PREFACE

The Olivetti L1 M20 Hardware Architecture and Function Manual describes the hardware design and provides interface information for the M20 computer.

This manual is intended for hardware and software designers, engineers and interested persons who need to understand the design and operation of the M20.

PREREQUISITES: Concepts of computer architecture

REFERENCES:

PCOS User Guide - 3982980 P (0)
Basic Language - 3982430 P (2)

DISTRIBUTION: General (G)

FIRST EDITION: July 1983

WARNINGS: Insertion of non-Olivetti certified modules may effect the M20's compliance to approval standards. Operation with such modules may constitute a safety risk to the user and result in interference to radio and TV reception.

Olivetti will not be liable for failure to conform to standards in cases where modules not produced by Olivetti have been connected.

The conditions of guarantee established in the M20 service contract will continue to apply only when upgrading is carried out in accordance with the indications laid down in the manual.

The Customer Technical Support Department of the local Olivetti subsidiary should be consulted on field service engineering problems.

PUBLICATION ISSUED BY:

Ing. C. Olivetti & C., S.p.A.
Direzione Documentazione
77, Via Jervis - 10015 IVREA (Italy)

CONTENTS

1. HARDWARE OVERVIEW

ABOUT THIS CHAPTER

This chapter gives a hardware overview of the M20 system and discusses the standard configurations and the various options, expansions and printers supported. It also deals with the environmental requirements and the A.C. input requirements.

CONTENTS

1. HARDWARE OVERVIEW

1.1 GENERAL

The Olivetti L1 M20 personal computer in any one of its standard configurations, Figure 1-1 shows configuration A, comprises two parts:

- Display
- Basic Module

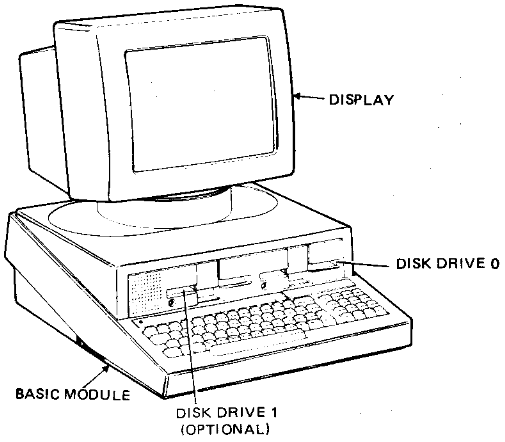


Fig. 1-1 Basic Module and Display - Configuration A

The Display houses a monochrome CRT unit or, as an option, a colour CRT unit. In the case of the monochrome unit, a single cable connects the Display to the Basic Module and in the case of the colour unit, two cables, one signal and one power, connect the Display to the Basic Module. An intensity control is accessable at the rear of the Display.

The Basic Module houses the keyboard, motherboard, power supply unit and, depending on the standard configuration, one 5.25 inch diskette drive, or one diskette drive and a hard disk unit. The diskette and hard disk drive units provide the M20 with the ability to use magnetic media for program or data storage.

The keyboard provides alphanumeric keys and a numeric keypad. In addition, the keyboard has a power-on lamp and a buzzer.

The motherboard is the printed circuit board which takes up most of the bottom of the Basic Module. The motherboard holds the microprocessor (CPU), all major circuitry for the M20 system and all logic, memory and control circuits for peripheral interfaces, such as the printer interface, and provides space for system expansion.

A reset button is provided to reset the M20 without the use of the power on/off switch. To avoid accidentally reseting the M20, this button is located behind a small hole on the right-hand side of the Basic Module. It is pressed by inserting into the hole a ball-point pen or similar object.

Three cable connectors are located at the rear of the Basic Module for the connection of the Display and other peripheral units. The cable connectors are:

Video connector
Parallel Input/Output connector (Centronics-like)
Serial Input/Output connector (RS-232-C)

Snap-out plates located above these connectors provide access to the mounting positions for connectors for optional interfaces such as the IEEE 488 interface or the TWIN RS-232-C interface.

Figure 1-2 shows a rear view of the M20 Basic Module.

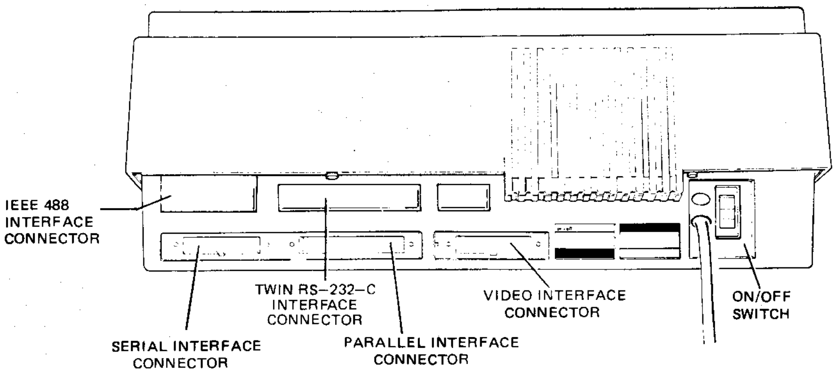

IEEE 488 INTERFACE CONNECTOR
TWIN RS-232-C INTERFACE CONNECTOR
VIDEO INTERFACE CONNECTOR
ON/OFF SWITCH
SERIAL INTERFACE CONNECTOR
PARALLEL INTERFACE CONNECTOR

Fig. 1-2 Basic Module - rear view

1.2 SYSTEM DESCRIPTION

The following summarizes the standard configurations, the peripheral options supported, and the expansion options.

STANDARD CONFIGURATION - A

- Monochrome Black and White 12 inch CRT Display with graphics capability
- Basic Module
  - Keyboard
  - 1 diskette drive with a storage capacity of 320 KB or 160 KB or 640 KB
  - Motherboard
    - Z8001 Microprocessor
    - 128 KB Random Access Memory
    - 8 KB Erasable Programmable Read Only Memory
    - Video Interface
    - Diskette Drive Interface
    - Parallel Input/Output Interface (Centronics-like)
    - Dual Communication Serial Interface (RS-232-C)
    - Timer
  - Power Unit

STANDARD CONFIGURATION - B

- Monochrome Black and White 12 inch CRT Display with graphics capability
- Basic Module
  - Keyboard
  - 1 Hard Disk unit with a storage capacity of 11.25 MB
  - 1 Diskette drive with a storage capacity of 320 KB or 640 KB
  - Motherboard
    - Z8001 Microprocessor
    - 128 KB Random Access Memory
    - 8 KB Erasable Programmable Read Only Memory
    - Video Interface
    - Hard Disk Controller
    - Diskette Drive Interface
    - Parallel Input/Output Interface (Centronics-like)
    - Dual communication Serial Interface (RS-232-C)
    - Timer
  - Power Unit

EXPANSIONS & OPTIONS

Colour 12 inch CRT Display
Second Diskette Drive (for configuration A)
3 32 KB RAM Memory Expansion Boards
giving a system memory of 224 KB
or
3 128 KB RAM Memory Expansion Boards
giving a system memory of 512 KB
IEEE 488 Interface Board
TWIN RS-232-C Interface Board
Alternate Processor Board (APB) 1086
Printers

A simplified block diagram of the M20 is shown in figure 1-3. The dotted lines show the optional modules. A brief explanation of each module follows.

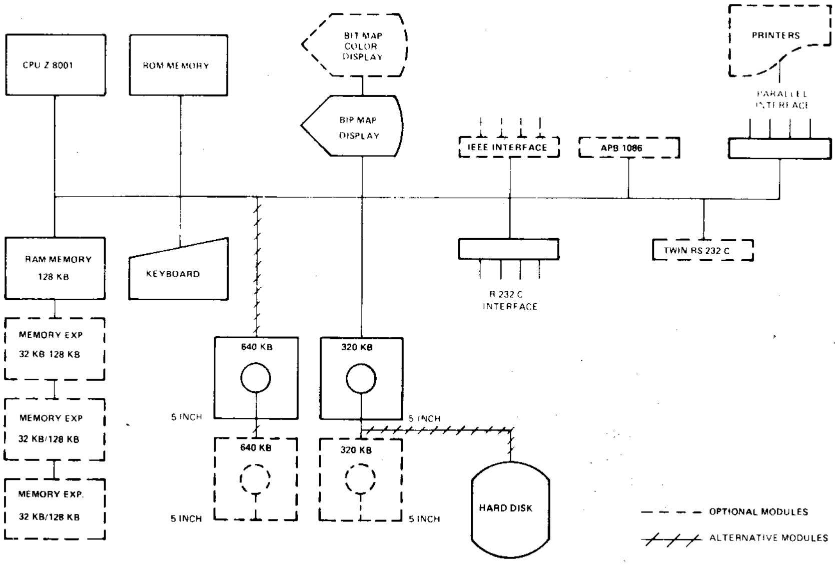


Fig. 1-3 M20 Simplified Block Diagram

1.2.1 DISPLAY

In its standard configuration the M20 uses a monochrome Display, and a bit-map technique for both graphics and text. The resolution is 512 by 256 dots. The character set includes upper-case and lower-case. Standard display attributes, all software defined, include reverse and hide capabilities.

The use of the bit-map technique allows two different page formats to be used, either a 1024 character format or a 2000 character format. The default format, determined by the system software, is 1024 characters. There is no hardware difference between the two formats.

For the colour display, available as an option, there are two different types of colour systems:

4 colour
8 colour

In the 4 colour system eight colours are available but only four may be viewed simultaneously. In the 8 colour system all the eight colours may be viewed simultaneously. The Display unit used for both systems is the same. The eight colour palette is composed of red, green, yellow, blue, magenta, cyan, black and white.

The resolution of the bit mapped graphics in the colour system is the same as that in the monochrome system but memory requirement is higher.

1.2.2 KEYBOARD

The keyboard has keys grouped into two sections, alphanumeric keys in a standard typewriter layout and a numeric keypad.

The M20 supports the following national key layouts:

Italian
German
French
British
USA ASCII
Spanish
Portuguese
Swedish and Finnish
Danish
Katakana
Yugoslavian
Norwegian
Greek
Swiss - French
Swiss - German

The national keyboards are jumper selectable or can be invoked by a PCOS command.

1.2.3 DISKETTE DRIVES

One 5.25 inch diskette drive must always be present on an M20 system. There are three types of diskette drives:

160 KB, unformatted capacity
320 KB, unformatted capacity
640 KB, unformatted capacity

If two diskette drives are used, they must be of the same type. Only 320 KB or the 640 KB diskette drives can be used as a hard disk back up device.

1.2.4 HARD DISK UNIT

The M20 uses a 5.25 inch Winchester type hard disk unit that has an unformatted storage capacity of 11.25 Mbytes.

1.2.5 CENTRAL PROCESSOR UNIT (CPU)

The M20 uses an advanced 16 bit microprocessor, the Z8001. Features of the Z8001 include:

sixteen 16 bit general registers
segmented addressing
handling of bit, byte, word and long word data
three types of interrupts: non-maskable, non-vectored, and vectored.

The support logic provides address translation for optimum flexibility in both hardware configuration and software utilization. The CPU and support logic are located on the motherboard.

1.2.6 MEMORY

In the standard configuration the M20 motherboard holds 128 KB of Random Access Memory (RAM) and 8 KB of Erasable Programmable Read Only Memory (EPROM). RAM is expandable to a maximum capacity of 224 KB or 512 KB.

The RAM stores user programs and data, and the BASIC interpreter and operating system software.

The EPROM stores the power-on bootstrap and power-on diagnostics. Information in EPROM is placed there during manufacture and being non-volatile is not lost when the M20 power is switched off.

1.2.7 VIDEO INTERFACE

The video interface provides the interface between the Display and the M20 system as well as the hardware associated with the generation of graphics and text. The video interface is located on the motherboard.

1.2.8 DISKETTE DRIVE INTERFACE

The diskette drive interface provides two major functions on the M20 system. It provides the logic and control circuitry needed to write data onto, or to read data from the diskettes and initially formats new diskettes. The diskette drive interface is located on the motherboard.

1.2.9 HARD DISK CONTROLLER

The hard disk controller interfaces the M20 to the hard disk unit. It ensures proper operation of the hard disk unit and error handling. It receives commands from the M20, decodes them, and then initializes and monitors the hardware as the commands are executed and returns status back to the M20. The hard disk controller is located on two printed circuit boards. One is located on the hard disk unit and one plugs into one of the expansion slots on the motherboard.

1.2.10 PARALLEL INTERFACE

The parallel interface provides the M20 with one Centronics-like parallel port for connecting a printer. The parallel interface is located on the motherboard.

1.2.11 DUAL COMMUNICATION SERIAL INTERFACE

The dual communication serial interface provides the M20 with one RS-232-C type serial port, used to interface the M20 to a modem or plotter, and one keyboard interface. The dual communication serial interface is located on the motherboard.

1.2.12 TIMER

The timer is a programmable device that has three independent channels. Two of the channels are used to set the keyboard and printer baud rates. The third channel is available to the user and can be programmed as an interval real-time clock. The Timer is located on the motherboard.

1.2.13 POWER SUPPLY UNIT

The power supply unit is housed in a metal case inside the Basic Module and provides all dc voltages required for the M20. A voltage selector jumper in the Power Supply Unit selects one of the following ac line input voltage ranges:

100V to 120V
200V to 240V

1.2.14 IEEE 488 INTERFACE BOARD

The IEEE 488 interface board plugs into one of the expansion slots on the motherboard to provide a means of transferring digital data among a group of instruments and system components. As implemented in the M20, the IEEE 488 option consists of a **Listener, Talker** and **Controller** plus line transceivers. It can be used with systems that use a byte-serial means of data transfer. The interface functions are described below.

Listener - A device capable of receiving data over the interface when addressed. Examples of this type of device are printers, display devices, programmable power supplies, programmable signal sources and the like.

Talker - A device capable of transmitting data over the interface when addressed. Examples of this type of device are tape readers, voltmeters that are outputting data, counters that are outputting data and so on.

Controller - A device capable of specifying talkers and listeners for data transfer. Examples of this are computers like the M20.

1.2.15 TWIN RS-232-C INTERFACE BOARD

The TWIN RS-232-C interface board plugs into one of the expansion slots on the motherboard to provide two communication channels with both RS-232-C and/or 20mA current loop options. It supports asynchronous and synchronous communication, can both receive and transmit data clocks for synchronous communication and various Baud rates are easily programmable for each channel. The board can be configured as follows:

2 RS-232-C Channels
2 Current Loop Channels
1 RS-232-C Channel and 1 Current Loop Channel

1.2.16 ALTERNATE PROCESSOR BOARD 1086

The Alternate Processor Board (APB) 1086 plugs into one of the expansion slots on the motherboard to allow the M20 to execute software written for an Intel 8086 microprocessor. The purpose of this board is to support the following two widely used operating systems:

CP/M 86 and **MS-DOS**

1.2.17 PRINTERS

The Olivetti printers that may be used with the M20 include the following:

Printer PR 2400	Non-impact dot matrix thermal printer
Printer PR 1450	Impact dot matrix printer
Printer PR 1471	Impact dot matrix printer
Printer PR 1481	Impact dot matrix printer
Printer PR 430	Impact daisy wheel printer
Printer PR 2300	Non-impact dot matrix ink jet printer

1.3 ENVIRONMENTAL & A.C. INPUT REQUIREMENTS

1.3.1 ENVIRONMENTAL REQUIREMENTS

The M20 operates reliably in a typical office environment but it is important to adhere to the following guide lines when choosing a suitable location.

The M20 should be plugged into an earthed power supply. Unearthed machines do not work properly and can be a safety hazard. If the M20 is not plugged into an earthed supply one can experience:

improper program operation
unreadable disks
expensive machine damage

The M20 should, when possible, be isolated from sources of electrical interference and from devices that cause excessive voltage level fluctuations. Some common sources of interference are:

air conditioners, large fans and blowers
large transformers and alternators
large brush type or induction motors such as those used for elevators
radio and TV transmitters, signal generators and high frequency security devices

Note: Office machines such as typewriters, copiers, calculators etc., may be connected to the same supply provided that they do not cause excessive interference.

The M20 should be placed in a relatively dust-free area. Airborne dust and smoke can cause excess wear on moving parts, short circuits (especially in the presence of high humidity) and read/write errors on disks.

The M20 should be placed away from heat and direct sunlight. Unusually high temperature coupled with low humidity may cause static problems.

The M20 is cooled by a fan at the rear of the Basic Module. Air is drawn in through vents at the front of the Basic Module and expelled through a vent at the rear. These areas must be kept clear of papers and other material that would obstruct air flow.

1.3.1.1 Environmental Characteristics

Operating Temperature Range:	10 to 40 degC
Operating Relative Humidity:	10% to 95%
Storage Temperature Range:	5 to 45 degC
Storage Relative Humidity:	5% to 95%

1.3.2 A.C. INPUT REQUIREMENTS

VOLTAGE RANGE	TOLERANCE	FREQUENCY RANGE	TOLERANCE
100V to 120V or 200V to 240V	10%	50 Hz to 60 Hz	+5%

The input voltage range is jumper selectable in the power unit.

2. HARDWARE

ABOUT THIS CHAPTER

This chapter describes in detail the M20 hardware. It deals with all the printed circuit boards including the expansions and options. It also gives the main characteristics of the magnetic media, power supply unit, colour and black and white displays as well as all the printers used by the M20 system.

CONTENTS

2. HARDWARE

2.1 BASIC MODULE

2.1.1 MOTHERBOARD

Figure 2-1 is a block diagram of the M20 motherboard. The following outlines the various modules and their function.

Central Processor Unit (CPU) - A 16 bit Microprocessor chip that contains arithmetic and logic circuits which extract program instructions from memory, one at a time, and execute them.

Random Access Memory - A volatile memory which stores the operating system software, BASIC interpreter, and all user programs and data. Information stored in RAM may be altered.

RAM Control & Timing - Circuitry which provides all the control and timing signals necessary to address memory and to control the transfer of data or instructions to and from memory.

RAM Address Multiplexer - Multiplexers used to address RAM.

Erasable Programmable Read Only Memory (EPROM) - EPROM is used to store the power-up diagnostics and bootstrap. It is a non-volatile type of memory and its capacity is 8 KB.

Mapping PROM - A fast bipolar PROM which provides dynamic memory segment relocation and makes software addresses as viewed by the programmer independent of physical memory addresses. The M20 uses a 1K x 8 mapping PROM perform this relocation process.

Status Decoder - Circuitry which interprets the status of the CPU.

Data Buffer - Bi-directional buffer used to interface the CPU to the system data bus.

Address Latches - Latches used to interface the CPU to the system address bus.

Interrupt Control Logic - Circuitry that handles the vectored priority interrupts to the CPU. It functions as an overall manager in the interrupt-driven system environment.

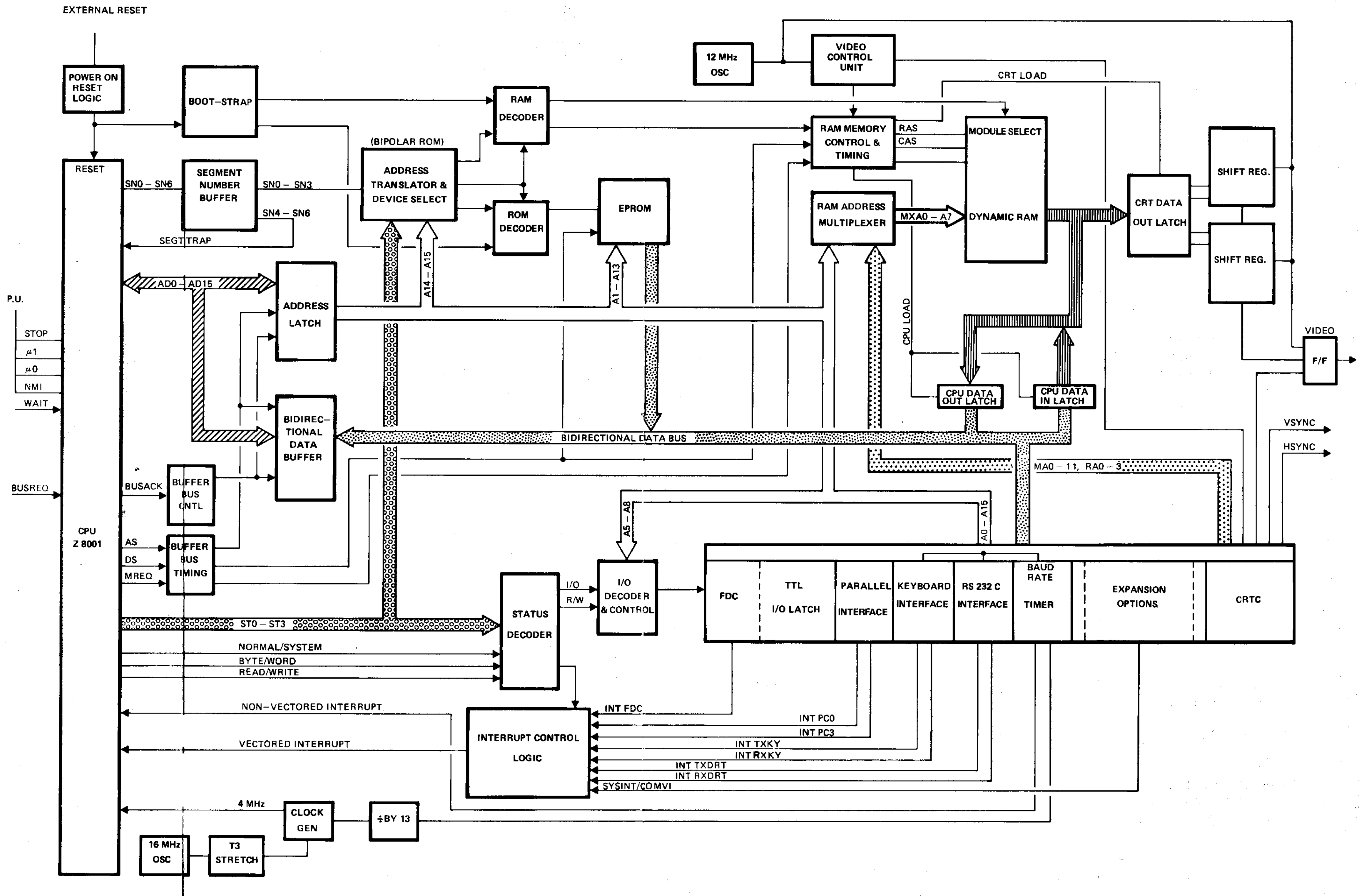


Fig. 2-1 Motherboard Block Diagram

Bootstrap & Reset Circuitry - Circuitry which starts the initialization procedure and causes the CPU to call for instructions from the EPROM.

Clock Circuitry - Circuitry which generates the clock signals for the system.

Video Control Unit - Circuitry for displaying the read data from refresh memory in step with the video-synchronizing pulses. It transforms the data into the video signal which is then displayed on the CRT.

CRT Latch - Latch used to interface the dynamic RAM to the video circuitry.

Video Interface - The CRT Controller provides the interface to raster scan the CRT and provides video timing and refresh memory addressing.

Diskette Drive Interface - Circuitry which provides all the logic and control necessary to write data onto, or read data from the 5.25 inch diskettes. It also initially formats new diskettes.

Dual Communication Serial Interface - The Serial (RS-232-C) interface, used to interface the M20 to a modem or plotter, and the keyboard interface, used to interface the M20 system to the keyboard, are grouped together and referred to as the Dual Communication Serial Interface.

Timer - Programmable device used to set the keyboard and printer baud rates and provide a real-time clock.

Parallel Interface - Centronics-like interface used to interface the M20 to one of the Olivetti printers.

Shift Registers - Registers which convert the parallel data from RAM into serial data. The serial data is then input to the video control unit.

TTL I/O Latch - The M20 uses two transparent latches to form an 8-bit input/output port. The bit assignment is as follows:

D0: 0 selects Drive 0; 1 deselects Drive 0
D1: 0 selects Drive 1; 1 deselects Drive 1
D2: not used
D3: 0 selects double density; a 1 selects single density
D4: uncommitted output; input is used for mapping PROM status
D5: uncommitted output; input is used for mapping PROM status
D6: uncommitted output; input is used for mapping PROM status
D7: uncommitted output; input is used for mapping PROM status

CPU Data Out Latch - Latch which interface the RAM memory to the system data bus.

CPU Data In Latch - Latch which interfaces the system data bus to the RAM memory.

T3 Stretch Circuitry - Circuitry used to modify the normal Z8001 timing.

2.1.1.1 Central Processor Unit

The Central Processor Unit (CPU) is the heart of the M20 computer, executing all transfers of data under control of the program. It consists of a Z8001 microprocessor chip that contains arithmetic and logic circuits which extract program instructions from memory, one at a time, and executes them.

The CPU is supported by additional logic, address decoding, timing and buffer elements which are necessary to address memory and control the transfer of data or instructions.

Pin Functions

Figure 2-2 shows the pin functions of the Z8001.

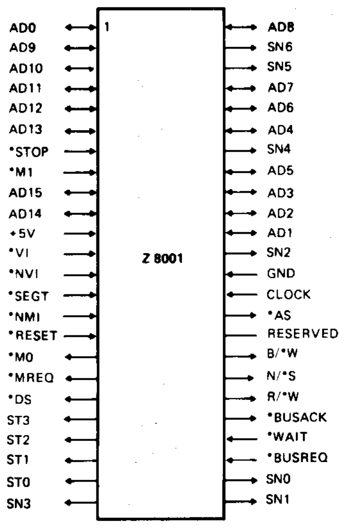


Fig. 2-2 Z8001 Pin Functions

The Z8001 pins can be grouped into categories according to their function.

Transaction Pins - These signals provide timing and data transfer for bus transactions.

AD0-AD15 Address/Data (output, active high, 3-state): These multiplexed address/data lines carry the input/output addresses, the offset portion of memory addresses and data during bus transactions.

SN0-SN6 Segment Number (output, active high, 3-state): These lines carry the encoded segment number of the memory address.

ST0-ST3 Status (output, active high, 3-state): These lines indicate the kind of transaction occurring on the bus and give additonal information about the transaction.

ST3	ST2	ST1	ST0	TRANSACTION
0	0	0	0	Internal Operation
0	0	0	1	Memory Refresh
0	0	1	0	Standard I/O
0	0	1	1	Special I/O
0	1	0	0	Segment trap acknowledge
0	1	0	1	Non-maskable interrupt acknowledge
0	1	1	0	Non-vectored interrupt acknowledge
0	1	1	1	Vectored interrupt acknowledge
1	0	0	0	Memory data request
1	0	0	1	Memory stack request
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Program reference, nth word
1	1	0	1	Instruction fetch, 1st word
1	1	1	0	Reserved
1	1	1	1	Reserved

B/*W Byte/Word (output, low = word, 3-state): This line indicates whether a byte or word of data is to be transmitted during a transaction.

***WAIT** (input, active low): A low on this line indicates that the responding device needs more time to complete a transaction.

***MREQ Memory Request** (output, active low, 3-state): A falling edge on this line indicates that the address/data bus is holding a memory address.

***AS Address Stobe** (output, active low, 3-state): The rising edge of *AS indicates the beginning of a transaction and shows that the Address, ST0-ST3, N/*W, R/*W and B/*W signals are valid.

***DS Data Strobe** (output, active low, 3-state): This line provides timing for data movement to or from the CPU.

R/*W Read/Write (output, low = write, 3-state): This line determines the direction of data transfer for memory input/output transactions. For memory read R/*W = high; for memory write R/*W = low. For input/output transactions the R/*W line indicates the direction of data transfer. Peripheral to CPU: Read R/*W = high; CPU to peripherial Write R/*W = low.

N/*S Normal/System Mode (output, low = system mode, 3-state): In system mode, all instructions can be executed and all CPU registers are accessed. This mode is intended for use by programs performing operating system functions. In normal mode some instructions may not be executed and the control registers of the CPU are inaccessible. In general this mode of operation is intended for use by application programs.

Bus Control Pins - These pins carry signals for requesting and obtaining control of the bus from the CPU.

***BUSREQ Bus Request** (input, active low): A low on this line indicates that a bus requester has obtained or is trying to obtain control of the bus.

***BUSACK Bus Acknowledge** (output, active low): A low on this line indicates that the CPU has relinquished control of the bus in response to a bus request.

Interrupt and Trap Pins - These pins convey interrupt and external trap requests to the CPU.

***NMI Non-maskable Interrupt** (input, edge activated): A high to low transition on *NMI requests a non-maskable interrupt. The *NMI interrupt has the highest priority of the three types of interrupts. NMI is reserved for events that require immediate attention.

***NVI Non-vectored interrupt** (input, active low): A low on this line requests a non-vectored interrupt. A non-vectored interrupt relies on the system software to determine its cause.

