazione e l'interfacciamento di I/O :

H1: Connettore a 26 poli dela PPI 82C55 (3 porte 8 bit + uscita alim. GND,+5V)

H2: Alimentazione (7...12VDC, 85 mA) + pulsante di reset

H3: 2 Porte seriali RS-232C + 1 porta seriale sincrona

H4: 1 porta seriale RS-485 + uscita alim. GND, +5V, + segnale di controllo porta

2.2 Jumpers

Sono presenti 6 jumpers di configurazione che permettono la selezione del tipo di EPROM o FLASH installata e la configurazione delle porte seriali.

2.2.1 FLASH / EPROM

JP1 JP2 JP3 FLASH / EPROM

1-2 1-2 1-2 FLASH 4 Mb (512 kB x 8)

NC NC 1-2 FLASH 1 Mb (128 kB x 8)

2-3 1-2 1-2 EPROM 2 Mb (256 kB x 8)

2-3 2-3 1-2 EPROM 1 Mb (128 kB x 8)

NC 2-3 1-2 EPROM 512 kB (64 kB x 8)

NC 2-3 2-3 EPROM da 64 a 256 kb (da 8 kB x 8 a 32 kB x 8)

2.2.2 Porte seriali

JP4 JP5 JP6 RS232 / RS485

1-2 1-2 NC 2 canali RS-232C a 3 linee

2-3 2-3 NC 1 canale RS-232C a 3 linee + 1 canale RS-485

2-3 2-3 1-2 1 canale RS-232C a 5 linee + 1 canale RS-485

2.3 Memoria

Il decodificatore di indirizzi permette la gestione del massimo spazio di indirizzamento di memoria fisica (1 MB) consentito dallo Z8S180 e di uno spazio di I/O contiguo all'area dei registri interni non rilocati. Lo spazio di indirizzamento è dato dalla seguente mappa:

Spazio di Indirizzi di I/O

da 0000h a 003Fh riservato ai Registri Z8S180 (non rilocati)

0040h PPI porta A

0041h PPI porta B

0042h PPI porta C

0043h PPI porta di controllo

Spazio di Indirizzi fisici di memoria

Gli indirizzi fisici sono ottenuti tramite la MMU (memory management unit) dello Z8S180 che in base ai parametri imposti nei registri CBAR, CBR e BBR effettua la traslazione dello spazio di indirizzi di memoria logica (indirizzi di 16 bit da 0000h a FFFFh) allo spazio di indirizzamento di memoria fisica (indirizzi di 20 bit da 0:0000h a F:FFFFh). Nella

configurazione di massima memoria installata si ha:

da 0:0000h a 7:FFFFh EPROM / FLASH

da 8:0000h a F:FFFFh SRAM / NVRAM

La versione base della ELC-180 è equipaggiata con una EPROM da 16 kB x 8 (32 kB x 8) contenente il Debug/monitor e una SRAM da 128 kB x 8; in questo caso gli indirizzi di memoria fisica sono:

da 0:0000h a 0:FFFFh EPROM

da A:0000h a C:0000h SRAM

3. Debug Monitor

3.1 Messaggio di apertura

Se la connessione con il terminale o PC è corretta, premendo il pulsante di RESET sulla scheda apparirà il seguente messaggio:

ELCOSYS

ELC-180 Single board computer

Zilog Z8S180 Debug Monitor Ver. 1.2

DEBUG MONITOR >>

3.2 Comandi

Se viene digitato un comando errato il Monitor emetterà un beep mentre per annullare qualsiasi comando digitato occorre premere il tasto di escape. I valori numerici devono essere forniti in Hex; se viene digitato un valore non Hex il comando viene annullato. Si possono digitare un numero qualsiasi di cifre, il monitor estrarrà solo quelle necessarie a partire dalla meno significativa. Per uscire dal monitor occorre digitare <CTRL C>.

