L A B T A M

MODEL 3000 COMPUTER

SYSTEM MANUAL

Registered Office:

43 Malcolm Road
Braeside
Victoria 3195
Australia
Phone: (03) 587 1444

PREFACE

This manual is written for those user who are interested in the technical aspect of the 3000 computer. It is a essential guide for those who intend to write system software for the 3000 computer.

The manual describes the system programming interfaces and facilities, such as communication buffer structures etc. Example programs are given wherever necessary.

Hardware setup information are also included for the benefit of field technicians.

CONTENTS

SECTION 1 OVERALL SYSTEM ARCHITECTURE

1.1 INTRODUCTION

The Labtam 3000 is a multi-processor desktop computer that includes both an 8-bit (Z80A) and a 16-bit (8086) microprocessors chip in the same cabinet to offer the best of both worlds to their user. All present CP/M-80 8-bit applications software can operate side by side with the new 16-bit CP/M-86 and the low cost microprocessors share expensive memory, power supply and peripheral equipment. The basic system includes a 5MHz Z80A single board computer card (Z80 SBC card) which is used for disk I/O handling and 8-bit software processing, and a 8MHz 8086 VDU/COMM card which is used for high resolution screen handling and 16-bit software processing in single user environment. However, in multi-user environment, for example, MP/M-86. the system is configured with an extra 8086 processor card for dedicated 16-bit program processing, the Z80 card is used for disk I/O and handles one terminal, and the other 8086 video card manages the high resolution screen and another three terminals as well as two printers.

1.2 BUS STRUCTURE

Labtam 3000 is a multiprocessor system which allows each processor to work asynchronously with respect to each other. Every processor card has its own internal bus for communicating with its on board memory and I/O options, and uses the Multibus for referencing additional memory and I/O options. Local accesses do not require Multibus communication, making the system bus available for use by other Multibus masters. This feature allows true parallel processing in a multiprocessor environment.

1.3 DISK TRACKS BUFFERING

This scheme utilizes 48K bytes of Multibus memory to accommodate 6 tracks which are copied from the disk. The number of track buffered can be specified by the user up to a maximum of 14 tracks. This technique effectively reduces rotational latency by reading the entire track at a time. The entire disk I/O procedure is interrupt driven and use DMA for all data transfers supervised by a Z80 DMA controller. During reading or writing a track, the system first reads the address of the first available sector. The track is then read or written from the next physical sector onwards. This enables an average of 1-1/4 turns to read or write an entire track. The 'Write Back' method - deferred write until no disk I/O activity for certain period of time, is adopted in order to reduce the number of disk accesses and the 'Least Recently Used' algorithm is implemented to determine which track should be swapped betwen the disk and the main memory. All error handling, retries and recalibration of the drive heads is carried out automatically by the Z80.

1.4 I/O HANDLING

This scheme utilizes 16K bytes Multibus memory as a buffer to interface a processor to a wide range of peripherals and the user does not (and probably should not) be aware of the physical address assignments of the I/O ports. All I/O operation is entirely interrupt driven and is transparent to the user. If the user wants to access a certain I/O device or process, he should only have to manipulate the device control blocks in common memory. These control blocks can be addressed through the logical channel number which is the index to the control block. All the logical channel numbers and the detailed discussion on the character I/O buffering technique can be found in Section 4. This scheme not only provides a consistent I/O handling algorithm, it also enhances the parallel processing in this multiprocessing environment.

1.5 PARALLELISM

'Principle of Parallelism' is the design philosophy our system mainly based on, two examples are outlined as follows:-

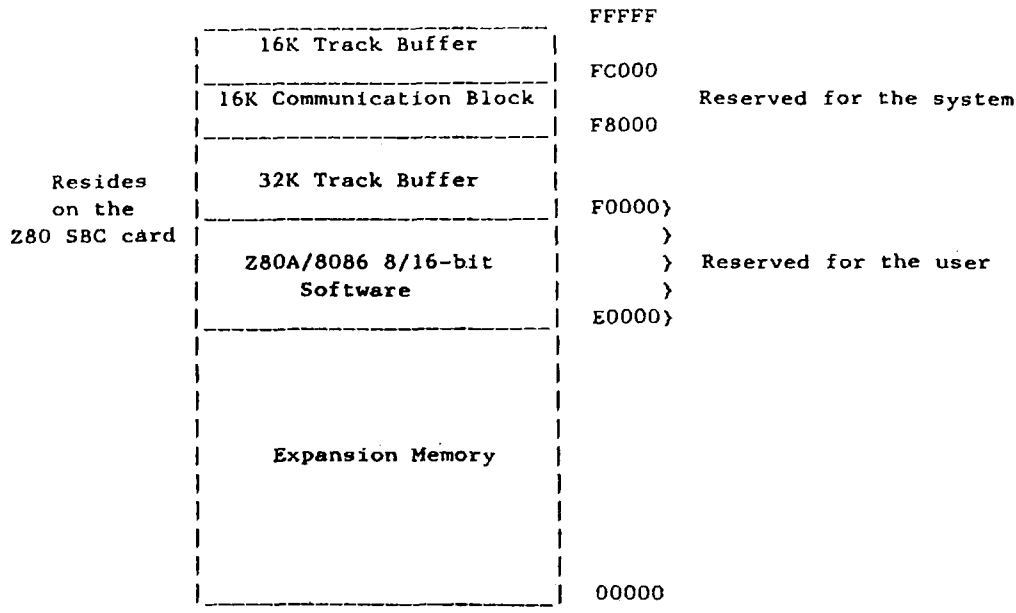
(i) Z80 is concurrently handling disk I/O activity ; executing user's program, attending one serial I/O channel and one data communication network channel, and handling real time clock request.

8086 is concurrently handling video display and keyboard interface, executing user's program, attending three serial I/O channels, and supervising all data transfers between the system and the parallel I/O channels (the output channel is used as parallel printer interface).

(ii) Automatic write back of 'dirty' disk tracks are occurred if no disk I/O request is taken place for 300 ms.

1.6 SYSTEM MEMORY MAP ORGRANIZATION

This system is designed basing on the Multibus architecture with 1M byte physical address spaces. The system memory map is organized as follows:-



At this stage, 128K bytes memory which is resided on the Z80A single board computer card is allocated for the Multibus memory, and is located at the top 128K bytes of the Multibus memory with the addresses range from E0000 to FFFFF. The top 64K bytes Multibus memory is reserved for the system use as the track and communication buffer. The remaining 64K bytes is used by the Z80A and 8086 to execute 8- and 16-bit software respectively. The relationship of the Z80A and 8086 processors with the Multibus memory are outlined as follows:-

Z80:-

The memory mapping hardware of the Z80A single board computer can map the 64K bytes Multibus memory with addresses range from E0000 to EFFFF as the user's area to run the 8-bit software. For example, when CP/M-80 programs are running on the Z80 processor, since all the CP/M functions can be handled by 8086 processor, only 2K byte of interface codes and data tables is required. The top 2K bytes of the user's area is reserved for this purpose and hence there is 62K bytes Transient Program Area made available for the user's software. This approximatly 10k bytes more TPA space then if the program was running under CP/M 2.2.

8086:-

The 8086 can access that 64K bytes Multibus memory by directly addressing its locations range from E0000 to EFFFF. For example, when CP/M-86 is running on the 8086 processor, CP/M-86 will occupy 16K bytes of local memory and hence the 64K bytes of Multibus memory as well as 32K bytes of local memory is available to the application software.

Graphics:-

This system provides a high resolution graphic capability in non-interlace mode (300 x 800 with 2 bits/pixel of intensity) and interlace (600 x 800 with 1 bit/pixel of intensity) raster scanning. The CRT controller generates the vertical sync and horizontal sync signals. It also generates a display enable signal which is fed to the count enable input line of the 16-bit display address counter. One of the five row address outputs from the CRT controller is used to determine the even or odd fields in non-interface mode. The video timing generator(PAL) generates all the timing for the video display memory, the shift register load pulses, and synchronizes the processor with the display memory and the CRT controller. The 64K words dynamic RAM is operated at page mode which allows for two-column addressing at the same row address. This enables 32-bits to be read from or written to the RAM in one memory cycle. Refreshing for the RAM is done automatically because the display cycle is executed frequently enough to satishfy the refresh requirment.

SECTION 2 MULTIBUS MOTHERBORAD

2 MOTHERBOARD (desk top unit)

For desk top units, the system motherboard has 6 Multibus slots with the priorities arranged in descending order (slot 5-slot 0), i.e., the lowest priority slot is the one closer to the video display unit and the highest priority slot is the one closer to the back panel. It also contains a PARALLEL priority bus exchange resolver which includes a priority encoder and decoder.

It also distributes power from the switch mode power supply (+24Vdc ,+/- 12 Vdc, +5Vdc and ground) to the disk drives (+24Vdc, +5 Vdc and ground), to the video display unit (+24Vdc and +12Vdc), the -5Vdc regulator which only supply to the Multibus, and the +/- 12V, +24V and +5V monitor lines to the rear panel LED displays.

2.1 BUS EXCHANGE TECHNIQUE

LABTAM 3000 is a multiprocessor system which allows each processor to work asynchronously with respect to each other. Multibus is used to hold all the hardware components together and since there can be many bus masters on a multi-master system bus, some means of resolving priority betyween bus master simultaneously requesting the bus must be provided. The PARALLEL priority bus exchange technique is used, which is based on a priority concept that at a given time one bus master will have priority above all the rest is implemented in the system. A parallel priority resolving technique has a separate bus request (BREQ\) line for each arbiter on the Multibus. Each BREQ\ line enters into a priority encoder which generates the binary address of the highest priority (BREQ\) line which is active at the inputs. The output binary address is decoded by a decoder to select the corresponding bus priority in (BPRN\) line to be returned to the highest priority requesting arbiter. The arbiter receiving priority (BPRN\ going low) then allows its associated bus master onto the multi-master system bus as soon as it becomes available. When one bus arbiter gains priority over another arbiter, it cannot immediately seize the bus, it must wiat until the present bus occupant completes its transfer cycle. Upon completing its transfer cycle, the present bus occupant recognizes that it no longer has priority and surrenders the bus, releasing BUSY\. BUSY\ is an active low OR-tied signal line which goes to every bus arbiter on the system bus. When BUSY\ goes high, the arbiter which presently has bus priority (BPRN\ low) then seizes the bus and pulls BUSY\ low to keep other arbiters off the bus. This multibus transaction is synchronized to the bus clock (BCLK) and allows for the parallel priority resolving circuitry time to settle and make a correct decision.

It must be noted that the detailed discussion of the Multibus elements and features could be found in the literature - Intel Multibus Specification (Manual Order Number: 9800683-02).

2.2 ARRANGEMENT OF CARDS

Z80A SBC card which is also used to handle the disk I/O process is assigned the highest priority in this multiprocessor system. Therefore, the Z80A SBC card is always plugged in the higher priority slot than the 8086 VDU/Comm cards. For example, the Z80A SBC card is plugged in No.4 slot and the 8086 VDU/Comm card is plugged in the No.2 slot. No. 2 being closer to video display unit in the desk top unit.

Only those cards which can drive the Multibus should need worry about its position relative to other masters. Any other cards, such as expansion memory etc., their positon on the motherboard is not important at all. However, even if the priorities of the masters are wrong, it can only degrade system performance, hardware damage can never occur.

2.3 MULTIBUS I/O PORT ASSIGNMENTS

In the basic system, all the boards are communicating via memory based buffers, there is very little requirements for Multibus i/o ports. The user should also consult the supplmentary documentations for other add-on i/o boards for their address assignments.

A channel attention port is implmented on the Z80 SBC card. This port is used for any external process to interrupt the Z80 processor, primary, this is used to signal the Z80 processor to perform required disk i/o or network communications.

Conversly, through this port, any external process can poll the status of the requested operation.

The following table summarizes this Multibus I/O assignments:-

Base address of this port is 00f0 (16 bit decoding used).

No	Name	Multibus Address	R/W	Comment
1	FDCCLR	Base +0	W	Reset floppy disk controller interrupt
2	NETCLR	Base +2	W	Reset NETWORK I/O channel interrupt
3	FDCATTN	Base +4	W	Inform Z80 to act on parameters put in Floppy Disk Controller or Network I/O buffers
4	FDCSTATUS	Base +8	R	Status of FDC and SIO interrupt Bit 0 = FDC Bit 1 = NETWORK I/O Bits 2 to 5 all ones.

where R/W = Read/Write

2.4 MULTIBUS INTERRUPTS

There are 8 interrupt lines (INT0 - INT7) available on the Multibus. Only INT3 is used (see section 2.3) in the basic system.

None of the other 7 lines are implemented in the basic system. Future additional processor will implement these lines, and make available for user applications. Consult factory for details.

SECTION 3 I/O DEVICES PROGRAMMING

3 INTRODUCTION

In order to achieve an expandable, consistant and flexable way to interface the 3000 processor to a wide range of peripherals, a communication buffer which is located at the top Multibus memory range , from F8000 - FBFFF was introduced. This section describes the existing I/O devices and the interprocessor communication.

All user i/o with standard 3000 hardware are performed via one of the following two ways,

a) direct access to i/o hardware, this may be diffcult due to 2 reasons,

i) requires different handling even only minor difference in hardware, interrupt handlying etc.

ii) the proccessor may not have direct or easy access of the hardware controlled by another processer.

b) access a set of generic buffers, which provides logically identical drivers, this minimize the effort reqired in handling bufferings, interrupts etc. Moreover, as high-level languages programming are becoming very popular, such buffers lend themself very easy to deal with in high level languages.

A block of 16K bytes of memory from F8000 - FBFFF is reserved in the Multibus for the communication buffers.

It should be noted here that the communications buffers were designed with the Zilog Z80 and Intel 8086 microprocessors in mind thus all multibyte values and pointers are little endian (least significant byte at the lowest address). This will only be of concern to those who wish to use big endian processors such as the Motorola 68000 in the LABTAM 3000. Also since mutual exclusion is required when accessing many of these buffers the processor card must have some method of locking the Multibus during the test and set of the free flags. Note that since the value one repesents free and zero busy, the M68000 TAS instruction cannot be used for this purpose.

3.1 TERMINOLOGIES

Throughout this section, all numeric constants are in hexadecimal values unless otherwise stated.

All relative pointers values are relative to the segment value of F000(sys_data_seg),i.e. absolute address F0000.

Positive logic is used throughout to name the semaphores, flags etc.

3.2 BUFFER SEMAPHORES

Semaphores are used to indicate the availability of associated buffer data, this is neccessary to resolve multi-processor access contentions. To access any data in a buffer which can be altered by another processor, the program must ensure that the buffer data is available by checking the semaphore.

In order to examine a semaphore, the LOCK prefix in 8086 code must be used to lockout the Multibus temporary to gain exclusive access. Similar lockout mechanism must be used for any other Multibus processors.