***VI Vectored Interrupt** (input, active low): A low on this line requests a vectored interrupt. A vectored interrupt causes 8 bits of the vector output by the interrupting device to be used to select a particular interrupt service procedure to which the program branches.

***SEGT Segment Trap** (input, active low): A low on this line requests a segment trap. Generated by addressing a non-existent segment.

Multi Micro Pins - These lines form a resource-request daisy chain that allows one CPU in a multi-processor system to access a shared resource. THESE TWO PINS ARE NOT USED BY THE M20.

*MI Multi-Micro In (Input, active low).
*MO Multi-Micro Out (Output, active low).

CPU Control Pins - These pins carry signals which control the overall operation of the CPU.

***STOP** (input, active low): This line is used to suspend CPU operation during the fetch of the first word of an instruction.

***RESET** (input, active low): A low on this line resets the CPU.

Register Organization

The Z8001 is organized around a general purpose register file. Figure 2-3 is a functional block diagram of the Z8001. All general purpose registers can be used as accumulators, and all but one as index registers or memory pointers.

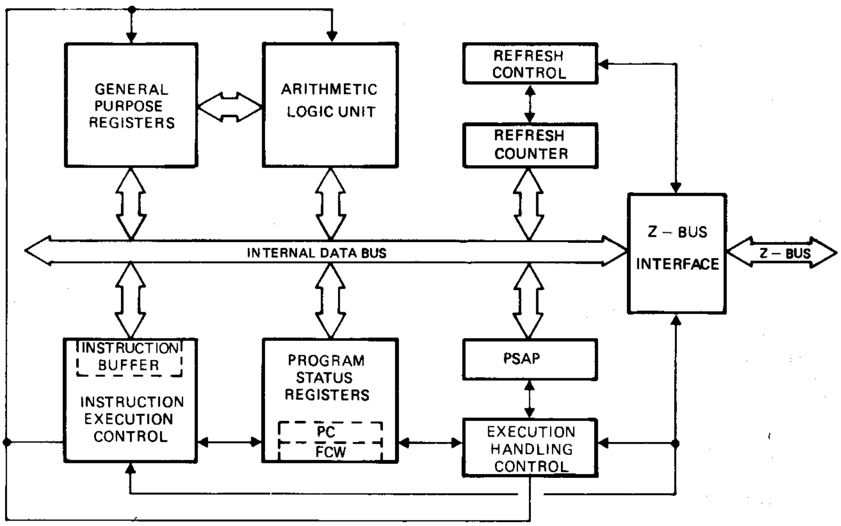


Fig. 2-3 Z8001 Functional Block Diagram

The Z8001 register file can be addressed as:

16 byte registers (occupying the upper half of the file)
16 word registers
8 long-word registers
4 quadruple-word registers
a mixture of the above

In addition to the general purpose registers the Z8001 also contains a number of special purpose registers. These include the Flag and Control Word (FCW) and the Program Counter (PC) which together are known as the Program Status Registers. The PC stores the address of the next instruction to be performed. The PC in the Z8001 consists of two words. Seven bits of the first PC word designate one of the 128 memory segments. The second word supplies the 16 bit offset that designates a memory location with the segment. The Program Status Registers are used to keep track of the state of the executing program.

The Z8001 also contains a programmable counter than can be used to refresh dynamic memory. This register is the Refresh Counter. Figure 2-4 shows the Program Status Registers formats.

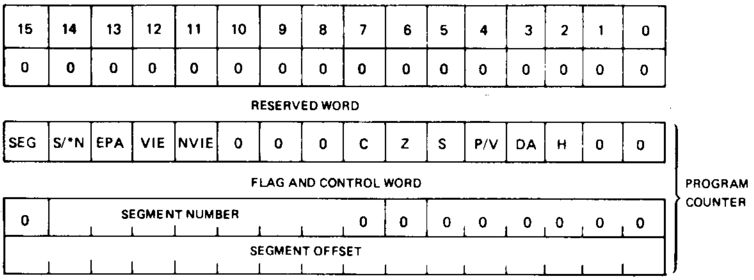


Fig. 2-4 Program Status Registers Formats

Interrupt and Trap Structure

The M20 provides a powerful interrupt and trap structure. Interrupts are external asynchronous events requiring CPU attention, and are generally triggered by peripherals needing service. Traps are synchronous events resulting from the execution of certain instructions. Both are processed in a similar manner by the CPU.

The M20 supports three types of interrupts (non-maskable, vectored and non-vectored), three internal traps (system call, unimplemented instruction, priveleged instruction) and a segment trap. Segment is the only external one. The descending order of priority for traps and interrupts is as follows:

non-maskable interrupts
segmentation trap
vectored interrupts
non-vectored interrupts

The M20 has 10 interrupts in all. There are eight vectored interrupts, a non-maskable and a non-vectored interrupt. The descending order of priority for interrupts is as follows:

```
NMI (non-maskable interrupt)
IR0 (INFTDC)   caused by floppy disk controller
IR1 (COMVI1)   caused by external daisy chain request
IR2 (COMVI2)   caused by external daisy chain request
IR3 (INTRxDRT) caused by DTE 8251A receive data
IR4 (INTRxDKY) caused by the keyboard 8251A Receive Data
IR5 (INTTxDRT) caused by DTE 8251A  transmit  data  or
               keyboard transmit data (jumper selectable)
IR6 (INTTxDRY) caused by the PPI 8255 PC0 or PC3 (jumper selectable)
IR7 (SYSINT)   caused by external daisy chain request
NVI (non-vectored interrupt) caused by timer
```

Interrupt and Trap Handling

The CPU response to an interrupt or trap request consists of five steps: acknowledging the external request (for interrupts and segment traps), saving the old program status area, loading the new program status area, extracting the service routine, and returning to the interrupted task.

Acknowledge Cycle - An external acknowledge cycle is required only for externally generated requests. The main effect of such a cycle is to receive from the external device a 16 bit identifier word, which will be saved with the old program status.

Status Saving - The old program status information is saved by being pushed on the system stack in the following order: Program Counter (PC: 16 bit offset followed by a word containing the 7 bit segment number); the Flag and Control Word (FCW); and finally the interrupt/trap service identifier word. The identifier word contains the reason or source of the trap or interrupt. For internal traps, the identifier is the first word of the trapped instruction. For segment trap or interrupts, the identifier is the value on the data bus read by the CPU during the interrupt-acknowledge or trap-acknowledge cycle. Figure 2-5 shows the format of the saved program status in the system stack.

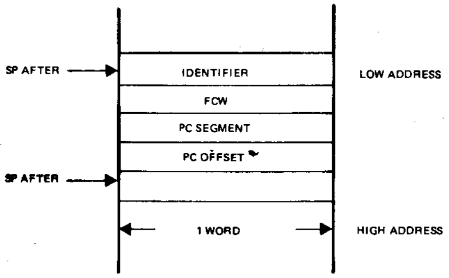


Fig. 2-5 Format of Saved Program Status

Loading New Program Status - After saving the current program status, the new program status (PC and FCW) is loaded from the Program Status area in the system program memory. The particular status words fetched from the Program Status Area are a function of the type of trap or interrupt and (for vectored interrupt) of the interrupt vector. Figure 2-6 shows the format of the Program Status Area.

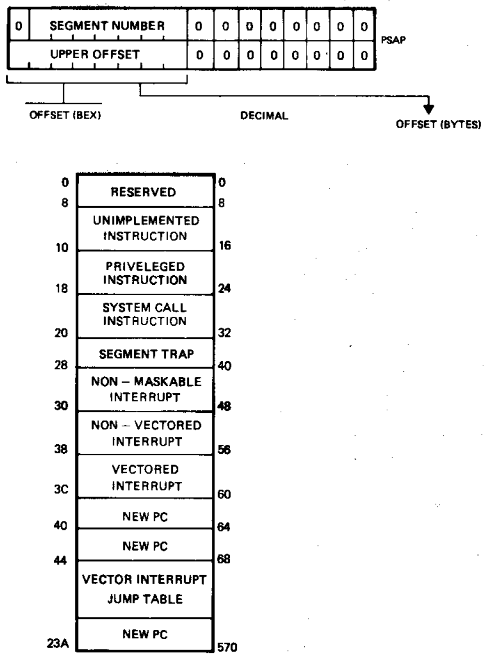


Fig. 2-6 Program Status Area

For each kind of interrupt, or trap other than a vectored interrupt, there is a single program status block that is loaded into the Program Status registers (which includes the FCW and the PC).

For vectored interrupts, the same FCW is loaded from the corresponding program status block. However, the PC value is selected from up to 128 different values in the Program Status Area. The low-order eight bits of the identifier placed on the data bus by the interrupting device is multiplied by two and used as an offset into the Program Status Area following the FCW for vectored interrupts. The identifier value 0 selects the first PC value, the value 2 selects the second PC value and so on up to the identifier 254, which selects the 128th PC value. All vectors must be even.

The Program Status Area is addressed by the Program Status Area Pointer (PSAP). As shown in Figure 2-6 the pointer contains a segment number and the high-order byte of a 16 bit offset address. The low-order byte is assumed to contain zeroes, thus the Program Status Area must start on a 256 byte address boundary. The PSAP is accessed using the Load Control Register Instruction (LDCTL).

Executing The Service Routine - Loading the new program status initializes the PC to the starting address of the service routine to process the interrupt or trap. This program is now executed. Because a new FCW was loaded, the maskable interrupts (NVI and VI) can be disabled for the initial processing of the service routine by a choice of FCW. This allows critical information to be stored before subsequent interrupts are handled.

Returning to the Interrupted Task - Upon completion, the service routine can execute an Interrupt Return Instruction (IRET), to case execution to continue at the point where the interrupt or trap occurred. IRET causes information to be popped from the system stack in the following order: the identifier is discarded, the saved FCW and PC are restored. The newly loaded FCW takes effect with the next fetched instruction, which is determined by the restored PC.

2.1.1.2 Interrupt Control Logic

The circuitry that handles the vectored priority interrupts to the CPU is known as the interrupt Control Logic. The LSI component used to perform this function is the Intel 8259A Programmable Interrupt Controller.

The Programmable Interrupt Controller (PIC) functions as an overall manager in the interrupt-driven system environment to establish the priority of the eight vectored interrupts. The PIC is also cascadable for up to 64 vectored priority interrupts without additional circuitry. The PIC accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest priority, ascertains whether the incoming request has a higher priority than the level currently being serviced, and issues an interrupt to the CPU based on this determination. Figure 2-7 shows a block diagram of the 8259A PIC.

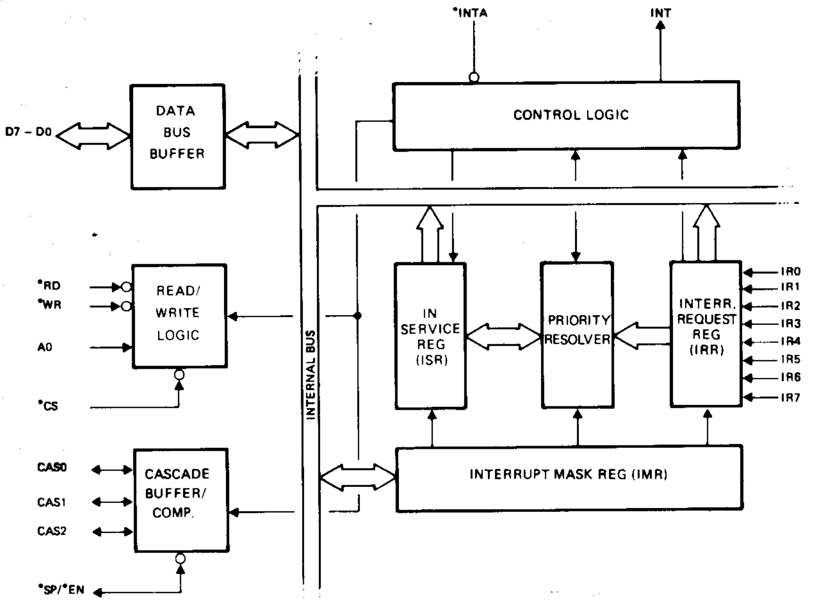


Fig. 2-7 8259A Block Diagram

Pin Functions

Figure 2-8 shows the pin functions of the 8259A PIC.

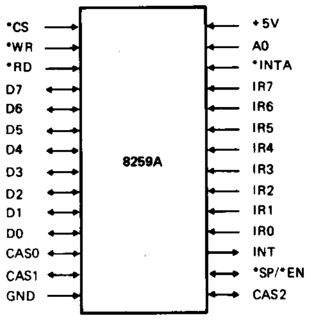


Fig. 2-8 8259A Pin Functions

***CS Chip Select**: A low on this input enables *RD and *WR communication between the CPU and the 8259A. *INTA functions are independent of *CS.

***WR Write**: A low on this input when *CS is low enables the 8259A to accept command words from the CPU.

***RD Read**: A low on this input when *CS is low enables the 8259A to release status onto the data bus for the CPU.

D0-D7 Data Bus: Control, status and interrupt-vector information is transferred on this bidirectional data bus.

CAS0-CAS2 Cascade: The CAS lines form a private 8259A bus to control a multiple 8259A structure. The CAS lines are outputs for a master 8259A and inputs for a slave 8259A.

***SP/*EN Slave Program/Enable Buffer**: When in the buffered mode this line can be used to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP=1) or slave (SP=0).

INT Interrupt: This output goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU.

IR0-IR7 Interrupt Request: Asyncronous inputs. An interrupt request is executed by raising an IR input (low to high) and holding it high until it is acknowledged (edge triggered mode), or just by a high level on an IR input (level triggered mode).

***INTA Interrupt Acknowledge**: This input is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.

A0 Address: This input acts in conjunction with the *CS, *WR and *RD inputs. It is used by the 8259A to decipher various command words the CPU writes and status the CPU wishes to read.

2.1.1.3 Initialization

A Power-up of the system starts the initialization sequence. A reset signal is generated that causes the CPU to call for instructions stored in EPROM to initialize system logic.

Reset may also be generated by pressing the Reset Button. Whenever a reset signal is generated the initialization sequence is immediately performed. It takes precedence over all other procedures.

Reset and Bootstrap

Figure 2-9 shows a logic diagram of the reset and bootstrap circuitry. A low on the CPU *RESET input starts the initialization sequence. CPU address and segment bus lines are reset, selected control and strobe lines are reset, and the status bits indicate that an internal process is taking place. External lines such as Read/Write and Byte/Word are not affected by the initialization sequence.

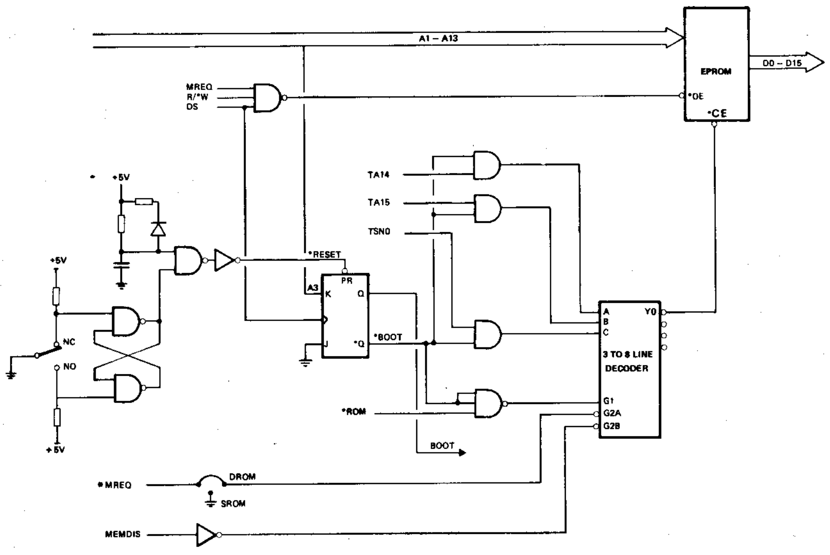


Fig. 2-9 Bootstrap Circuitry Logic Diagram

The *RESET signal presets the boot ROM enable flip-flop. The *BOOT signal inhibits the action of the translated address signals TA14, TA15, TSNO and *ROM and takes the ROM decoder select inputs A,B & C low via the AND gates and the G1 input high via the NAND gate. This sets the decoder data output Y0 low and enables the EPROM. The EPROM is then addressed to start the bootstrap operation. Three consecutive memory read cycles are executed in the system mode. The first reads the FCW from location 0002, the second reads the 7 bit PC segment number from location 0004 and the third reads the 16 bit PC offset from location 0006 (see Figure 2-10). The last IF cycle starts the initialization program. On completion of the program, address bit A3 is active, the boot ROM enable flip-flop is reset, and normal addressing is resumed.

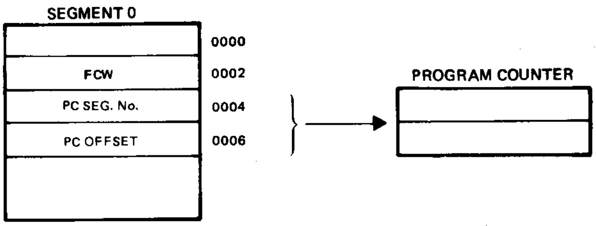


Fig. 2-10 Initialization Sequence

2.1.1.4 Erasable Programmable Read Only Memory

Memory storage for the Power-up Bootstrap and Power-up Diagnostics is provided by the EPROM. Two 4K x 8 bit EPROMs provide the 8 KB of memory. The data outputs of the EPROM are connected to the bidirectional data buffer lines D0-D7 and the address lines are connected to the address latch output lines A1-A13. The EPROM is controlled by the chip enable signal (*CE) and output enable signal (*OE). The EPROM is only accessed by the CPU.

2.1.1.5 Random Access Memory

Memory storage for user programs and data and the operating system software and BASIC interpreter (loaded from diskette), is provided by the Random Access Memory (RAM). In its standard configuration the M20 holds 16 RAM IC's organized 65,536 words x 1 bit to provide a total of 128 KB of memory. The RAM's used are the NEC uPD41664/3 or the Hitachi 4864/2. The two types cannot be mixed on the same motherboard.

Of the total RAM available, the bit-mapped display for the monochrome system is assigned 16 KB, the operating system software and BASIC interpreter 64 KB and the remaining 48 KB for programming use. Of this, at least 32 KB is available for user programs.

The memory expansion board options provide the means to increase RAM to a maximum capacity of 224 KB or 512 KB. For the colour Display options memory expansion is necessary. The RAM requirement for the bit-mapped Display increases to 32 KB for the 4 colour system and 48 KB for the 8 colour system.

Address Spaces

Programs and data may be in the main memory of the computer system or in peripheral devices. In either case, the location of the information must be specified by an address before the information can be accessed. A set of these addresses is called an address space. The M20 supports two different types of addresses and thus two categories of address space.

Memory Addresses - which specify locations in main memory. This space is subcategorized into:

> Instruction space (normal or system mode). These spaces typically address memory that contains user programs (normal mode) or system programs (system)
>
> Data space (normal or system mode). These spaces may be used to address the data that user or system programs operate on.
>
> Stack Space (normal or system mode). These spaces can be used to address the system and normal program stacks.

Input/Output Addresses - which specify the ports through which peripheral devices are accessed.

Segmented Memory Address

Segmentation is a means of partitioning memory into segments so that a variety of useful functions may be implemented including protection mechanisms that prevent a user from referencing data belonging to others, attempting to modify read-only data or overflowing a stack. The M20 uses the Z8001 with segmented address capability. Three segment lines and 16 address lines are used to address a total of 256 KB of RAM and ROM. Each segment can hold a maximum of 64 KB.

Address Translation

The entire memory resources of the M20 can be visualized as a series of 16 KB memory blocks, whether they be RAM or ROM. The address translation/mapping or relationship between the logical memory addresses as viewed by the programmer, and the physical memory elements as addressed by hardware is accomplished through a mapping PROM (a fast bipolar ROM). Figure 2-11 is a logic diagram of the translation PROM showing the input/output functions.

Due to the address mapping required in the monochrome configuration, the mapping PROM is used to perform limited address translation, as well as device selection functions. This PROM can be viewed as a simple Memory Management Unit without operating system control.

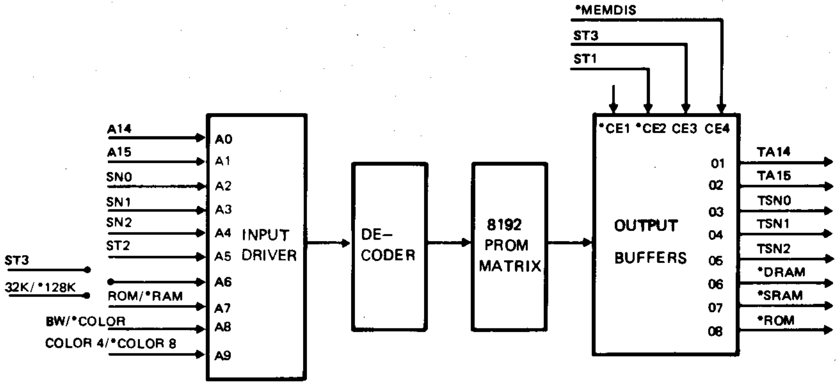


Fig. 2-11 Mapping PROM Logic Diagram

The PROM address inputs consists of CPU address lines A14, A15, CPU status line ST2, CPU segment lines SN0-SN2, and operating system control lines *COLOUR and *RAM. Data output of the PROM consists of translated address lines TA14, TA15, translated segment lines TSN-TSN2, and device select lines *ROM, *DRAM, *SRAM. *DRAM selects the dynamic RAM, that is the RAM on motherboard and expansion boards while *SRAM selects the RAM on the videotex board.

Memory Management has two functions: the efficient allocation and reallocation of memory space to executing tasks so as to optimize overall memory usage; and the protection of memory contents from unintended or unauthorised accesses by executing tasks. The memory management unit takes the logical addresses and translates them into physical addresses for accessing memory.

M20 memory is configured according to the following scheme:

SEGMENT	CONTENTS
0	PCOS kernel
1	Basic Interpreter and PCOS utilities
2	PCOS variables, Basic stock and tables, user memory
3	Screen Bit-map
4	Diagnostics and Bootstrap

Memory Organization

Although M20 is word organized, memory is addressed as bytes. All instructions are word aligned using even addresses. Memory is divided into two banks: an upper byte bank (even addresses), and a lower byte bank (odd addresses). Figure 2-12 is a simplified logic diagram of the RAM memory, organized in upper and lower byte banks, and circuitry for generating signals. (*WEL Write Enable Lower; *WEU Write Enable Upper; *CASU Column Address Strobe Upper; *CASL Column Address Strobe Lower; *RASL Row Address Strobe Lower; *RASU Row Address Strobe Upper.)

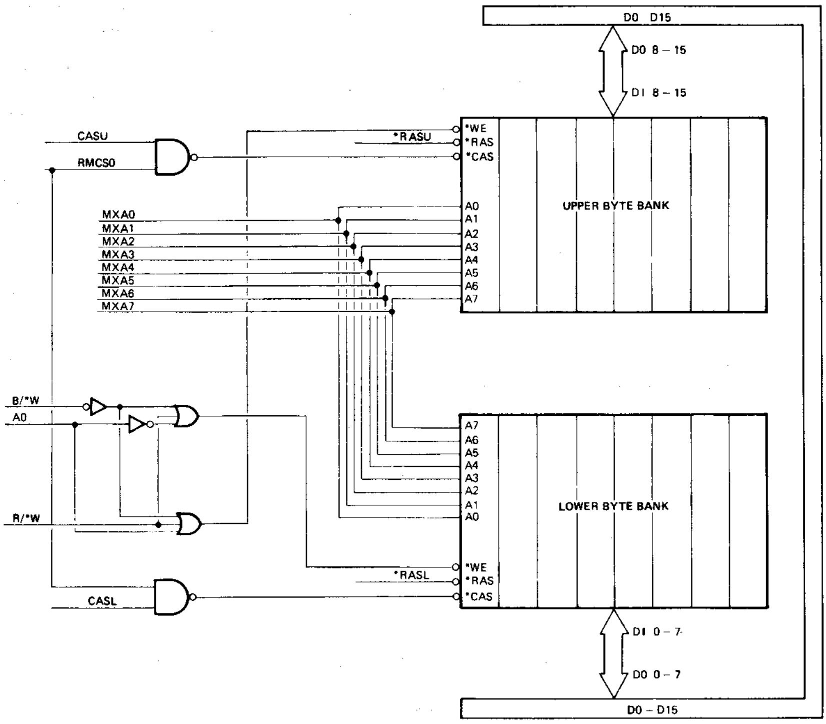


Fig. 2-12 Memory Simplified Logic Diagram

When RMCS0 is active RAM memory on the motherboard is being addressed. When RMCS1, RMCS2, RMCS3 are active, memory on the first, second and third plug-in boards are respectively addressed.

Refresh of the dynamic cell matrix is accomplished by Row Address Strobe (RAS).

The 16 address bits required to decode 1 of the 65,536 cell locations within each 64K bit RAM are multiplexed onto the 8 memory address inputs and latched into the on-chip address latches by applying two clocks. The first clock, the Row Address Strobe (*RAS) latches the 8 row addresses into the chips. The second clock, the Column Address Strobe (*CAS), subsequently latches the 8 column addresses into the chips. Each of these clocks, *RAS, *CAS, triggers a sequence of events which are controlled by different delayed internal clocks. The two clock chains are linked together logically in such a way that the address multiplexing operation is performed outside of the critical path timing sequence for read data access.

Data to be written into a selected cell is latched into an on-chip register by a combination of *WRITE and *CAS while *RAS is active. The latter of *WRITE or *CAS to make its negative transition is the strobe for the Data In (DIN) register. This permits several options in the write cycle timing.

In a write cycle, if the *WRITE input is brought low (active) before *CAS, DIN is strobed by *CAS and the set-up and hold times are referenced to *CAS. If the input data is not available at *CAS time or if it is desired that the cycle be a read-write cycle, the *WRITE signal will be delayed until after *CAS has made its negative transition. In this "delayed write cycle" the data input set-up and hold times are referenced to the negative edge of *WRITE rather than *CAS.

Data is retrieved from the memory in a read cycle by maintaining *WRITE in the inactive or high state throughout the portion of the memory cycle in which *CAS is active (low). Data read from the selected cell is available at the data output within the specified access time.

2.1.1.6 Timing

The basic timing periods of the Z8001 are; a clock cycle, a bus transaction, and a machine cycle. These are illustrated in Figure 2-13. A clock cycle (sometimes called a T state) is one cycle of the CPU clock, starting with a rising edge. A bus transaction covers a single data movement on the CPU bus and will last for three or more clock cycles, starting with a falling edge of the Address Strobe (*AS) and ending with a rising edge of the Data Strobe (*DS). A machine cycle covers one basic CPU operation and always starts with a bus transaction. A machine cycle can extend beyond the end of a transaction by an unlimited number of clock cycles.

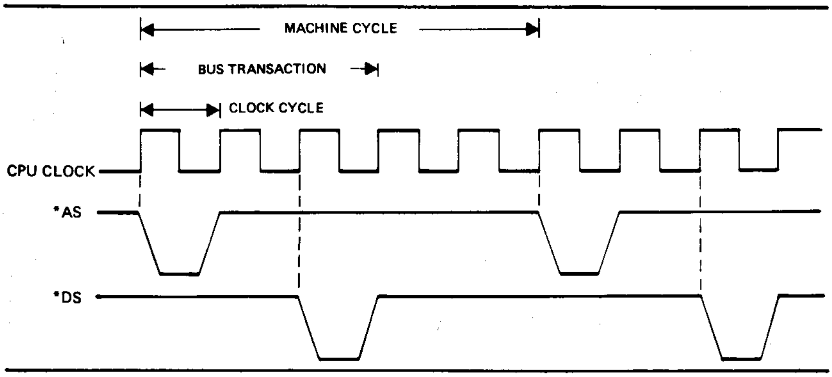


Fig. 2-13 Basic Timing Periods

The following is divided into two parts:

Normal Z8001 Timing
Modified Timing as used by the M20

Normal Z8001 Timing

The Z8001 executes instructions by stepping through a sequences of basic CPU operations. These include:

Memory read or write
Input/Output device read or write
Interrupt acknowledge and internal execution
Bus request acknowledge

Each basic operation can take three to ten clock cycles to execute. Instructions that require more clock cycles to execute are broken up into several machine cycles. Thus no machine cycle is longer than ten clock cycles, unless lengthened to accomodate slow devices, and fast response to a bus request is obtained.

Memory Read or Write - Figure 2-14 shows the memory read and write timing for the Z8001. Memory transactions move data to or from memory. They are used to fetch instructions from memory and fetch and store memory data. They also store old program status and fetch new program status during interrupt and trap handling and after reset. Memory read or write and instruction fetch operations are identical, except for the status information on the status lines.