3.2.1 Alterazione di memoria - A

Il comando 'A' permette di modificare dei bytes nella memoria logica. Il monitor indica l'indirizzo, il dato corrente e la direzione del display. Quando viene digitato un byte esso viene scritto all'indirizzo indicato e l'indirizzo successivo nella direzione appropriata viene visualizzato. Il tasto t (toggle) cambia la direzione di scrolling La t non viene visualizzata, ma l'indicatore di direzione cambia di segno. Se il tasto t viene premuto dopo aver immesso un dato ma prima del <return>, il dato NON viene scritto in memoria.

Esempio:

DEBUG MONITOR >> Alter Memory From: 8000

8000 FF +

8001 FF + aa

8002 FF + t (non visualizzato e linea sovrapposta con la prossima)

8002 FF -

8001 AA - <esc>

DEBUG MONITOR >>

3.2.2 Inserimento di un breakpoint - B

Può essere inserito solo un breakpoint per volta. Il codice all'indirizzo del breakpoint fornito è sostituito con l'istruzione RST 038h (opcode 0FFh). Quando viene raggiunto il breakpoint, viene generato automaticamente un reset e vengono visualizzati i registri utente. Quando viene digitato <B> si ottiene:

Esempio:

DEBUG MONITOR >> Breakpoint At What Address? abcd

DEBUG MONITOR >> Go from Adress: 9000

A F B C D E H L A' F' B' C' D' E' H' L' I IX IY SP PC

FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00 FFFF FFFF 2000 abcd

3.2.3 Confronto di memoria - C

Confronta un dato numero di bytes di memoria e visualizza le differenze. Quando viene digitatato <C> si ottiene:

Esempio:

DEBUG MONITOR >> Compare Data At Address: 0000

With Data At Adress: 8000

How Many Bytes To Compare? 3

0000 = 18 : 8000 = 00

0001 = 3c : 8001 = FF

0002 = 00 : 8002 = FF

DEBUG MONITOR >>

3.2.4 Visualizzazione memoria - D

Visualizza bytes di memoria in formato Hex ed ASCII. La visualizzazione può essere sospesa digitando ^S. Digitando un ulteriore ^S si visualizza una ulteriore linea. Ogni altro tasto elimina la pausa. Il tasto <ESC> termina la visualizzazione dopo aver visualizzato la linea corrente.

Esempio:

DEBUG MONITOR >> Display Memory From : 0

How Many Bytes? 20

0000 18 3C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 *. < . . . . . *

0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 *. . . . . *

DEBUG MONITOR >>

3.2.5 Riempi celle di memoria - F

Riempie celle di memorie per uno specifico range di indirizzi con uno dato byte. Quando >F> viene premuto si ottiene:

Esempio:

DEBYUG MONITOR >> Fill Memory From: 9000

How Many Bytes? 28

With What Data? 55

DEBUG MONITOR >> Display Memory From : 9000

How Many Bytes? 30

9000 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 *UUUUUUUUUUUUUUUU*

9010 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 *UUUUUUUUUUUUUUUU*

9020 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 55 *UUUUUUUUUUUUUUUU*

DEBUG MONITOR >>

(ATTENZIONE: l'area di memoria da 8000h a 8400h è utilizzata dal Debug Monitor come area di lavoro. I comandi 'F' e 'M' non funzionano se applicati in questo range)

3.2.6 Comando GO - G

Esegue programmi. Se è fornito un indirizzo, esso viene utilizzato come Program Counter (PC) utente per eseguire il programma. Se non è dato un indirizzo, viene utilizzato il valore corrente di PC come indirizzo di partenza.

Esempio:

DEBUG MONITOR >> Breakpoint At What Address? 9008

DEBUG MONITOR >> Go From Address: 9000

A F B C D E H L A' F' B' C' D' E' H' L' I IX IY SP PC

FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00 FFFF FFFF 2000 9008

Se si prova ad eseguire il comando GO all'indirizzo del breakpoint viene visualizzato un messaggio di attenzione.