The following example loops until semaphore is free (i.e =1), before continuing to access data in the associated buffer,

```
LOOP:   SUB  AL,AL              ;CLEAR ACC, TO BE MOVE TO SEMAPHORE
        LOCK XCHG,SEMAPHORE     ;LOCK MULTIBUS, XCHG AL WITH SEMAPHORE BYTE
        TEST AL,AL
        JZ   LOOP
        access data
        MOV  SEMAPHONE,1        ;FREE BUFFER
```

It is highly recommanded that interrupt be disabled during buffer access to prevent dead lock between processes and resource hogging.

3.3 COMMUNICATION BUFFER CONTROL BLOCK

For any processor in the system that wishes to create buffer(s) for whatever reason, the buffer control block located at F8000 provides information regarding,

a) start of first free location in communication buffer block
b) end of free area

The buffer control block format is (at base address F8000)

```
base+0    free          ;control block free semaphore, 1=free
base+2    end_of_mem    ;2 bytes, end pointer of free area
base+4    start_of_mem  ;2 bytes, start pointer of free area
```

3.4 STANDARD I/O BUFFER ASSIGNMENTS

All the I/O devices and processes has a logical channel numbers which are the indexes to the communication buffer block. The following table specifies the logical process number of the existing I/O devices and processes in the system.

I/O devices or channel name	Logical channel number	
Z80 - process	0	
Floppy disk	1	
Network channel	3	
Real time clock	4	
8086 - process	8	
Video - console	9	
Serial printer	10	
Parallel printer	11	
Serial channel 1	14	
Serial channel 2	12	
Serial channel 3	13	
Serial channel 4	2	
8086 - MPM cpu	16	)
Serial channel 5	18	)
Serial channel 6	19	)
Serial channel 7	20	)only on 8086 MPM card
Serial channel 8	17	)
Parallel i/o	21	)

NOTE: all unused channel no. are reserved.
Early MP/M 86 Card (pre NOV,83) used channel no. 15 for CPU control.

A buffer pointer block is located at F8010, sequence of 4 byte blocks are used to hold the follwoing information for each logical channel,

byte 0,1	pointer to buffer-control-block for the channel
byte 2	channel type, 4 - disk 1 - serial i/o 2 - parallel i/o 3 - processor

Thus the Multibus address to the communication buffer can be obtained by the following formula:-

Pointer to Base Address of the Buffer = (F8010)+ (4xLCN)

where LCN = Logic Channel Number.

In the following sections, the format and access method of each buffer are described.

3.5 Z80 and 8086 PROCESS BUFFER

As there are multiple processors in the system, and they may be called upon by the master processor to execute a particular section of code, e.g. the 8086 processor loads in a section of Z80 code in certain area of memory, then it signals the Z80 processor to execute the code. Also during the execution, the master process can also stop the execution at any time.

The process buffer format is organized as follows:-

Attribute	Offset to base address
Address valid	0
Process running	1
Process stop	2
Filler	3 (not used)
Physical address	4 (8086 style Instruction Pointer: Segment) 4 bytes, offset:segment format.

Every processors in the system has a background task which regularly scans the associated buffer. The background task checks whether the "Address valid" has been set to high. If it is, the process would be initiated and executed from the starting address which is stored at the "Physical Address" field. The address valid field is cleared by the processor after starting the execution. The "Process running" field would be set to high in order to indicate the process is in execution mode. The process could be stopped by setting the "Process stop" field to high and waiting until the "Process running" field goes low.

NOTE: As it only a single Z80 processor which is running, the users program should not disable interrupts for longer than is absolutely required. Pre-empting the Z80 by disabling interrupts will disable the writing of "dirty" track buffers to disk as well as the real-time clock and serial channel.

Similarly, disabling interrupts on the 8086 Video card turns off most of the character I/O and the sceen updating.

3.6 DISK BUFFERS

The Z80 card handles all disk i/o functions, disk types includes 8" and 5.25" floppy disks using WD279X and 5.25 W/disk using WD1002 controllers. A buffer is defined in the communication block for disk i/o.

The disk buffer format is organized as follows:-

Attribute	Offset to base address	*Z80 Register	Comment
Flag	0		0=Idle FF=Busy
Command/Status	1	a	
Drive	2	c	Floppy = drive 0..3 Hard = drive 4..7
Cylinder (low Cyl)	3	b	
Side (head:high Cyl)	4	e	Floppy Side number 0..1 Hard D0..3 = M.S.Nybble of Cyl D4..7 = L.S.Nybble of head
Sector	5	d	
Sector Length)	6	l	
)	7	h	
)	8	ix	
32 bit linear)	9		
Address field)	10	iy	
)	11		
Error	16		
Dirty	17		
Changed	23		Advise changed state
RDY 0	24		)
RDY 1	25		)floppy disk ready state
RDY 2	26		)
RDY 3	27		
RDY 4	28		)
RDY 5	29		)w/disk ready state
RDY 6	30		)
RDY 7	31		)
TBL Free	32		
Buffers	34		no. of buffers
Buffer ptrs	36		) 28 bytes of buffer ptrs for max. of 14 buffers
Buffer headers	64		

*Z80 Registers - these are the register used when calling the disk controller functions in the Z80 kernal calls.

Drives 0 to 3 are defined as floppy disk drives (either 8 inch or 5.25 inch) and drives 4 to 7 are defined as hard disks (Winchesters). The meaning of the clyinder and side field changes between these two types of drives. For a hard disk these fields are low cylinder and head/high cylinder respectively. The low cylinder field is the least significant byte of the cylinder number and the head/high cylinder field is itself divided into two fields. The 4 highest bits of this field is the required head number and the 4 lowest bits are the 4 high order bits of the 12 bits cylinder number. The WD1002 Hard Disk Controller used in the 3000 can only support 3 drives with a maximum of 1024 cylinders and 8 heads.

The changed flag is set to 1 if any drive READY line changed. This and the individual flags can be used to detect disk changes.

The disk I/O process would be active if the "Flag" field in the buffer is set to active. This process is running with the highest priority in the system. Command can be issued through the command/status field and the response to that command also can be found at the field. The status fields for winchesters differ from that for floppy diskettes. The following table summaries the status for the commands.

for Floppy Disk

Bit	Name	Meaning
D7	ERROR	An error requires service.
D6	Write Protected	The disk was write protected.
D5	READY	The drive is ready. (An error only if not set)
D4	Record Not Found	The sector ID field returned was not found.
D3	CRC error	If D4 is set, an error was found in one or more ID fields; otherwise it indicates error in a data field.
D2	Overrun/Underrun	Data was not read/written to the WD279X in time.
D1	Seek Error	
D0	Not Used	

Also the drive, cylinder, side and sector fields contain the location of the sector in error, the low byte of the length contains the number of retries and the high byte of the length contains the last WD279X command executed.

for Hard disks

Bit	Name	Meaning
D7	ERROR	An error requires service.
D6	READY	The drive is ready. (An error only if not set)
D5	Write Fault	Hardware error.
D4	Seek Complete	The R/W heads have settled after a seek.
D3	Not Used	
D2	Corrected	The ECC corrected an error in the data field
D1	Not Used	
D0	ERROR	Same as D7.

Also the drive, cylinder, side and sector fields contain the location of the sector in error, the low byte of the length contains an additional error byte and the high byte of the length contains the last WD1001/WD1002 command executed.

The additional error byte definition is:

Bit	Name	Meaning
D7	Bad Block	The bad block bit in a sector ID read was set.
D6	Uncorrectable	Data field CRC error or > 5 bit error burst.
D5	ID field CRC Error	Sector ID field is unreadable
D4	Record Not Found	The sector ID returned was not found.
D3	Not Used	
D2	Aborted Command	Drive not ready, write fault or seek incomplete.
D1	Track 0 Error	Drive could not be restored in 1024 steps
D0	DAM not found	Could not find a valid data field

The first error byte uses some bit combinations to request additional error states. These are,

FF illgeal command
FE buffers not flushed. Drive went not ready with valid data still in buffer.

The disk function commands are summarised as follows:-

No.	Name	Description
0	Ignore Error	After an error status is returned this command causes the error to be ignored.
1	Retry Operation	After an error status is returned this command causes the operation that returned the error to be retried.
2	Abort Operation	After an error status is returned this command causes the operation that returned the error to be aborted. (N.B. The above three commands are NOPs if an error has not occurred. On the return of an error status one of these commands MUST be given.)
3	Reset Buffers	Clear all buffers. Any pending writes will first be performed. Also specifies:- 1) The number of buffers to use (min. 2, max. 14), in the Track No. byte (Offset 3, or Z80-b) -1 = default (6 tracks) 2) Offset 2(Z80, c)= If read after write verification is to be used, 0 = no verify 3) Offset 5(Z80, d)= If write through is required, 0 = no write through

4	Flush Buffers	Flush any pending writes from the track buffers.
5	Read Sector	Read a sector.
6	Write Sector	Write a sector. A preread of the track takes place. These two routines are the normal interface to the disk drives. The users program need not be aware that multiple track buffering is actually occurring.
7	Read Track	Read a track.
8	Write Track	Write a track. No preread of the track occurs. These two routines read/write tracks without going through the track buffers. They are used for fast diskette duplication and format verification.
9	Format Track	Format a track. Data is an image of the required format in the form required by the WD179X/279X family for floppy disks and that required by the WD1001/1002 family for hard disks. Data pointer is in buffer bytes 8 to 11.
10	Get Drive Specifications	Return the drive description tables in buffer, see summary following.
11	Put Drive Specifications	Replace the drive description tables in memory. Typically used before selecting a new disk type or formatting a diskette. The user should issue a Get Drive Specification command, modify only the parameters required and issue a Put Drive Specification command to update the controller tables.
12*	Read Drive Specifications	Reads the specified drive, cylinder and side in order to establish the diskette format and returns the default drive and disk specifications for that format. The specifications returned are as for get drive specification.
13	Seek Track	Seeks the head to the specified track. Used to exercise the heads for cleaning and to park the winchester disk.
14*	Raw Read	Reads the specified track including gaps, id fields and data fields with no error checking. Used to examine the format of unknown diskettes and for diagnostics.

* Not implemented for hard disk drives.

The drive description table is summarized as follows:-

Name	Offset to base adr.	Z80 regr.	Description
drvno	2	(c)	The drive number
trk	3	(b)	The current track (FF = track unknown, forces restore)
dens	4	(e)	The current density (0 = double density FF = single density)
drvtyp	5	(d)	The drive type (D7 = 8 inch, D6 = double sided, D5 = double tracking, D4 = Hard disk, D3 = write protected, D1 = changed, D0 = ready)
rate	6	(d)	The drive stepping rate
secsize	7	(h)	The current sector size for the drive
maxsec	8	(ix-1)	The highest sector number on the disk
minsec	9	(ix-h)	The lowest sector number on the disk
pretrk	10	(iy-1)	The first cylinder to begin precompensation.

Floppy disks: 0 = 3mS, 1 = 6, 2 = 10, 3 = 15, 4 = 6, 5 = 12, 6 = 20, 7 = 30

Hard disks: 0 = buffered seek, 1 = 0.5mS, 2 = 1.0, 3 = 1.5, 4 = 2.0, 5 = 2.5, 6 = 3.0, 7 = 3.5, 8 = 4.0mS, 9 = 4.5, 10 = 5.0, 11 = 5.5, 12 = 6.0, 13 = 6.5, 14 = 7.0, 15 = 7.5

0 = 128 bytes
1 = 256 bytes
2 = 512 bytes
3 = 1024 bytes

Floppy disks: Cylinder No.

Hard disks: Cylinder No. / 4

The sectors per track table is summarized as follows:-

Disk	128byte/sect	256byte/sect	512byte/sect	1024byte/sect
Single density 5.25"	18	10	5	2
Double density 5.25"	30	18	8	5
Single density 8"	26	15	8	4
Double density 8"	52	26	16	8

NOTE: Due to the 8K byte size of the track buffers, formatted track capacities of greater then 8K bytes cannot be used.

The default disk drive setup values are summarized as follows:-

Drive size	Stepping Time (ms)	Bytes/Sector	Sector/track
8"	3	1024	8
5.25"	3	1024	5

The supervisory calls use a comman buffer format to pass parameters except in a Z80 environment, where the registers are used to store the actual parameters and on exit the status is returned in register A.

In the case of another Multibus master requesting a disk I/O operation, the external processor has to fill in the appropriate fields in the disk buffer while the flag field is Idle, and set it to Active before output to the attention port. When the Z80 processor is interrupted, disk I/O operation will be performed through the system software. Once it is finished, the system software also clears the interrupt and set the Flag field back to Idle.

The Z80A single board computer card responds to a 16-bit I/O address from the Multibus. These I/O ports are used to signal the Z80A that a disk I/O command is available and to inform the requesting processor of the Z80 status.

The following table summarizes the Multibus I/O assignments for this:-

Base address of this port is 00F0 (16 bit decoding).

No	Name	Multibus Address	R/W	Comment
1	FDCCLR	Base +0	W	Reset floppy disk controller interrupt
2	NETCLR	Base +2	W	Reset NETWORK I/O channel interrupt
3	FDCATTN	Base +4	W	Inform Z80 to act on parameters put in Floppy Disk Controller or Network I/O buffers
4	FDCSTATUS	Base +8	R	Status of FDC and SIO interrupt Bit 0 = FDC - connects to MULTIBUS PIN 40 (INT 3) ** Bit 1 = NETWORK I/O Bits 2 to 5 all ones.

where R/W = Read/Write

** Bit 0 of FDCSTATUS is jumpered to pin 40 of Multibus (INT 3). This provides an optional interrupt signal to any other processor.

3.7 REAL TIME CLOCK

There is a battery backed-up real time calendar clock available in t system under the direct control of the Z80 processor. The state of this clo is duplicated in a buffer in the communication buffer area for any Multib master to read or set. The data in the buffer are updated once every 10th a sec.

The real time clock buffer is organized as follows:-

Attribute	Offset to base - address	Comment
Free flag	0	If equal 0, buffer not avail.
Change flag	1	D7=set time, D0=time has changed
100th & 10th of a Sec	2	)
Seconds	3	)
Minutes	4	)
Hours	5	) All the times are in packed BCI
Day of week	6	)
Day of month	7	)
Month	8	)
Year	9	)

To set the real time clock, check if Free flag (use Semaphore handling) is 1,then set Free Flag to 0 to obtain exclusive use of buffer. Set D7 and reset D0 of Change Flag, program all time registers, release buffer by setting Free flag. D0 of Change Flag will go to 1 after the real time clock is updated.

D0 of Change flag only get set every 10th of a sec. To sychronize wtih next 10th sec period, reset D0 and wait for it to go 1.