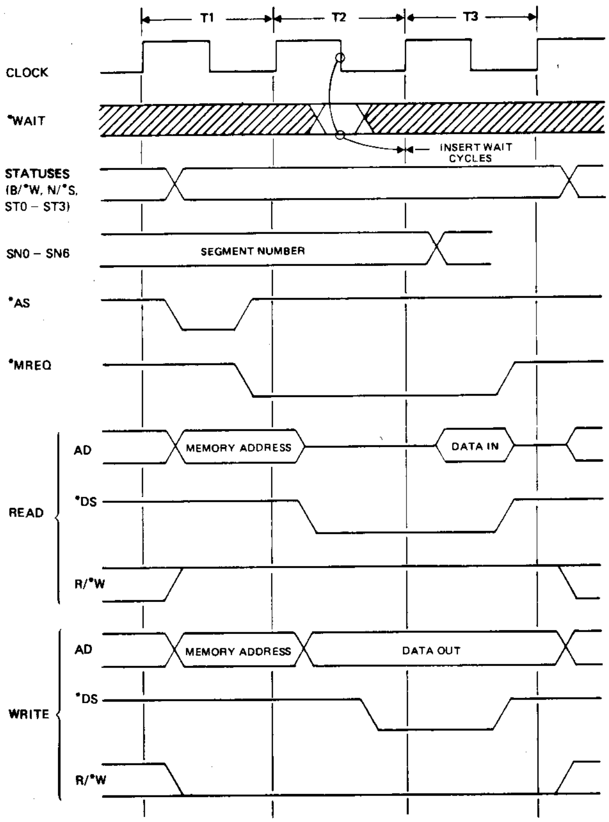


Fig. 2-14 Memory Read and Write Timing

During a memory read cycle, a 16-bit address is placed on the address lines AD0-AD15 early in the first clock cycle. The 7-bit segment number is output on status lines SN0-SN6 one clock cycle earlier than the 16-bit address offset to compensate for the delay in the memory management unit. A valid address is indicated by the rising edge of Address Strobe *AS. Status and mode information becomes valid early in the memory access cycle and remain stable throughout. The *WAIT line is sampled in the middle of the second clock cycle by the falling edge of the clock. If *WAIT is low, a wait cycle is inserted between T2 and T3. *WAIT is sampled again in the middle of this wait cycle, and additonal wait cycles can be inserted. This allows interfacing to slow memories. No control outputs change during wait cycles.

Input/Output Read or Write - Figure 2-15 shows the input/output timing for the Z8001.

Input/output transactions move data to or from peripherals or CPU supported devices. They are used during the execution of I/O instructions. I/O timing is similar to memory read/write timing, except one wait cycle is inserted between T2 and T3.

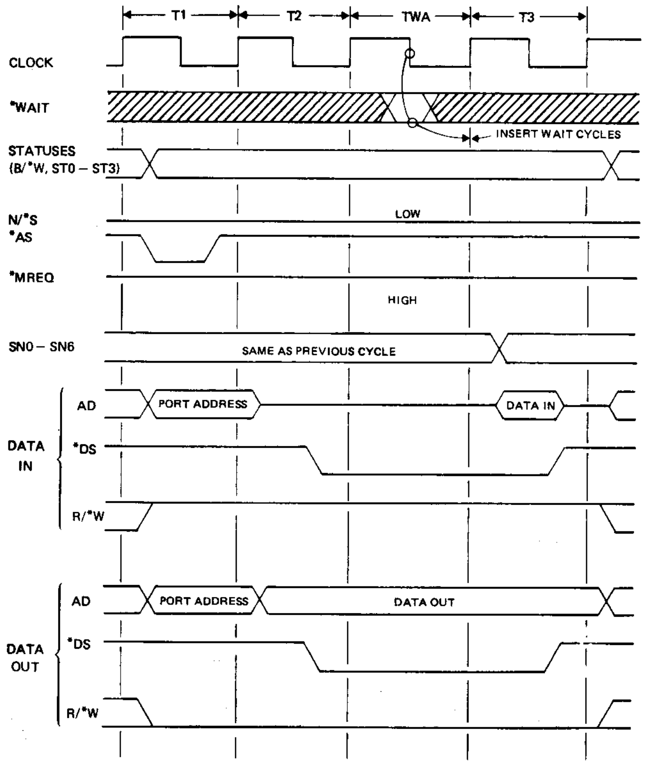


Fig. 2-15 Input/Output Timing

Interrupt and Segment Trap Request/Acknowledge - Figure 2-16 shows the interrupt and segment trap request/acknowledge timing for the Z8001.

The Z8001 CPU supports three interrupts: non-maskable (*NMI), vectored (*VI), and non-vectored (*NVI), and one segment trap (*SEGT). A high to low transition on the *NMI input is asynchronously edge detected and the internal *NMI latch is set. At the beginning of T3 in the last machine cycle of any instruction the *VI, *NVI and *SEGT lines are sampled together with the state of the internal *NMI latch. If an interrupt or trap is detected, the subsequent instruction fetch cycle is exercised, but aborted. The program counter is not updated, but the system stack pointer is decremented.

The next machine cycle is the interrupt acknowledge transaction. This machine cycle has five wait cycles inserted between T2 and T3.

After the last wait cycle, the CPU reads the 16-bit identifier on the address lines AD0-AD15 and stores it temporarily, to be saved on the stack later in the acknowledge cycle.

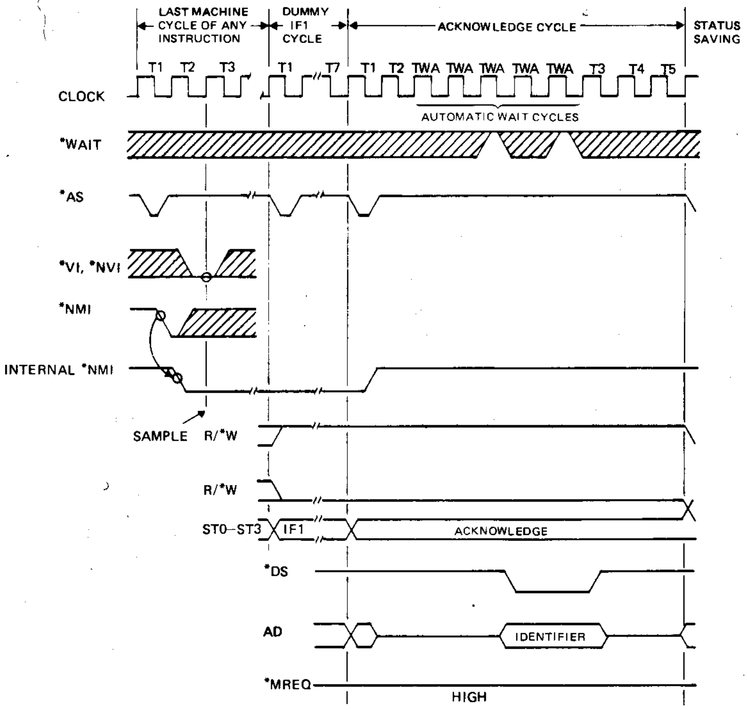


Fig. 2-16 Interrupt and Trap Request/Acknowledge Timing

For the non-vectored and non-maskable interrupts, all 16 bits can represent peripheral device status information. For the vectored interrupt, the low byte is the jump vector, and the high byte can be extra user status information. For the segment trap, the high byte is the memory management unit identifier and the low byte is undefined. Note that the Mapping PROM in the M20 performs the same work as a conventional memory management unit.

After the acknowledge cycle, the N/*S output indicates the change to system mode.

Status Saving Sequence: The machine cycles following an interrupt acknowledge or segment trap acknowledge cycle push the old status information on the system stack in the following order:

program counter 16-bit offset
program counter 7-bit segment number
flag and control word
interrupt/trap identifier word

Subsequent machine cycles fetch the new program status from the program status area, and then branch to the interrupt/trap service routine.

Bus Request/Acknowledge - Figure 2-17 shows the bus request/acknowledge timing for the Z8001.

A low on the *BUSRQ input indicates to the CPU that another device is requesting the Address/Data and Control buses. If the external *BUSRQ line is low at the beginning of any machine cycle, an internal synchronous *BUSRQ signal is generated, which after completion of the current machine cycle, causes the *BUSAK output to go low and all CPU outputs to go into the high impedance 3-state. The requesting device can then control all buses. When BUSRQ is released, it is synchronized with the rising clock edge and the *BUSAK output goes high one clock period later, indicating that the CPU will again take control of the bus.

M20 Timing

As described, Z8001 transactions can be lengthened to accomodate a slow device by the insertion of wait cycles. If a responding device needs more time to complete a transaction, wait cycles are inserted between T2 and T3 by making the *WAIT input active (low). The CPU then waits for periods of 250 ns. As these wait periods are too long for certain M20 operations, the Z8001 *WAIT input is normally held inactive (high) and T3 stretched to produce effective wait periods of 62.5 ns.

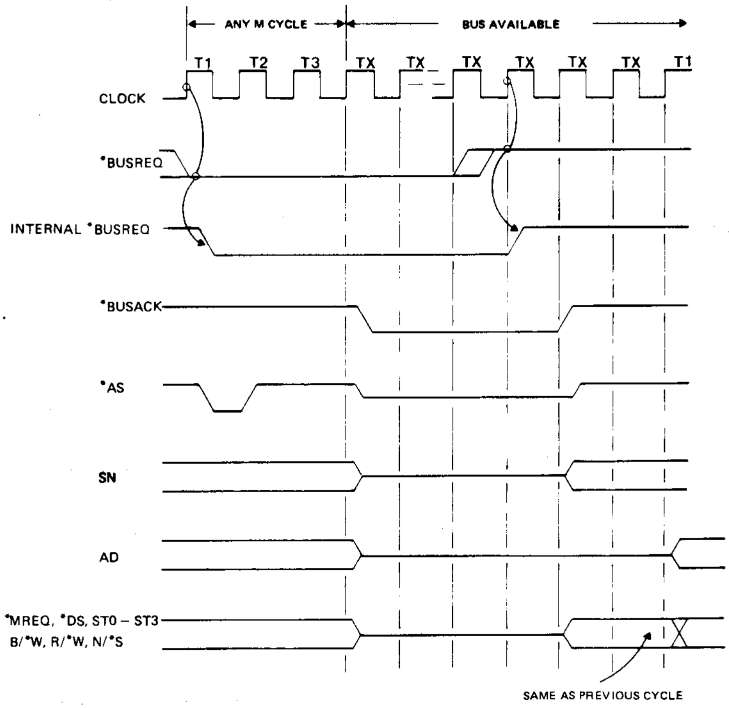


Fig. 2-17 Bus Request/Acknowledge Timing

T3 Stretching - Figure 2-18 shows the T3`stretch' timing and Figure 2-19 shows a logic diagram of the T3`stretch' circuitry.

When the CPU signal *MREQ (Memory Request) goes active, the BOOT output of the Boot ROM enable flip-flop is not active. *DRAM (dynamic RAM) is also low. As a result the On Board Memory Select (*OBMS) signal goes active. Note that *OBMS is asserted whenever the CPU requests memory service. Since *BOOT is high the T3 Stretch Latch 1 is set and SWAIT goes high. *SWAIT goes active low and the output *Q of T3 Stretch Latch 2 goes high. The two inputs of the Counter Disable NAND gate are form by the T3 Stretch Latch 2 *Q output and the Data Strobe Latch *Q output.

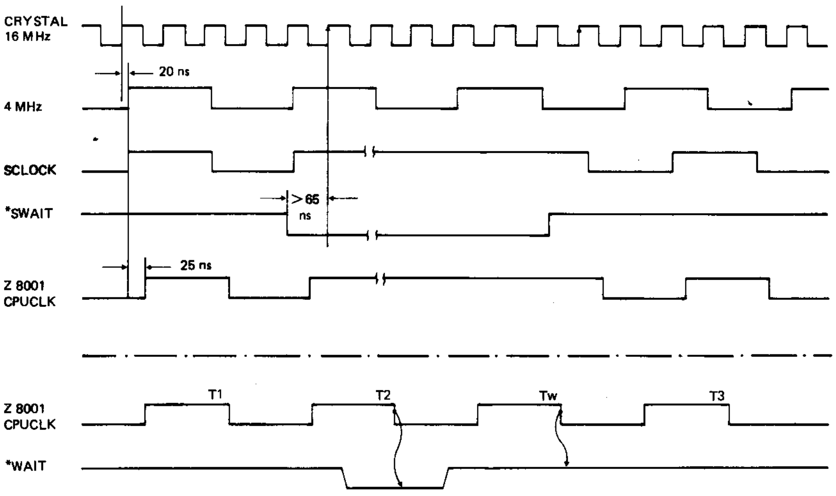


Fig. 2-18 T3 Pulse Stretch Timing

The Signal *RAMDTK is active whenever the CPU is using the memory and inactive whenever the CRT is using the memory.

In normal operation the *RAMDKT signal is active before the data strobe signal (*DS) is active. In this case, *RAMDTK resets the T3 Stretch Latches, that is output *Q of T3 Stretch L1 goes low and *Q of T3 Stretch Latch L2 also goes low and the counter is not disabled.

If *RAMDTK is asserted after *DS, then both inputs to the Counter Disable NAND gate go high, the counter is disabled and the CPU clock held at T3 time. When *RAMDTK is asserted, the T3 Stretch latches are reset, and the counter enabled. The CPU clock then starts to advance.

Figure 2-20 shows timing waveforms. Figure 2-21 shows a logic diagram of the Memory Timing circuitry.

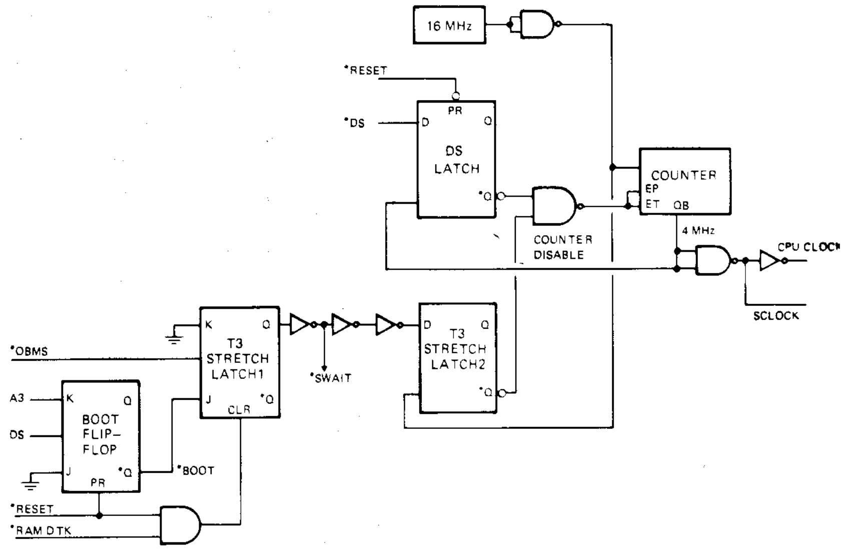


Fig. 2-19 Pulse Stretch Logic Diagram

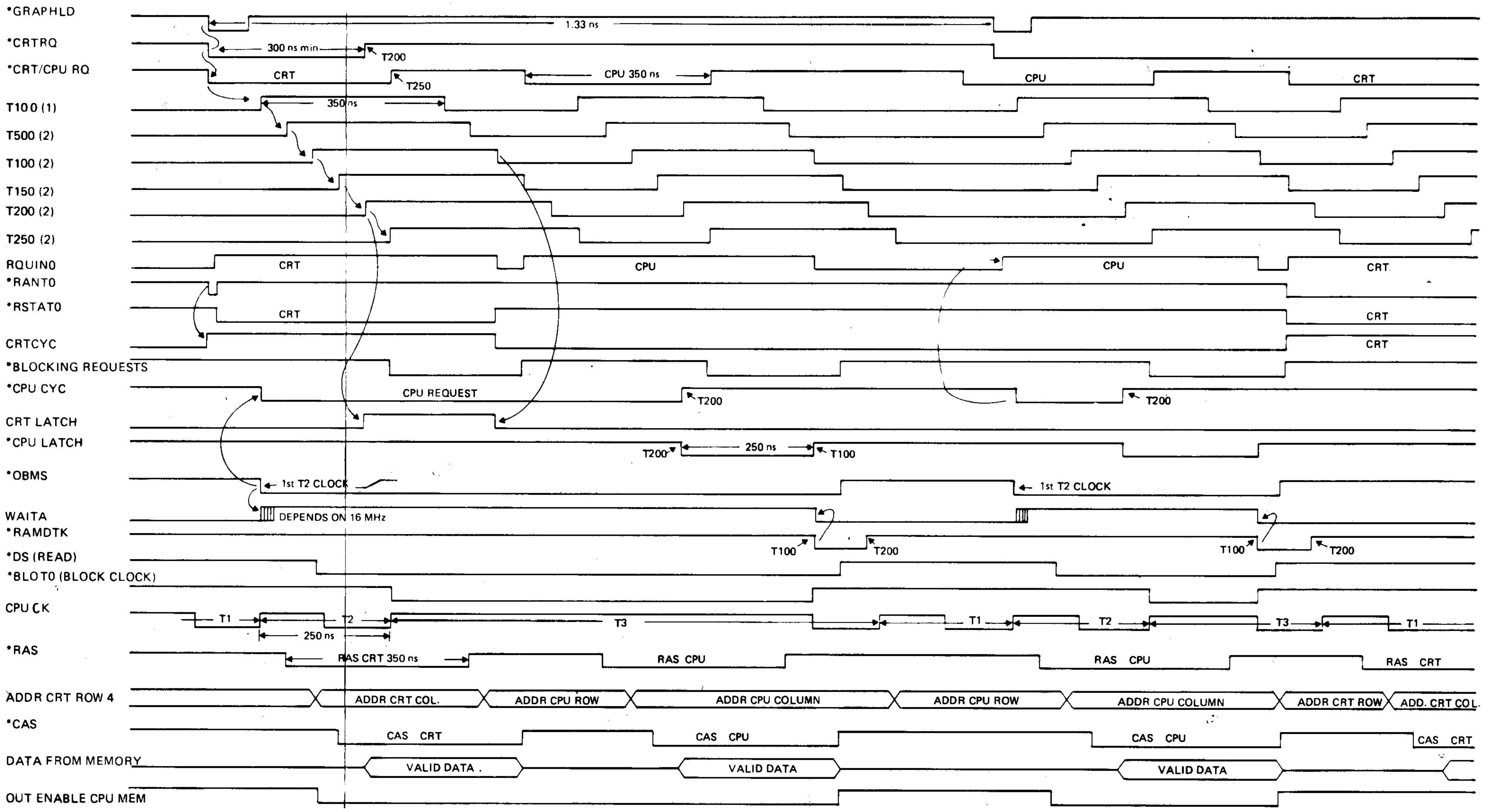


Fig. 2-20 Timing Waveforms

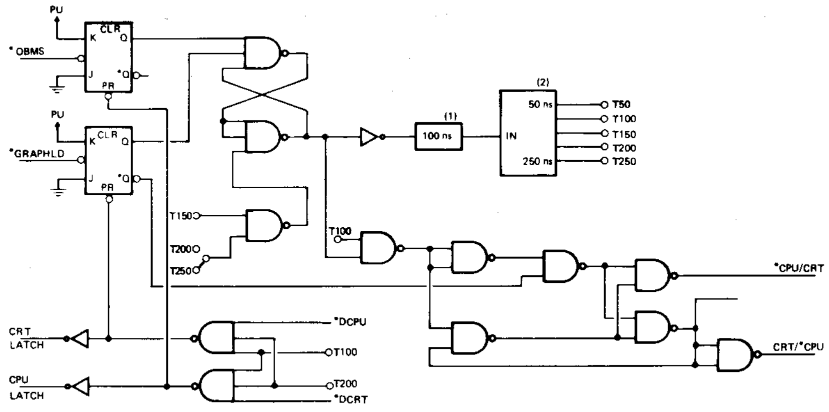


Fig. 2-21 Memory Timing Circuitry - Logic Diagram

2.1.1.7 Dual Communication Serial Interface

The Dual Communication Serial Interface uses two Intel 8251A Programmable Communication Interfaces (PCI) to provide the M20 with one RS-232-C type serial interface and one keyboard interface. The RS-232-C type serial interface is used to interface the M20 with a modem or plotter.

The PCI, also referred to as a Universal Synchronous/Asynchronous Receiver/Transmitter (USART), accepts data characters from the M20 CPU in parallel format and then converts them into a continuous stream of serial data for transmission. It can simultaneously receive serial data streams, as in the case of the keyboard interface, and then convert them into parallel format for the M20 CPU. It signals the CPU whenever it can accept a new character for transmission, or whenever it has received a new character. Figure 2-22 is a logic diagram of the Dual Communication Serial Interface.

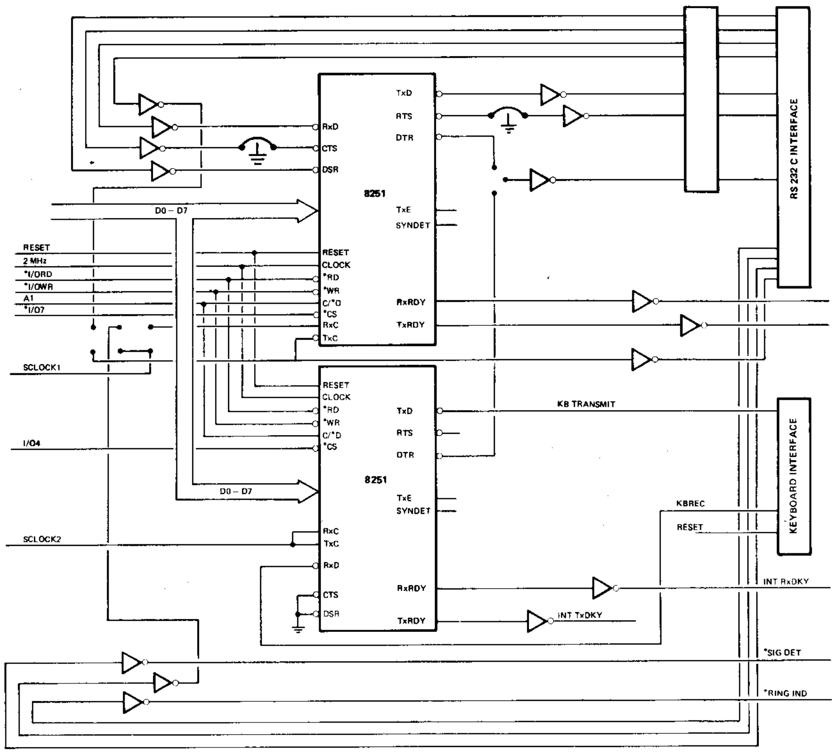


Fig. 2-22 Dual Communications Serial Interface Logic Diagram

Programming of the 8251A is performed by the M20 system software which issues a set of control words, mode and control, to initialize the 8251A to support M20 communication formats.

For the M20, the 8251A operates with a communication format of; 2 stop bits, even parity, parity disable, character length of 8 bits, baud rate factor 16X and a command format of; no hunt mode, no internal resets, Request to Send=0, error reset, no break character, receive enabled, Data Terminal Ready=0, transmit enabled.

Programmable Communication Interface (PCI)

The PCI 8251A features include:

Full duplex, double buffered, transmitter and receiver.
Synchronous or asynchronous operation.
Wide range of Baud rates.
Error detection, Parity, Overrun and Framing.

The 8251A has double-buffered data paths, with separate Input/Output registers for control, status, Data in and Data out. Figure 2-23 shows a simplified block diagram of the 8251A.

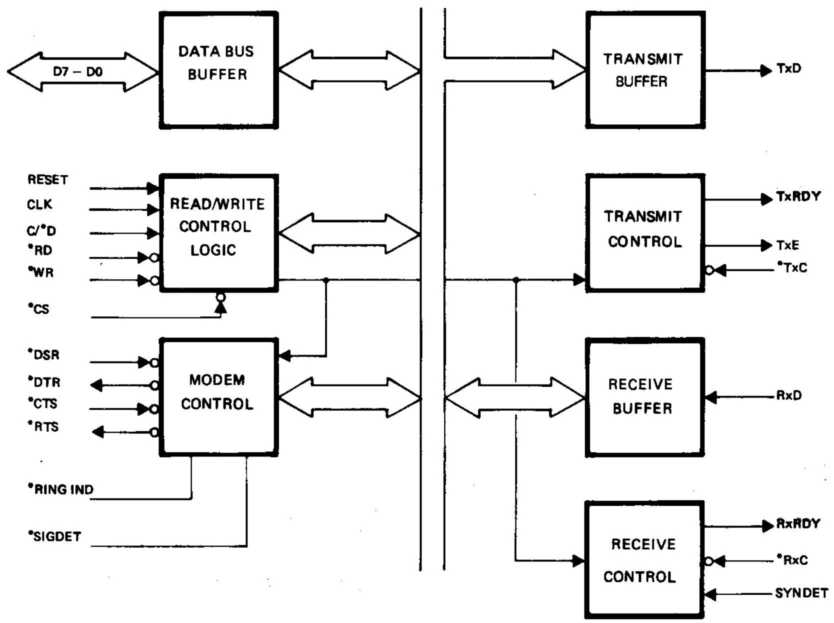


Fig. 2-23 8251A Block Diagram

Pin Functions

Figure 2-24 shows the pin functions of the 8251A. The 8251A pins can be grouped according to the functional units as outlined in Figure 2-23.

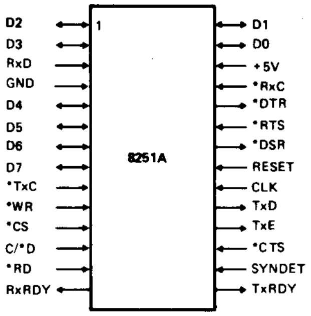


Fig. 2-24 8251A Pin Functions

Data Bus Buffer Pins

DO-D7 Data: The 8251A interfaces with the M20 CPU via this 3-state bi-directional data bus. Data is transmitted or received by the internal buffers upon execution of input or output instructions from the CPU. Control words, command words, and status information are also transferred via the Data Bus Buffer.

Read/Write Control Logic Pins

RESET: A high on this input forces the 8251A into an "idle mode". It will remain in this mode until a new set of control words are written into it.

CLK Clock: This input is used to generate internal device timing.

***WR Write:** A low on this input informs the 8251A that the M20 CPU is writing data or control words to the 8251A.

***RD Read:** A low on this input informs the 8251A that the M20 CPU is reading data or status information from the 8251A.

C/*D Control/Data: This input with the *WR and *RD inputs, informs the 8251A that the word on the data bus is either a data character, control word or status information.

***CS (Chip Select):** A low on this input selects the 8251. When it is high *RD and *WR will have no effect.

The following truth table summarizes the logic functions of the above inputs:

C/*D	*RD	*WR	*CS	FUNCTION
0	0	1	0	8251A Data ==> Data Bus
0	1	0	0	Data Bus ==> 8251A Data
1	0	1	0	Status ==> Data Bus
1	1	0	0	Data Bus ==> Control
X	1	1	0	Data Bus ==> 3-State
X	X	X	1	Data Bus ==> 3-State

Modem Control Pins - The Modem Control Unit is used to simplify the interface of the M20 CPU to almost any modem. Its signals are general purpose.

***DSR Data Set Ready:** This input is normally used to test modem conditions such as Data Set Ready.

***DTR Data Terminal Ready:** This output is normally used for modem control such as DT Ready or Rate Select.

***RTS Request To Send:** This output is used for modem control, such as Request To Send.

***CTS Clear To Send:** This input enables the 8251A to transmit serial data if the Tx enable bit in the command byte is set to ONE.

Transmit Buffer Pin

TxD Transmit Data: This output is a composite serial stream of data. The buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream and inserts the character bits (depending upon the communication technique).

Transmit Control Pins - This unit manages all activities associated with the transmission of serial data.

TxRDY Transmitter Ready: This output tells the M20 CPU that the transmitter is ready to accept data.

TxE Transmitter Empty: This output goes high when the 8251A has no characters to transmit. It resets automatically upon receiving a character from the M20 CPU.

***TxC Transmitter Clock:** This input controls the rate at which the character will be transmitted. In the synchronous transmission mode, the Baud rate is equal to the *TxC frequency. In the asynchronous transmission mode, the baud rate is a fraction of the *TxC frequency. This factor can be 1, 1/16, or 1/64 of the TxC.

Receiver Buffer Pin

RxD Receive Data: Serial data is input to this pin. The Receive Buffer accepts serial data and converts the serial input to parallel format. It also checks for the bits and characters that are unique to the communication technique, and produces an 'assembled' character for transmission to the M20 CPU.

Receiver Control Pins - This unit manages all receiver-related activities, including initialization, false start, parity and error response.

***RxRDY Receiver Ready:** This output indicates that the 8251A contains a character that is ready for input to the M20 CPU. (Failure to read character before assembly of next character will set overrun condition error)

***RxC Receiver Clock:** This input controls the rate at which the character is to be received. In the asynchronous transmission mode, the Baud rate is a fraction of the *RxC frequency. A portion of the mode instruction selects this factor: 1, 1/16, or 1/64 of the *RxC.