3.2.7 Help - H

Digitando 'h' si ottiene un help dei comandi:

DEBUG MONITOR >> Help Display

A - (A)lter memory from logical address

B - set a (B)reakpoint

C - (C)ompare logical addresses

D - (D)isplay logical memory

F - (F)ill logical memory

G - (G)o from logical address

H - (H)elp screen

I - (I)nput byte from 16-bit I/O address

L - (L)oad an Intel Hex file from the host

J - (J)ump to Device Driver Software

M - (M)ove memory blocks

O - (O)utput byte to 16-bit I/O address

R - display / modify cpu (R)egister

U - specify the (U)art delay

V - display the software (V)ersion

X - e(X)amine the memory map

DEBUG MONITOR >>

3.2.8 Ingresso da indirizzo - I

Effettua la lettura di un dato di 8 bit da un indirizzo di I/O a 16 bit. Quando viene premuto <I> si ottiene:

Esempio:

DEBUG MONITOR >> Input from a Port Address : 0040

0040 : 54

DEBUG MONITOR >>

3.2.9 Caricamento di un file Intel Hex - L

Questo comando è riconosciuto dal PC e dopo il return digitato alla fine della linea di comando il PC invierà il file richiesto alla scheda. In caso di errore occorre premere una sequenza di <esc> e <cr> fino alla emissione di un beep, per uscire dalla fase di caricamento.

Esempio:

DEBUG MONITOR >> l program.hex

DEBUG MONITOR >>

3.2.10 Spostamento di blocchi di memoria - M

Trasferisce lo specificato numero di bytes da un indirizzo all'altro.

DEBUG MONITOR >> Move Memory From: 3

To: 9000

How Many Bytes? 1

DEBUG MONITOR >>

(ATTENZIONE: l'area di memoria da 8000h a 8400h è utilizzata dal Debug Monitor come area di lavoro. I comandi 'F' e 'M' non funzionano se applicati in questo range)

3.2.11 Uscita ad indirizzo - O

Effettua la scrittura di un dato a 8 bit ad un indirizzo di I/O a 16 bit.

Esempio:

DEBUG MONITOR >> Output to a Port Address : c3

Data to Write : 00

DEBUG MONITOR >>

3.2.12 Visualizzazione e modifica registri - R

Viene richiesto il registro di partenza. Se non viene fornito nessun nome verranno visualizzati tutti i registri della CPU. Se viene fornito un nome, la visualizzazione parte da quel registro, ed il valore può essere modificato. <cr> muove il display al prossimo registro e <esc> termina la visualizzazione. Il registro PC non può essere modificato con il comanto 'R'.

Esempio:

DEBUG MONITOR >> Display/Modify Registers (cr for all) :

A F B C D E H L A' F' B' C' D' E' H' L' I IX IY SP PC

FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00 FFFF FFFF 6000 8000

DEBUG MONITOR >> Display/Modify Registers (cr for all) : h'

H' FF : 00

L' FF : 00

I 00 : <esc>

Display/Modify Registers (cr for all) :

A F B C D E H L A' F' B' C' D' E' H' L' I IX IY SP PC

FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00 00 00 FFFF FFFF 6000 8000

DEBUG MONITOR >>

3.2.13 Ritardo UART - U

Il monitor si avvia con un ritardo inserito prima di inviare qualsiasi carattere in uscita. Ciò permette anche ai PC più lenti la corretta visualizzazione dei caratteri. Nel caso di PC veloci o di terminali ASCII, questo ritardo può essere rimosso. La sola opzione del comando è Y (sì) o N (no).