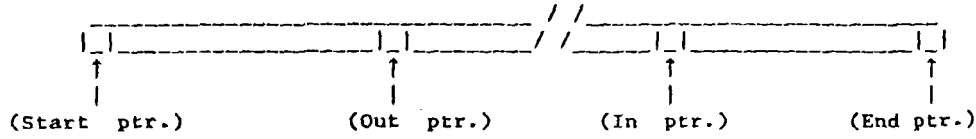
The real time counter format is organized as follows:-

Counter addressed	Units D0 D1 D2 D3	Max BCD Code	Tens D4 D5 D6 D7	Max BCD Code
100th and 10th of seconds	D0 D1 D2 D3	9	D4 D5 D6 D7	9
Seconds	D0 D1 D2 D3	9	D4 D5 D6 -	5
Minutes	D0 D1 D2 D3	9	D4 D5 D6 -	5
Hours	D0 D1 D2 D3	9	D4 D5 - -	2
Day of the week	D0 D1 D2 D3	7	- - - -	0
Day of the month	D0 D1 D2 D3	9	D4 D5 - -	3
Month	D0 D1 D2 D3	9	D4 - - -	1
Year (1978 = year zero)	D0 D1 D2 D3	9	D4 D5 D6 D7	9

3.8 DATA STRUCTURE FOR ALL CHARACTER I/O BUFFERS

The parallel and serial I/O buffer are used to interface between the processors and all character devices. The buffers used are of circular type with the following structure,

CIRCULAR BUFFER



(i)	Start pointer	- a pointer points to the beginning of the buffer
(ii)	End pointer	- a pointer points to the end of the buffer
(iii)	In pointer	- a pointer points to the next empty position in the buffer into which the next character should be placed.
(iv)	Out pointer	- a pointer points to the next character to be output.

The buffer size is allocated before run time by the appropriate processor responsible for the channel, separate buffers are allocated for both the input and output of each channel. Buffers cannot be reallocated by a user program.

When the next "In" pointer (In + 1) is equal to the "Out" pointer, this indicates that the buffer is FULL. On the other hand, while the "In" pointer is equal to the "Out" pointer, this indicates that the buffer is empty. When either the In-pointer or Out-pointer reaches the End-pointer, they will wrap around back to the Start-pointer.

A semaphore which is called "Free" is used in resolving conflicting access to the buffer. "Free = 1" means the resource is available and "Free = 0" means the resource is busy. This scheme can provide the system with the following advantages:-

(i) A consistent character I/O handling algorithm.

(ii) A type ahead capability - no interrupts to other processors for each character and fully automatic handshaking via software (XON/XOFF) or hardware (CTS/DTR), all these are transparent to user.

(iii) Extremely easy future I/O enhancement of the system.

3.8.1 SYCHRONOUS COMMUNICATIONS

This buffer scheme is suitable for both asychronous or sychronous communications. However, with sychronous communications, the following points must be noted,

a) The user software should not free the buffer when loading one block of characters into the buffer.

b) If the data length is longer then one buffer length, there is no guarantee that the I/O processor can maintain block continuity.

3.8.2 BUFFER CONTROL BLOCK

For all channels, there is a BUFFER CONTROL BLOCK which provides information about the states of a particular channel. The base address of this block can be obtained by the method stated in SECT. 3.4.

The block is divided into 3 parts,

1) input buffer block
2) output buffer block
3) Serial channel programming control and status.

Not all channels have valid data in the third block, e.g. the Parallel printer channel does not have the programming control and status blocks.

3.8.2.1 INPUT BUFFER CONTROL BLOCK

This is the block that describes the input buffer.

Name	Offset to Base-Address	Comment
i_free	0	1 => free ; 0 => busy
i_stopped	1	1 => free ; 0 => busy
i_in	2	next character goes here
i_out	4	next charcter from here
i_size	6	buffer size
i_buff_ptr	8	16 bit offset address of system buffer
i_buff_end	10	Offset of last byte in the buffer

The initiating processor will set the above fields in the buffer to the following default values:-

free = 1
stopped = 0
in = out = buff-ptr
size = 256 bytes
buff-end = buff-ptr + 255 bytes

Note that 'stopped' is the flag used to indicate whether the 'XON/XOFF'

control characters has been sent. If 'XON' is sent, 'stopped' will be set to '0' and if 'XOFF' is sent, 'stopped' will be set to '1'.

3.8.2.2 OUTPUT BUFFER CONTROL BLOCK

This is the block that describes the output buffer.

Name	Offset to Base-Address	Comment
o_free	12	1 => free ; 0 => busy
o_stopped	13	1 => free ; 0 => busy
o_in	14	next character goes here
o_out	16	next charcter from here
o_size	18	buffer size
o_buff_ptr	20	16 bit offset address of system buffer
o_buff_end	22	Offset of last byte in the buffer

The initiating processor will set the above fields in the buffer to the following default values:-

free = 1
stopped = 0
in = out = buff_ptr
size = 256 bytes (1K bytes for serial and parallel printers)
buff_end = buff_ptr + 255 bytes

Note that 'stopped' is the flag used to indicate whether the 'XON/XOFF' control characters has been received. If 'XON' is received 'stopped' will set to '0' and if 'XOFF' is received, 'stopped' will set to '1'.

3.8.3 SERIAL I/O PROGRAMMING CONTROL AND STATUS

All serial i/o channels used in the 3000 are either Zilog SIO or Intel 8274 (NEC 7201) multi-protocol USARTs. This block is used to program and maintains a image of the control and status registers of these chips.

Attribute	Offset to Base-Address	Comment
status_flag	24	Bit 0 & 1 indicates rr0 & rr1 are set or not respectively.
rr0	25	read register 0 status
rr1	26	read register 1 status
parameter_flag	27	indicates parameters have changed.
wr0	28	write register 0)
wr1	29	write register 1)
wr2	30	write register 2)
wr3	31	write register 3)Z80 SIO
wr4	32	write register 4) or
wr5	33	write register 5) Intel
wr6	34	write register 6) 8274
wr7	35	write register 7)format

rec_baud_rate	36	encoded (0-15) to (0-19200) baud
trx_baud_rate	37	encoded (0-15) to (0-19200) baud
cr_null_fill	38	No. of nulls to transmit after CI
lf_null_fill	39	No. of nulls to transmit after LI
out_stop_char	40	XOFF - output to device
out_rest_char	41	XON - output to device
inp_stop_char	42	XOFF - input to device
inp_rest_char	43	XON - input to device
inp_char_mask	44	defines which bits are valid
lower_limit	45	defines no. of char. left in output buffer for XON to be sent
upper_limit	47	defines no. of char. filled in output buffer of XOFF to be sent

At this stage, the buffering scheme is only suitable for operating in the asynchronous mode. The buffer must be initialized with the following information: character length (wr3;D7,D6 and wr5;D6,D5), number of stop bits (wr4;D3,D2), odd, even or no parity (wr4;D1,D0), and receiver (wr3;D0) or transmitter (wr5;D3) enable. When loading these parameters into the buffer, wr4 information must be written before the wr3, wr5 parameters/commands. The baud rate, number of nulls to transmit after CR or LF, XON, and XOFF fields also can be changed as wished. Parameter-flag field indicates whether the parameters have been changed. Rr0 and rr1 are the status fields and bit 0 and bit 1 of status-flag field indicates rr0 and rr1 are set or not respectively.

All serial i/o channel are initialized to (except keyboard input)

9600 baud, 8 data bit, 1 stop bit, no parity.

The user should refer to Z80 SIO or Intel 8274 Data Sheet for detailed information on the write registers and read registers.

3.8.3 CHARACTER I/O SAMPLE PROGRAMMING

This section describes the algorithms for

a) get a character from input buffer (GETCHAR)
b) put a character into output buffer (PUTCHAR)

All the descriptions following are heavly C Language flavoured, in particular, the assembler example routine is designed to interface with the C-Compiler by MARK WILLIAMS INC.

3.8.3.1 GETCHAR (get character from input buffer)

```
/* Buffer Full: In + 1 = Out,  Buffer Empty:  In = Out  */

GETCHAR                          /* Get a character from a buffer  */

{ 1: wait (free)                 /* Get semaphore and set "busy" */
     If (empty) {                /* Test whether the buffer is empty */
     Signal (free);              /* Set semaphore 'Available' */
     Goto 1;
               }
     char = *out;                /* Use resource; output the character which is
                                    pointed by the 'out' pointer */
     If (out ++>end) out=start;/* Wrap around situation in circular buffer */
     Signal (free)               /* Set semaphore 'Available' */
     }
```

Assembler routine, on entry, the logical channel number is in stack. Cross reference all names with buffer control block definition in previou sections.

```
                sys_data_seg   equ  f000h     ;segment adr. for comm. buffers
                ptrs_base      equ  8010h     ;base adr. for buffer control
                                              ;block pointers.
get_char::
          push bp                  /* Save caller's stack fram pointer*/
          mov  bp, sp              /* Point to old-bp */
          push si                  /* Save compiler register variable */
          push di                  /* Save compiler register variable */

          push ds                  /* Save Data-segment */
          mov  ax, sys_data_seg
          mov  ds,ax               /* (ds) = f000 */

one:      mov  di, 4[bp]           /* Get channel # from stack */
          shl  di, 1               /* Multiplied by 2 */
          shl  di, 1               /* Multiplied by 2 */
          mov  di, ptrs_base[di]           /* Pointer to base address of the
                                      buffer-control-blcok */
          pushf                    /* Push flags onto stack */+++
          cli                      /* Clear interrupt flag, disable int. */+++

two:      subb al, al               /* (al) = o */
          lock xchgb al, i_free[di] /* Obtain the current value of
                                       semaphore */
          Testb al, al
          jz    two                /* If busy, then wait */
          mov  si, i_out[di]       /* Assign si  as  a  'out' pointer */
          cmp  si, i_in[di]        /* Buffer empty? */
          jne char_avail
          movb i_free[di], 1       /* Set semaphone available */
          popf                     /* Pop flags off stack */+++
          jmp one                  /* wait */
```

```
char_avail:
        movb al, [si]           /* get the character which is pointed by
                                   the 'out' pointer */
        subb ah, ah             /* (ah)=0 */
        inc si                  /* Increment 'out' pointer */
        cmp si, i_buff_end[di]  /* 'out' pointer > 'end' pointer? */
        jle four
        mov si, i_buff_ptr[di]  /* Wrap around situation in circular
                                   buffer */

four:   mov i_out[di], si       /* Update 'out' pointer */
        movb i_free[di], 1      /* Set semaphore 'Available' */
        popf                    /* Pop flag off stack */+++
        pop ds                  /* Restore old environment */
        pop di
        pop si
        pop bp
        ret                     /* Return from procedure, character in AL*/
```

+++ these statements are necessary in any multiasking or interrupt driven system. These are used to prevent resource hogging and dead lock. e.g. In MP/M system, if a process gets buffer byte is then suspended for 1/4 sec, video card tries to open buffer but stops to wait for the MP/M process. All character i/o on the video card stops.

3.8.3.2 PUTCHAR (put a character into an output buffer

```
PUTCHAR                         /* Put a character into a buffer */

{ 1: wait (free)                /* Get semaphore and set 'Busy' */
     If (full) {                /* Test whether the buffer is full */
     Signal (free);             /* Set semaphore "Available' */
     Goto 1;
              {

     *in = char                 /*  Use resource; input a character and place to
                                   the position pointer by the 'in' pointer */
     If (in ++>end) in=start;   /* Wrap around situation in circular buffer */
     Signal (free)              /* Set semaphore 'Available' */
     }
```

Assembler code interfaced to C Compiler, cross reference all names with the structure definition of the buffer control block in previous sections.

On entry, the character and channel # are in stack respectively,

```
sys_data_seg    equ  f000h     ;data seg. for comm. buffers
pts_base        equ  8010h     ;base adr. for pointers
                               ;to buffer control block.

PUTCHAR:
        push bp                 /* Save caller's stack frame pointer */
        mov bp, sp              /* Point to old bp */
        push si                 /* Save compiler register variable */
        push di                 /* Save compiler register variable */
        push ds                 /* Save data-segment */
        mov  ax,sys_data_seg
        mov ds, ax              /* (ds) = f000 */

one:    mov di, 6[bp]           /* Get channel # */
        shl di, 1               /* Multiplied by 2 */
        shl di, 1               /* Multiplied by 2 */
        mov di, ptrs_base[di]   /* Pointer to base address of the buffer
                                   control block */
        pushf                   /* Push flags onto stack */+++
        cli                     /* Clear interrupt flag */+++

two:    subb al, al             /* (al)=0 */
        lock xchagb al, o_free[bi]  /* Obtain the curent value of
                                       semaphore */
        testb al, al
        jz two                  /* If busy, then wait */
        mov si, o_in[di]        /* Assign si as a 'in' pointer */
        mov bx, si
        inc bx                  /* 'bx' pointer = 'in' pointer + 1 */
        cmp bx, o_buff_end[di]  /* 'bx' pointer >'end' pointer */
        jle three
        mov bx, o_buff_ptr[di]  /* Wrap around situation in circular buffer

three:  cmp bx, o_out[di]       /* Buffer full? */
        jne four
        popf                    /* Pop flags off stack */+++
        movb o_free[di], 1      /* Set semaphore available */
        jmp one                 /* Wait */

four:   Mov ax, 4[bp]           /* get character from stack */
        movb [si], al           /* Place the character to the position
                                   pointed by the 'In' pointer */
        mov o_in[di], bx        /* Increment 'In' pointer */
        movb o_free[di], 1      /* Set semaphore available */
        popf                    /* Pop flag off stack */+++
        pop ds
        pop di
        pop si
        pop bp
        ret                     /* Return from procedure */
```

+++ See similar explanations in GETCHAR.

SECTION 4 Z80 SBC CARD

ARCHITECTURE

The Z80 SBC (Single Borad Computer) is an Multibus based card with following sub-systems,

a) Z80A Microprocessor Unit
b) local EPROM/RAM memory,
c) 128K byte Multibus memory,
d) floppy disk controller with DMA channel,
e) Winchester disk controller i/f with DMA channel,
f) LABNET high speed serial channel,
g) RS232C multiprotocol serial channel,
h) real time calendar clock with battery back-up,
i) Memory Management Unit for Z80A to access 1 Mbyte Multibus Memory.

The Z80A processor controls all of the above modules with an multitasking rvisor implemented in the local EPROM. The Z80A operates at 4MHZ clock with provision for 5 MHZ, it can address any 64K byte within the first 1 ibus memory, moreover, it can run any 8 bit Z80A program residing in any of this memory.

This board is part of the basic 3000 computer backbone which provides the >wing functionalities,

1) floppy disk controller (8" and/or 5.25" units),
2) 5.25" Winchester Disk controller (with external WD1002 controller card,
3) hardware driver controls for LABNET and an serial channel,
4) real time calendar clock manager
5) executioner of any 8 bit Z80A programs.

In the following sections, the functional descriptions of each sub-es will be described, plus programming information which are applicable ose who wishes to access these hardware in the Z80 environment. Hardware procedures and field adjustments are also documented for the benefit of technician.

To draw specific attention, there is a section devoted to the system available for the multi-tasking supervisor. In particular, for those who nterested in the following programmings in Z80,

a) character i/o to any serial port on 3000 basic computer,
b) access to an 32 bit timer,
c) disk controller functions.

4.1 I/O PORT ASSIGNMENTS

The following table summarizes the I/O port assignments under normal Z80 i/o operations.