SYNDET/BRKDET Sync Detect/Break Detect: This line is used in the synchronous mode for either input (external synchronous mode) or an output (internal synchronous mode). The Break Detect is used in the asynchronous mode only.

Keyboard Interface

When a key on the keyboard is pressed, the key-code complete with control information is transmitted to the keyboard interface port J11. The key-code is transmitted in a serial half-duplex mode at a Baud rate of 1200. Transmission is asynchronous and the message is made up of one start bit, eight data bits, two stop bits.

This serial data is received on the 8251A RxD line and converted into parallel format by the receive buffer. The receive buffer also checks for bits and characters that are unique to the communication technique. It then informs the M20 CPU that it has a character to send by forcing low the *RxRDY output which in turn is the INT RXDKY (IR4) input to the interrupt control logic. Data is then sent to the M20 CPU on data lines D0-D7.

The 8251A TxD line is used to transmit the keyboard interrupt.

Serial Interface RS-232-C

The Electronic Industries Association (EIA) recommended standard RS-232-C defines the electrical characteristics for interfacing Data Terminal Equipment (DTE) to Data Communications Equipment (DCE). The DTE is the terminal for the timeshare user, while the DCE is a modem.

The European equivalent of the RS-232-C is Recommendation V24 of the International Consultative Committee for Telephone & Telegraph (CCITT).

A list of the RS-232-C interchange circuits available on the M20 Serial Interface Port J7 showing category as well as CCITT identification in accordance with Recommendation V24 is shown in Figure 2-25.

INTERCHANGE CIRCUIT	C.C.I.T.T. EQUIVALENT	DESCRIPTION	GND	DATA		CONTROL		TIMING	
				FROM DCE	TO DCE	FROM DCE	TO DCE	FROM DCE	TO DCE
AA	101	PROTECTIVE GROUND	*						
AB	102	SIGNAL GROUND/COMMON RETURN	*						
BA	103	TRANSMITTED DATA			*				
BB	104	RECEIVED DATA		*					
CA	105	REQUEST TO SEND					*		
CB	106	CLEAR TO SEND				*			
CC	107	DATA SET READY				*			
CD	108.2	DATA TERMINAL READY					*		
CE	125	RING INDICATOR				*			
CF	109	RECEIVED LINE SIGNAL DETECTOR				*			
DA	113	TRANSMIT SIGNAL ELEMENT TIMING							*
DB	114	TRANSMIT SIGNAL ELEMENT TIMING						*	
DD	115	RECEIVE SIGNAL ELEMENT TIMING						*	

Fig. 2-25 RS-232-C Interchage Circuits

If a printer is to be connected to the Serial Interface Port, the question arises as to which device is DTE and which is DCE. In this case neither the M20 nor the printer is a terminal or modem. A peripheral cable is available to permit such a configuration. A modem cable is also available to adapt the Serial Interface pin configuration to that defined in the RS-232-C standard.

On the transmit and receive data circuits the M20 uses +12V to represent a binary 0 and -12V to represent a binary 1. These levels apply only to data circuits.

For asynchronous serial data communications a start bit indicates the beginning of the character. This is followed by the character data bits, least significant first, and two stop bits. The stop bits allow the receiver time to assemble the serial data, send the assembled character on, and prepare for the next character. A parity bit may be inserted before the stop bits for error detection.

The DTE asserts Request to Send when it has data to transmit. It then waits for the DCE to assert Clear to Send before transmitting. RTS and CTS are handshake lines. However, it is the DTE and the communications link that are handshaking, not the DTE and DCE. Handshake is on a message basis not by character.

The Ring Detector and Signal Detector circuits are monitored by ports PB6 and PB7 of the Parallel Interface.

2.1.1.8 Parallel Interface

The parallel interface uses an Intel 8255A Programmable Peripheral Interface (PPI) to provide the M20 with one Centronics-like port for connecting to a printer.

The functional configuration of the 8255A can be programmed to operate in any of the following modes:

Mode 0	Basic Input/Output
Mode 1	Strobed Input/Output
Mode 2	Bi-directional Bus

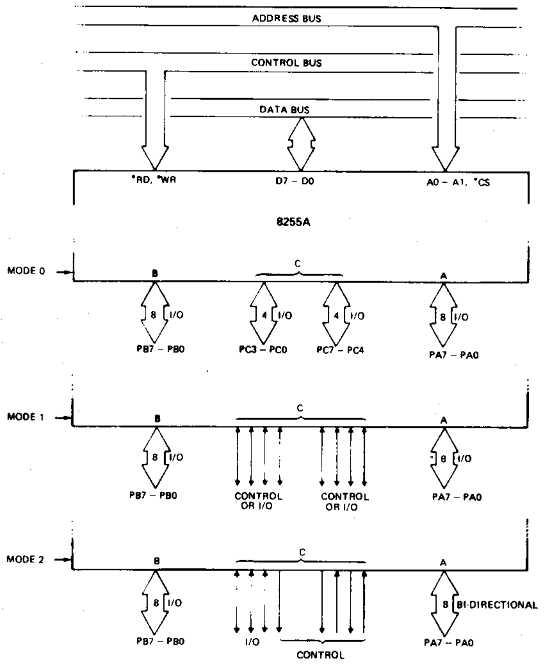


Fig. 2-26 8255A Basic Modes

Programming of the 8255A is performed by the M20 system software which issues a set of control words, mode, bit set, bit reset etc., to initialize the 8255A to support M20 requirements.

For the M20, Port-A operates in Mode 1 strobed output configuration; Port-B operates in Mode 1 strobed input configuration; Port-C is used by Port-A and Port-B to generate or accept hand-shaking signals.

Programmable Peripheral Interface (8255A)

The 8255A has 24 input/output pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation, Mode 0, Mode 1 and Mode 2. In Mode 0, each group of 12 input/output pins may be programmed in sets of 4 to be input or output. In Mode 1, each group of 12 input/output pins may be programmed to have 8 lines of input or output (see Figure 2-26). Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. Mode 2 is a bidirectional bus mode which uses 8 lines for a bidirectional bus and 5 lines (borrowing one from the other group) for handshaking.

Figure 2-27 shows a simplified block diagram of the 8255A. The following descriptions refer to the signals and elements shown in this diagram.

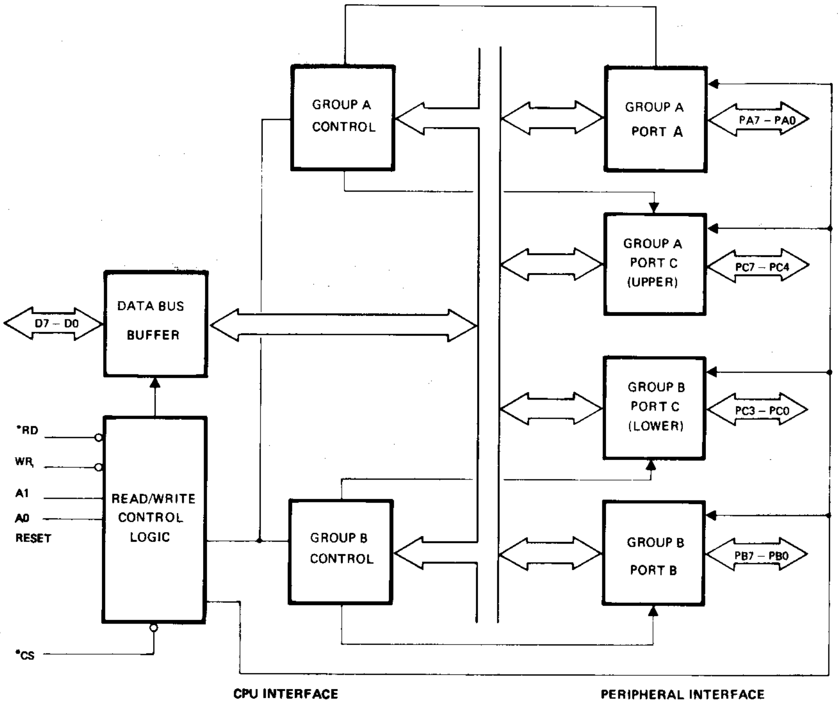


Fig. 2-27 8255A Block Diagram

Pin Functions

The pin functions of the 8255A are shown in figure 2-28.

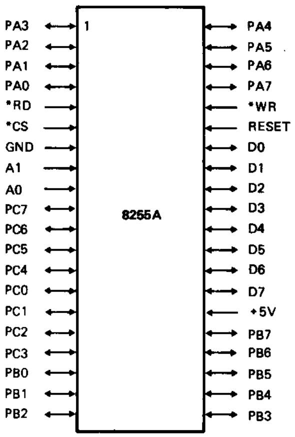


Fig. 2-28 8255A Pin Functions

Data Bus Buffer - The 3-state data bus buffer interfaces the 8255A with the M20 system data bus. Data is transmitted or received on D0-D7 upon execution of input or output instructions. Control words and status information is also transferred through the data bus buffer.

Read/Write & Control Logic - The read/write logic and contol manages all internal and external transfers of both data and control or status words. It accepts inputs from the CPU address and control busses in turn, issues commands to both control groups.

***RD Read:** A low on this input enables the 8255A to send status or data to the CPU.

***WR Write:** A low on this input enables the CPU to write data into the 8255A

***CS Chip Select:** A low on this input enables communication between the 8255A and the CPU.

A0, A1 Address: These two address inputs are used with the RD and WR inputs to control selection of one of the three ports or the control word registers.

The following truth tables summarize the logic functions of the above inputs:

A1	A0	*RD	*WR	*CS	Input Operation
0	0	0	1	0	Port A ==> Data Bus
0	1	0	1	0	Port B ==> Data Bus
1	0	0	1	0	Port C ==> Data Bus

A1	A0	*RD	*WR	*CS	Output Operation
0	0	1	0	0	Data Bus ==> Port A
0	1	1	0	0	Data Bus ==> Port B
1	0	1	0	0	Data Bus ==> Port C
1	1	1	0	0	Data Bus ==> Control

A1	A0	*RD	*WR	*CS	Disable Function
X	X	X	X	1	Data Bus ==> 3-state
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Data Bus ==> 3-state

RESET: A high on this input clears the control register and all three ports are set to the input mode.

Group A & Group B Controls - Each of the control blocks accepts commands from the read/write logic, receives control words from internal data bus and issues commands to its ports.

Control Group A	-	Port A and Port C upper (C7-C4)
Control Group B	-	Port B and Port C lower (C3-C0)

No read operation of the control word register is allowed.

Ports A,B & C - Port A has one 8 bit data output latch/buffer and one 8 bit data input latch. Port B has one 8 bit data input/output latch buffer, and one 8 bit data input buffer. Port C has one 8 bit data output latch/buffer and one 8 bit data input buffer (no latch for input). This port can be used for control signal outputs and status signal inputs with ports A and B.

Interface Functions

Figure 2-29 is a logic diagram of the parallel interface.

The following interface signals are used:

Data Strobe - Used to transfer character data from the interface to the printer.

Data Lines - Carry the character data from the interface to the printer.

Acknowledge - A low level indicates that the current character has been accepted by the printer and the printer is now available for a new character.

Busy - A high level indicates that the printer is busy and cannot accept a new character.

Paper Empty - A high level indicates that the printer is out of paper.

Select - A high level indicates that the interface has been selected and is available to transfer data.

Fault - A low level indicates that a printer fault condition exists.

Demand - A low indicates that the printer cannot receive data (used in the Data Products like interface).

Output Buffer Full - A low indicates that Port A0-A7 (DATA1-DATA8) holds data for transfer to the printer.

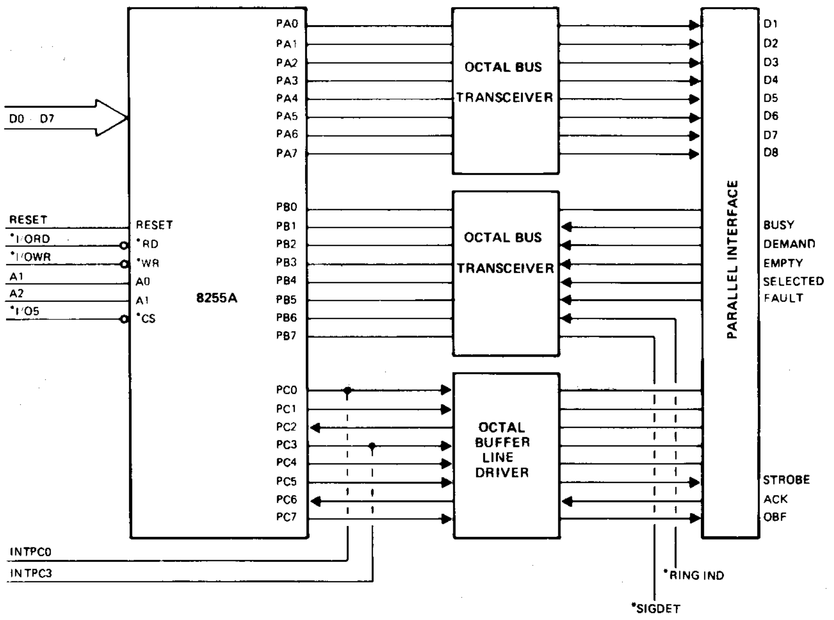


Fig. 2-29 Parallel Interface

2.1.1.9 Video Interface

The M20 uses the bit-map technique to generate both graphics and text. The picture element for display is a single dot, and the dots making up the graphic and textual display are stored in the M20 RAM.

For the monochrome display, the bit-map resolution is 512 x 256 dots. The memory requirement is therefore 512 x 256 bits (16 KB) of RAM for storage and refresh.

All textual characters are generated by the system software. In the textual mode, the display format can be either 16 rows of 64 characters, with a gross character matrix of 8 x 16, or 25 rows of 80 characters, with a gross character matrix of 6 x 10. The default format is 16 rows of 64 characters.

Figure 2-30 shows a logic diagram of the video interface for the monochrome display together with associated logic.

The dot oscillator operates at 12 MHz. The dot counter divides the dot rate by 16 to provide the character clock for the CRT Controller (CRTC). The CRTC generates memory addresses (MA0-MA11) and raster addresses (RA0-RA3) that together form the RAM address. These are multiplexed onto the 8 address inputs of the RAM.

The 16-bit output of the RAM is directed to the shift registers, which serialize the data to develop the video signal. The video signal together with with horizontal and vertical sync signals form the video interface output.

A full scan line of 512 dots is made up of 32 16-bit data words. These words are stored in sequential addresses in RAM and are addressed by MA0-MA4. The 16 rasters making up one 'line' are addressed by RA0-RA3 and the 16 rows making up one screen are addressed by MA5-MA9.

Figure 2-31 shows the video page format.

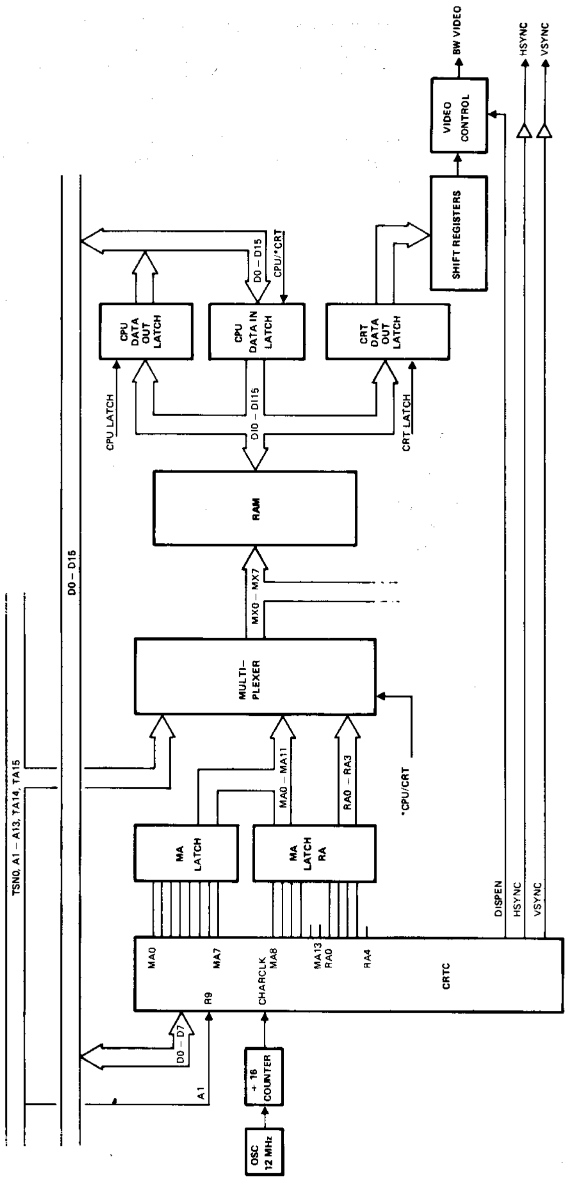


Fig. 2-30 Video Interface Logic Diagram

CRT Controller (MC6845)

The MC6845 generates the memory addresses, raster addresses, video timing and display enable signals.

Programming of the MC6845 internal register file R0-R17, to establish screen format and CRTC functions, is performed by the M20 system software through the data bus D0-D7 and the control inputs R/*W, *CS, RS and E.

The following lists the MC6845 internal register file functions and the programmed values.

REG No.	REGISTER NAME	PROGRAMMED VALUE
R0	Horizontal Total	39 Character
R1	Horizontal Displayed	32 Character
R2	Horizontal Sync Position	33 Character
R3	Horizontal Sync Width	6 Character
R4	Vertical Total	17 Character Row
R5	Vertical Scan Line Adjust	- Scan Line
R6	Vertical Displayed	16 Character Row
R7	Vertical Sync Position	16 Character Row
R8	Interlace Mode	-
R9	Max Scan Line Address	16 Scan Line
R10	Cursor Start	-
R11	Cursor End	-
R12	Start Address (H)	0
R13	Start Address (L)	0
R14	Cursor (H)	-
R15	Cursor (L)	-
R16	Light Pen (H)	-
R17	Light Pen (L)	-

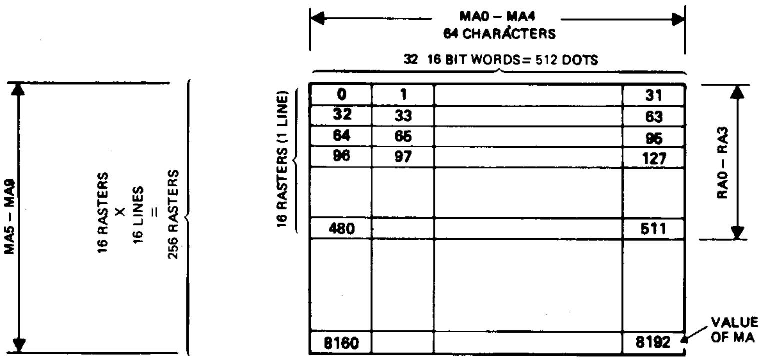


Fig. 2-31 Video Page Format

Pin Functions

Figure 2-32 shows the pin functions of the MC6845.

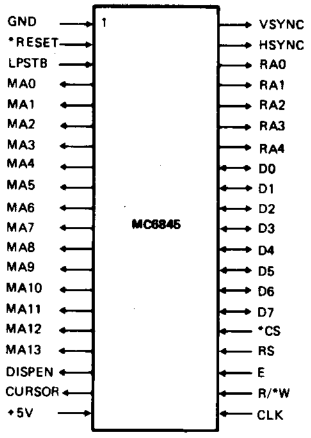


Fig. 2-32 MC6845 CRTC Pin Functions

CPU Interface Pins - The CRTC interfaces to the CPU bus on the bi-directional data bus (D0-D7) using *CS, RS, E and R/*W for control signals.

D0-D7 Data Bus: The 3-state data bus buffer interfaces the MC6845 with the M20 system data bus. Data is transmitted or received on D0-D7 upon execution of input or output instructions.

E Enable: A high to low transition of this input enables the data bus input/output buffers and clocks data to and from the CRTC.

***CS Chip Select:** A low on this input selects the CRTC to read or write the internal register file. This signal should only be active when there is a valid stable address being decoded from the CPU.

RS Register Select: This input selects the address register (RS=0) or one data register (RS=1) of the internal register file.

R/*W Read/Write: This input determines whether the internal register file is written to or read from. A write is active low.

CRT Control Pins - The CRTC provides horizontal sync, vertical sync, and Display Enable signals.

VSYNC Vertical Sync: This output determines the vertical position of the displayed text.

HSYNC Horizontal Sync: This output determines the horizontal position of the diplayed text.

DISPEN Display Enable: This output indicates that the CRTC is providing addressing in the active display area.

Refresh Memory Addressing Pins

MA0-MA13 Refresh Memory Addresses: These 14 outputs are used to refresh the CRT screen with pages of data within the 16 KB block of refresh memory.

RA0-RA4 Raster Addresses: These 5 outputs are used to address the 16 rasters making up one "line" (a line of characters).

Other Pins

CLK Clock: The clock input used to synchronize all CRT control signals.

CURSOR: This output indicates cursor display to external video processing logic.

LPSTR Light Pen Strobe: This input latches the current refresh address in the resgister file.

***RES**: This input is used to reset the CRTC. A low on this input forces the CRTC into the following status:

All the counters in the CRTC are cleared and the device stops the display operation.

All outputs go to a low level.

Control Registers in the CRTC remain unchanged.

The *RES input has the capability of reset when LPSTB is low. The CRTC starts the display operation immediately after the release of *RES.

***RESET**: A low on this input resets the CRTC.

2.1.1.10 Diskette Drive Interface

The Diskette Drive interface provides the logic and control circuitry necessary to control and record data onto, or read data from diskettes and initially formats new diskettes.

The interface circuitry controls the read/write head position and, acting in response to commands issued by the CPU under operating system control, selects the requested sector and track for each read or write operation. Phase-locked loop circuits provide accurate timing for data recovery.

The ECMA 70 STANDARD is used for recording. All recording is double density modified frequency modulation (MFM), with 256 bytes of data recorded on each sector (for serial data a byte is defined as 8 consecutive bit cells). The interface can control up 2 diskette drives.

Figure 2-33 shows a logic diagram of the Diskette Drive interface. The major elements of the Diskette Drive interface include:

Floppy Disk controller/formatter
Floppy Support Logic
VCO clock generator
Two data latches
Two monostable multivibrators

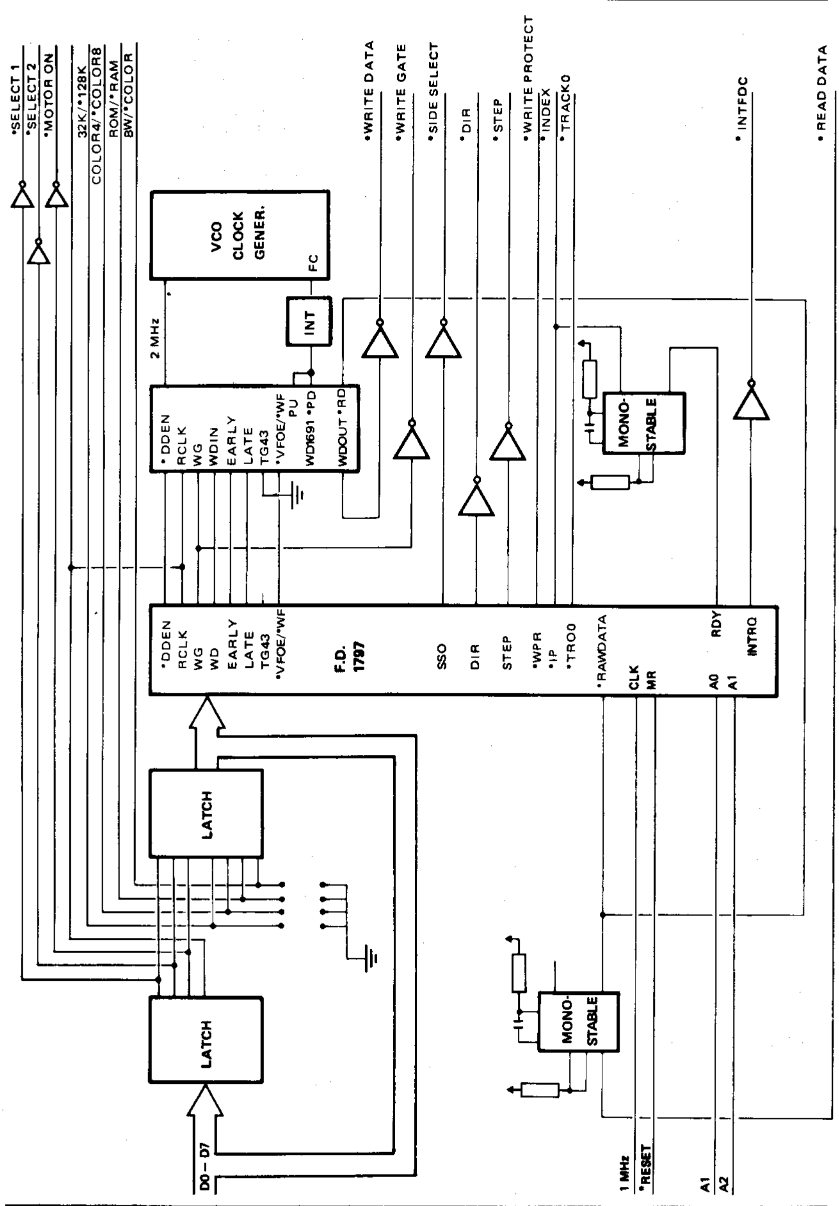


Fig. 2-33 Diskette Drive Interface Logic Diagram

The Floppy Disk Controller/Formatter (FD1797), in either the write or read mode, provides track seek, with verification, and sector search. All the operating system must do is issue the SEEK command, and identify the sector and track. The FD1797 provides direction (DIRC) and step (STEP) signals to the diskette drive unit. The FD1797 also provides write formatting, status and error checking for the system, disk-side selection, and double density or single density selection.

The two data latches are used to save incoming commands to enable both read and write operations to be performed on the same data bus.

The WD1691 together with the VCO provides a phase-lock loop data separator in order to accurately recover data from the composite (data and clock) signal (READ DATA) transmitted from the diskette drive. This is achieved by ensuring that the Read Clock (RCLK), derived from the 2 MHz output of the VCO, is in phase with the READ DATA (RDD) pulses. The WD1691 monitors the READ DATA (RDD) pulses and increases or decreases the frequency of the VCO according to whether they occur early or late relative to the RCLK window. The phase comparison circuit in the WD1691 produces a Pump-up (PU) or Pump-down (PD) signal, which is integrated to form the frequency control voltage (FC) for the VCO. The nominal 2 MHz output of the VCO, fed back to the VCO input of the WD1691, completes the loop.

Of the two one-shot multivibrators one provides the READY signal to the FD1797 to indicate that a diskette is mounted in the drive, and that the drive is turning. This one-shot is triggered by the INDEX pulse from the diskette drive unit, which occurs every 200ms. If the period between the pulses is approximately 250ms, the one-shot will provide a continuous or dc READY signal to the FD1797.

The second one-shot multivibrator is used to condition the READ DATA from the diskette drive. The one-shot provides a 300ns nominal pulse to the RAW DATA input of the FD1797 and the RDD input of the WD1691.

READ/WRITE Operations

To READ a file, the sequence of operations is as follows:

The operating system issues a SEEK Command.

The head is positioned to the correct track.

The READ command to 1797 is issued by CPU.

Data is read from the correct sector.

The OS checks the CRC (Cyclic Redundancy Check character - used in a modified cyclic code for error detection and correction)

If correct, data is transferred into M20 System RAM Memory.

SEEK command issued to get next sector.

To WRITE a file, the sequence is essentially the same except for the third step. In this step, data is written onto the diskette and then its CRC is checked. If not correct, the write attempt is repeated a set number of times or until no error is observed. If the error persists, the error is flagged to the operator.

Floppy Disk Controller (FD1797)

Figure 2-34 shows a simplified block diagram of the FD1797.