Esempio:

DEBUG MONITOR >> Is padding needed for host comms? (y or n): No, UART is

full speed

DEBUG MONITOR >>

3.2.14 Numero versione - V

Digitando V viene visualizzato il numero di versione del monitor:

DEBUG MONITOR >> Z8S180 Monitor Version 1.2

DEBUG MONITOR >>

3.2.15 Analisi della mappa di MMU - X

Questo comando serve per comprendere il funzionamento della MMU (Memory Management Unit) del processore Z8S180. Dando i valori dei 3 registri di controllo della MMU (CBAR, CBR e BBR), si avrà la corrispondente mappa di memoria fisica.

Esempio:

DEBUG MONITOR >> X

Examine MMU register values

Value for MMU Common/Bank Area Register (CBAR: Addr 3Ah) ?84

Value for MMU Bank Base Register (BBR: Addr 39h) ?B4

Value for MMU Common Base Register (CBR: Addr 38h) ?F0

Above Mapped to

Logical addr Physical Addr

Common Area 1 : 08000h ==> F8000h

Bank Area : 04000h ==> B8000h

Common Area 0 : 00000h ==> 00000h

Se il valore fornito per CBAR nono è accettabile, allora compare un messaggio che ricorda che i 4 bit superiori di CBAR devono essere maggiori od uguali ai 4 inferiori. E viene richiesto un nuovo valore:

Value for MMU Common/Bank Area Register(CBAR: Addr 3Ah) ?0F

Upper nibble must be greater than or equal to Lower nibble!

Value for MMU Common/Bank Area Register(CBAR: Addr 3Ah) ?

A questo punto occorre immettere il nuovo valore per CBAR.

3.2.16 Altre caratteristiche

(a) NMI

Se si connette a massa la linea di NMI del processore si ha un rientro al monitor con il seguente messaggio:

NMI received !!!

A F B C D E H L A' F' B' C' D' E' H' L' I IX IY SP PC

FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF 00 FFFF FFFF 6000 8000

DEBUG MONITOR >>

(b) TRAP

Lo Z8S180 ha la peculiarità di generare un interrupt di tipo TRAP se viene eseguito un codice non definito, cioè non corrispondente ad una istruzione Z180. In questo caso, sia ha un rientro al monitor preceduto dal messaggio:

TRAP !!!

Undefined instruction found at location 9000

DEBUG MONITOR >>

APPENDICE A - Tabella connessioni ELC-180

H1 1 PA0

2 PA1

3 PA2

4 PA3

5 PA4

6 PA5

7 PA6

8 PA7

9 PB0

10 PB1

11 PB2

12 PB3

13 PB4

14 PB5

15 PB6

16 PB7

17 PC0

18 PC1

19 PC2

20 PC3

21 PC4

22 PC5

23 PC6

24 PC7

25 GND (OUT)

26 VCC (OUT)

H2 1 GND

2 GND

3 VDC (IN 7...12VDC, 85 mA)

4 VDC

5 /PBRST

6 *** unconnected ***

7 *** unconnected ***

8 *** unconnected ***

9 *** unconnected ***

10 *** unconnected ***

H3 1 GND

2 TX0

3 RX0

4 TX1/RTS0

5 RX1/CTS0

6 GND

7 RX1/CTS0

8 TX1/RTS0

9 /DCD0

10 GND

11 TXS

12 RXS

```
13 CKS

14 *** unconnected ***

H4 1 GND (OUT)

2 VCC (OUT)

3 A

4 B

5 PC7

6 *** unconnected ***

JP1 1 A18

2 S1

3 VCC

JP2 1 A17

2 S2

3 VCC

JP3 1 A15

2 S3

3 VCC

JP4 1 R2OUT

2 RXA1

3 RO

JP5 1 TXA1

2 T2IN

3 /RTS0

JP6 1 /CTS0

2 R2OUT

RN1 1 VCC
```

```
2 /NMI

3 /INT0

4 /INT1

5 /INT2

6 /BREQ

7 /WAIT

8 /DREQ1

9 *** unconnected ***

SW1 P$1 /PBRST

P$2 GND

P$3 *** unconnected ***

P$4 *** unconnected ***
```