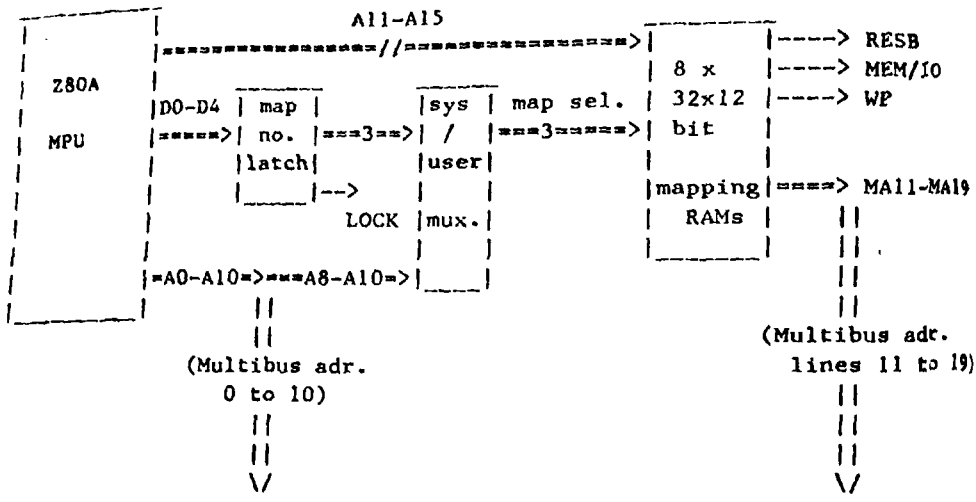
No	Name	Device	Address	Comment
1	fdcset	SR Flip Flop (U81)	00h	Set floppy disk controller interrupt.
2	serset	SR Flip Flop (U81)	08h	Set serial I/O channel interrupt
3	mapwr0	RAM (U52, U58)	10h	Write the memory mapping RAM
4	mapwr1	RAM (U40)	18h	Write the memory mapping RAM
5	intswt	Flip Flop (U91)	20h	Switch to user map
6	mapnum	D Flip Flop (U82)	28h	Memory map number
7	drive0	)	30h	Drive 0 selected
8	drive1	) 8 bit addressable	31h	Drive 1 selected
9	drive2	) Latch (U54)	32h	Drive 2 selected
10	drive3	)	33h	Drive 3 selected
11	fltrest		38h	Reset drive fault
12	fdccmd	)	40h	Status (RE)/Command (WE) Register
13	fdctrk	) WD 2793	41h	Track Register
14	fdcsec	) Floppy Disk	42h	Sector Register
15	fdcdat	) Controller	43h	Data Register
16	siodata	)	48h	Channel A Data
17	sidcmd	) Z80 sio/2	49h	Channel A Commands/Status
18	siobdata	) (U19)	4Ah	Channel B Data
19	siobcmd	)	4Bh	Channel B Commands/Status
20	ctc-data	)	50h	Data R/W, transfers communicate
		) AM9513 System		with int. reg.
21	ctc-cmd	) Timing Controller	51h	Control R/W, Status/command reg.
22	pic-data	) AM9519	58h	Data R/W internal reg. or memory
23	pic-cmd	) Interrupt Control	59h	Control R/W, Status/Command
24	hdata	)	60h	Data Register
25	herrvr	)	61h	Error(RE)/Write Precompensation (WR)
26	hscount	)	62h	Sector Count
27	hsnum	) WD1001	63h	Sector Number
28	hclow	) Winchester	64h	Cylinder Low
29	hchigh	) Disk Controller	65h	Cylinder High
30	hdrive	)	66h	Size/Drive/Head
31	hstatus	)	67h	Status (RE) /Command (WR)
32	drvstatus	8 bit latch (U60)	68h	
33	maprdo	RAM (U52, U56)	70h	Read the memory mapping RAM
34	maprd1	RAM(U40)	78h	Read the memory mapping RAM
35	fdcdma	Z80A DMA0 (U30)	80h	Floppy disk DMA controller
36	siodma	Z80A DMA1 (U35)	a0h	SIO DMA controller

37	rtc-count1)		e0h	Counter-ten thousands of seconds.
38	rtc-count2)		e1h	Counter-Hundredths & 10th of Sec.
39	rtc-count3)		e2h	Counter-Seconds
40	rtc-count4)		e3h	Counter-Minutes
41	rtc-Count5)		e4h	Counter-Hours
42	rtc-counts)	MM58167A	e5h	Counter-Day of week
43	rtc-count7)	Real time clock	e6h	Counter-Day of Month
44	rtc-count8)	(U3)	e7h	Counter-Month
45	rtc-ram1)		e8h	RAM -)
46	rtc-ram2)		e9h	RAM -)
47	rtc-ram3)		eah	RAM -) Reserved
48	rtc-ram4)		ebh	RAM -)
49	rtc-ram5)		ech	RAM -)
50	rtc-ram6)		edh	RAM -)
51	rtc-ram7)		eeh	RAM -) Reserved
52	rtc-ram8)		efh	RAM -)
53	rtc-instat)		f0h	Interrupt Status Register
54	rtc-intcon)		f1h	Interrupt control Register
55	rtc-reset1)	MM58167A	f2h	Counter Reset
56	rtc-reset2)	Real Time Clock	f3h	RAM Reset
57	rtc-stat)	(U3)	f4h	Status Bit
58	rtc-go)		f5h	GO Command
59	rtc-Standint)		f6h	STANDBY INTERRUPT
60	rtc-test)		ffh	Test Mode

4.2 MEMORY MANAGEMENT OF Z80 SBC CARD

A special feature of the Z80 SBC card is the memory mapping hardware. This unit allows the Z80A to address any 64K memory in the Multibus first 1 Mbyte addressing space. The method employed is PAGING, any 2K pages of phyical memory can be linked together through the definition of the map content of a bank of 256 x 12 bits high speed bipolar RAMs. The content of the mapping RAM unit can be altered through i/o instructions. The advantage of PAGING method allows flexible manipulation of non-continuous physical memory without any speed degradation.

The top five logical address lines (A11-A15) from Z80 are mapped by the memory mapping circuitry. The lower eleven logical address lines (A0-A10) from Z80 and physical address lines from Multibus are identical and form the within-page address. Logical address are then mapped in 2K pages.



Z80 MEMORY MANAGEMENT UNIT

Four extra address lines which are generated by the mapping extend the physical addressing capability to 1M bytes. The Z80A can still only address 64K of logical address space at any one time, however, this 64K can be obtained by selecting any combination of thirty-two 2K physical pages out of a total of 1M and these thrity-two physical pages can be rearranged at will within the 64K logical address space by software in Z80A. The map RAM also produces three attribute bits which are as follows,

(i) RESB - Resident bus select
(ii) MEM/IO - Memory/IO (1 for Memory access, 0 for I/O access)
(iii) WP - write protect bit

The RESB bit selects local memory access. The MEM/IO bit allows the Z80 to access either Multibus memory or I/O ports (64K ports). The WP bit disables any Multibus memory access when activated.

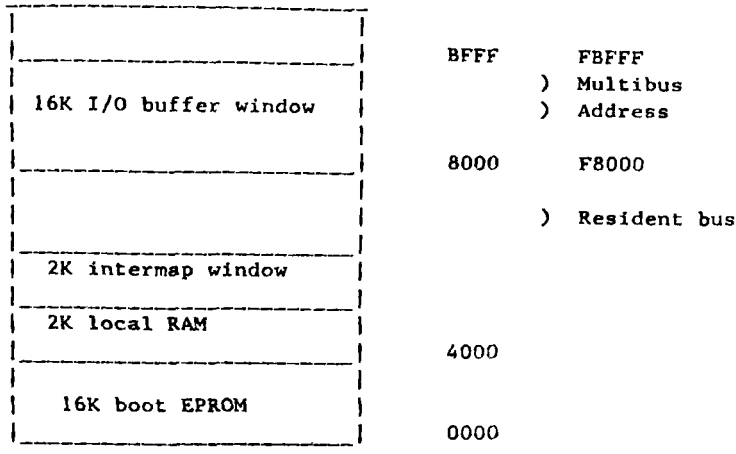
These 3 attribute bits together with the base address (A11-A19) of each 2K pages are programmed into the memory mapping RAM. Altogether 8 different maps are available.

The MUX unit (U73 and U68) multiplexes the map number selections,

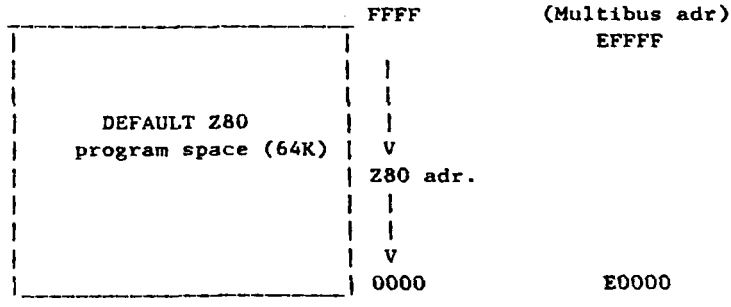
a) system map (MAP 0) selected automatically by interrupt cct.,
b) user map no. latched in U82,
c) DMA map (MAP 7) selected automatically by DMA acknowledgment cct,
d) map no. selected for writing data into, this is relying on the Z80 characteristics of putting register BC pair onto the address line when performing register indirect i/o.

Map 0 and 7 are reserved for system use, Map 1 is used for running 8 bit CPM programs. The mapping of these are described below. Maps 2 - 6 are not initialized and available for user applications.

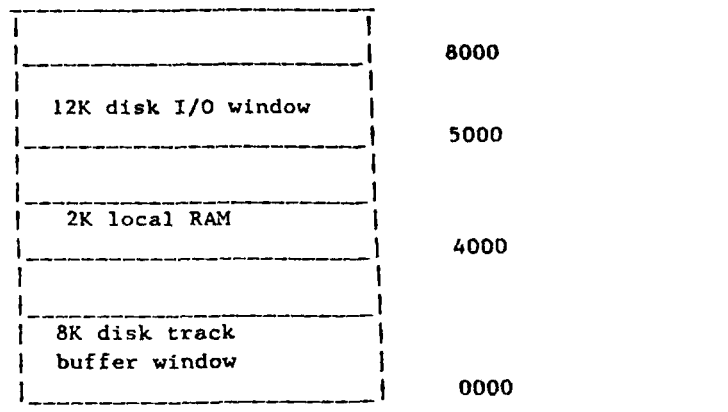
MAP 0 - SYSTEM MAP



MAP 1 - DEFAULT USER MAP FOR 8 BIT PROGRAMS



MAP 7 - DMA MAP



4.2.1 MEMORY MAP PROGRAMMING

The content of the map RAM can be read and written through the following I/O port physical address:-

I/O Port Name	Address	RAM Contents	Comments
MAPRD0	70h	MA 11,12,13,14,15,16,17,18	Read the memory mapping RAM (U52, U58)
MAPRD1	78h	Map No. MA19, WP,MEM/IO RESB	Read the memory mapping RAM(U40) and D F/F(U82)
MAPWR0	10h	MA 11,12,13,14,15,16,17,18	Write memory mapping RAM(U52, U56)
MAPWR1	18h	(lower 4 bit not used) MA 19, WP, MEM/IO, RESB	Write memory mapping RAM (U40)

The memory mapping hardware relies on the Z80 characteristic of putting the register pair B and C on the 16 address lines during register indirect I/O to avoid address multiplexers for the upper address lines. Therefore when programming the contents of the mapping RAM, the map no. must be in register B. A typical routine to map 64K bytes Multibus memory into map is described as follows:-

To load the memory mapping rams for a specified map number (USRMAP) with data for the logical address in HL, the DE pair is the data derived from the base address of the physical memory, using the following rule.

a) ignore A0 - A10 of the physical address,
b) load A11 - A18 into register E,
d) load A19 into bit 4 of register D,
e) set bit 5 - 7 of register D according to values for WP, MEM/IO and RESB respectively,
f) lower 4 bit of register D is don't care.

```
Entry:  b  = map number
        de = map ram data
        hl = logical address in map USRMAP
Exit:   de, hl are unchanged

     MAPMASK   EQU  07H
     MAPWR0    EQU  10H
     MAPWR1    EQU  18H

                                ;assumes de has map ram data
          ld   b, USRMAP
          ld   hl,0             ;starts at logical adr 0
          call lddata                ;map into map USRMAP


nxtusr:   inc  e                     ;increment low address
          jr   nz,nxturl
          ld   a, 10h
          add  a,d                   ;increment A19 if overflow
          ld   d,a

nxturl:   ld   a, MAPMASK +1
          add  a,h                   ;increment hl by 2k
          ld   h,a
          ld   b, USRIMAP
          call lddata
          ld   a, h
          cp   of8h                  ;mapped last block
          jr   nz, nxturs            ;no


lddata:   ld   a,h                   ;MSB of address in current map.
          and not MAPMASK
          or   b                     ;merge with map number
          ld   b,a
          ld   c, MAPWR0
          out (c), e
          ld   c, MAPWR1
          out (c), d
          ret

          end
```

4.2.2 Z80 ACCESSING MULTIBUS I/O PORTS

The Z80 processor can only directly address i/o ports as those documented in section 4.1 using the normal i/o instructions. For accessing any Multibus i/o ports, the memory management unit is required.

First assign an window in the 64K Z80 address space (in 2K blocks) for this purpose. Set up a map (or modify the normal user map i.e. MAP 1) so that the Multibus i/o address (16 bit) is mapped as if it is memory address within that window. However, the MEM/IO attribute of that window must be set to 0.

Once this is set up, the Multibus i/o port can be addressed as an memory location within that window.

4.2.3 HOW TO SELECT MAP

To bring any one of the 6 user maps (MAP 1 - 6) into system, simply output the map number to port MAPNUM (028h).

If a multiple number of map is required to be switched in and out often, the following example provides the fastest possible method.

First, initialize the auxillary registers of the Z80 registers with the following parameters,

```
MAPNUM    EQU    028h      ;map select latch port adr.
USER1MAP  EQU    01        ;map number 1
USER2MAP  EQU    02        ;map number 2
USER3MAP  EQU    03        ;map number 3
USER4MAP  EQU    04        ;map number 4

mapinit:  :      :
          :      :
      program mapping ram with data
          :      :
          exx                    ;switch to aux. register set
          ld   c,MAPNUM          ;load map sel. port adr. into C'
          ld   d,USER1MAP        ;ld map nos.  into D',E',H' and L'
          ld   e,USER2MAP
          ld   h,USER3MAP
          ld   l,USER4MAP
          exx
          ret


    ;select map 1
sel1:     exx
          out  (c),d
          exx
          ret
```

And similarly for selecting the other maps.

4.3 REAL TIME CALENDAR CLOCK

The real time calendar clock implemented on the card is the MM58167 clock chip. The device includes an addressable real time counter, 56 bits of RAM, and two interrupt outputs. The backup batteries allows the device to be disabled from the rest of the system for standby low power operation. The addressable portion of the counter is from 1/10,000 of a second to months. The RAM is formatted the same as the real time counter, i.e. 4 bits per digit, 14 digits. The first of the two interrupt outputs is the 'Interrupt Output'. It can be programmed to provide 8 different output signals. They are: 10Hz,

1Hz, once per minute, once per hour, once a day, once a week, once a month and when a RAM/real time counter comparison occurs. The second interrupt is 'Standby Interrupt'. This interrupt occurs when enabled and when a RAM/real time counter comparison occurs and is not connected on the 3000.