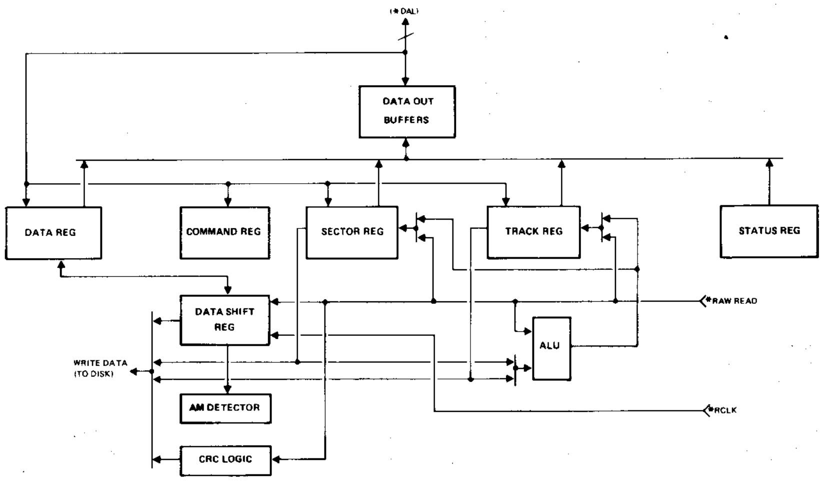


Fig. 2-34 FD1797 Block Diagram

Pin Functions

The pins of FD1797, shown in Figure 2-35, can be grouped according to fuction.

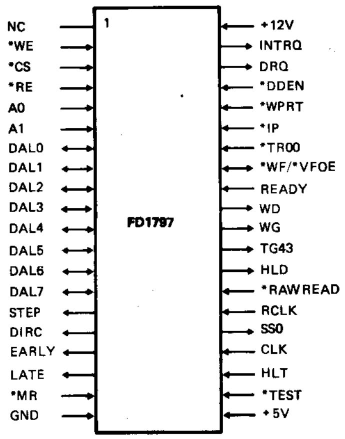


Fig. 2-35 FD1797 Pin Functions

CPU Interface Pins - The FD1797 is designed to operate on a multiplexed bus with other bus-oriented devices, but the floppy disk controller always has the highest interrupt priority. Its interface with CPU consists of an 8 bit bi-directional bus, plus additional lines for chip select, interrupt, clock register select, write enable, read enable and master reset.

DAL0-DAL7 Data Access Lines: These lines are an 8 bit bi-directional data bus, used for transferring data, control and status words. The bus is receiver enabled by the *RE line; write enabled by the *WE line.

***CS Chip Select:** A low on this input enables CPU communication with the FD1797.

***WE Write Enable:** A low on this input gates data on the DAL into the selected register when *CS is low.

***RE Read Enable:** A low on this input controls the placement of data from the selected register on the DAL lines when *CS is low.

A0, A1 Register Select: These lines select internal registers in the FD1797 to receive or transfer data on the DAL lines under *RE and *WE control.

A1	A0	*RE	*WE
0	0	Status Register	Command Register
0	1	Track Register	Track Register
1	0	Sector Register	Sector Register
1	1	Data Register	Data Register

The **DATA Register** is used as a holding register during the READ and WRITE operations.

The **TRACK Register** holds the track number of the current head position. It is incremented or decremented, according to received STEP instructions. Position is always verified by comparison with the data recorded in each sector.

The **SECTOR Register** holds the address of the desired sector position. The contents of this register are compared with the sector address recorded in each sector during Read or Write operations.

The **COMMAND Register** stores the command presently being executed. It is loaded from the DAL.

The commands accepted by the FD1797 are:

RESTORE
SEEK
STEP
STEP IN
STEP OUT
READ SECTOR
WRITE SECTOR
READ ADDRESS
READ TRACK
WRITE TRACK
FORCE INTERRUPT

The **STATUS Register** holds the device status information. The meaning of the stored bits depends upon the type of command previously executed. The contents of this register can be read onto the DAL lines.

CLK Clock: A 1 MHz square wave input for internal timing reference.

INTRQ Interrupt Request: This input is set on completion of any command. It is reset after the status register is read or the command register is written to.

***MR Master Reset:** This resets the FD1797 and performs a RESTORE operation.

DRQ Data Request: This line indicates that the DR contains assembled data in Read operations, or the DR is empty in write operations.

Diskette Drive Interface Pins - As the block diagram shows, the interface to the diskette drive unit consists of serial-data transfer lines, plus additional control, status and timing lines.

***RAW READ Raw Read:** During a read operation, the FD1797 requires as an input the Read Data signal and the Read Clock. The Read Data signal comes from the drive, and consists of a negative pulse for each recorded flux transition. These are shifted into and stored in the Data Register during a read operation. The Read Data signal is also used by the WD1691 (RDD) for phase comparison.

RCLK Read Clock: The read clock signal is derived from the data stream by the phase-lock loop (WD1691). The phasing (relative to the Read Data signal) is important.

WG Write Gate: For Write operations, the Write Gate output is activated, permitting current to flow to the write head in the drive unit.

WPRT Write Protect: Writing is prohibited if the Write Protect input is low, indicating that the disk is write protected. The Write Gate and the Write Data signals both come from the FD1797, but are transferred to the disk drive via the WD1691.

***WD Write Data:** The Write Data signal consists of a 250 ns pulse for each flux transition plus unique address marks and clock pulses. The composite Write Data signal is read from the Data Register during Write operations.

***IP Index:** This input tells the FD1797 when the index hole is encountered on the disk, and thus identifies the starting point.

DIRC Direction: Direction output refers to whether the head is stepping in or out. The output is high when the head is stepping out, and low when stepping in.

STEP Step: The step output contains a pulse for each step.

HLD Head Load: This output controls the movement of the read/write head. This signal is generated whenever the drive is selected, so that the head will be loaded.

HLT Head Loading Time: When logic high, the input is assumed to be engaged.

***TR00 Track 00:** This input informs the FD1797 that the Read/Write head is positioned over Track 00.

SSO Side Select Output: The logic level of this output is controlled by the S flag. When S = 1, SSO is set to a logic 1; when S = 0, SSO is set to a logic 0.

EARLY Early: Indicates that the write data pulse occuring while Early is high should be shifted early for write precompensation.

LATE Late: Indicates that the write data pulse occuring while Late is high should be shifted late for write precompensation.

TG43 Track Greater Than 43: Informs the drive that the Read/Write head is positioned between tracks 44-76.

WD Write Data: Contains the address marks as well as data and clock.

READY Ready: Indicates disk readiness and is sampled for a logic high before Read/Write commands are performed.

***WF/*VFOE Write Fault/VFO Enable:** This bi-directional line is used to signify writing faults at the drive, and to enable the external PLO data separator.

***DDEN Double Density:** This line selects either single or double density operation.

Floppy Disk Support Logic (WD1691)

The WD1691 is designed to minimize the external logic required to interface the FD1797 to a disk drive. With the use of an external VCO the WD1691 generates the RCLK signal for the FD1797, provides pump pulses to control the VCO frequency, and VFOE/WF de-multiplexing and write precompensation. Figure 2-36 shows a simplified block diagram of the WD1691.

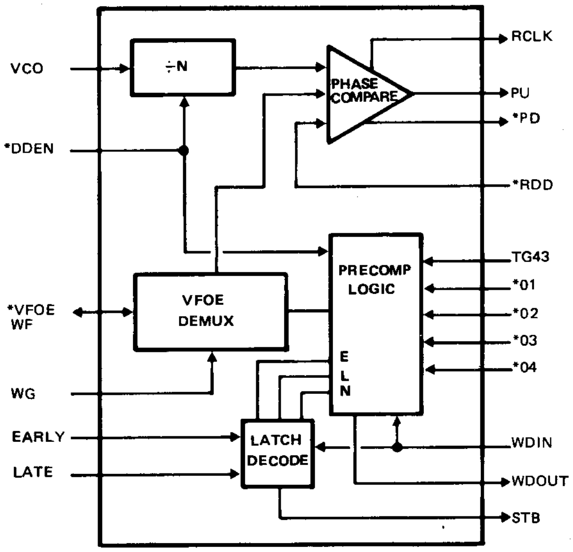


Fig. 2-36 WD1691 Block Diagram

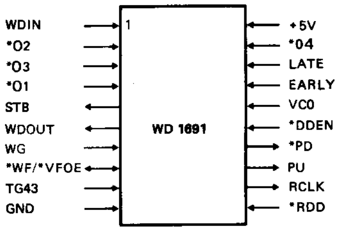


Fig. 2-37 WD1691 Pin Functions

Pin Functions

The pins of the WD1691, shown in Figure 2-37, have the following functions.

WDIN Write Data Input: Ties directly to the FD1797 WD pin.

***01-*04 Phase 01-04:** Four phase inputs to generate write compensation delay. NOT USED BY M20.

STB Strobe: Strobe output. NOT USED BY M20.

WDOUT Write Data Output: Serial, precompensated write data to be sent to the disk drives WD line.

WG Write Gate: Ties directly to the FD1797 WG pin.

***VFOE/*WF VFO Enable/Write Fault:** Ties directly to the FD1797 *VFOE/*WF pin.

TG43 Track 43: Ties directly to the FD1797 TG43 pin, if write precompensation is required on Tracks 44-76.

***RDD Read Data:** Composite clock and data stream input from drive.

RCLK Read Clock: RCLK signal generated by the WD1691, to be tied to the FD1797 RCLK input.

PU Pump Up: 3-state output that will be forced high when the WD1691 requires an increase in VCO frequency.

***PD Pump Down:** 3-state output that will be forced low when the WD1691 requires an decrease in VCO frequency.

***DDEN Double Density Enable:** Double density select input. When inactive (high), the VCO frequency is internally divided by two.

VCO Voltage Controlled Oscillator: Nominal 2 MHz master clock input.

EARLY, LATE: Early and Late signals from the FD1797, used to determine write precompensation.

Data Recording Format

Figure 2-38 shown the disk track format.

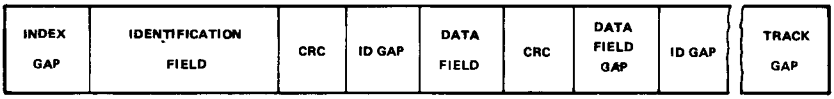


Fig. 2-38 Disk Track Format

Index Gap - This gap separates the index pulse from the Identification (ID) field. It consists of 32 bytes of 4EH.

Identification Field - The identification field is made up of the ID Address Mark, Track number, side number, sector number, and sector length. It consists of 20 bytes.

CRC - Two cyclic redundancy check (CRC) bytes are generated during a read/write operation of the ID field. The bit pattern of the CRC bytes is determined by the bit pattern of the ID Field.

ID Gap - This gap is used to separate the ID field from the data field. It consists of 22 bytes of 4EH.

Data Field - The data field contains the record data.

CRC - Two cyclic redundancy check (CRC) bytes are generated during a read/write operation of the data field. The bit pattern of the CRC bytes is determined by the bit pattern of the data field.

Data Field Gap - This gap separates one sector from another. This gap contains 48 bytes of 4EH.

Track Gap - This gap occurs after the last record of the last sector of a track and separates that record from the index pulse.

2.1.1.11 Timer

The Timer uses an Intel 8253 Programmable Interval Timer to provide the transmitter clocks for the two 8251A PCI of the Dual Communication Serial Interface and a real-time clock.

The 8253 is a general purpose multi-timing element that is treated as an array of Input/Output ports by the system software. Instead of timing loops being set in the system software, the 8253 can be configured to match program requirements.

The 8253 contains three counters identical in operation but fully independent of each other. The 8253 can be programmed so that each counter can work in any of the following modes:

Mode 0	Interrupt on terminal Count
Mode 1	Programmable one-shot
Mode 2	Rate generator
Mode 3	Square wave generator
Mode 4	Software triggered strobe
Mode 5	Hardware triggered strobe

Programming of the 8253 is performed by the M20 system software which issues a set of control words, mode and count value, to initialize the 8253 to support M20 requirements.

For the M20, Counter 0 operates in Mode 3 (square wave generator mode) to generate TxC for the RS-232-C Serial Interface; Counter 1 operates in Mode 3 to generate the TxC for the Keyboard Interface; Counter 2 operates in Mode 2 (rate generator mode) for program interrupt. The following details these modes of operation.

Mode 2 Rate Generator. Divide by N counter. The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses, the present period will not be affected, but the subsequent period will reflect the new value. The GATE input when low, will force the output high. When the Gate input goes high, the counter will start from the initial count. Thus, the GATE input can be used to synchronize the counter. When this mode is set, the output will remain high until after the count register is loaded. The output then can also be synchronized by software.

Mode 3 Square Wave Generator. Similar to MODE 2 except that the output will remain high until one-half the count has been completed (for even numbers) and go low for the other half of the count. If the count is odd, the output will be high for (N + 1)/2 counts and low for (N - 1)/2 counts. If the count register is reloaded with a new value during counting, this new value will be reflected immediately after the output transition of the current count.

Programmable Interval Timer (8253)

Figure 2-39 shows a simplified block diagram of the 8253. The following descriptions refer to the signals and elements shown in this diagram. The pin functions for the 8253 are shown in figure 2-40.

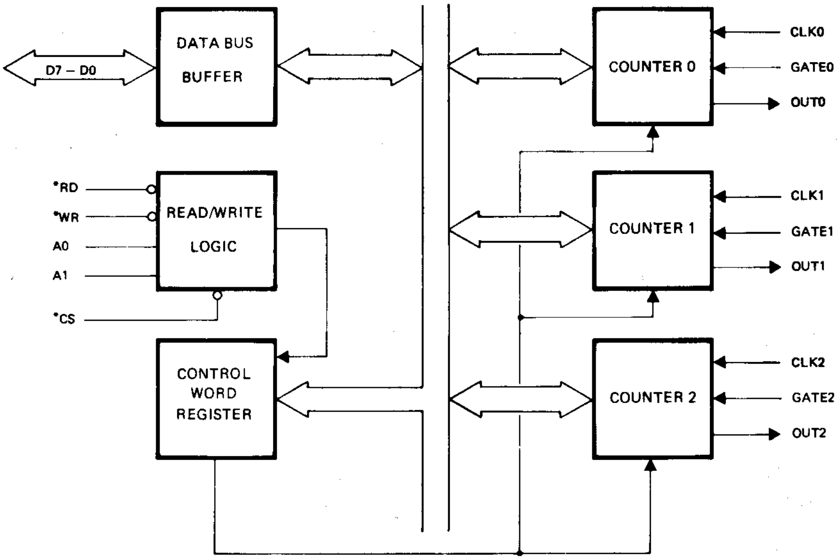


Fig. 2-39 8253 Block Diagram

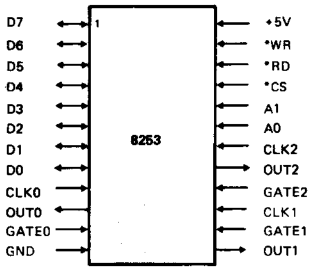


Fig. 2-40 8253 Pin Functions

Data Bus Buffer - The 3-state data bus buffer interfaces the 8253 with the M20 system data bus. Data is transmitted or received on D0-D7 upon execution of input or output instructions. The data bus has three functions:

Programming the modes of the 8253
Loading the count registers
Reading the count values

Read/Write Logic - The Read/Write logic accepts inputs from the system bus and generates control signals overall device operation. It is enabled by the Chip Select signal so that no operation can occur to change the function unless called for by the system logic.

***RD Read:** A low on this input informs the 8253 that the CPU is inputting data in the form of counters value.

***WR Write:** A low on this input informs the 8253 that the CPU is either outputting mode information or loading counters.

A0, A1 Address: These inputs are connected to the address bus and are used to select one of the three counters to be operated on and to address the control word register for made selection.

***CS Chip Select:** A low on this input enables the 8253. No reading or writing will occur unless the device is selected.

The following truth table summarizes the logic functions of the above inputs:

*CS	*RD	*WR	A1	A0	FUNCTION
0	1	0	0	1	Load counter 0
0	1	0	0	1	Load counter 1
0	1	0	1	0	Load counter 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter 0
0	0	1	0	1	Read Counter 1
0	0	1	1	0	Read Counter 2
0	0	1	1	1	No-op 3 state
1	X	X	X	X	Disable 3 state
0	1	1	X	X	No-op 3 state

Control Word Register - This register is selected when A0,A1 are 11. It then accepts information from data bus buffer and stores it in a register. The information stored controls the operation mode of each counter, selection binary or Binary Coded Decimal (BCD) counting, and the loading of each count register. The Control Word Register can only be written into; no read operation of its contents is available.

Counters - The Counters are identical in operation. Each consists of a single 16-bit, pre-settable, down counter that can operate in either binary or BCD. Input, gate and output are configured by the selection of modes stored in the Control Word Register.

The three counters are fully independent of each other. The reading of the contents of each counter is done with a READ operation by the system software. Reading can be done without blocking or interrupting the count.

2.1.2 KEYBOARD

The keyboard has keys grouped into two sections, alphanumeric keys providing a standard typewriter layout in various national versions and a numeric keypad to allow convenient input of numeric data strings.

The keyboard incorporates a Power-on indicator lamp and a buzzer. The buzzer provides audible feedback for certain M20 operations.

The keyboard supports the following national key layouts:

Italian
German
French
British
USA ASCII
Spanish
Portuguese
Swedish - Finnish
Danish
Katakana
Yugoslavian
Norwegian
Greek
Swiss - French
Swiss - German

The layout of the 72 key version of the keyboard is shown in Figure 2-41. The layout of the Katakana version differs in that it contains 75 keys.

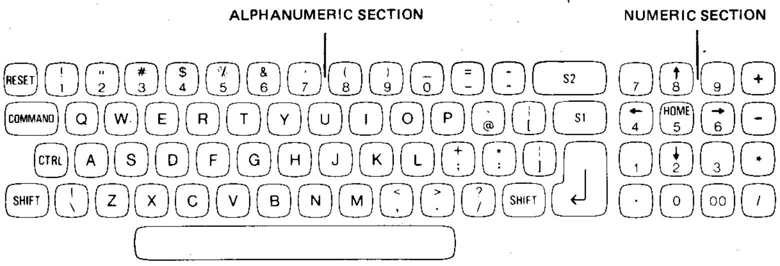


Fig. 2-41 Keyboard Layout

The two **SHIFT** keys are used for the selection of upper or lower case for alpha character keys and for the selection of the upper symbol of the two symbol keys. The **COMMAND** key when used together with the ?/ key provides a shift lock for letters A to Z. The **CTRL** key when used together with certain character keys provides a variety of BASIC operations and when used together with the **RESET** key provides a logical reset.

The main component of the keyboard is an 8048 microcomputer, a totally self sufficient 8-bit parallel computer that contains a 1K x 8 bit program memory, a 64 x 8-bit RAM data memory, 27 input/output lines and an 8-bit timer/counter in addition to an on board oscillator and clock circuits. Figure 2-42 shows a logic diagram of the keyboard.

The COMMAND, CTRL and two SHIFT keys connect directly to Port P20-P23 while the remainder of the keys employ an 9-row x 8-column matrix to encode the key information onto the Bus DB0-DB7. The BDC output at Port P27-P30 is decoded by the BDC to Decimal Decoder and used to sequentially scan for row and column information, this information being read onto the Bus from the outputs of the Comparators.

Once a key is pressed and its position encoded and validated, its key-code complete with relevant control information, SHIFT, CTRL or COMMAND, is transmitted from Port P24 in a serial asynchronous half duplex mode. The key-code transmitted consists of one start bit, eight data bits and two stop bits. Parity control is not employed.

The code to identify the national keyboard employed is set-up by means of the selectable jumpers on Ports P14-P17 or invoked by a PCOS command.

The keyboard buzzer is driven from Port P27 and emits a 3100Hz tone to provide the user with audible feedback upon certain M20 operations and upon erroneous multiple key operation. The duration of the tone is used to distinguish between the two conditions: approximately 10ms for feedback and approximately 50ms for errors.

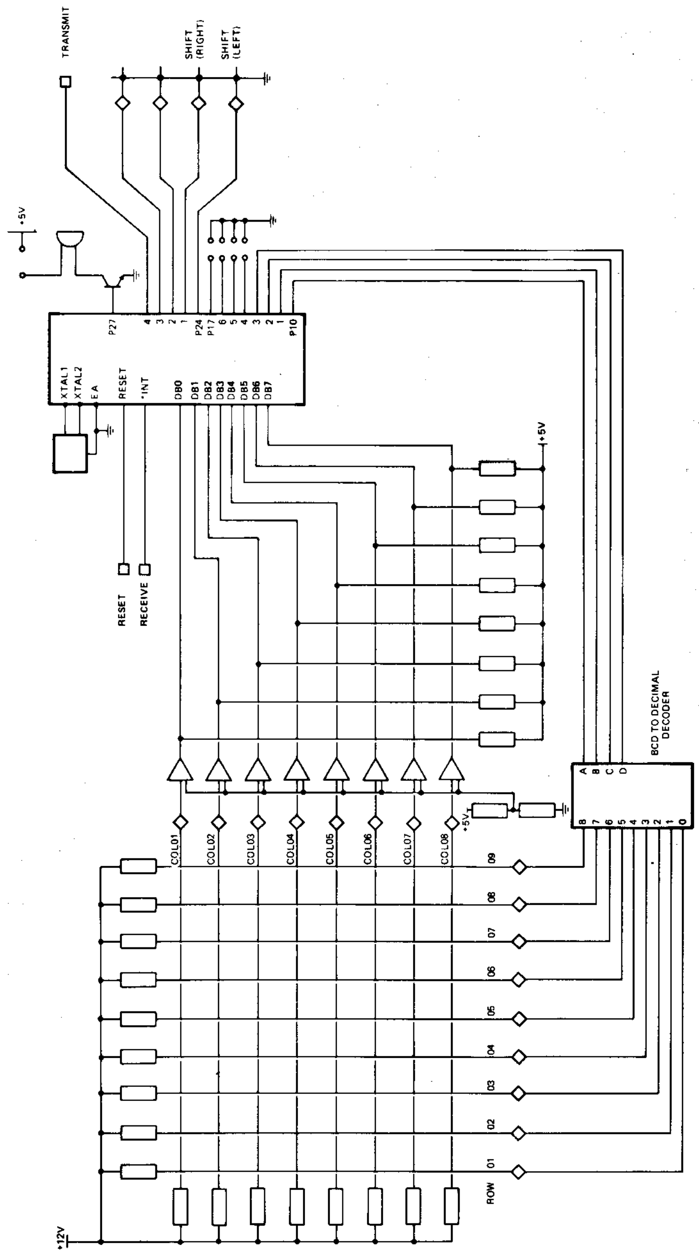


Fig. 2-42 Keyboard Logic Diagram

2.1.3 DISKETTE DRIVE

The Diskette Drive is a random access magnetic storage unit that uses removable 5.25 diskettes for data storage.

Inserting a diskette into the drive and closing the cover clamps the disk to a self centering hub that is driven by a servo controlled dc motor. Two read/write heads, one for each surface of the disk, are mounted on a carrier that is moved over the surfaces of the disk by means of a spiral cam attached to the shaft of a stepper motor.

The LED on the face of the drive lights to indicate that the drive has been selected.

Opening the drive cover releases the disk from the hub and the disk is ejected. The drive cover must not be opened while the drive is selected.

The drive incorporates:

A track zero sensor, which consists of a microswitch that detects when the head carriage is at track 00.

A write protect sensor, which consists of a microswitch that detects whether the write protect notch in the disk cover is present (a disk is write protected when this notch is covered).

An index hole sensor, which consists of a photodiode and phototransistor that detects when the index hole in the disk, and thus the starting point, is encountered.

2.1.4 HARD DISK CONTROLLER

The Hard Disk Controller is used to interface the Olivetti 5.25 inch Winchester hard disk drive to the M20.

It comprises two Printed Circuit Boards; the Hard Disck Controller board, which is mounted at the bottom of the disk drive assembly, and the Hard Disk Transition board, which plugs into one of the system bus slots (J3 or J4) on the motherboard.

The Hard Disk Controller circuitry may be divided into the following functional elements:

Processor Functions
Serial Data Separation
Data Conversion and Checking
Serial Data Generation
Host Interface Functions

Figure 2-43 shows a block diagram of the Hard Disk Controller.

Processor Functions

All functions of the Hard Disk Controller are controlled by the 8X300 is a microprocessor that operates at a basic clock rate of 8MHz. The architecture of this microprocessor is different from most standard microprocessors in that no common data or address bus is provided to be shared by ROM or RAM (on the Hard Disk Controller board). Instructions are fetched from the Hard Disk Controller ROM via a dedicated instruction address and data bus.

The program storage of the Hard Disk Controller is limited to 1K words. Program data is input to the 8X300 on the Instruction Bus (ID0-ID15) as 16 bit words which are decoded to perform the desired operation.

Circuity associated with the 8X300 perform the following functions:

FAST IO SELECT CIRCUITRY to provide port access decoding on an instruction by instruction basis.

RESET CIRCUITRY to hold the 8X300 reset for approximately 40ms. after initial power on.

RAM MEMORY through which all data read from disk or written to the disk passes in order to provide the buffering required for asynchronous data transfer between the M20 and disk. RAM also serves as a scratchpad storage during program execution.

RAM ADDRESSING CIRCUITRY

RAM ACCESSING CIRCUITRY

ROM MEMORY for program storage

MAC CONTROL PORT which consists of a 6 bit control port that controls the various functional sections of the Hard Disk Controller.

Serial Data Separation

In order to provide maximum data recording density and therefore maximum storage efficiency, data is recorded on the disk using a Modified Frequency Modulation (MFM) technique. This technique requires clock bits to be recorded when two successive data bits are missing in the serial data stream. This reduces the total number of bits required to record a given amount of information on the disk. This results in an effective doubling of the amount of data capacity, hence the term "double density". The fact that clock bits are not recorded with every data bit cell requires circuitry that can remain in sync with data during the absence of clock bits. Synchronous decoding of MFM data streams requires the decoder circuitry to synthesize clock bit timing when clocks are missing and synchronize to clock bits when they are present. This is accomplished by using a phase locked oscillator employing an error amplifier/filter to sync onto and hold a specific phase relationship to the data and clock bits and to shift the resultant serial data into registers for parallelization into bytes.

Data Conversion and Checking

MFM data which has been separated to form No Return to Zero (NRZ) data and clocks is processed through specialized circuitry to prepare it for parallel processing by the 8X300. Circuitry associated with this processing perform the following functions:

Address Mark (AM) detection
Serial to Parallel Conversion
CRC Checking Circuit

Serial Data Generation

The Hard Disk Controller records data on the disk in the MFM format. In order to produce the proper data format, the Hard Disk Controller uses a several specialized devices to process the parallel data supplied by the M20 into a serial MFM data stream. The data supplied by the M20 is temporarily stored in the buffer RAM until the correct sector is located for the data to be written.

The process of writing is essentially the opposite of reading except that the data separator circuitry is not required and the generation of the MFM data stream is produced by synchronous clocking techniques. Circuitry associated with serial data generation perform the following functions:

Parallel to Serial Conversion
CRC Generation
MFM and Precompensation

Host Interface

The Host interface circuitry is contained on the Hard Disk Controller board and the Hard Disk Transition board. The Hard Disk Transition board generates the signals Card Select (CS), Write Enable (WE) and Read Enable (RE). All data transfers between the M20 and the Hard Disk controller take place over the 8 bit bi-directional data bus (D0 - D7). The source or destination register is selected by three address lines (A0 - A2). Note that the signals output from the M20 are A1, A2 and A3 and are then renamed as A0, A1, and A2 by the Hard Disk Transition board. All accesses to the Hard Disk Controller are controlled by the Card Select (*CS), Read Enable (*RE), and Write Enable (*WE). Since the access time for any particular read or write operation will vary, the Hard Disk Controller provides a not ready signal (*WAIT). Accessing the Hard Disk Controller is essentially like accessing variable speed RAM. The M20 must provide a valid address in A0-2 along with *CS. Immediately or after a short set up time, the M20 may assert *RE or *WE. If access time on the Hard Disk Controller is greater than 100ns, then *WAIT will be asserted. The M20 must keep all address lines and strobes stable while *WAIT is asserted. When the Hard Disk Controller de-asserts *WAIT, the data has been accepted on a write or the data is on the data bus D0-D7 on a read.

2.1.5 HARD DISK UNIT

The Hard Disk Unit is a random access magnetic storage unit that uses three non-removable 5.25 in diameter disks for data storage.

Six read/write heads, one for each disk surface, are mounted on a carrier that is moved across the disk surfaces by a stepper motor. The disks are mounted on a spindle that is directly driven by a dc brushless motor. The disks, heads and carrier are mounted in a sealed die cast alluminium enclosure. Contamination protection of the disks, heads and carrier is provided by recirculating the air within the enclosure through a class 100 absolute filter.

The LED on the face of the Hard Disk Unit lights to indicate that the drive has been selected.

Characteristics

Unformatted capacity		
Track:	10.417	KB
Surface:	1.875	MB
Unit:	11.25	MB
Sectors/track:	33	
Data surfaces:	6	
Access times		
Track-track:	1.1	ms
Average:	66.0	ms
Maximum:	198.0	ms
Damping:	20.0	ms
Average latency:	8.33	ms
Transfer rate:	5	MB/s
Spindle speed:	3600	±1%
Linear density:	7820	BPI
Track density:	254	TPI
Cylinders:	180	
Tracks:	1080	
Read/write heads:	6	
Acceleration time:	14	s
Breaking time:	5	s

2.1.6 POWER SUPPLY UNIT

The power supply circuitry required to provide all dc voltages for the operation of the M20 is housed in a metal case mounted along the left-hand side of the basic module. This case also houses the system on/off switch, accessable at the rear of the basic module, and the terminal strips for the connection of the basic module ac power cable, the colour CRT display ac power cable and the cooling fan cabling.