The real time clock can be accessed through the communications buffer using other Multibus micrprocessors or I/O ports from the Z80. The user should refer to section 3.7 for external access via communication block in Multibus memory. The section only discuss the access method via Z80A.

Later in this section, the calibiration procedure of the real time clock and battery replacement procedure are also discussed.

4.3.1 SAMPLE PROGRAMMING FOR ACCESSING THE REAL TIME CLOCK

This section, we only discuss the direct i/o accessing of the clock chip using Z80 assembler programming. There is also a 32 bit timer counter accessible via the resident multi-tasking supervisor call, see section 4.5.2.

For better understanding, the user is suggested to read the documentation published by NATIONAL SEMICONDUCTORS for the MM58167A clock chip.

I/O port definitions for the clock chips are,

```
RTC.1SEC    equ  0e1h     ;adr. for 1/100 and 1/10 sec counter
RTCSEC      equ  0e2h     ;adr. for sec. counter
RTCMIN      equ  0e3h     ;adr. for min. counter
RTCHOUR     equ  0e4h     ;adr. for hour counter
RTCDAY      equ  0e5h     ;adr. for day of week counter
RTCDATE     equ  0e6h     ;adr. for day of month counter
RTCMONTH    equ  0e7h     ;adr. for month ctr.
RTCSTS      equ  0f4h     ;adr. for status port, (bit 0 = 0 for
                          ;         stable data)

          ; all address in between 0e7 - 0f4 plus 0f5 - 0ff are used
          ;by the clock chip.  Read MM58167A documentation.
```

All counters of the real time clock are in BCD format, refer to section 3.7 or MM58167A documentation for details. To read any counter, simple address the counter, however, validity of data must be checked by testing bit 0 of the status port (port 0f4). When this bit is 0, the data just read is stable and valid.

e.g. to read the data for the hour counter,

```
rdhour:     ld   c,RTCHOUR
            call rdrtc                  ;read read time counter
            :      :
            :      :
            ret

rdrtc:      in   b,(c)                  ;read counter data, adr. in C
```

```
in   a,(RTCSTS)          ;read status port
bit  0,a                 ;check bit 0 of status port
ret  z                   ;ret. if data stable, data in a
jr   rdrtc               ;data not valid, read again.
```

4.3.2 CALIBRATION OF REAL TIME CLOCK

1) You will require a frequency meter and trimmer capacitor adjustment tool.
2) Connect frequency meter probe to pin 12 on I.C. 3.
3) Adjust C16 to give a frequency of 10Hz.
4) Check voltage at pin 24 on I.C. 3: it should be between 2 and 3 volts d.c. If voltage is too low, the batteries may need replacing or may simply need recharging. Leave unit running for 24 hours to recharge batteries.
5) Utility routines running under CP/M and LABDOS will set the actual day, date and time of the real time clock.

4.3.3 BATTERY REPLACEMENT FOR REAL TIME CLOCK

The life of the back-up battery is dependent on the power on/off pattern of any particular computer, however, it is resonable to assume that a battery life of 1 - 3 years for normal usage pattern. When the real time clock starts to go irregular after power off, it is time to change the battery. The battery used are Nical rechargeable types, 1.25V 40mA-Hour, two required.

4.4 Z80 MULTITASKING SUPERVISORY FUNCTIONS

The following supervisory calls are available for the system programmer:-

Name	Number	Comment	
charmgr	0	Character manager	
timer	1	Return time value	
newtask	4	Start a new task	
kill	5	Delete task	
newcond	6	Allocate a condition variable	
delcond	7	Deallocate a condition variable	
entermon	8	Enter critical region	
exitmon	9	Exit critical region	
empty	0a	Test if condition empty	
signal	0b	Signal to a waiting task	
wait	0c	Wait for a signal	
dispatch	0d	Dispatch next ready task	
exit	0f	Kill current task	
no-fer	10	Ignore error	)
no-xfer	11	Retry last operation	)
no-xfer	12	Abort	)
resbuff	13	Reset buffering	)
flshbuff	14	Flush buffers	)
secread	15	Read sector	) Floppy and
secwrite	16	Write sector	) Winchester disk I/O
trkread	17	Read track	)
trkwrite	18	Write track	)

```
trkformat   19      Format track                    )
gtspec      1a      Get drive spec.                 )
wrspec      1b      Specify disk format             )
dskid       1c      Read format from disk           )
rdspec      1d      Read drive spec.                )
rawrd       1e      raw read                        )
```

Only the commonly required ones are documented in the following sections, these include charmgr, timer and the disk controller calls.

All supervisory calls are done via this macro,

```
SVC     macro   TYPE        ;TYPE is the call number
        di
        ld      a,05Fh
        out     (59),a      ;generates a Z80 interrupt
        ld      a,TYPE
        ei
        halt

        endm
```

All other parameters are passed via the registers.

4.4.1 CHARACTER I/O MANAGER

The character manager performs all character i/o operation for all the serial or parallel character i/o channels on the 3000 computer. The channel assignments are,

channel	number (decimal)	
Serial channel 1	14	
Serial channel 2	12	
Serial channel 3	13	
Serial channel 4	2	
Serial printer (send only)	10	
parallel printer (send only)	11	
Serial channel 5	20	)
Serial channel 6	19	)
Serial channel 7	18	) on MP/M 86 card only
Serial channel 8	17	)

For Z80 programs, this is the easiest means to do character i/o for most channels, except Serial channel 4. Serial channel 4 is on board and can alternatively be accessed via direct i/o.

A number of functions are available under the character i/o manager call, the function number is passed via register B, register E holds the channel number. The following table details the functions available,

FUNCTION	NO.	PARAMETERS
Output character	0	C = character to be output
Output status	1	status returned in A, 0 = output buffer full Off = output buffer available
Input character	2	charater returned in A
Input status	3	status returned in A, 0 = input buffer empty Off = character available
Device present status	4	status returned in A, 0 = channel does not exist Off = channel exists
Clear output buffer	5	
Clear input buffer	6	

4.4.1.1 SAMPLE PROGRAMMING FOR CHARACTER I/O MANAGER

The following example routines are for

a) getting a character from serial channel 1
b) sending a character to serial printer

a) GETCHAR ;get a character from serial channel 1

```
SER1     EQU     0eh      ;channel # for serial 1
CHARMGR  EQU     0        ;supervisory call #
INSTS    EQU     3        ;input status function #
GETC     EQU     2        ;input character function #


SVC      macro   TYPE     ;TYPE is the call number
     di
     ld   a,05Fh
     out  (59),a
     ld   a,TYPE
     ei
     halt
          endm

GETCHAR:  ld   e,SER1
          ld   b,INSTS     ;see if any character avail.
          SVC  CHARMGR
          and  a
          jr   z,GETCHAR   ;loop if no character
          ld   e,SER1
```

```
                ld   b,GETC
                SVC  CHARMGR        ;now get character
                ret                 ;character returned in A
```

b) PUTCHAR ;send character to serial printer

```
SERPTR    EQU    0ah   ;serial printer channel #
CHARMGR   EQU    0
OUTSTS    EQU    0
OUTC      EQU    1

SVC       macro  TYPE  ;TYPE is the call number
          di
          ld   a,05Fh
          out  (59),a
          ld   a,TYPE
          ei
          halt
          endm
```

; on entry, assume character already in C

```
PUTCHAR:  push bc               ;save character to be sent
PUT0:     ld   e,SERPTR
          ld   b,OUTSTS         ;see if output buffer empty
          SVC  CHARMGR
          and  a
          jr   z,PUT0           ;loop if buffer full
          pop  bc               ;restore char.
          ld   e,SERPTR
          ld   b,OUTC
          SVC  CHARMGR
          ret
```

4.4.2 32 BIT TIMER MANAGER

There is a constantly running 32 BIT timer available for time keeping purposes, each unit of this timer is 50 mS. The user can set the initial value of this timer, only reading is possible.

To read this timer, just use the macro call SVC 1. Registers D,E,H and L will have the value of the timer returned, DE pair being the most significant part of the timer value.

4.4.3 DISK CONTROLLER SUPERVISORY CALLS

Supervisory call no. 16 to 30 (10 - 1E) are for the disk controller functions, these functions correspond to those disk function commands 0 - 14 detailed in section 3.6.

All parameters are passed via the Z80 registers as that documented in section 3.6. In fact the operations of the 2 sets of functions are identical, only difference is means of activating the functions.

4.5 CALIBRATION OF FLOPPY DISK CONTROLLER CHIP

This calibration method is used to adjust the data separater timing of the floppy disk controller circuitry. This should only be attempted by personnal who have some experience with floppy disk drives.

Note: This calibration should be done when the unit is COLD!

1) You will require a C.R.O., trimpot adjuster and a boot prom for the Z80AFDC card.

2) Power up system. Regardless of any error messages, insert jumper at E18. Connect C.R.O. probe to pin 31 of I.C. 36 (WD 2793) and adjust VR1 to give a 120 ns positive (write precompensation) going pulse.

3) Reconnect C.R.O. probe to pin 29. Adjust VR2 to give a 225 ns (read pulse width) positive going pulse.

4) Reconnect C.R.O. probe to pin 16. Check logic level at pin 25 (or pin 17) or I.C. 36 with logic probe (or C.R.O.), if a logic one, adjust VC1 to give a square wave with a period of 1.9uS. If logic zero adjust for a period of 3.8uS.

5) REMOVE JUMPER FROM E18!!!

6) Press reset button. System should boot correctly.

4.6 I/O CONNECTIONS OF Z80 CARD

Connector J1 - 50 pins RS-232C and Networking serial I/O connector. A 50 pin male header is used to attach from the Z80A SBC board to the back panel, this leads the signal to the NETWORK and SERIAL CHANNAL 4.

Pin No.	Signal	I/O	Description	
1	TSET	O	Transmitter Clock (DTE)	)
2	STF	O	Transmitter frequency	)
3	CD	I	Data Carrier Detect	)
4	DTR	O	Data Terminal Ready	)
5	GND	O	Signal Ground	)
6	DSR	I	Data Set Ready	)
7	CTS	I	Clear to send	) SERIAL CH. 4
8	RSET(DCE)	I	Receiver Clock (DCE)	)
9	RTS	O	Request to send	)
10	RD	I	Receive Data	)
11	TSET(DCE)	I	Transmitter clock (DCE)	)
12	TD	O	Transmit data	)
13	GND	O	Signal Ground	)
14	GND	O	Signal Ground	)

15	RTS-	O	Request to send (-)
16	RTS+	O	Request to send (+)
17	TDATA-	O	Transmit Data (-)
18	TDATA+	O	Transmit Data (+)
19	TCLK-	I	Transmitter Clock (-)
20	TCLK+	I	Transmitter Clock (+)
21	NVM-	I	Non-valid Manchester Code (-)
22	NVM+	I	Non-valid Manchester Code (+)
23	RDATA-	I	Receive Data (-)
24	RDATA+	I	Receive Data (+)
25	RCLK-	I	Receiver Clock (-)
26	RCLK+	I	Receiver Clock (+)
27	GND	O	Signal Ground
28	GND	O	Signal Ground

4.6.1 CONNECTOR J2

Connector J2 - 50 pins 8" floppy disk drive connector. A 50 pin male header is used to attach from one to four 8" disk drives to the Z80A SBC board.

Pin No.	Signal	I/O	Description
2	LOW CURRENT	O	R/W head is positioned between tracks 44-76
4	UNSAFE RESET	O	Disk drive reset
6	FILE UNSAFE	O	
8	-		
10	TWO SIDED	I	Indicates single sided or two sided disk
12	-		
14	SIDE SELECT	O	Directly controlled by the 'S" flag in type II or III commands
16	-		
18	HEAD LOAD	O	Controls the loading the R/W head against the media
20	INDEX	O	Index hole is encountered on the diskette
22	READY	I	Indicates disk readiness and is sampled before R/W cmd is performed
24	-		
26	DRV 0	O	Drive select 0
28	DRV 1	O	Drive select 1
30	DRV 2	O	Drive select 2
32	DRV 3	O	Drive select 3
34	DIR	O	Director of R/W head - stepping in or out
36	STEP	O	Contains a pulse for each step
38	WRITE DATA	O	Contains the unique Address marks as well as data and clock
40	WRITE GATE	O	Valid before writing is to be performed on the diskette
42	TRACK 00	I	R/W head is positioned over Track 00
44	WPRT	I	Controls the amount of delay in Write precompensation
46	RDATA	I	Ran data read

48	WRITE FAULT	O	
50	FAULT RESET	O	Disk drive reset

All the odd numbered pins are signal ground and all the signals are active low.

4.6.2 CONNECTOR J3

Connector J3 - 34 pin 5-1/4" mini floppy disk drive connector. A 34 pin male header is used to attach from one to four 5-1/4" drives to the Z80A SBC board

Pin No.	Signal	I/O	Description
2	SPARE		
4	USE	O	An LED indicator lights when this interface goes low
6	DRV 3	O	Drive select 3
8	INDEX	O	Index hole is encountered on the diskette
10	DRV 0	O	Drive select 0
12	DRV 1	O	Drive select 1
14	DRV 2	O	Drive select 2
16	MOTOR ON	O	Starts the spindle motor
18	DIRECTION	O	Controls the inward or outward direction of the R/W head
20	STEP	O	A pulse signal for moving the R/W head
22	WRITE DATA	O	Data to be written on the diskette is sent to this interface line
24	WRITE GATE	O	Valid before write driver becomes active
26	TRACK 00	I	R/W head is positioned over Track 00
28	WRITE PROTECT	I	Notifies the host system of a diskette with a write protect notch
30	READ DATA	I	Raw Data Read
32	SIDE SELECT	O	Selects which of the two sides of the diskette should be read/written
34	READY	I	Indicates disk readiness and is sampled before R/W cmd is performed

All the odd numbered pins are signal ground.

4.6.3 CONNECTOR J4

Connector J4 - 40 pin Winchester Interface connector. A 40 pin male header is used to connect the Z80A SBC board to the Winchester controller.

Pin No.	Signal	I/O	Description
1	DAL 0	I/O)	
3	DAL 1	I/O)	
5	DAL 2	I/O)	
7	DAL 3	I/O)	
9	DAL 4	I/O)	8 bit bi-directional Data Access Lines
11	DAL 5	I/O)	
13	DAL 6	I/O)	
15	DAL 7	I/O)	
17	A 0	I)	These three address lines are used to
19	A 1	I)	select one of eight registers in the Task
21	A 2	I)	File.
23	$\overline{CS}$	I	Makes a transition for each byte read from or written to task file
25	$\overline{WR}$	I	When $\overline{WR}$ is active along with $\overline{CS}$, the host may write data to selected register
27	$\overline{RD}$	I	When RD is active along with CS, the host may read data from selected register
29	$\overline{WAIT}$	I	Returns to the inactive state
31	N/C		
33	N/C		
35	INTQ	I	Interrupt Request Line is activated whenever CmD has been completed.
37	DRQ	I	Data Request Line is activated whenever the sector buffer contains data
39	$\overline{MR}$	I	Master Reset Line initialises all internal logic

4.7 JUMPER SETTINGS FOR Z80 CARD

This procedure outlines the basic hardware settings for the Z80 SBC (part # 820822 revision C) card. It should be noted that all of the following items are normally checked and set during production testing and therefore, should not require changes by the user.

JUMPER		SETTING/COMMENT
E1	PROCESSOR CLOCK SPEED-	Jumper AB for 4Mhz operation. Jumper BC for 5Mhz operation. Current production units are shipped at 5Mhz.

E2 DMA ENABLE FOR HARD DISK- Jumper AB to enable DMA transfer from the hard disk controller interface. Jumper BC to disable.

E3 INTERRUPT DAISY CHAIN- If the DMA1 (I.C. 35) chip is not fitted, jumper AB to complete the interrupt daisy chain.

E4 BPRO/- Bus priority out signal. Do not jumper for parallel multibus systems.

E5 not used Do not jumper.

E6 BCLK/- Bus clock. Normally jumpered.

E7 CCLK/- Constant clock. Normally jumpered.

E8 MULTIBUS INTERRUPT JUMPERS-

E8
A = INT7/ (multibus pin 36)
B = INT0/ (" " 35)
C = INT6/ (" " 38)
D = INT1/ (" " 37)
E = INT5/ (" " 40)
F = INT2/ (" " 39)
G = INT4/ (" " 42)
H = INT3/ (" " 41)
I = SERINT/
J = FDCINT/

At present there should be a wire wrap link connecting E8-H to E8-J.

E9 ON BOARD INTERRUPT JUMPERS-

E9 A = INT3 (BUFFERED MULTIBUS SIGNAL)
B = INT2 (" " ")
C = INT1 (" " ")
D = INT0 (" " ")
E = INT4 (" " ")
F = INT5 (" " ")
G = INT6 (" " ")
H = INT7 (" " ")
I = IREQ7 (9517 INTERRUPT INPUT)
J = IREQ3 (" " ")
K = IREQ1 (" " ")
L = IREQ0 (" " ")

Wire wrap links will go from either of E9 A,B,C,D,E,F,G,H,M or N to either of E9 I,J,K or L. At present there are no links necessary.

E10 RTS/CTS- Request to send and clear to send for network port. Not yet defined.

E11 TCLK- Network transmit clock. Do not jumper.

E12 TCLK+ Network transmit clock. Do not

jumper.

E13 RS-232c CLK — RS-232c port transmit clock. For internal clock select BC, for external clock select AB. Normally select BC.

E14 RS-232c CLK — RS-232c port receive clock. For internal clock select BC. For external clock select AB. Normally select BC.

E15 ADDRESS SELECT FOR MEMORY AND I/O- (see below)

I/O address select

The group of jumpers E15A, E15B and E15C form the upper 12 address bits of the 8 multibus I/O ports on the Z80ASBC card. The next lower 3 address lines are further decoded by I.C.88 (74LS138) to give each of the 8 I/O ports. When running CP/M86, the video card uses these I/O ports to gain the attention of the Z80ASBC card. The base address of these I/O ports is normally set to 00fXh. The jumper settings are shown below:

Group E15A (lower 4 bits of the address)

A=	(AD7)	not jumpered
B=	(AD6)	" "
C=	(AD5)	" "
D=	(AD4)	" "

Group E15B (next 4 bits of the address)

A=	(AD11)	jumpered
B=	(AD10)	"
C=	(AD9)	"
D=	(AD8)	"

Group E15C (uppermost 4 bits of the address)

A=	(AD15)	jumpered
B=	(AD14)	"
C=	(AD13)	"
D=	(AD12)	"

4.7.1 MEMORY ADDRESS SELECT JUMPERS

The group of jumpers E15E and E15D define the base address the two on board multibus memory banks. Each memory bank is a contiguous 64K byte memory bank which can be setup so that its base address is on one of 16 64K byte boundries. Table 1 shows all the possible address settings for the base address and the associated jumper settings. In the standard system one of the memory banks must be set to E0000h and the other set to F0000h. Care should be taken to ensure that no two memory banks have same base address.

table 1: 'x'= jumper is to be inserted at this position.

Base address	Jumper settings A	B	C	D
00000h	x	x	x	x
10000h	x	x	x	
20000h	x	x		x
30000h	x	x		
40000h	x		x	x
50000h	x		x	
60000h	x			x
70000h	x			
80000h		x	x	x
90000h		x	x	
a0000h		x		x
b0000h		x		
c0000h			x	x
d0000h			x	
e0000h				x
f0000h				

Example of jumper settings for standard system:-

Group E15D (memory bank 1)
A= (AD19) not jumpered
B= (AD18) " "
C= (AD17) " "
D= (AD16) " "

Group E15E (memory bank 0)
A= (AD19) not jumpered
B= (AD18) " "
C= (AD17) " "
D= (AD16) jumpered

E16 Upper 8 bits of I/O address enable. Insert jumper to enable 16 bit I/O addressing. When jumper is not inserted, the group of jumpers E15B and E15C are disabled and therefore,only 8 bit I/O addressing is necessary. The boards are normally setup for 16 bit I/O addressing.

E17 CAS delay select- Normally select BC. Selecting AB will result in more wait states for memory access to the on board multibus memory.

4.7.2 FLOPPY DISK CONTROLLER JUMPERS

E18 FDC test link- Jumper should only be inserted when calibrating floppy disk controller chip (WD 2793, I.C. 36) See below for details.

E19 FDC READY- This jumper selects which ready line to use for mini floppy disk drives. Jumper AB to select pin 6 (of J3) or BC to select pin 34 (of J3). For Mitsubishi drives select BC.

E20 .- not fitted to rev. C cards -

E21 OPTION SELECTION

A = Jumper to indicate that FD0 is a mini floppy.

B = not defined

C = Jumper to indicate that FD1 is mini floppy.

D = not defined

E = Jumper to indicate that FD3 is a mini floppy.

F = not defined

G = Jumper to indicate that FD3 is a mini floppy .

E22 DRIVE RESET- Select BC for UNSAFE RESET/ (pin 4). Select AB for FAULT RESET (pin 50). For Mitsubishi and NEC disk drives select UNSAFE RESET/

E23 DRIVE SELECT FOUR ENABLE- This jumper allows drive 3 to be setup as a mini floppy drive, providing that the ready pin is pin 34 (J3). This jumper allows pin 6 (J3) to be used as a drive 3 select line. Unless the system has a drive 3 mini floppy, do not use this jumper.

table 1: 'x'= jumper is to be inserted at this position.

Base address	Jumper settings A	B	C	D
00000h	x	x	x	x
10000h	x	x	x	
20000h	x	x		x
30000h	x	x		
40000h	x		x	x
50000h	x		x	
60000h	x			x
70000h	x			
80000h		x	x	x
90000h		x	x	
a0000h		x		x
b0000h		x		
c0000h			x	x
d0000h			x	
e0000h				x
f0000h				

Example of jumper settings for standard system:-

Group E15D (memory bank 1)
A= (AD19) not jumpered
B= (AD18) " "
C= (AD17) " "
D= (AD16) " "

Group E15E (memory bank 0)
A= (AD19) not jumpered
B= (AD18) " "
C= (AD17) " "
D= (AD16) jumpered

E16 Upper 8 bits of I/O address enable. Insert jumper to enable 16 bit I/O addressing. When jumper is not inserted, the group of jumpers E15B and E15C are disabled and therefore, only 8 bit I/O addressing is necessary. The boards are normally setup for 16 bit I/O addressing.

E17 CAS delay select- Normally select BC. Selecting AB will result in more wait states for memory access to the on board multibus memory.

4.7.2 FLOPPY DISK CONTROLLER JUMPERS

E18 FDC test link- Jumper should only be inserted when calibrating floppy disk controller chip (WD 2793, I.C. 36) See below for details.

E19 FDC READY- This jumper selects which ready line to use for mini floppy disk drives. Jumper AB to select pin 6 (of J3) or BC to select pin 34 (of J3). For Mitsubishi drives select BC.

E20 - not fitted to rev. C cards -

E21 OPTION SELECTION

A = Jumper to indicate that FD0 is a mini floppy.

B = not defined

C = Jumper to indicate that FD1 is mini floppy.

D = not defined

E = Jumper to indicate that FD3 is a mini floppy.

F = not defined

G = Jumper to indicate that FD3 is a mini floppy .

E22 DRIVE RESET- Select BC for UNSAFE RESET/ (pin 4). Select AB for FAULT RESET (pin 50). For Mitsubishi and NEC disk drives select UNSAFE RESET/

E23 DRIVE SELECT FOUR ENABLE- This jumper allows drive 3 to be setup as a mini floppy drive, providing that the ready pin is pin 34 (J3). This jumper allows pin 6 (J3) to be used as a drive 3 select line. Unless the system has a drive 3 mini floppy, do not use this jumper.

SECTION 5 8086 VDU/COMM CARD

5.1 GENERAL DESCRIPTION

The 8MHz 8086 VDU COMM card is designed for graphic display and communications control. It includes a 16-bit CPU, 256 bytes of non-volatile RAM, 64K bytes EPROM, 128K bytes video display RAM, three serial communication interface, a programmable parallel input port, a keyboard interface, serial and parallel printer interfaces, a CRT controller, a video display address generator, a video display RAM timing, a priority interrupt controller, and Multibus arbitrator. It communicates with other masters in the system via the Multibus.

This card is also designed to be able to work in stand-alone operation, the same card is used in the LABTAM 3000DT graphic terminals, giving identical operation to the system console.

8086 Architecture features:-

The architecture includes four 16-bit byte addressable data registers, two 16-bit memory base pointer registers and two 16-bit index registers, all accessed by a total of 24 operand addressing modes for complex data handling and very flexible memory addressing. A 6-byte instruction queue provides pre-fetching of sequential instruction and the stack oriented architecture facilitates nested sub-routines and co-routines, reentrant code and powerful interrupt handling. The memory expansion capabilities offer a 1 Mbyte addressing range. The dynamic relocation scheme allows ease in segmentation of pure procedure and data for efficient memory utilization. For segment registers (code, stack, data, extra) contain program loaded offset values which are used to map 16 bit address to 20-bit addresses. Each register maps 64K bytes at a time and activation of a specific register is controled explicitly by program control and is also selected implicitly by specific functions and instructions.

Bus Structure:-

The 8086 VDU/COMM card has an internal bus for communicating with on-board memory and I/O options, and uses the Multibus for referencing additional memory and I/O options. Local accesses do not require Multibus communication, making the system bus available for use by other Multibus masters. This feature allows true parallel processing in a multiprocessor encironment.

Non - Volatile RAM :-

The 8086 VDU/COMM card contains 256 bytes non-volatile RAM which is used to store the VDU setup parameters. A store cycle transfers a complete copy of all RAM storage locations into the corresponding locations of the overlaid non-volatile EEPROM memory. The Reset Circuit causes the non-volatile data stored in the EEPROM to be copied back into the internal RAM. Once the EEPROM data is back in the RAM it can then be accessed by normal RAM read or write cycles.

EPROM :-

Four sockets are provided for up to 64k bytes of non-volatile read only memory on the 8086 VDU/COMM card with address range from 70000 to 7FFFF. It contains the terminal control program.

RAM :-

The 8086 VDU/COMM card contains 128K bytes of dynamic read/write memory. Memory refresh is guaranteed by the video timing generator. This 128K bytes memory is located within the addresses 00000 to 1FFFF. The upper half of this memory is reserved for the screen bit mapping and the lower half is for interrupt vectors, data and stack areas and for downloaded 8086 program.

Parallel I/O interface:-

The 8086 VDU/COMM card contains a parallel input port and a parallel printer interface. The printer interface is CENTRONICS compatible. The parallel input interface is used for grpahic tracker ball input.

Serial I/O:-

Programmable communications interface using the Intel 8274/NEC7201 Multi-Protocol Serial Controllers (MPSC) are contained on the 8086 VDU/COMM card. Three of the serial I/O channels can fully support Asynchronous mode protocols. I/O buffering technique is implemented by a circular buffer. The user should refer to section 3 for the detailed discussion on communication buffer. 3 full duplex channels with RS232C V.24 interfaces are implemented, provison for synchronous communication is also built in.

Keyboard Interface:-

A serial asynchronous interface method is used to receive character from the keyboard Input buffering technique is implemented by a circular buffer to enable "type ahead" capability. The user should refer to section 3 for the detailed discussion on communication buffer.

Serial Printer Interfaces:-

The serial printer interface is operated in asynchronous mode. Output buffering technique is implemented by a circular buffer. The user should refer to section 3for the detailed discussion on communication buffer.

CRT controller:-

The CRT controller generates the signals necessary to interface the system to a raster scan CRT display. The data to be displayed is stored in the refresh screen memory by the CPU controlling the data processing system. All timing in the CRT controller is derived from the 1M Hz clock input. The processor communicates with the CRT controller through an 8-bit data bus by reading or writing into the 19 registers.

Video Display Address Generator:-

The video start address is explicitly controlled by the user program. A 16 bit up counter is used to generate the video address which is multiplexed to address the 64K bytes screen memory.

Video Display RAM Timing:-

A PAL is used to generate all the timing for the screen memory, the shift register load pulses, and to synchronizes the processor with the display memory and the CRT controller.

Interrupt circuitry:-

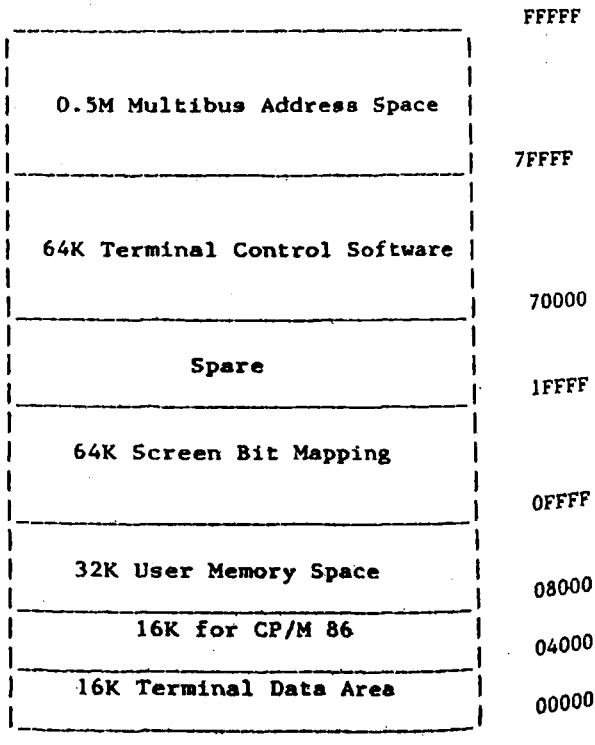
The 8259A programmable Interrupt Controller provides 8 vectored interrupt levels. The interrupt controller accepts interrupt requests from the CRT controller, the programmable parallel and serial I/O interfaces and light-pen. The interrupt controller then determines which of the incoming requests is of the highest priority determines whether this request is of the higher priority than the level currently being serviced, and, if appropriate, issues an interrupt to the CPU. Any combination of interrupt levels may be masked, via software, by storing a single byte in the interrupt mask register of the interrupt controller. The controller generates a unique memory address for each interrupt level.