Power Generation & Regulation

The power supply is a switching converter type the main characteristic of which is the method of obtaining voltage regulation on the primary winding of the output transformer. This is achieved by placing a saturable reactor, the impedance of which is varied according to the current in its control winding, in series with the primary winding of the output transformer.

The voltages on the secondary winding of the output transformer are full wave rectified and used to form the +5V, +12V and -12V power supplies. The +5V and +12V supplies are filtered and used directly as output whereas the -12V supply is filtered and passed via a series regulator before being used as output.

Regulation of the +5V and +12V supplies is obtained by comparing them with a reference voltage by means of a differential amplifier. The resultant difference voltage is amplified and used as the control source to vary the current in the control winding of the saturable reactor in the switching regulator.

A.C. Input Characteristics

VOLTAGE RANGE	TOLERANCE	FREQUENCY RANGE	TOLERANCE
100V to 120V or 200V to 240V	10%	50 Hz to 60 Hz	+5%

The input voltage range is jumper selectabe in the power unit.

D.C. Output Characteristics

VOLTAGE	TOLERANCE	CONTINUOUS CURRENT	RIPPLE
+ 5V	5%	3.3A min, 9.0A max	50mV p-p
+12V	3%	2.0A min, 5.6A max	100mV p-p
-12V	5%	-- 0.7A max	100mV p-p

2.2 CRT DISPLAY

The display houses a monochrome black & white or colour CRT and all associated circuitry. It has a circular base that allows the screen to be revolved and tilted to a convenient viewing position. The display can either sit on top of the basic module or on an adjacent horizontal surface. A brightness control, behind the top right edge of the display housing, is the only control accessible to the user. Other controls commonly associated with a CRT display are located within the display housing. Both types of display, monochrome and colour, have etched screens to reduce glare.

The monochrome display is connected to the basic module by a single cable that carries both signal and +12V power from the basic module. This cable is hard wired to the display and plugs into the video interface connector at the rear of the basic module. Within the display a +12 V regulator further stabilizes the +12 V supply in order to eliminate oscillations on the screen that would otherwise be caused by power fluctuations during start-up and stepping of the disk drives.

The colour display is connected to the basic module by two cables, one signal and one power. The power cable is hard wired to the basic module and plugs into the power socket at the rear of the display while the signal cable can be unpluged from both basic module and display.

Display Formats

In alphanumeric mode, two different page formats are available, a 1024 character format and a 2000 character format. The character set includes upper-case and lower-case and display attributes include reverse and hide.

The two formats have the following characteristics:

1024 Character Format

characters per line:	64
number of lines:	16
net character matrix:	5 x 7
gross character matrix:	8 x 16

2000 Character Format

characters per line:	80
number of lines:	25
net character matrix:	5 x 7
gross character matrix:	6 x 10

In graphics mode, 512 x 256 addressable dots (pixels) are available. The resolution of the monochrome and colour displays are the same.

2.3 PRINTERS

The Olivetti printers that may be used with the M20 include the following:

- Printer PR 2400
- Printer PR 1450
- Printer PR 1471
- Printer PR 1481
- Printer PR 430
- Printer PR 2300

Printer - PR 2400

The PR 2400 is a non-impact dot matrix thermal printer. The printing mechanism consists of 80 thermal electrodes, one per character, that prints elementary rows with a back-and-forth raster motion so that each electrode builds up a dot matrix character. The matrix size is 7-rows x 5-columns for alphanumeric characters and 10-rows x 7-columns for plotting. When used in the plotting mode, all 560 dots can be printed on each elementary row and any number of rows can be printed along the length of the paper supply.

Printing Speed:	240 lines/min
Character Pitch:	10 chars/in
Maximun Length of Print Line:	80 chars
Paper Feed:	friction feed
Paper Feed-in:	rear
Types of Document Handled:	paper rolls

The PR 2400 supports the following character sets:

- USA ASCII
- French
- Norwegian - Danish
- Spanish
- British
- Portuguese
- German
- Swedish - Finnish
- Katakana

Printer - PR 1450

The PR 1450 is an impact dot matrix printer using normal paper. The dot matrix is 9-rows x 7-columns (4+3) and printing is monodirectional with rapid carriage return from any print position.

A boldface feature which doubles the character width can be selected. In this case the line character capacity is halved.

The printer can operate in plotter mode (Bit Image Mode) which allows printing of graphs, tables, histograms, images, etc., by associating the print needle configuration with the received data bits.

Printing Speed:	100 chars/s
Character Pitch:	10 chars/in 16.6 chars/in
Maximum Length of Print Line:	80 chars (10 chars/in) 132 chars (16.6 chars/in)
Paper Feed:	friction-feed, pin-feed (optional adjustable sprocket feed)
Paper Feed-in:	rear (front optional)
Types of Document Handled:	single document, paper rolls, cards, fan-fold forms

The PR 1450 supports the following character sets:

USA ASCII
International
German
Portuguese
Spanish
Norwegian - Danish
French
Italian
Swedish - Finnish
Swiss
British

Printer - PR 1471

The PR 1471 is an impact dot matrix printer using normal paper. The dot matrix is 9-rows x 7-columns (4+3) and printing is bidirectional with optimized head runs.

Basic features include horizontal and vertical tabulation programs that are maintained in memory during power interruptions by an integral back-up battery supply.

A boldface feature which doubles the character width can be selected. In this case the line character capacity is halved.

The printer can operate in plotter mode (Bit Image Mode) which allows printing of graphs, tables, histograms, images, etc., by associating the print needle configuration with the recieved data bits.

Printing Speed:	140 chars/s
Character Pitch:	10 chars/in 12 chars/in 16.6 chars/in
Maximun Length of Print Line:	132 chars (10 chars/in) 159 chars (12 chars/in) 220 chars (16.6 chars/in)
Paper Feed:	sprocket feed
Paper Feed-in:	rear
Types of Document Handled:	fan-fold forms

The PR 1471 supports the following character sets:

- USA ASCII
- International
- German
- Portuguese
- Spanish
- Norwegian - Danish
- French
- Italian
- Swedish - Finnish
- Swiss
- British
- Yugoslavian
- Icelandic
- Katakana
- Greek
- Russian
- Delta

Printer - PR 1481

The PR 1481 is an impact dot matrix printer using normal paper. The dot matrix is 9-rows x 7-columns (4+3) and printing is bidirectional with optimized head runs.

Basic features include horizontal and vertical tabulation programs that are maintained in memory during power interruptions by an integral back-up battery supply.

A boldface feature which doubles the character width can be selected. In this case the line character capacity is halved.

The printer can operate in plotter mode (Bit Image Mode) which allows printing of graphs, tables, histograms, images, etc., by associating the print needle configuration with the received data bits.

Printing Speed:	140 chars/s
Character Pitch:	10 chars/in
	12 chars/in
	16.6 chars/in
Maximum Length of Print Line:	132 chars (10 chars/in)
	158 chars (12 chars/in)
	220 chars (16.6 chars/in)
Paper Feed:	friction-feed (sprocket-feed optional)
Paper Feed-in:	rear (front optional)
Types of Document Handled:	options include: single forms, cards, fan-fold forms, journal roll

The PR 1481 supports the following character sets:

- USA ASCII
- International
- German
- Portuguese
- Spanish
- Norwegian - Danish
- French
- Italian
- Swedish - Finnish
- Swiss
- British
- Yugoslavian
- Icelandic
- Katakana
- Greek
- Russian
- Delta

Printer - PR 430

The PR 430 is an impact daisy wheel printer using normal paper. Printing is bidirectional with optimized head runs. The daisy wheel, which has 100 character carrying radii, is available in various national character sets and in various styles.

Boldface printing is obtained by overprinting characters 1/120 in out of phase.

Printing Speed:	30 chars/s (10 chars/in)
	31 chars/s (12 chars/in)
	32 chars/s (15 chars/in)
Character Pitch:	10 chars/in
	12 chars/in
	15 chars/in
Maximum Length of Print Line:	150 chars (10 chars/in)
	180 chars (12 chars/in)
	225 chars (15 chars/in)
	128-225 chars (proportional spacing)
Paper Feed:	friction-feed (sprocket-feed optional)
Paper Feed-in:	rear
Types of Document Handled:	options include: single and multi-copy documents,fan-fold forms.

Printer - PR 2300

The PR 2300 is a non-impact dot matrix printer which prints on normal paper using dry spark ink jet technology. The dot matrix is 7-rows x 7-columns for 10 and 12.2 characters/in and 5-rows x 7-columns for 18.3 characters/in. Printing of elementary rows is bidirectional with optimized head runs. When used in the plotting mode, 880 dots can be printed on each elementary row and any number of rows can be printed along the length of the paper supply.

Printing Speed:	50 lines/s (80 chars/line)
Character Pitch:	10 chars/in
	12.2 chars/in
	18.3 chars/in
Maximum Length of Print Line:	80 chars (10 chars/in)
	97 chars (12.2 chars/in)
	147 chars (18.3 chars/in)
Paper Feed:	friction feed, pin feed
Paper Feed-in:	rear
Types of Document Handled:	paper rolls, fan-fold forms

The PR 2300 supports the following character sets:

USA ASCII
German
Spanish
Norwegian - Danish
French
Italian
Swedish - Finnish
British

2.4 EXPANSION OPTIONS

2.4.1 MEMORY EXPANSION BOARDS

There are four types of memory expansion board:

32 KB Black & White
32 KB Colour
128 KB Black & White
128 KB Colour

The expansion boards plug into the memory expansion slots (J10, J9 & J8) on the motherboard. The first expansion board plugs into J10, the second into J9 and the third into J8. One cannot install a 32 KB board together with a 128 KB in the same system. However, one can install a colour memory board and a black & white memory board in the same system.

Memory is divided into two banks: an upper byte bank (even addresses), and a lower byte bank (odd addresses). The operation of these boards is similar to that of the RAM on the motherboard. The signals RMCS1, RMCS2 and RCMS3, generated on the motherboard, address memory on the first, second and third expansion boards respectively.

The circuitry and operation of the black & white and colour memory expansion boards are similar. The major difference between the two is that the memory data output lines of the colour memory expansion boards are not directly connected to the memory data input lines. They are connected together via a buffer.

		MOTHERBOARD		MEMORY EXPANSION											
		USER MEMORY		FIRST EXPANSION	USER MEMORY		SECOND EXPANSION	USER MEMORY		THIRD EXPANSION	USER MEMORY		BIT MAP 1	BIT MAP 2	BIT MAP 3
		REL1-1	REL2		REL1-1	REL2		REL1-1	REL2		REL1-1	REL2			
32KB BOARDS	B/W	41KB	37KB	B/W 32KB BOARD	57KB	69KB	B/W 32KB BOARD	57KB	101KB	B/W 32KB BOARD	57KB	133KB	16KB	—	—
	4 COLOUR	—	—	COLOUR 32KB BOARD	57KB	53KB	B/W 32KB BOARD	57KB	85KB	B/W 32KB BOARD	57KB	117KB	16KB	16KB	—
	8 COLOUR	—	—	COLOUR 32KB BOARD	—	—	COLOUR 32KB BOARD	—	69KB	B/W 32KB BOARD	—	101KB	16KB	16KB	16KB
128KB BOARDS	B/W	41KB	37KB	B/W 128KB BOARD	—	165KB	B/W 128KB BOARD	—	293KB	B/W 128KB BOARD	—	421KB	16KB	—	—
	4 COLOUR	—	—	COLOUR 128KB BOARD	—	149KB	B/W 128KB BOARD	—	277KB	B/W 128KB BOARD	—	405KB	16KB	16KB	—
	8 COLOUR	—	—	COLOUR 128KB BOARD	—	—	COLOUR 128KB BOARD	—	261KB	B/W 128KB BOARD	—	389KB	16KB	16KB	16KB

1 – BIT MAP ON MOTHERBOARD
2 – BIT MAP ON 1st EXPANSION
3 – BIT MAP ON 2nd EXPANSION

Fig. 2-44 Possible Combination of Memory Expansion Boards

32KB Memory Expansion Boards

These memory expansion boards hold 16 RAM IC's organized 16,384 words x 1 bit to provide a total of 32 KB of memory. The RAM's used are the Mostec MK4116-2. Insertion of three memory expansion boards provides the M20 with a total of 224 KB of RAM.

128KB Memory Expansion Boards

These memory expansion boards hold 16 RAM IC's organized 65,536 words x 1 bit to provide a total of 128 KB of memory. The RAM's used are the NEC uPD41664/3 or the Hitachi 4864/2. The two types cannot be mixed on the same expansion board. Insertion of three memory expansion boards provides the M20 with a total of 512 KB of RAM.

Memory Expansion Requirement for Colour Systems

The M20 can support two different types of colour systems:

4 colour
8 colour

The 4 colour system requires 32 KB of RAM for the Bit-Mapped Display. 16 KB on the motherboard and 16 KB on the first colour memory expansion board.

The 8 colour system requires 48 KB of RAM for the Bit-Mapped Display. 16 KB on motherboard, 16 KB on first colour memory expansion board and, 16 KB on second colour memory expansion board.

Figure 2-44 illustrates the possible combination of memory expansion boards and the amount of user memory available.

2.4.2 IEEE 488 INTERFACE BOARD

The IEEE 488 interface board provides the M20 with a general purpose interface bus (GPIB) to allow connection to a variety of devices. These devices must be able to handle the protocol defined in the IEEE 488 standard. This standard defines the methods and types of messages by means of which a CONTROLLER (the M20) can control a TALKER (instruments, disk drives, etc) and one or more LISTENERS (recording devices, printers, disk drives terminals etc).

The bus structure of the IEEE 488 interface is organized into three sets of signal lines as shown in Figure 2-45.

Data Bus

DIO1-DIO8 Data Input/Output: Message bytes are carried on this bidirectional data bus in a bit-parallel byte-serial form.

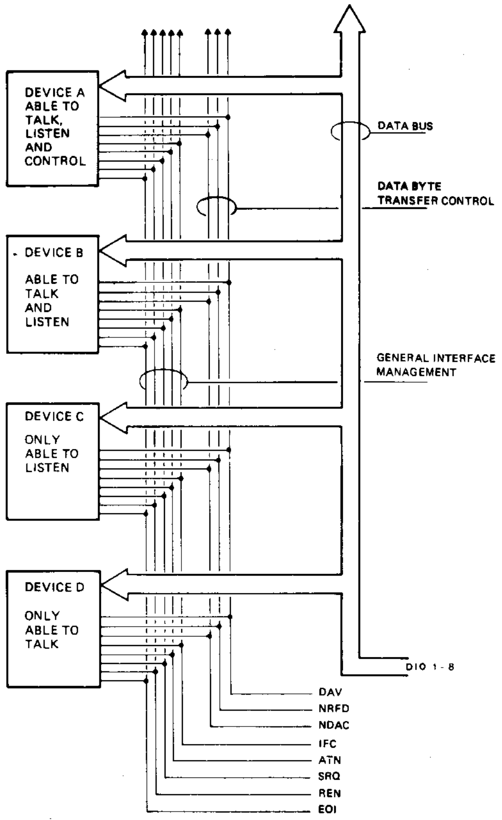

DEVICE A ABLE TO TALK, LISTEN AND CONTROL
DEVICE B ABLE TO TALK AND LISTEN
DEVICE C ONLY ABLE TO LISTEN
DEVICE D ONLY ABLE TO TALK
DATA BUS
DATA BYTE TRANSFER CONTROL
GENERAL INTERFACE MANAGEMENT
DIO 1 - 8
DAV
NRFD
NDAC
IFC
ATN
SRQ
REN
EOI

Fig. 2-45 IEEE 488 Interface Bus Structure

Data Byte Transfer Control Bus - Signal lines used to effect the transfer of each byte of data on the DIO lines from a talker or controller to one or more listeners.

DAV Data Valid: Signal line used to indicate the condition (availability and validity) of information on the DIO signal lines.

NRFD Not Ready For Data: Signal line used to indicate the conditon of readiness of device(s) to accept data.

NDAC Not Data Accepted: Signal line used to indicate the condition of acceptance of data by device(s).

The DAV, NRFD and NDAC signal lines operate in what is called a three wire (inter-locked) handshake process to transfer each byte across the interface.

General Interface Management Bus - Signal lines used to manage an orderly flow of information across the interface.

ATN Attention: Signal line used by controller to specify how data on the DIO lines are to be interpreted and which devices must respond to the data.

IFC Interface Clear: Signal line used by the controller to place the interface system, portions of which are contained in all interconnected devices, in a known quiescent state.

SRQ Service Request: Signal line used by a device to indicate the need for attention and to request an interruption of the current sequence of events.

REN Remote Enable: Signal line used by the controller (with other messages) to select between two alternate source of device programming data.

EOI End Of Identify: Signal line used (by a talker) to indicate the end of a multiple byte transfer sequence or, with ATN (by a controller) to execute a polling sequence.

The IEEE 488 interface board plugs into one of the System Bus slots (J3 or J4) on the motherboard. A ribbon cable connects the board to an IEEE 488 standard connector at the rear of the basic module. Figure 2-46 shows the System Bus interface signals used by the IEEE 488 interface board. Figure 2-47 shows the IEEE 488 interface signals.

Figure 2-48 shows a block diagram of the IEEE 488 interface board. The major elements include:

GPIB Talker/Listener 8291A
GPIB Controller 8292
GPIB Transceivers 8293
Programmable Interrupt Controller 8259

The 8292 Controller is a microprocessor controlled chip designed to function with the 8291A Talker/Listener and two 8293 Transceivers to form an IEEE Standard 488 interface bus. The electrical interface is provided by the transceivers that are hardware programmed to one of four modes of operaton. These modes allow the 8293 to be configured to support both a Talker/Listener/Controller environment and a Talker/Listener environment.

Mode 0	Talker/Listener Control
Mode 1	Talker/Listener Data
Mode 2	Talker/Listener/Controller Control
Mode 3	Talker/listener/Controller Data

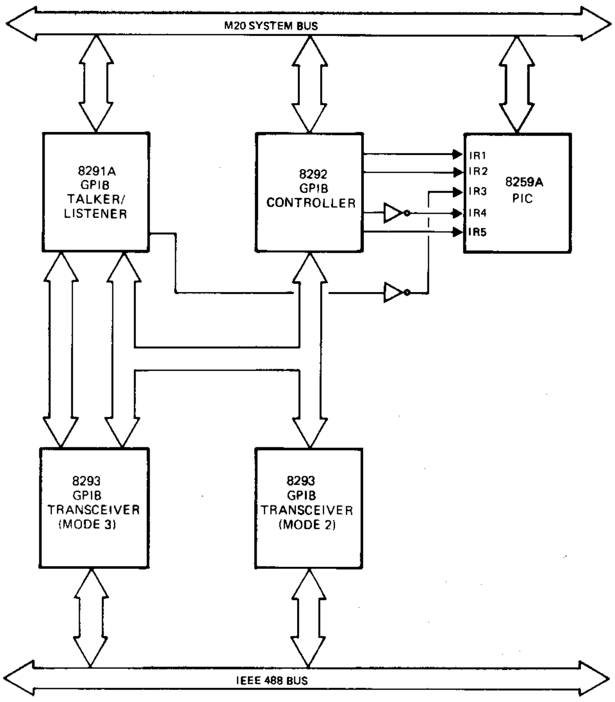
Mode 2 and Mode 3 are used to support the M20 Talker/Listener/Controller environment.

NAME IEEE BOARD	NAME MOTHERBOARD	PIN
M	GND	1,2,3,4,95,97,98,99,100
T	+5V	5,6,7,8,93,94
D0	D0	14
D1	D1	13
D2	D2	16
D3	D3	15
D4	D4	18
D5	D5	17
D6	D6	20
D7	D7	19
D8	D8	22
A1	A1	29
A2	A2	32
A3	A3	31
A5	A5	33
A6	A6	36
A7	A7	35
A8	A8	38
A11	A11	39
A12	A12	42
A13	A13	41
A14	A14	44
A15	A15	43
IOREQ	*I/OREQ	49
RESET	*RESET	50
DS	*DS	53
VINTA	*VINTACK	68
INTOB	*COMVI1	69
CAS0	CAS0	74
CAS1	CAS1	73
CAS2	CAS2	71
CLK	4MHz	76
RW	R/*W	81

Fig. 2-46 System Bus Signals Used By IEEE 488 Interface

NAME IEEE BOARD	PIN	NAME IEEE CONNECTOR	PIN
DIO1	24	DIO1	1
DIO2	23	DIO2	2
DIO3	22	DIO3	3
DIO4	21	DIO4	4
DIO5	1	DIO5	13
DIO6	2	DIO6	14
DIO7	3	DIO7	15
DIO8	4	DIO8	16
EOI	20	EOI	5
DAV	19	DAV	6
NRFD	18	NRFD	7
NDAC	17	NDAC	8
IFC	16	IFC	9
SRQ	15	SRQ	10
ATN	14	ATN	11
REN	5	REN	17
SHIEL	13	SHIELD	12
M	6	DAV GROUND	18
M	7	NFRD GROUND	19
M	8	NDAC GROUND	20
M	9	IFC GROUND	21
M	10	SQR GROUND	22
M	11	ATN GROUND	23
M	12	GROUND	24

Fig. 2-47 IEEE 488 Interface Signals



M20 SYSTEM BUS
8291A GPIB TALKER/ LISTENER
8292 GPIB CONTROLLER
IR1
IR2
IR3
IR4
IR5
8259A PIC
8293 GPIB TRANSCEIVER (MODE 3)
8293 GPIB TRANSCEIVER (MODE 2)
IEEE 488 BUS

Fig. 2-48 IEEE 488 Block Diagram

The IEEE 488 interface utilizes the Intel 8259A Interrupt Controller chip to handle interrupts. The Task Complete Interrupt (TCI) and the Special Interrupt (SPI) outputs of the 8292 are connected to the IR1 and IR2 inputs respectively. The Interrupt Request (*INT) output of the 8291A is connected to IR3 through an inverter. The Input Buffer Not Full Interrupt (*IBFI) and Output Buffer Not Full Interrupt (*OBFI) outputs of the 8292 are connected to IR4 and IR5 respectively. The interrupt priorities are under program control. The 8259A outputs an interrupt request (*COMVI1) to the interrupt controller on the M20 motherboard.

Transferring Information

Information transfer takes places as follows.

All active listeners indicate on the Not Ready For Data (*NRFD) line their state of readiness to acept a new piece of information. If an active listener is not ready it pulls the *NRFD line down to 0V. The active talker observes the state of the *NRFD line and will not start the data transfer until the signal line has gone high.

The active talker observes that the *NRFD line has gone high and places a data byte on the data lines and waits 2 us. It then asserts Data Valid (*DAV) by pulling it low. The 2 us delay allows the data to reach valid logic levels on the data lines. The assertion of *DAV is a signal to the active listener(s) to read information on the data bus. The listeners acknowledge the assertion of *DAV by immediately pulling down to *NRFD.

Until now the active listeners have held Not Data Accepted (*NDAC) low. When *DAV is asserted and all of the active listeners accept the data on the data lines, they will release *NDAC. As the slowest active listener releases *NDAC, *NDAC will go high.

The active talker observes that the *NDAC line has gone high and acknowledges the listeners' acceptance of data by releasing *DAV. The release of *DAV signals to the listeners that the data transfer is complete. The listeners again pull NDAC low in preparation for the next transfer of data.

Figure 2-49 shows a timing diagram of the complete handshake.

NOTE: The control of Data transfer is effected by the active talkers and listeners.

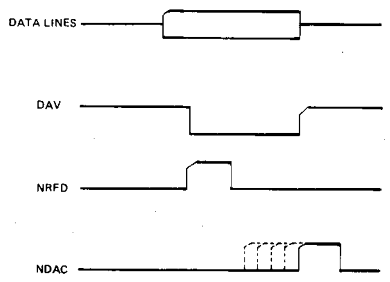


Fig. 2-49 Timing diagram of handshake

Once the controller has configured the bus, it takes no part in subsequent transactions until reconfiguration is desired.

Configuring the Bus

One of the interface management lines is called Attention (*ATN). The active controller manages this line. *ATN signifies whether the transactions on the bus are data or control information. The active talker supervises data transactions but the active controller supervises control transactions.

When the controller wishes to configure the bus it asserts *ATN. This causes any active talker to relinquish control of the *DAV line. Transmission of control information occurs in the same manner as transfer of data. The difference is that when *ATN is asserted, the active controller takes the place of the active talker, and both talkers and listeners accept the information. All devices, whether active or not, accept information. transmitted by the controller when *ATN is asserted.

The active talker and active listeners may be designated during the transmission of control information. The data lines carry control information after *ATN has been asserted. When *ATN is negated, it will assume control of *NDAC and *NRFD. Listeners that do not observe their listen addresses in a control transfer do not change state, remaining as they were before the controller asserted *ATN.

Polling

A poll is the controller request for status information. The controller may request the status of any device individually by addressing the device as a talker and sending that device a Serial Poll Enable command. This constitutes one of the bus commands a controller can send when it asserts *ATN. Using the serial poll, the controller can obtain 8 bits of status information from the addressed device. The controller then sends a Serial Poll Disable command to the device, returning it to data mode.

Address Assignments

Table below shows the address decoding used on the IEEE 488 board. Address bits A11-A15 are all required to be low (logic 0). Address bits A5-A7 select between the various devices such as the 8259A, 8292 or 8291A. Address bits A1-A3 provide the appropriate register selection within each selected device. Note that all I/O addresses are odd, that is A0 is a logic level 1.

ADDRESS DECODING

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	D	D	1	x	x	x	D	y	y	y	D

D = Don't Care

xxx =	000 for 8292
	011 for 8291A
	101 for 8259A

yyy = used to select individual registers

The 8292 Controller System Controller (SYC) input monitors the system controller switch. The IEEE 488 board provides a jumper to enable the user to keep SYC input at logic 0 or 1 as needed. The jumper should be in the 1 position to indicate to the 8292 that it is an active controller.

2.4.3 TWIN RS-232-C INTERFACE BOARD

The TWIN RS-232-C Interface board provides two communication channels with both RS-232-C and 20mA current loop options. The board can be configured as follows:

2 RS-232-C channels
2 Current loop channels
1 Current loop channel + 1 RS-232-C channel

Each of the above configurations requires a specific cable for the connection of peripheral equipment. Figure 2-50 shows the two signal standards on the transmit and receive data circuits. For the Current Loop option, 20 mA represents a binary 1 and 0 mA represents a binary 0. For the RS-232-C option, negative voltage represents a binary 1 and positive voltage represents a binary 0.

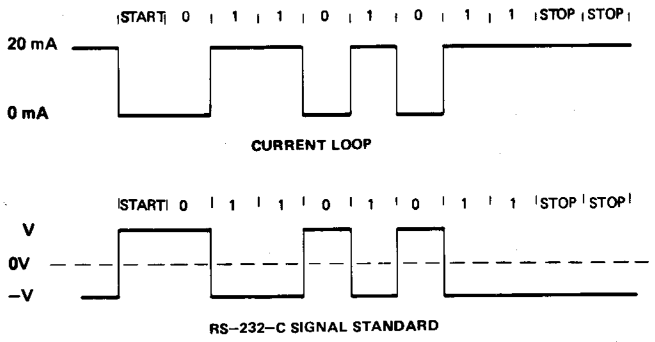

START 0 1 1 0 1 0 1 1 STOP STOP
20 mA
0 mA
CURRENT LOOP
START 0 1 1 0 1 0 1 1 STOP STOP
V
0V
–V
RS–232–C SIGNAL STANDARD

Fig. 2-50 20 mA Current Loop and RS-232-C Signal Standards

The Twin RS-232-C interface board plugs into one of the System Bus slots (J3 or J4) on the motherboard. A ribbon cable connects the board to an edge connector at the rear of the basic module. Figure 2-51 shows the System Bus interface signals used by the RS-232-C interface board. Figure 2-52 shows the RS-232-C interface signals.