Visual Display Capabilities:-

Two types of raster scanning are used in visual display unit, interlace and non-interface. In non-interface mode scanning, it consists of two planes of 800 x 300 pixels and each plane is corresponding to 32K bytes memory bank. This two bits per pixel scheme enables four different intensity levels (i.e. black, dim, normal and bold) in the display. However, in interlace mode scanning, it consists only a single plane of 800 x 600 pixels which is corresponding to a 64K bytes memory bank. This one bit per pixel scheme enables 2 different intensity levels (i.e. black and normal).

5.2 MEMORY MAP OF 8086 VDU/COMM CARD

The 128K bytes of RAM resided on the card can only be accessed by the on board internal bus. 64K bytes memeory space is used for the screen bit mapping, 16K bytes data area is used for terminal handling software, 16K bytes memory space is used for running CP/M86 and the remained 32K bytes is made available for the user's program. The other memory on the card is the 64K bytes of non-volatile read only memory which contains the terminal control software. The memory map is organized as follows:-



8086 VDU/COMM card Memory Map

5.3 ACCESS TO MULTIBUS MEMORY AND I/O

Memory:-

To allow for future expansion, the 8086 VDU/COMM card reserves the bottom half M bytes of the 8086's address space for its own use. Thus only half of the Multibus IM bytes memory space is available to the 8086 at one time. The WINDOW signal determines which half of the memory space is available, i.e. 0 to 7FFFF for WINDOW = low and 80000 to FFFFF for WINDOW = high.

I/O:-

The 8086 VDU/COMM card uses partial I/O address decoding with A15 high selecting the onboard I/O devices. Thus only the bottom 32K of the Multibus I/O space (0 to 7FFF) is visible to the 8086 MPU.

5.4 I/O PORT ASSIGNMENT

When the 8086 address line A15 is high and the three status lines are set to I/O Read or Write, this means that 8086 is performing a resident bus I/O operation. The following table summarizes the resident bus I/O assignment.

No	Name	Device	Physical Address	Comment
1	crt-adr	CRT Controler	0e000h	CRT Controller is selected
2	crt-req	MC6845 (U51)	0e002h	Selects either the address register or one of the data register or the internal register
3	pic-a0	programmable interrupt Controller 8259A (U19)	0e400h	Programmable interrupt Controller is selected
4	pic-a1		0e402h	Interrupt mask register can be loaded or read.
5	comm0-a-data	Multi Protocol Serial Controller 8274 (U27)	0e600h	Channel A Data
6	comm0-a-cmd		0e602h	Channel A Commands/Status
7	comm0-d-data		0e604h	Channel B Data
8	comm0-d-amd		0e606h	Channel B Commands/Status
9	comm1-a-data	Multi Protocol Serial Controller 8274 (U35)	0e800h	Channel A Data
10	comm1-a-amd		0e802h	Channel A Commands/Status
11	comm1-b-data		0e804h	Channel B Data
12	comm1-b-and		0e806h	Channel B Commands/Status
13	ctc0-data	System Timing	0ea00h	Data R/W transfers communicate with int. reg.
14	ctco-cmd	Controller AM9513 (U45)	0ea02h	Control R/W, Status/Command registers.
15	ctc1-data	System	0ec00h	Data R/W transfers communicate

16	ctcl-cmd	) Timing) Controller) AM9513 (U40)	0ec02h	with int. reg. Control R/W, Status/Command registers.
17	nvram	Non-volatile RAM (U20, U13)	0ee00h	Non Volatile RAM is selected
18	pwr-fail	)	0f000h	Enable Multibus access
19	window	)	0f002h	0=lower 512K, 1=upper 512K
20	swap	)	0f004h	Swap data & display segments
21	vdouble	) 8 bit	0f006h	Enable double verticle resolution
22	half	) Addressable	0f008h	Enable half intensity
23	click	) Latch 74LS259	0f00ah	Disable key click
24	blank	) (U7)	0f00eh	Disable blanked screen
25	start-lo	Latch 74LS374 (U65)	0f200h	Lower byte of display start address
26	start-hi	Latch 74LS374 (U57)	0f400h	Upper byte of display start address
27	joy-start	Connect J2 pin 39	0f600h	Joystick start pulse
28	print-out	Latch 74LS374 (U52)	0f800h	Parallel printer output
29	print-strobe	Connector J2 Pin 1	0fa00h	Printer data strobe
30	store-array	Nonvolatile RAM (U20,U13)	0fe00h	Store in nonvolatile RAM
31	in-switch	Multiplexer 74ls25 (U15)	0f00h	Read setup switches
32	mouse-in	Latch 74LS373 (U60)	0f600h	Parallel data input
33	mouse-ack	Connector J2 Pin 27	0f800h	Mouse data input acknowledg

5.5 I/O CONNECTIONS OF 8086 VDU/COMM CARD

Connector J1 - 50 pin 4 channel serial port connector. A 50 pine male header is used to attach 4 RS-232C serial channel to the 8086 VDU/COMM card. The pin-out of the header is shown below.

PIN NO Serial-1	Serial-2	Serial-3	Serial printer	RS232C Connector	Signal	I/O	Description
1	14	26	39	24	TSET(DCE)	O	Transmitter Clock (DCE)
2	15	27	40	11	STF	O	Transmitter Frequency
3	16	28	41	8	CD	I	Data Carrier Detect
4	17	29	42	20	DTR	O	Data terminal ready
5	18	30	43	7	GND	O	Signal ground
6	19	31	44	6	DSR	I	Data set ready
7	20	32	45	5	CTS	I	Clear to send
8	21	33	46	17	RSET(DCE)	I	Receiver clock (DCE)
9	22	34	47	4	RTS	O	Request to send
10	23	35	48	3	RD	I	Receive Data
11	24	36	49	15	TSET(DTE)	O	Transmitter Clock (DTE)
12	25	37	50	2	TD	O	Transmit Data
13		38			SPARE		

Connector J2 - 50 pin parallel printer and parallel input connector. A 50 pin male header is used to attach the parallel printer and parallel input terminals to the 8086 VDU/COMM card. The pin-out of the header is shown below.

PIN NO	Signal	I/O	Description	Device
1	PSTB	O	Parallel interface printer strobe	)
2	GND	O	Signal ground	)
3	DB0	I/O	)	)
4	DBI	I/O	)	)
5	DB2	I/O	)	)
6	DB3	I/O	) 8 bit bi-directional data access	) Parallel printer
7	DB4	I/O	) lines	) Interface (J3)
8	DB5	I/O	)	)

	DB6	I/O	)		)
9	DB7	I/O	)		)
10					
	PACK	I		Parallel Printer Interface	)
11				Acknowledge	)
	GND			Signal Ground	)
12	(PBSUY)	I		Parallel Interface printer busy	)
13	(FAULT)	I		Parallel Interace printer fault	)
14	GND	O		Signal ground	)
15					
	GND	O		Signal ground	)
16	STB	O		Parallel input device strobe	)
17	GND	O		Signal ground	)
18					
19	DB0	I/O	)		)
20	DBI	I/O	)		)
21	DB2	I/O	)		)
20	DBI	I/O	)		)
21	DB2	I/O	)		) Parallel Input
22	DB3	I/O	)	8 bit bi-direction Data Access	) (J7)
23	DB4	I/O	)	Lines	)
24	DB5	I/O	)		)
25	DB6	I/O	)		)
26	DB7	I/O	)		)
27	ACK	I		Parellel Input Device	)
				acknowledge	)
28	GND	O		Signal ground	)
29	GND	O		Signal ground	)
30	LPSTROBE	I		Light Pen strobe	)
31	GND	O		Signal Ground	) light Pen (J8)
32	LPSPARE				)
33	GND	O		Signal Ground	)
34	GND	O		Signal Ground	)
35	XTIME	I		X-direction pulse width	)
36	GND	O		Signal Ground	)
37	YTIME	I		Y - direction Pulse width	) Joystick (J4)
38	GND	O		Signal Ground	)
39	JSTART	O		Trigger Joystick Monostable	)
				Multivibrator	)
40	GND	O		Signal Ground	)
41	GND	O		Signal Ground	)
42	KB DATA	I		Keyboard Data	) Keyboard (J12)
43	GND	O		Signal Ground	)
44	GND	O		Signal Ground	)
45	BELL	O		Keyboard Bell	)
46	CLICK	O		Keyboard Click	)
47	-				
48	-				
49	-				
50	-				

Connector J3 - 16 pin monitor interface connector. A 16 pin male header is used to attach the monitor terminal to the 8086 VDU/COMM card. The pin-out of the header is shown below:-

PIN NO	Signal	I/O	Description
2	HD	O	Horizontal deflection
4	HD	O	Horizontal deflection
6	SIGAL	O	Video Signal
8	SIGAL	O	Video Signal
10	VD	O	Vertical deflection
12	VD	O	Vertical deflection
14	-		
16	-		

All the odd numbered pins are signal ground.

Connector J4 - BNC output for coaxial cable with characteristic impedance 50 ohm. This carries the composite video signal.

5.6 JUMPER SETTINGS

E1 - LOCK/ Jumper to connect LOCK/ to the MULTIBUS. For the 3000 computer this signal is normally not jumpered.

E2 - BPRO/ BUS PRIORITY OUT signal. This signal is only jumpered when the MULTIBUS is configured as a serial priority system. As the 3000 computer uses a parallel priority system this signal is not jumpered.

E3 - ANYRQST Normally jumper AB. This means the 8086 will surrender the MULTIBUS to a lower priority bus master as if it were a higher priority master.If BC is jumpered then normal bus priority will apply.

E4 - OPTION SELECT

E4-A Jumper to set bit 1 of keyboard select.

E4-B Jumper to disable graphics.

E4-C Jumper to set bit 0 of keyboard select.

E4-D Jumper to enable killer.

E4-E Jumper to force reprogramming of the non volatile rams.

E4-F not defined.

E4-G Jumper to select stand alone terminal mode.

E4-H not defined.

E5 - Channel 2 RS-232-C receive clock select. Jumper AB for external clock source or BC for internal clock source. For normal operation select BC.

E6 - Channel 2 RS-232-C transmit clock select. Jumper AB for external clock source or BC for internal clock source. For normal operation select BC.

E7 - Keyboard/serial channel data selection for channel 3. Jumper AB to get data from the keyboard or BC to operate channel 3 as a normal RS-232C port. Normally select AB.

E8 - Channel 3 carrier detect option. Normally jumper AB when channel 3 operates as the keyboard port. Jumper BC to operate as a normal RS-232C serial port.

E9 - Channel 0 receive clock select. Jumper AB for external clock source or BC for internal clock source Normally jumper BC.

E10 - Channel 0 transmit clock select. Jumper AB for external clock source or BC for internal clock source. Normally jumper BC.

E11 Channel 1 receive clock select. Jumper AB for external clock source or BC for internal clock source. normally jumper BC.

E12 - Channel 1 transmit clock select. Jumper AB for external clock source or BC for internal clock source. Normally jumper BC.

E13 - Horizontal deflection select for C.R.T. Jumper AB for an active low signal or BC for an active high signal. normally select AB. (see below for more details.)

E14 - Vertical deflection select for C.R.T. Jumper AB for active low signal or BC for active high signal. Normally select BC.(see below for more details.)

5.7 CALIBRATION OF HORIZONTAL AND VERTICAL SYNC CCT.

The adjustment will depend on the type of C.R.T. used. For the Chuomusen (QDM-9M/12W) display make the following adjustments to the VDU/COMM card:

1)Set jumper E13 to AB. Set jumper E14 to BC.
2)Connect C.R.O. probe to E13 and adjust RV1 for a positive going pulse width of 4 to 40usec.
3)Connect C.R.O. probe to E14 and adjust RV2 for a negative going pulse width of 300usec. to 1.4msec.

When the VDUCOMM card has been adjusted, it still may be necessary to make some adjustments to the C.R.T.'s E.H.T card. When making these adjustments, run the program VTEST. This program dumps a grid pattern onto the screen.

1)vertical sync. frequency:- adsjust VR101 so that vertical synchronization is stable.

2)vertical height:- adjust VR102 to ogtain the correct image size.

3)vertical linearity:- adjust VR103 so that the squares at the bottom of the picture are the same height as those at the top.It may be necessary to repeat step 2 again.

4)horizontal centering:- adjust L101 to center the picture.Further adjustments can be made by rotating the magnet rings on the yoke.

5)horizontal width:- adjust L102 to obtain the desired width.

6)horizontal linearity:- adjust L103 so that the width of the squares on the left of the screen is the same as the those on the right.

7)sub brightness:- if the external brightness control does not cover the range required, adjust VR104. VR104 should be set so that the raster is just visable when the external brightness control is at its maximum setting.

8)focus:- adjust VR105 so that the dots in the center of the screen and at the edge are as clear as possible.

SECTION 6 128 KBYTE MEMORY CARD

The 128 Kbyte memory card is a de-populated version of the Z80 SBC CARD, only the 128 Kbyte memory circuitry is used. Therefore, the jumper settings are basically the same.

Memory address select jumpers

The group of jumpers E15E and E15D define the base address the two multibus memory banks. Each memory bank is a contiguous 64K byte memory bank which can be setup so that its base address is on one of 16 64K byte boundries. Table 1 shows all the possible address settings for the base address and the associated jumper settings. In the standard system with only one extra 128K memory card one of the memory banks must be set to d0000h and the other set to c0000h. Care should be taken to ensure that no two memory banks have same base address.

table 1: 'x'= jumper is to be inserted at this position.

Base address	Jumper settings A	B	C	D
00000h	x	x	x	x
10000h	x	x	x	
20000h	x	x		x
30000h	x	x		
40000h	x		x	x
50000h	x		x	
60000h	x			x
70000h	x			
80000h		x	x	x
90000h		x	x	
a0000h		x		x
b0000h		x		
c0000h			x	x
d0000h			x	
e0000h				x
f0000h				

Example of jumper settings for standard system:-

Group E15D (memory bank 1)
A= (AD19) not jumpered
B= (AD18) " "
C= (AD17) jumpered
D= (AD16) jumpered

Group E15E (memory bank 0)
A= (AD19) not jumpered
B= (AD18) " "
C= (AD17) jumpered
D= (AD16) not jumpered

E17 CAS delay select- Normally select BC. Selecting AB will result in more wait states for memory access to the on board multibus memory.

SECTION 7 BACK PANEL CONNECTION ASSIGNMENTS

This section documents the cable connections between the rear panel and the various cards.

7.1 CABLE CONNECTIONS

i) Z80A SBC Board:-

(a) Connector J1 - 50 pins RS-232C and Networking serial I/O connector. A 50 pin male header is used to attach from Z80A SBC board to the back panel.

(b) Connector J2 - 50 pins 8" floppy disk drive. A 50 pin male header is used to attach from one to four 8" disk drives to the Z80A SBC BOARD.

(C) Connector J3 - 34 Pin 5-1/4" mini-floppy disk drive connector. A 34 pin male header is used to attach from one to four 5-1/4" drives to the Z80A SBC board.