NAME RS-232-C BOARD	NAME MOTHERBOARD	PIN
M	GND	1,2,3,4,95,97,98,99,100
P5	+5V	5,6,7,8,93,94
P12	+12V	91,92
M12	GND	89,90
DATA0	D0	14
DATA1	D1	13
DATA2	D2	16
DATA3	D3	15
DATA4	D4	18
DATA5	D5	17
DATA6	D6	20
DATA7	D7	19
DATA8	D8	22
ADRON	A0	29
ADR2	A2	32
ADR5	A5	33
ADR6	A6	36
ADR7	A7	35
ADR11	A11	39
ADR12	A12	42
ADR13	A13	41
ADR14	A14	44
ADR15	A15	43
NORIO	*1/OREQ	49
RESET	*RESET	50
SYSDS	*DS	53
VITAK	*VINTACK	68
SYINT	*SYSINT	70
ICAS0	CAS0	74
ICAS1	CAS1	73
ICAS2	CAS2	71
SYSRW	R/*W	81

Fig. 2-51 System Bus Signals Used By RS-232-C Interface

NAME CHANNEL 1 RS-232-C BOARD	PIN	NAME CHANNEL 1 RS-232-C CONNECTOR	PIN	NAME CHANNEL 2 RS-232-C BOARD	PIN	NAME CHANNEL 2 RS-232-C CONNECTOR	PIN
-	-	FRAME GROUND	4	-	-	FRAME GROUND	41
M	1,2	LG1GD	3	M	20,21	LG2GD	21,24
LPSL1	3	LPSL1	6	LPSL2	22	LPSL2	23
TXD01	4	TXD01	5	TXD02	23	TXD02	26
DTR01	5	DTR01	8	DTR02	24	DTR02	25
RTS01	6	RTS01	7	RTS02	25	RTS02	28
BUSY1	7	BUSY1	10	BUSY2	26	BUSY2	27
DSR01	8	DSR01	9	DSR02	27	DSR02	30
CTS01	9	CTS01	12	CTS02	28	CTS02	29
RXD01	10	RXD01	11	RXD02	29	RXD02	32
T1CLK	11	T1CLK	14	T2CLK	30	T2CLK	31
TX1CK	12	TX1CK	13	TX2CK	31	TX2CK	34
SDET1	13	SDET1	16	SDET2	32	SDET2	33
BUSY1	14	RGID1	15	BUSY2	33	RGID2	36
TCL01	15	TCL01	18	TCL03	34	TCL03	35
TCL02	16	TCL02	17	TCL04	35	TCL04	38
RCL01	17	RCL01	20	RCL03	36	RCL03	37
RCL02	18	RCL02	19	RCL04	37	RCL04	40
R1CLK	19	R1CLK	22	R2CLK	38	R2CLK	39

Fig. 2-52 RS-232-C Interface Signals

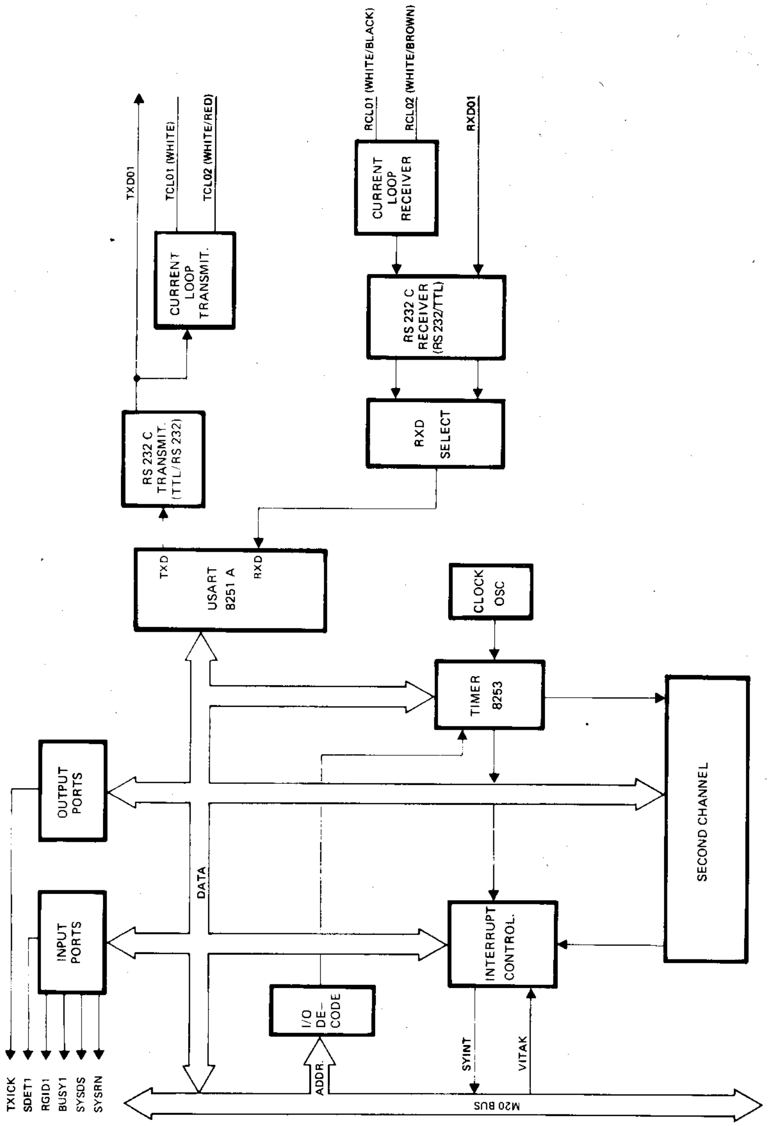


Fig. 2-53 TWIN RS-232-C Interface Block Diagram

Figure 2-53 shows a block diagram of the TWIN RS-232-C interface board. The major elements include:

Programmable Communication Interface (PCI) - Two 8251A PCIs are used to provide the two communicaton channels. Programming of the 8251A is performed by the M20 system software to initialize the 8251A to support M20 communication formats. The PCI, also referred to as a Universal Synchronous/Asynchronous Receiver/Transmitter (USART), accepts data characters from the M20 CPU in parallel format and then converts them into a continous stream of serial data for transmission. It can simultaneously receive serial data streams and then convert them into parallel format for the M20 CPU. It signals the CPU whenever it can accept a new character for transmission, or whenever it has received a new character. For asynchronous serial data communication, as implemented for the M20, a start bit indicates the beginning of the character. This is followed by the character data bits, least significant first, and two stop bits (see Figure 2-50).

Programmable Interval Timer - A 8253-5 Programmable Interval Timer is used to provide the transmitter clocks for the two 8251A PCI. The PCI contains three fully independent counters. Two are used for the PCIs and one can be jumpered to an interrupt input.

Programmable Interrupt Controller (PIC) - The 8259A Programmable Interrupt Controller accepts interrupt requests from the RxRDY and TxRDY outputs of the 8251A PCIs and from the 8253-5 Timer and outputs an interrupt request (SYSIN) to the Interrupt Controller on the M20 motherboard. The 8259A PIC of the TWIN RS-232-C Interface board is a slave to that on the M20 motherboard and only outputs an interrupt request when addressed through ICAS0-ICAS2.

RS-232-C Transmitter - The transmitter circuitry converts the TTL signal levels into RS-232-C signal levels.

```
TTL Binary 0 =  0 V,  RS-232-C Binary 0 = +12 V
TTL Binary 1 = +5 V,  RS-232-C Binary 1 = -12 V
```

RS-232-C Receiver - The receiver circuitry converts the RS-232-C signal levels into TTL signal levels.

Current Loop Transmitter - The transmitter circuitry converts the RS-232-C signal levels into 20mA current loop levels. Two modes of operation are available:

The M20 RS-232-C Interface supplies the 20mA loop current.

The User equipment supplies the 20mA loop current.

Current Loop Receiver - The receiver circuitry converts the 20mA current loop levels into RS-232-C signal levels. Optical couplers are used in order to provide isolation.

Received Data select - The receive data select circuitry detects whether the data received is from current loop circuitry or from RS-232-C circuitry.

2.4.4 ALTERNATE PROCESSOR BOARD 1086

The Alternate Processor Board (APB) 1086 enables the M20 to execute software written for an Intel 8086 microprocessor and thereby work in a CP/M-86 or MS-DOS operating system environment. These operating systems are widely used and a brief outline of their structure follows.

The CP/M and MS-DOS operating systems may each be seperated functionally as follows:

The CONSOLE COMMAND PROCESSOR (CCP) contains the routines which request, obtain and interpret user commands, providing the interface between the CP/M (or MS-DOS) and the user.

The BASIC DISK OPERATING SYSTEM (BDOS) contains all the routines required to allow access to and from the disk drives.

The BASIC INPUT OUTPUT SYSTEM (BIOS) contains all the routines required to allow access to the input/output devices such as Display and keyboard. In particular, BIOS allows the user to print characters on the screen either individually or in text strings, to obtain from the keyboard either single characters or lines of text, and to query the availaility of input from the keyboard.

The APB 1086 plugs into one of the system bus slots (J3 or J4) on the M20 motherboard. The APB 1086 is initialized by the M20 system software. Figure 2-54 shows a flow diagram of the initialization sequence that takes place when the M20 is switched on with an APB 1086 board installed on the motherboard.

After power-on or reset the M20 performs the power-up diagnostics assuring proper operation of the M20. The user is then asked if the M20 is to run under the control of the APB 1086. If the user chooses to run the M20 under the control of the APB 1086, the system is then initialized (country codes fetched from keyboard, screen memory initialized etc.). Then a series of tests on the system are performed. The tests performed (in the sequence shown) are:

8086 CPU Test
Memory Test
LSI tests (identical to those done by Z8001 on power-up)
Keyboard test
Disk Drive Test

If the above tests are successful the CP/M-86 (or MS DOS) operating system is booted from drive 1 and the operator can now run the M20 in a CP/M environment.

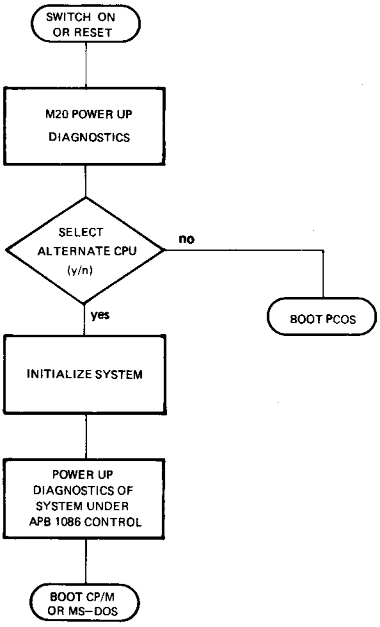


Fig. 2-54 Initialization Sequence

Figure 2-55 shows a block diagram of the APB 1086. The major elements include:

8086 Activation Circuits
Address Decoder
8086 CPU
Read Only Memory
Address Translator and Decoder
Status Generation Logic
State Sequencer

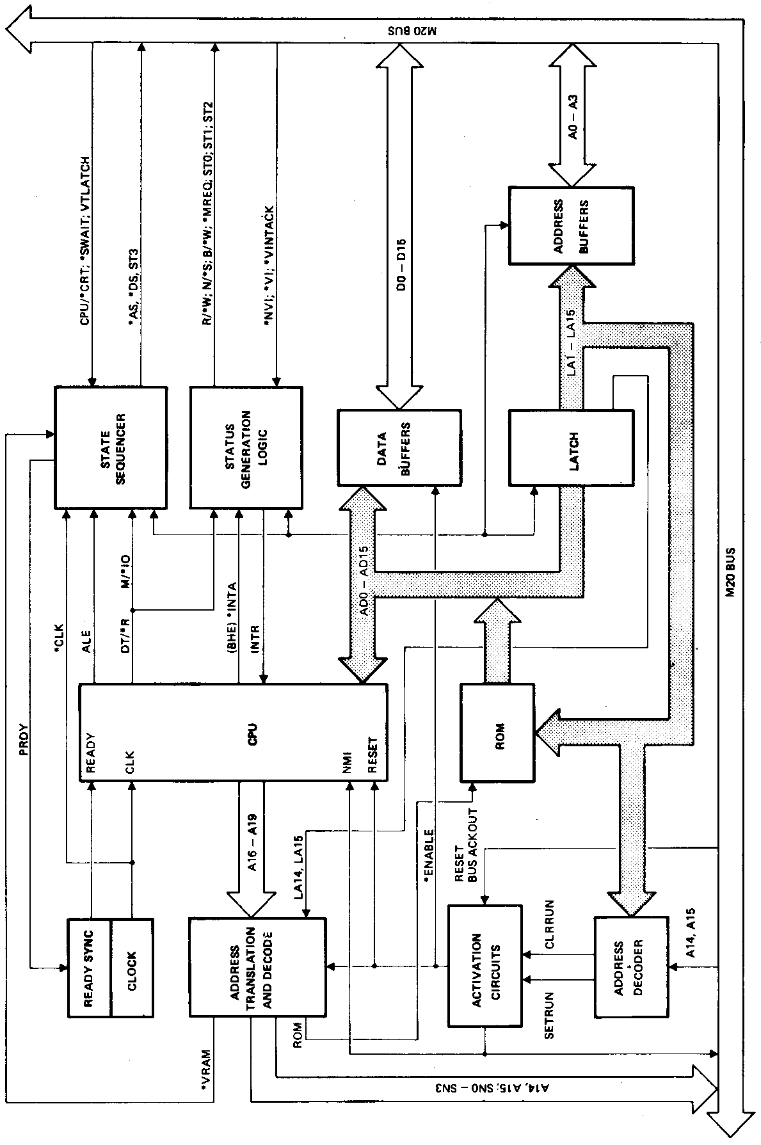


Fig. 2-55 APB 1086 Block Diagram

8086 Activation Circuits

Operation of the APB 1086 is controlled by these activation circuits. These circuits are responsible for starting and stopping the 8086 CPU as well as the CPU of the M20. They also insure that all the APB 1086 bus drivers are in the high impedance state whenever the 8086 CPU is de-activated, and that these drivers are not turned on until the M20 CPU drivers have relinquished the bus.

The 8086 CPU can be de-activated via a software command or a hardware reset occurring over the M20 bus such as occurs upon power on or pressing the M20 reset button. The APB 1086 can only be activated by a software command from the M20 CPU.

Address Decoder

The address decoder detects the presence of a valid I/O instruction on the M20 system bus. Its only criteria for proper operation is that the address and I/O REQ lines on the M20 system bus be stable at least 15ns before and after the *DS line goes low and that the latch outputs, LA1 to LA15, must remain low for at least 50ns. Its purpose is to drive the *SETRUN and *CLRRUN lines for the 8086 activiation circuitry. It decodes address lines A1-A15. It responds to any valid I/O instruction occurring over the M20 bus regardless of which CPU or DMA device is in control of the bus.

Any I/O operation to address 7FFC or 7FFD hex activates *SETRUN and the APB 1086. Any I/O operation to address 7FFA or 7FFB activates *CLRRUN, forces the M20 *NMI line low, and de-activates the 8086.

8086 CPU

The 8086 is the primary element of the APB 1086. This processor has attributes of both 8 and 16 bit microprocessors. The internal functions of the 8086 processor are partitioned logically into two processing units. The first is the Bus Interface Unit (BIU) and the second is the Execution Unit (EU). Figure 2-56 shows a block diagram of the 8086 CPU.

These units can interact directly but for the most part perform as separate asynchronous operation processors. The bus interface units provides the functions related to instruction fetching and queuing, operand fetch and store, and address relocation. This unit also provides the basic bus control.

The execution unit (EU) receives pre-fetched instructions from the Bus Interface Unit (BIU) queue and provides un-relocated operand addresses to the BIU. Memory operands are passed through the BIU for processing by the EU, which passes results to the BIU for storage.

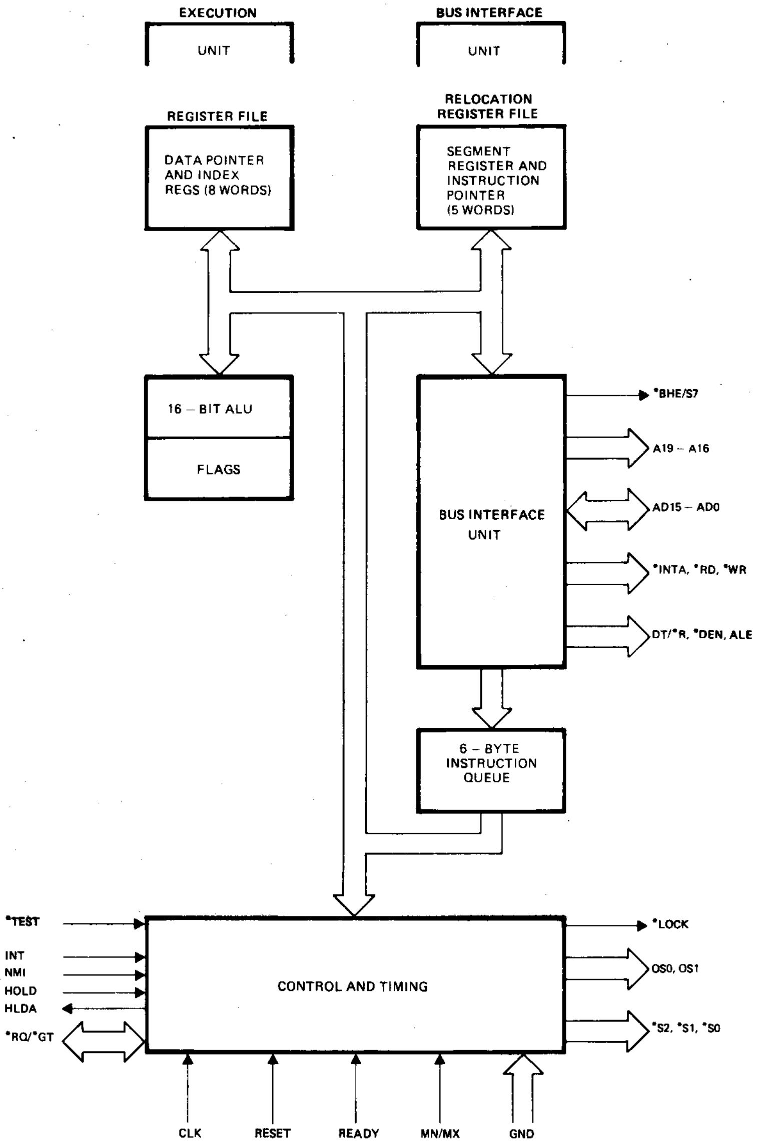


Fig. 2-56 8086 Block Diagram

Read Only Memory (ROM)

Two 2764 (8K X 8) EPROMS are used by the APB 1086 to provide initialization software, boot diagnostics, and operating system core routines for use by 8086 software. The 2764's are arranged for a 16 bit (2 byte) word access, one provides the least significant (odd address) byte of data while the other provides the most significant (even address) byte. Enabling of these ROMs is controlled by the address translator.

Address Translator and Decoder

The address translator and decoder consists of a PAL16L8 Programmable Logic Array. The purpose of this device is to remap the segmented address mapping produced by the M20 mapping PROM so that the 8086 CPU "sees" a continuous memory address space. It also maps the memory reserved for the CRT display into a fixed location within the 8086 address space. It accomplishes this task by translating the most significant 6 address bits (LA14-LA19) from the 8086 onto M20 address lines A14, A15, and SN0-SN3 which are then "mapped" by the M20 mapping PROM to the proper 16 K block of memory.

This device also provides address decoding in that it "knows" which 8086 addresses are possible accesses to memory controlled by the M20 motherboard, which addresses refer to the onboard ROMs and which addresses are non- existent. Accesses to M20 controlled memory are indicated by the *VRAM output going low, *ROM accesses cause the *ROM output to go low. Accesses to non-existent memory are treated as accesses to the upper 16K bytes of segment 1 of M20 controlled memory.

Status Generation Logic

The status generation logic translates 8086 cycle type information supplied by the 8086 on line DT/*R, M/*10, BHE, *INTA into Z8001 signals R/*W, B/*W, *MREQ, ST0, ST1, ST2 and ST3. For 8086 operation, the Z8001 N/S signal is tied permanently high to indicate normal mode. The Z8001 R/*W line is a simple inversion of the 8086 DT/*R line.

Since the 8086 recognizes one type of interrupt, the status generation logic acts as a priority decoder to give priority to non-vectored requests. *NVI and *VI requests are "ORed" to produce signal *INTR and passed to the 8086.

State Sequence

State Sequence consists of a PAL16R6 Programmable Array Logic. The purpose of this device is to generate the basic timing signals required by the M20. The timing sequence produced depends on cycle type.

3. INTERFACING

ABOUT THIS CHAPTER

This chapter deals with all the connectors found on the M20 system and explains the signal names, functions, levels and loading of these connectors. It also illustrates with the use of example programs, typical uses for the IEEE 488 and RS-232-C interfaces. This chapter also contains timing waveforms for memory read/write operations, input/output operations, and interrupt acknowledge operations.

CONTENTS

3. INTERFACING

3.1 PARALLEL INTERFACE - J6

The parallel interface provides the M20 with one Centronics-like parallel interface for the connection of a printer.

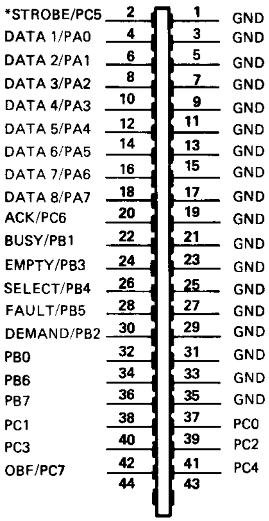


Fig. 3-1 Parallel Interface Connector - J6

3.1.1 SIGNAL NAMES & FUNCTIONS

GND	System Ground.
STROBE (PC5)	Data Strobe - Used to transfer character data from the interface to the printer.
DATA1-DATA8 (PC0-PC7)	Data Lines - Carry the character data from the interface to the printer.

ACK (PC6)	Acknowledge - A low level indicates that the current character has been accepted by the printer and the printer is now available for a new character.
BUSY (PB1)	Busy - A high level indicates that the printer is busy and cannot accept a new character.
EMPTY (PB3)	Paper Empty - A high level indicates that the printer is out of paper.
SELECT (PB4)	Select - A high level indicates that the interface has been selected and is available to transfer data.
FAULT (PB5)	Fault - A low level indicates that a printer fault condition exists.
DEMAND (PB2)	Demand - A low indicates that the printer cannot receive data (used in the Data Products like interface).
OBF (PC7)	Output Buffer Full - A low indicates that Port **A0-A7** (DATA1-DATA8) holds data for transfer to the printer.

3.1.2 LEVELS & LOADING

LEVELS

All Inputs and Outputs are TTL compatible.

LOADING

Outputs - Each output is driven from a high driving capability chip, a LS245 Bus Transceiver or a LS244 Buffer and Line Driver (IOH = -15 mA, IOL = 24 mA) and includes a 1 KΩ pull-up resistor to +5V.

Inputs - Each input connects directly to a single LS245 Bus Transceiver or LS244 Buffer and Line Driver (IIH = 20 μA, IIL = - 200 μA) and includes a 1 KΩ pull-up resistor to +5V.

3.2 RS-232-C COMMUNICATION SERIAL INTERFACE - J7

The RS-232-C Communication Serial Interface provides the M20 with a RS-232-C type serial port for the connection of a modem or plotter.

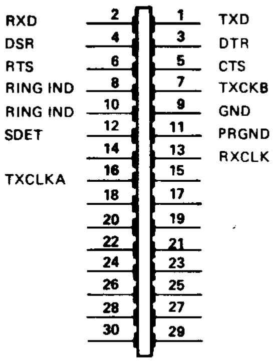


Fig. 3-2 RS-232-C Communication Serial Interface Connector - J7

3.2.1 SIGNAL NAMES & FUNCTIONS

PRGND	Protective Ground - Connected to basic module frame.
GND	Signal Ground/Common Return - Common ground reference for all interchange circuits (except protective ground).
TxD	Transmitted Data, to DCE - Generated by data terminal equipment and transferred to local transmitting signal converter for transmission of data to peripheral data terminal equipment.
RxD	Received Data, from DCE - Generated by receiving signal converter in response to data signals received from peripheral data terminal equipment via peripheral transmitting signal converter.
RTS	Request To Send, to DCE - Used to condition the local data communication equipment for data transmission.
CTS	Clear To Send, from DCE - Used to indicate whether or not the data set is ready to transmit data.
DSR	Data Set Ready, from DCE - Used to indicate the status of the local data set.
DTR	Data Terminal Ready, to DCE - Used to control the switching of data communication equipment to the communication channel.
RING IND	Ring Indicator, from DCE - Indicates that a ringing signal is being received on the communication channel.

SDET	Received Line Signal Detector, from DCE - Indicates that the data communication equipment is receiving a signal which meets its suitability criteria.
TXCKA	Transmitter Signal Element Timing, to DCE - Used to provide the transmitting signal converter with signal element timing information.
TXCLB	Transmitter Signal Element Timing, from DCE - Used to provide the data terminal equipment with the signal element timing information.
RXCLK	Receiver Signal Element Timing, from DCE - Used to provide the data terminal equipment with received signal element timing information.

3.2.2 LEVELS & LOADING

The RS-232-C Communication Serial Interface meets all electrical characteristics defined in the EIA Standard RS-232-C.

3.3 VIDEO INTERFACE - J5

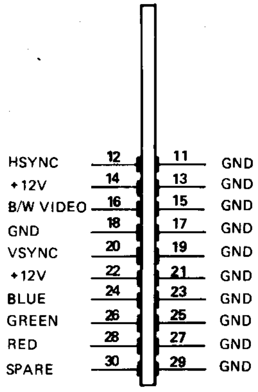


Fig. 3-3 Video Interface Connector - J5

3.3.1 SIGNAL NAMES & FUNCTIONS

GND	System Ground.
+12V	+12V Power Supply.
HSYNC	Horizontal Sync - An active high signal that drives the Display directly to determine the horizontal position of the displayed text.
VSYNC	Vertical Sync - An active high signal that drives the Display directly to determine the vertical position of the displayed text.
B/W VIDEO	Black & White Video
BLUE,GREEN, RED,SPARE	Colour Video

3.3.2 LEVELS & LOADING

LEVELS

All Outputs are TTL compatible.

LOADING

Outputs - The Colour Video outputs are driven from a moderate/high driving capability S175 Quad D-Type Flip-Flop (IOH = -1 mA, IOL = 20 mA)

The Black & White Video output is driven from a 75452B open collector NAND Peripheral Driver (IOH = 100 μA, IOL = 300 mA) employing a pull-up resistor of 30Ω to +5V.

The Horizontal and Vertical Sync outputs are driven from 75451B open collector AND Peripheral Drivers (IOH = 100 μA, IOL = 300 mA) each employing a pull-up resistor of 1KΩ to +5V.

3.4 SYSTEM BUS INTERFACE - J3,J4

3.4.1 IEEE 488 INTERFACE BOARD

The IEEE 488 interface board plugs into one of the System Bus slots (J3 or J4) on the motherboard. A ribbon cable connects the board to an IEEE 488 standard connector at the rear of the basic module.

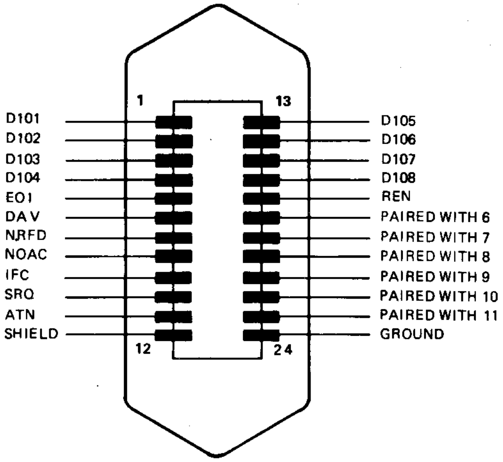


Fig. 3-4 IEEE 488 Interface Connector

3.4.1.1 SIGNAL NAMES & FUNCTIONS

DI01-DI08	Data Input/Output - Data lines used to carry message and data bytes in a bit-parallel byte-serial form.
DVA	Data Valid - Handshake control line; indicates the availability and validity of the information on the DIO.
NDAC	Data Not Accepted - Handshake control line; indicates the condition of acceptance of data by connected device.
NRFD	Not Ready For Data - Handshake control line; indicates the condition of readiness of connected device to accept data.
REN	Remote Enable - Control line; used by active controller to select between two sources of device programming data.
IFC	Interface Clear - Control line; used by active controller to place interface into known quiescent state.
SRQ	Service Request - Control line; indicates need for attention and requests the active controller to interrupt current sequence of events.
ATN	Attention - Control line; used by controller to specify how data are to be interpreted and which device must respond to data.

EOI	End Or Identify - Control line; used by talker to indicate the end of a multiple byte transfer sequence or, by a controller in conjunction with ATN to execute a polling sequence.

3.4.1.2 LEVELS & LOADING

The IEEE 488 Interface meets all electrical characteristics defined in the IEEE Standard 488.

3.4.1.3 PROGRAMMING EXAMPLES

The PCOS command IEEE 488 should be PLOADed before the following programs are run.

USING A PLOTTER

This example requires a plotter to draw a circle and lable it "OLIVETTI M20". The program relates to a specific plotter, and the form of instructions sent may differ if another type of plotter is used. What these instructions are, and how they are to be interpreted, is made clear by comments in the program.

Figure 3-5 shows the example program to carry out the above.