(d) Connector J4 - 40 pin Winchester Interface connector. A 40 pin male header is used to connect the Z80A SBC board to the Winchester Controller.

ii) 8086 VDU/COMM Card:-

(a) Connector J1 - 50 pin 4 channel serial port connector. A 50 pin male header is used to attach 4 RS-232C serial channel to the 8086 VDU/COMM card.

(b) Connector J2 - 50 pin parallel printer and parallel input connector. A 50 pin male header is used to attach the parallel printer and parallel input terminals to the 8086 VDU/COMM card.

(c) Connector J3 - 16 pin monitor interface connector. A 16 pin male header is used to attach the monitor terminal to the 8086 VDU/COMM card

7.2 RS-232C CONNECTIONS

The following is the pin out connections on all DB-25 RS232C connections, this include all the Serial channels, however, the Serial printer channel has no RD (receive data) line.

PIN NO	I/O	Signal
1	O	Chassis Gnd
2	O	TD (Transmit Data)
3	I	RD (Receive Data)
4	O	RTS (Request to send)
5	I	CTS (clear to send)
6	I	DSR (Data set ready)
7	O	Signal Gnd
8	I	CD (Data carrier detect)
9		
10		
11		
12		
13		
14		
15	I	TCLK (Transmitter clock)
16		
17	I	RCLK (Receiver clock)
18		
19		
20	O	DTR (Data Terminal Ready)
21		
22		
23		
24	O	TCLK (Transmitter clock)
25		

7.3 PARALLEL INPUT CONNECTOR - DB-15(Male)

PIN NO	I/O	Signal
1	O	Chassis Gnd
2	O	STROBE
3	I/O	DB0)
4	I/O	DB2)
5	I/O	DB4) Data
6	I/O	DB6)
7	O	Signal Gnd
8	O	Signal Gnd
9		-
10	O	Signal Gnd
11	I/O	DB1)
12	I/O	DB3)

	I/O	DB5	) Data
13	I/O	DB7	)
14			
	I/O	ACK	
15			

LIGHT-PEN CONNECTOR - DB-9 (Female)

7.4

PIN NO	I/O	Signal
	O	Chassis Gnd
1	O	Signal Gnd
2	O	Signal Gnd
3	O	Signal Gnd
4	O	Signal Gnd
5	O	+5v
6	O	Signal Gnd
7	I	LPSTROBE
8		LPSPARE
9		

7.5 KEYBOARD CONNECTOR

PIN NO A	B**		I/O	Signal
1	1	(Red)	O	+5v
2	4	(Blue)	I	KB DATA
3	5	(Black)	O	KB GND
4	6	(White)	O	BELL
5	2	(Green)	O	Signal Gnd
6	3	(Brown)	O	CLICK

** 2 types of connectors used
A = small green plastic type
B = metal shell type

7.6 NETWORK CONNECTOR - DB-15 (Female)

PIN NO	I/O	Signal
1	O	Shield
2	O	RTS+ (Request to send)
3	O	TDATA+ (Transmit data)
4	O	Signal Gnd
5	I	TCLK+ (Transmitter clock)
6	I	NVM+ (Non-valid Manchester code)
7	I	RDATA+ (Receive data)
8	I	RCLK+ (Receiver clock)
9	O	RTS- (Request to send)
10	O	TDATA- (Transmit data)
11	O	+12V
12	I	TCLK- (Transmitter clock)
13	I	NVM- (Non-valid Manchester code)
14	I	RDATA- (Receive data)
15	I	RCLK- (Receiver clock)

7.7 PARALLEL PRINTER CONNECTOR

PIN NO	I/O	Signal
1	O	STB
2	I/O	DB0
3	I/O	DB1
4	I/O	DB2
5	I/O	DB3
6	I/O	DB4
7	I/O	DB5
8	I/O	DB6
9	I/O	DB7
10	I	ACK
11	I	BUSY
12		N/C
13		N/C
14	O	Signal Gnd
15		N/C
16	O	Signal Gnd
17	O	Chassis Gnd
18		N/C
19	O	Signal Gnd
20	O	Signal Gnd
21	O	Signal Gnd
22	O	Signal Gnd
23	O	Signal Gnd
24	O	Signal Gnd
25	O	Signal Gnd
26	O	Signal Gnd
27	O	Signal gnd
28	O	Signal Gnd
29	O	Signal Gnd
30		N/C
31	I	FAULT

		N/C
32		N/C
33		N/C
34		N/C
35		N/C
36		

7.8 JOYSTICK CONNECTOR- DB-9 (Male)

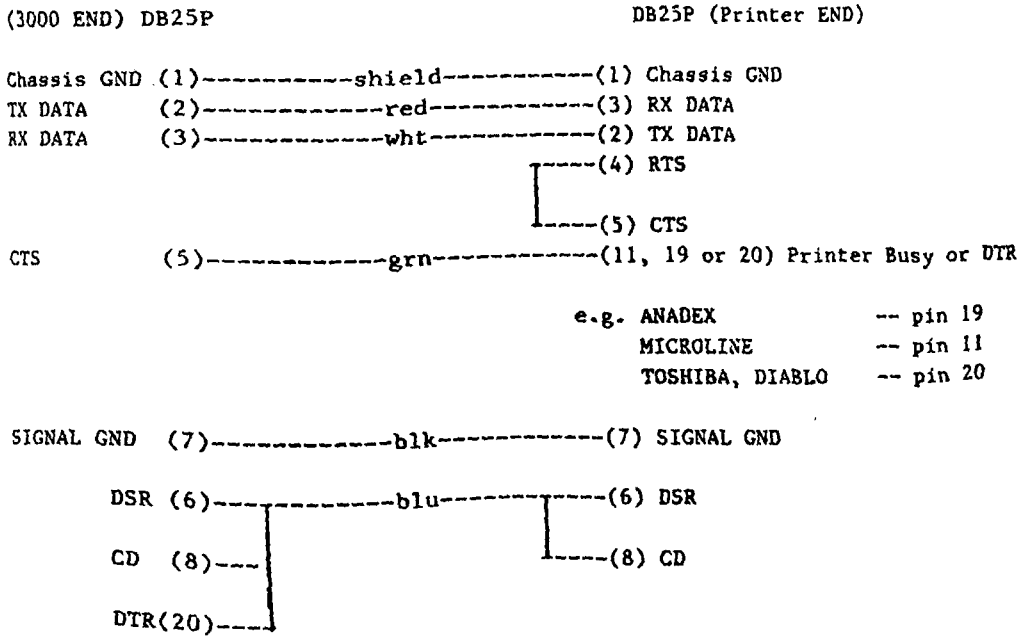
PIN NO	I/O	Signal
	O	Chassis Gnd
1	O	Signal Gnd
2	O	Signal Gnd
3	O	Signal Gnd
4	O	Signal gnd
5	O	+5v
6	I	XTIME
7	I	YTIME
8	O	JSTART
9		

APPENDIX A

SERIAL PRINTER CONNECTIONS

The serial printer connector (J6) on the rear panel of the 3000 computer is a SEND-ONLY RS232C connection. The following are the recommended printer cable connections using V24 hand shaking.

Note that the signal CT5 (5) from 3000 can be connected to different pins on the printer depending on the particular printer. See your printer manual for the PRINTER BUSY signal for this connection. If no PRINTER BUSY signal is used, connect to pin 20(DTR).

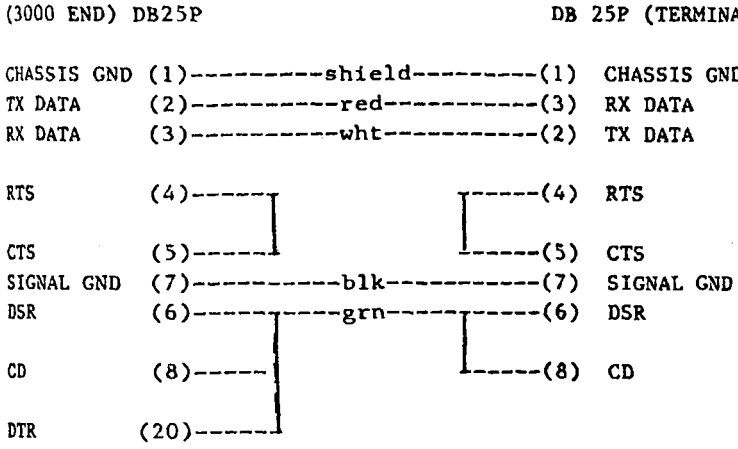


---- END APPENDIX A ----

APPENDIX B

SERIAL TERMINAL CONNECTION

The following diagrams detail the cable connections for connecting external terminals to any of the serial channels of the 3000 computer.



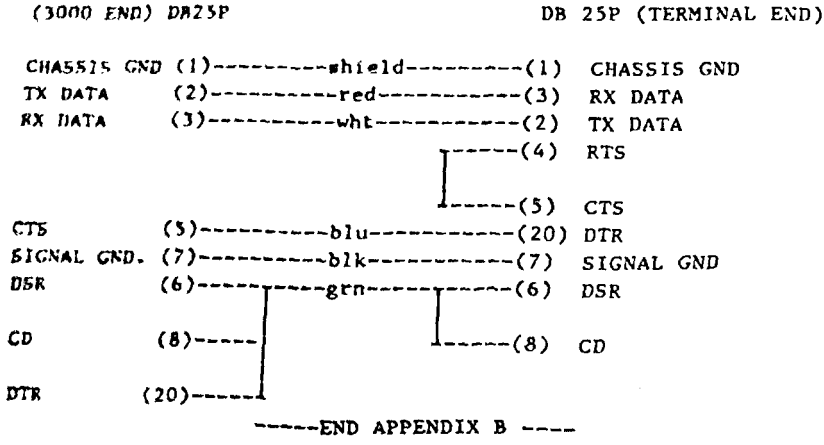
The above cable does not use any handshaking signals, software protocol such as XON-XOFF should be used for handshaking.

If hardware handshaking is required, use the following alternate connection.

(see next page)

Cable connections for using RS232C hardware handshaking

NOTE: this is the connection used for 3000 terminals.



-----END APPENDIX B ----

APPENDIX C

PRINTER SETUP

This appendix details the dip-switch setup for the printers supported currently by LABTAM, these are:-

(i) ANADEX 9500
(ii) MICROLINE 83A
(iii) MICROLINE 84
(iv) MICROLINE 83A/84 with high speed serial I/F option
(v) TOSHIBA NIC-2100H or P1350

(i) ANADEX DP-9500

The switch settings for paralle and serial interface to LABTAM 3000 computer are outlined as follows:-

Switches S1 and S2 (Printer Options):-

Switch	Setting	Description	Switch	Setting	Description
S1-1	OFF	)	S2-1	OFF	)
S1-2	ON	)	S2-2	OFF	) No perforation skip
S1-3	ON	) For 11"	S2-3	OFF	Spare
		) long paper			
S1-4	OFF	)	S2-4	OFF	6 lines/inch
S1-5	ON	)	S2-5	ON	) XON-XOFF communication
S1-6	OFF	10 Char/inch	S2-6	OFF	) protocol
S1-7	ON	7.5 sec. time out	S2-7	OFF	Wraparound Overlength Lines
S1-8	OFF	'#' Character selected	S2-8	OFF	No auto line feed

Switches S3 and S4 (Communication Options):-

All switches of S3 must be 'OFF' for parallel interface operation.

Switch	Setting	Description
S3-1	ON	Serial interface
S3-2	OFF	13.2 inch/line
S3-3	ON	)
S3-4	ON	) 9600 Baud
S3-5	ON	)
S3-6	OFF	)
S3-7	ON	8 Data bits
S3-8	OFF	1 stop bit
S3-9	OFF	) No parity
S3-10	OFF	)

S4 switched to RS232C operation (for serial i/f only).

(ii) MICROLINE 83A

The switch settings for parallel and serial interfaces to LABTAM 3000 computer are outlined as follows:-

Dip Switches on Operation Panel Circuit Board

Switch	Setting	Description
SW1	OFF	)
SW2	OFF	) US ASCII
SW3	OFF	)
SW4	OFF	)
SW5	OFF	8 bits
SW6	OFF	Won't move paper up one line as CR is received
SW7	OFF	Printer ignores DEL code
SW8	ON/OFF	Serial/Parallel interface

Dip Switches on Control Circuit Board

Switch	Setting	Description
SW1	ON	Space = ready, Mark = busy
SW2	ON	)
SW3	OFF	) 1200 BPS
SW4	ON	)
SW5	OFF	Spare
SW6	OFF	Without parity

(iii) MICROLINE 84

The switch settings for parallel interface to LABTAM 3000 computer are outlined as follows:-

DIP Switches on HLPA Circuit Board

Switch	Setting	Description
1	OFF	8 bits
2	OFF	Won't move paper up one line as CR is received
3	OFF	Printer ignores DEL code
4	OFF	Spare
5	OFF	)
6	OFF	) US ASCII
7	OFF	)
8	OFF	)

(iv) High Speed Serial Interface Board for MICROLINE 83A/84

This section describes a high-speed serial interface to be packaged in MICROLINE 83A/84 as an option for use with a start-stop synchronization serial communication circuit. The DIP switches SW1-SW6 on the serial interface circuit board are set as follows:-

DIP SW1

Switch	Setting	Description
SW1	OFF	}
SW2	ON	} Character format: 1 start bit + 8 data bits + no-parity
SW3	OFF	}
SW4	OFF	}
SW5	ON	} 4800 BPS
SW6	ON	}
SW7	ON	CR code
SW8	OFF	Shift in/out code : valid

DIP SW2

Switch	Setting	Description
SW9	OFF	}
SW10	OFF	} Protocol : Centronics Unblocked
SW11	ON	}
SW12	OFF	DTR Signal : SPACE at power on
SW13	OFF	RS232C : MARK at BUSY state ; Current loop : SPACE at BUSY state
SW14	OFF	Receivable regardless of CD signal MARK/SPACE
SW15	OFF	RS232C interface
SW16	OFF	Current loop: 3-wire/4-wire

(v) TOSHIBA NIC-2100H or P1350

SWITCH S1

-1	OFF	;	print position not moved on paper advance
-2	OFF	;	print start on paper movement
-3	OFF	;	no LF on auto word wraparound
-4	OFF	;	line truncated on overflow
-5	OFF	;	for friction feed mode
-6	OFF	;	cr code function only (no automatic LF)
-7	OFF	;	Bi-directional logic seek printing
-8	ON	;	printer enabled on power on

SWITCH S2 (set all switches on p.c.b. to OFF, so that front panel switch becomes the over-riding one)

-1	OFF		
-2	OFF	;	default FONT 0
-3	OFF		
-4	OFF	;	10 CPI pitch

SWITCH S3

-1	OFF		
-2	OFF		
-3	ON		
-7	OFF	;	serial I/F (READY/BUSY) handshaking
-8	ON	;	

SWITCH S4

-1	OFF		
-2	OFF		
-3	OFF		
-4	ON	;	9600 baud
-5	OFF	;	8 data bit
-6	OFF	;	one stop bit
-7	ON	;	neglect parity
-8	ON		

---- END APPENDIX C ----