The Plotter Example

Line 10 sets all devices on the interface to Remote Enable.

Line 20 sets the plotter address to 6.

Line 30 sets the variable XY$ equal to "1000,50". These are the co-ordinates of a point that will be sent to the plotter.

Line 40 instructs the plotter to move the pen to the absolute co-ordinate "1000, 50" with the pen up.

Line 50 sets the plotter character size to 15 times the normal size.

Line 60 sets the plotter print orientation.

Line 70 instructs the plotter to print the character string "OLIVETTI M20".

```
Ø5 REM PROGRAM SENDING DATA TO A PLOTTER
1Ø ISET REN
2Ø PA=6 'plotter address
3Ø XY$="1ØØØ,5Ø"
4Ø PRINT@ PA;"M"+XY$
5Ø PRINT@ PA;"S15"
6Ø PRINT@ PA;"QØ"
7Ø PRINT@ PA;"POLIVETTI M2Ø"
8Ø R=1ØØØ
9Ø XY$=STR$(18ØØ+R*COS(Ø))+","+STR$(13ØØ+R*SIN(Ø))
1ØØ PRINT@ PA;"M"+XY$ 'move to first point on circle w/pen up.
11Ø FOR X=Ø TO 6.3 STEP .1
12Ø XY$=STR$(18ØØ+INT(R*COS(X)))+","+STR$(13ØØ+INT(R*SIN(X)))
13Ø PRINT@ PA;"D"+XY$
14Ø NEXT
15Ø PRINT@ PA;"H" 'home w/pen up.
```

Fig. 3-5 Using a Plotter

Lines 80 to 140 contain the equation to draw a circle.

Line 80 sets R to "1000"

Line 90 sets XY$ equal to the string (1800 + R * Cos(0)), plus the string (1300 + R * Sin(0)) to define the first point of the circle.

Line 100 instructs the plotter to move the pen to this point with the pen up.

Line 110 defines the loop start and end values and step value for X. The value of X will therefore change 63 times so that the circle will be made up of 64 straight lines.

Line 120 is similar to line 90 with the Sin and Cos values replaced with the X value set at line 110.

Line 130 instructs the plotter to draw a straight line from the point at which the pen is resting to the co-ordinate set by XY$.

Line 140 completes the loop.

Line 150 instructs the plotter to return the pen to the home position with the pen up.

USING A DIGITAL VOLTMETER

This example requires a Digital Voltmeter (DVM) to take a voltage reading each time the DVM service request (SRQ) is triggered and display the result on the M20. For demonstration purposes this could be done manually, using an external trigger, so that readings are taken whenever required. The program relates to a specific DVM, and the form of the instructions may differ if another type of DVM is used.

Figure 3-6 shows the example program to carry out the above.

```
  05 REM PROGRAM TRIGGERING AND READING VOLTMETER OUTPUT
  10 CLEAR
  20 DEFINT A-Z
  30 DVM = 1
  40 ISET REN
  50 WBYTE 33,4; 'Send SDC
  60 PRINT @ DVM; "F1R7M3T2D1A0"
  70 ON SRQ GOSUB 1000
  80 PRINT "Waiting for srq"
  90 'P=POS(1) 'Returns row of text cursor.
 100 CURSOR (1,2) : I=I+1 : PRINT I : GOTO 100
 110 END
1000 'SRQ ROUTINE
1010 PASS = PASS + 1
1020 POLL DVM,STATUS
1030 CURSOR (1,10)
1040 PRINT "STATUS=";STATUS,"PASS=";PASS
1050 LINE INPUT@ DVM ; VOLTS$
1060 PRINT "READING IS",VOLTS$
1070 PRINT @ DVM;
1080 ON SRQ GOSUB 1000 : RETURN
```

Fig. 3-6 Using a Digital Voltmeter

The DVM Example

Line 10 closes all previously open files.

Line 20 defines that all variables starting with an alpha character will be integer variables.

Line 30 sets the DVM address to 1.

Line 40 sets all devices on the interface to Remote Enable.

Line 50 sends Selected Device Clear (SDC) to the DVM. (33, the DVM address plus listen 1+32, followed by 4, the interface message for SDC).

Line 60 sends the string "F1R7M3T2D1A0" that sets the DVM to:

F1	DC Volts
R7	Auto Ranging
M3	Maths Off
T2	Trigger External
D1	Data Ready RQS (sets SRQ on valid data)
A0	Auto Calibration Off

Line 70 transfers control to the the SRQ subroutine at line 1000.

Line 80 displays waiting for SRQ and line 90 repositions the cursor on the next line.

Line 100 sets up a waiting loop that counts from 1, and displays the count value, until the SRQ is triggered and repositions the cursor after each count so that the previous count value is overwritten.

Line 110 defines the end of the main body of the program.

Line 1000 The start of the SRQ subroutine.

Line 1010 counts the number of times the subroutine has been called.

Line 1020 polls the DVM for status.

Line 1030 repositions the cursor so that the following instruction to print status and pass information does not overwrite the count value.

Line 1040 displays the status and pass information.

Line 1050 instructs the DVM to output its present voltage reading.

Line 1060 displays the voltage reading.

Line 1070 is a dummy PRINT instruction that causes the DVM to become a listener in readyness for the next SRQ trigger.

Line 1080 returns control to the main body of the program. By having GOSUB and return instructions on the same line, the interrupt routine is completed before another can be initiated.

USING A DIGITAL VOLTMETER AND A PLOTTER

This example uses a DVM to take readings of a mains supply voltage and a plotter to plot a graph of voltage against time.

Figure 3-7 shows the example program to carry out the above.

```
  0 REM PROGRAM USING TWO PERIPHERALS
  1 TIME = 1000
  5 ON ERROR GOTO 9000
  7 CALL "pl ie"
 10 XMIN = -30
 20 XMAX = 140
 30 INPUT "Enter your A.C. line voltage: ";ACV
 40 YMAX = .2 * ACV
 50 YMIN = -YMAX
 60 MAXY = YMAX / 2
 70 MINY = -MAXY
 80 NUMTIC = MAXY - MINY
 90 TICINC = INT(1300/NUMTIC)
100 PLOTTER = 6
105 'Functions FNX$ & FNY$ return plotter units in string form that is
    suitable for sending to plotter.
110 DEF FNX$(X) = STR$(INT((X-XMIN)*3600/(XMAX-XMIN)))
120 DEF FNY$(Y) = STR$(INT((Y-YMIN)*2600/(YMAX-YMIN)))
130 C$ = "Move" : X = 0 : Y = MINY : GOSUB 1000
140 GOSUB 2000
170 C$ = "Move" : X = 0 : Y = 0 : GOSUB 1000
175 X0% = 0 : X1% = 100
180 X0$ = FNX$(X0%)
182 X1$ = FNX$(X1%)
184 TICINC = INT((X1%-X0%)/10)
186 PRINT  PLOTTER;"X1," +STR$(TICINC) + ",10"
188 GOSUB 2100
200 DVM = 1
210 ISET REN
220 WBYTE 63,95;
230 PRINT@ DVM;"F2R7T2T3"
240 WBYTE 33,8;
250 INPUT@ DVM;ACV1
255 WBYTE 63; 'UNL
260 Y = ACV - ACV1
270 C$ = "Move": X = 0 : GOSUB 1000
280 FOR X = 1 TO 100
285 C$ = "D"
290 WBYTE 33,8; 'GET
300 INPUT@ DVM;ACV1
310 Y = ACV - ACV1
320 GOSUB 1000
322 C$ = "Move" : GOSUB 1000
325 FOR K = 1 TO TIME : NEXT K
330 NEXT X
340 PRINT@ PLOTTER;"H"
350 END
1000 'Sub plot
1010 CODE$ = LEFT$(C$,1) 'Get rid of all but the first character.
1020 IF CODE$ = "H" THEN PLOT$ = CODE$ : GOTO 1070 'Home needs no
     parameters
```

Fig. 3-7 Using a Digital Voltmeter and a Plotter (cont.)

```
1Ø25 'X$ & Y$ are the plotter units in string form while X% & Y% are the
     same values in numeric form.
1Ø3Ø X$ = FNX$(X)
1Ø4Ø Y$ = FNY$(Y)
1Ø5Ø PLOT$ = CODE$ + X$ +"," + Y$ 'Build string to send to plotter.
1Ø7Ø PRINT@ PLOTTER ; PLOT$ 'Send string to plotter.
1Ø75 WBYTE 63; 'UNL
1Ø8Ø RETURN
2ØØØ 'Sub Y-axis
2ØØ5 PRINT@ PLOTTER;"XØ" + "," + STR$(TICINC) + "," + STR$(NUMTIC)
2Ø1Ø C$ = "Move" : X =-4
2Ø2Ø PRINT@ 6,"S2" 'Small character size
2Ø3Ø FOR Y = MAXY TO MINY STEP-1
2Ø4Ø GOSUB 1ØØØ
2Ø5Ø PRINT@ PLOTTER; "P" + STR$(Y)
2Ø6Ø NEXT Y
2Ø65 WBYTE 63; 'UNL
2Ø7Ø RETURN
21ØØ 'Sub Xaxis
21Ø5 XT = 1ØØ
211Ø C$ = "Move" : Y =-1
212Ø FOR X = 98 TO 8 STEP-1Ø
213Ø GOSUB 1ØØØ
214Ø PRINT@ PLOTTER;"P" + STR$(XT)
2145 XT = XT-1Ø
215Ø NEXT X
216Ø RETURN
9ØØØ 'Error recovery
9Ø1Ø RESUME NEXT
```

Fig. 3-7 Using a Digital Voltmeter and a Plotter

Line 1 sets the time interval between voltage readings. If this is to be more than 10 seconds the program should be amended to lift the pen after each point is plotted so that ink does not leak onto the paper. This can be done by using a dummy move instruction to move the pen to the co-ordinates at which it is resting. The pen will be lifted but remain at the same co-ordinates until the next draw instruction.

Line 5 prevents the program being halted. The error recovery subroutine at line 9000 causes the program to go on to the next plot.

Line 7 calls and PLOADs the IEEE 488 package.

Line 10 and 20 define the upper and lower limits of the X axis.

Line 30 asks the user to enter the nominal voltage of the mains supply to be monitored.

Line 40 and 50 define the limits of the Y axis as ±20% of the mains supply voltage.

Line 60 and 70 define the limits to which the Y axis will be plotted.

Line 80 and 90 set the number of increments and their interval for the Y axis.

Line 100 sets the plotter address to 6.

Line 110 and 120, as described in line 105, return the plotter units in string form for sending to the plotter.

Line 130 instructs the plotter to move the pen to X=0 and the minimum value of Y and sends transfers control to the sub plot routine at line 1000.

Sending Co-ordinates to the Plotter

Line 1000 the start of the sub plot routine.

Line 1010 strips CODE$ of all but the first character.

If the character left in CODE$ is "H", PLOT$ = CODE$ and lines 1030 to 1050 are jumped. "H" (Home) does not reqire parameters X and Y.

Line 1030 and 1040 assign the values of X and Y to plotter units.

Line 1050 strings the three values (CODE$ + X$ + Y$) into PLOT$.

Line 1070 sends this string to the plotter.

Line 1075 sends a universal listen (UNL) so that the system is ready to receive further instructions.

Line 1080 returns control to the main body of the program.

Drawing the Y Axis

Line 140 transfers control to the sub Y axis subroutine at line 2000.

Line 2000 is the start of the subroutine to draw the Y axis and the increment numbers beside it.

Line 2005 instructs the plotter to draw the Y axis. (X0 for the plotter used in this example) with NUMIC number of segments drawn at TICINC increments

Line 2010 sets C$ to move and X to -4, so that the scale is drawn to the left of the Y axis.

Line 2020 sets the plotter character size. For efficiency, since the Y axis was drawn from bottom to top, the scale is drawn from top to bottom, so that the pen does not travel the length of the axis to start.

Line 2030 starts the loop, decrementing Y by 1 each time; beginning with the value MAXY and ending with MINY.

Line 2040 transfers control to the sub plot subroutine at line 1000. The string variable PLOT$ is built with X constant at -4 as set at line 2010 and the loop varies Y to move the pen vertically down the page.

Line 2050 addresses and instructs the plotter to draw the value of Y.

Subroutine 1000 sets the pen to the next position and line 2050 again instructs the plotter to draw the new value of Y, and so on.

When all values of Y have been drawn, line 2065 sends a Universal Listen (UNL) to all peripherals.

Line 2070 returns control to the main body of the program.

Line 170 sets C$ to move and X and Y to 0. Subroutine 1000 moves the pen to 0,0.

Drawing the X Axis

Line 175 assigns X0% and X1% variables.

Lines 180 and 182 set the string variables X0$ and X1$ to the minimum and maximum values of X, in plotter units, for sending to the plotter.

Line 184 sets TICINC to the integer part of the value (X1%-X0%)/10. Thus there are to be 10 intervals along the X axis.

Line 186 is similar to line 2005, but this time the X axis is to be drawn, with 10 ticks, at intervals of TICINC.

Line 188 transfers control to subroutine 2100.

Line 2100 is the start of the subroutine that draws the X axis and the increment numbers under it.

Line 2105 sets the start point of the X co-ordinate.

Line 2110 sets C$ to move and Y to -1 (so that the scale is drawn under the X axis.

A loop is set up at line 2120, starting at 98 so that the numbers are directly under the tick marks.

Line 2130 transfers control to subroutine 1000.

Subroutine 1000 moves the pen to the co-ordinates set, with Y constant at -1 and X decreasing by 10 at each pass.

Line 2140 instructs the plotter to draw the new value of XT.

Line 2145 changes the value of XT at each pass.

Line 2150 completes the loop.

When all values have been drawn line 2160 returns control to the main body of the program.

Setting Up the DVM

Line 200 sets the DVM address to 1.

Line 210 sets all devices on the interface to Remote ENable.

Line 220 sends a universal listen UNL (63) and univeral talk UNT (95), so that the plotter will not listen to the proceeding instructions.

Line 230 sends the string "F2R7T2T3" that sets the DVM to:

F2	AC Volts
R7	Auto Ranging
T2-T3	Enables the DVM to respond to a Group Execute Trigger (GET) instruction

In order that the first point on the graph is a real point, not where the pen happens to be resting (normally 0), the first GET is triggered outside the main loop, at line 240. This addresses the the DVM as a listener (1+32) and sends the GET instruction (8).

The GET will be sent each time line 290 is reached.

Plotting the Graph

Line 250 accepts input from the DVM and stores it as variable ACV1.

Line 260 computes the difference between the mean line voltage set at line 30 (ACV) and the measured voltage (ACV1) and assigns this value to the Y co-ordinate.

Line 270 sets C$ to move, X to 0 and transfers control to subroutine 1000 that positions the pen at this point.

Line 280 starts the loop. 100 voltage readings are to be taken and the loop variable is used to increment the X co-ordinate.

Line 285 sets C$ to Draw.

Line 290, 300 and 310 correspond to 240, 250 and 260, triggering and accepting voltage readings from the DVM. This information is used by subroutine 1000 to reposition the pen, but this time the pen is drawing.

Line 325 runs the time interval set at Line 1.

Line 330 defines the end of the loop and the process repeats until all 100 points have been plotted (101 including the first).

Line 340 sends the plotter the Home instruction.

Line 350 terminates the program.

Figure 3-8 shows an example of a plot produced using this program.

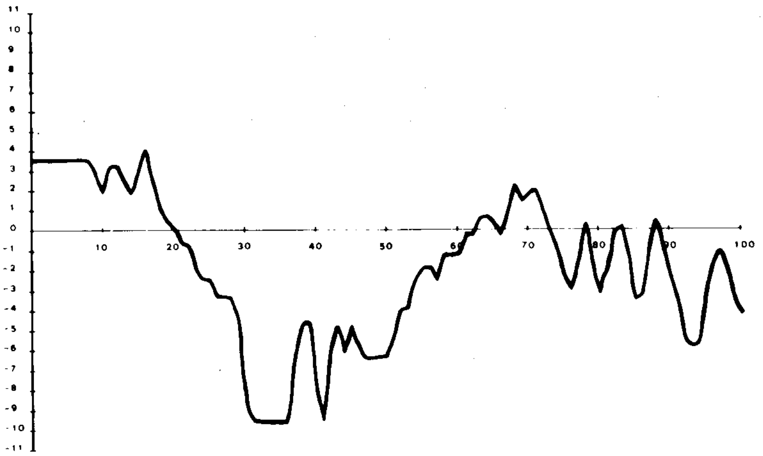


Fig. 3-8 Sample Plot of Voltage against Time

3.4.2 TWIN RS-232-C INTERFACE BOARD

The Twin RS-232-C interface board plugs into one of the System Bus slots (J3 or J4) on the motherboard. A ribbon cable connects the board to an edge connector at the rear of the basic module.

3.4.2.1 SIGNAL NAMES & FUNCTIONS

FM1GD,FM2GD	Protective Ground - Connected to basic module frame.
LG1GD,LG2GD	Signal Ground/Common Return - Common ground reference for all interchange circuits (except protective ground).
TxD01,TxD02	Transmitted Data, to DCE - Generated by data terminal equipment and transferred to local transmitting signal converter for transmission of data to peripheral data terminal equipment.
RxD01,RxD02	Received Data, from DCE - Generated by receiving signal converter in response to data signals received from peripheral data terminal equipment via peripheral transmitting signal converter.

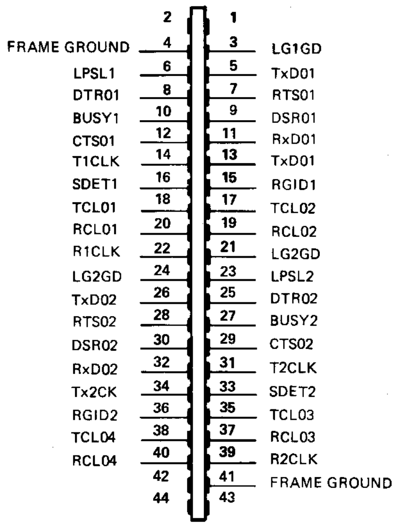


Fig. 3-9 RS-232-C Interface Connector

RTS01,RTS02	Request To Send, to DCE - Used to condition the local data communication equipment for data transmission.
CTS01,CTS02	Clear To Send, from DCE - Used to indicate whether or not the data set is ready to transmit data.
DSR01,DSR02	Data Set Ready, from DCE - Used to indicate the status of the local data set.
DTR01,DTR02	Data Terminal Ready, to DCE - Used to control the switching of data communication equipment to the communication channel.
RGID1,RGID2	Ring Indicator, from DCE - Indicates that a ringing signal is being received on the communication channel.
SDET1,SDET2	Received Line Signal Detector, from DCE - Indicates that the data communication equipment is receiving a signal which meets its suitability criteria.
T1CLK,T2CLK	Transmitter Signal Element Timing, to DCE - Used to provide the transmitting signal converter with signal element timing information.

TX1CK,TX2CK	Transmitter Signal Element Timing, from DCE - Used to provide the data terminal equipment with the signal element timing information.
R1CLK,R2CLK	Receiver Signal Element Timing, from DCE - Used to provide the data terminal equipment with received signal element timing information.
BUSY1,BUSY2	Busy Signal, to DTE - Non-standard circuit, used to transmit busy or anomaly information.
TCL01,TCL02	20 mA Current Loop - Transmitted data and Return (Ch 1).
RCL01,RCL02	20 mA Current Loop - Received data and Return (Ch 1).
TCL03,TCL04	20 mA Current Loop - Transmitted data and Return (Ch 2).
RCL03,RCL04	20 mA Current Loop - Received data and Return (Ch 2).

3.4.2.2 LEVELS & LOADING

The RS-232-C Interface meets all electrical characteristics defined in the EIA Standard RS-232-C.

3.4.2.3 PROGRAMMING EXAMPLES

The PCOS command RS232 should be PLOADed before the following programs are run.

SETTING UP TRANSMISSION PARAMETERS

This example sets up the transmission parameters by defining a Set Communication port (SCOMM) command according to required transfer characteristics.

Figure 3-10 shows the example program to carry out the above.

Program Set-up

In accordance with normal CI and BASIC programming practice, all variables are defined as integers before usage. Three arrays are dimensioned to contain port numbers (line 40), Baud rates (line 70) and parity check values (line 100).

Making the CI Resident

Line 130 asks the user if the RS-232-C driver is resident. If not, it is made resident by the EXEC 'rs' instruction at line 150.

Line 160 asks the user if CI is resident. If not, it is made resident by the EXEC 'ci' instruction at line 180.

```
1Ø REM
2Ø REM program to set up transmission parameters
3Ø REM
4Ø DIM PN$ (3)
5Ø DATA "com:","com1:","com2:"
6Ø FOR I = 1 TO 3 : READ PN$(I):NEXT I
7Ø DIM R$(8)
8Ø DATA "5Ø","11Ø","3ØØ","6ØØ","12ØØ","24ØØ","48ØØ","96ØØ"
9Ø FOR I = 1 TO 8 : READ R$(I) : NEXT I
10Ø DIM P$(3)
11Ø DATA "none", "odd", "even"
12Ø FOR I = 1 TO 3 : READ P$(I) : NEXT I
13Ø PRINT :INPUT "'rs' already executed? (yes-no) ",A$
14Ø IF A$<>"no" THEN 16Ø
15Ø EXEC "rs"
16Ø PRINT :INPUT "'ci' already resident? (yes-no) ",A$
17Ø IF A$<>"no" THEN 19Ø
18Ø EXEC "pl ci"
19Ø PRINT :INPUT "enter port number (0-1-2)", PN%
2ØØ IF PN%<Ø OR PN%>2 THEN PRINT "wrong selection" : GOTO 19Ø
21Ø PRINT  :INPUT  "enter  baud  rate  (5Ø-11Ø-3ØØ-6ØØ-12ØØ-24ØØ-48ØØ-96ØØ)
    ",BR$
22Ø FOR I=1 TO 8:IF R$(I)=GRI THEN 25Ø
23Ø    NEXT I
24Ø    PRINT "wrong selection" : GOTO 21Ø
25Ø PRINT:INPUT "enter parity (none-odd-even)", PA$
26Ø FOR I=1 TO 3:IF P$(I)=PA$ THEN 29Ø
27Ø    NEXT I
28Ø    PRINT "wrong selection":GOTO 25Ø
29Ø PRINT :INPUT "enter no. of stop bits (Ø=1 bit 1=1.5 bits 2=2 bits)"
    SB$
3ØØ IF SB$<"Ø" OR SB$>"2" THEN PRINT "wrong selection" : GOTO 29Ø
31Ø PRINT :INPUT "enter no. of data bits (5-6-7-8) ", DB$
32Ø IF DB$<"5" OR DB$>"8" THEN PRINT "wrong selection":GOTO 31Ø
33Ø PRINT:INPUT "enter duplex (full-half) ", DU$
34Ø IF DU$<>"full" AND DU$<>"half" THEN PRINT "wrong selection": GOTO 33Ø
35Ø PRINT :INPUT "enter handshake (on-off) ", HA$
36Ø IF HA$<>"on" AND HA$<>"off" THEN PRINT "wrong selection":GOTO 35Ø
37Ø PRINT :INPUT "enter buffer size (>Ø) ", BS$
38Ø IF BS$<= "Ø" THEN PRINT "wrong selection" :GOTO 37Ø
39Ø ES$  =  "sc   "+PN$(PN%+1)+","+BR$+","+PA$+","+SB$+","+DB$+","+DU$+",
    "+HA$+","+BS$
4ØØ EXEC ES$                                        'set communications port
41Ø E% = Ø
42Ø CALL "ci" (PN%,"o",@E%)            'open port
43Ø IF E%=Ø THEN GOTO 46Ø
44Ø    PRINT "port open error = ";E%
45Ø    STOP
46Ø PRINT:INPUT "default transmission parameters ok? (yes-no)", A$
47Ø IF A$<>"no" THEN 65Ø
48Ø HS% = Ø
```

Fig. 3-10 Program to Set-up Transmission Parameters (cont.)

```
490 A$ = "transmit enable":GOSUB 700
500 HS% = HS% + I%
510 A$ = "data terminal ready" : GOSUB 700
520 HS% = HS% + 2*I%
530 A$ = "receive enable":GOSUB 700
540 HS% = HS% + 4*I%
550 A$ = "send break character":GOSUB 700
560 HS% = HS% + 8*I%
570 A$ = "error reset" : GOSUB 700
580 HS% = HS% + 16*I%
590 A$ = "request to send" : GOSUB 700
600 HS% = HS% + 32*I%
610 CALL "ci" (PN%,"sw",@E%,,@HS%)                'set hardware status
620 IF E%=0 THEN GOTO 650
630    PRINT "status write error = "; E%
640    STOP
650 PRINT : PRINT "end of setup program"
660 END
700 PRINT
710 PRINT A$;" (1=yes,0=no)";
720 INPUT I%
730 IF I%<>0 AND I%<>1 THEN PRINT "wrong selection": GOTO 700
740 RETURN
```

Fig. 3-10 Program to Set-up Transmission Parameters

Selecting Transmission Parameters

Lines 190 to 370 ask the user to enter a value for each of the SCOMM parameters; port number, Baud rate, parity, number of stop bits, number of data bits, half or full duplex, handshake on or off, buffer size.

In each case the user is asked to choose from a set of valid values and wrong selection displayed if the user enters an invalid value.

Line 420 opens the selected port and line 440 displays any port open error and value.

Selecting Hardware Status

Lines 460 to 600 ask the user to make a selection from all the available defaults in the hardware status byte. If the default transmission parameters are not OK (line 460), the user is asked to enter a value (1=yes, 0=no) for each of the transmission parameters; transmit enable, data terminal ready, receive enable, send break character, error reset, request to send, wrong selection displayed if the user enters an invalid value.

When the selection is completed the status byte is written on the selected port at line 610.

Should such a set-up not be possible, line 630 displays any status write error and value.

Line 650 displays end of set-up program.

TRANSMISSION OF DATA FROM THE M20

This example deals with the transmission of data from the M20. The example program, listed in Figure 3-11, transmits a string of characters entered at the keyboard and is considered complete when a carriage return is entered. To interrupt the program the user enters **CTRL C.**

```
10 REM
20 REM     program to transmit data
30 REM
40 PN% = 0
50 E% = 0
60 LINE INPUT "enter string",A$
70 CALL "ci" (PN%, "w",@E%, A$, 13)
80 IF E%<>0  THEN PRINT "write error ="; E% :STOP
90 GOTO 60
```

Fig. 3-11 Program for the Transmission of Data from the M20

Line 60 asks the user to enter a string of characters.

The string, followed by a carriage return (13), is sent to the transmission line by the CI command at line 70.

Error Check

Line 80 displays any return write error and value.

TRANSMISSION OF DATA TO THE M20

The following examples deal with data received by the M20. The example programs, listed in Figure 3-12 and Figure 3-13, receive a string of ASCII character codes on the RS-232-C interface and considers the string complete when a carriage return code is received.

TRANSMISSION OF DATA WITHOUT STATUS CHECKS

Data Transfer

When set-up is complete, line 80 displays ready to receive.

The CI command at line 120 reads the character codes one by one over the interface.

Lines 140 to 160 build these codes into a string of characters for display once a carriage return code is received.

```
10 REM
20 REM  program to receive data (without status check)
30 REM
40 PN% = 0
50 E% = 0
60 BC% = 0
70 C$ = SPACE$(1)
80 PRINT : PRINT "ready to receive"
90 CALL "ci" (PN%, "sr",@E%,,,@BC%)         'check one character
100 IF E%<>0  THEN PRINT "status read error = ";E% :STOP
110 IF BC%=0 THEN GOTO 90
120 CALL "ci" (PN%, "r",@E%,@C$, 1)         'receive one character
130 IF E%<>0  THEN PRINT "read error = ";E% : STOP
140 IF ASC(C$)<>13  THEN GOTO 170
150 IF A$ = "END"  THEN GOTO 190
160 PRINT A$ : A$ = "" : GOTO 80
170 A$ = A$ + C$
180 GOTO 90
190 CALL "ci" (PN%, "c",@E%)                'close port
200 IF E%<>0  THEN PRINT "port close error = ";E% :STOP
210 PRINT : PRINT "end of receive program"
220 END
```

Fig. 3-12 Transmission of Data to the M20 without Status Checks

Receipt Check

Lines 90 to 110 make use of the buffer count in the CI Status Read command to detect when a character code has been received.

Line 130 displays any read error and value.

End of Data

When the END string is received at line 150, control is returned to the keyboard and the interface port is closed, line 190.

Line 200 displays any port close error and value and line 210 displays end of receive program.

```
10 REM
20 REM  program to receive data (with status check)
30 REM
40 PN% = 0
50 E% = 0
60 HS% = 0 : DS% = 0
70 BC% = 0
80 C$ = SPACE$(1)
90 PRINT :PRINT "ready to receive"
100 CALL "ci" (PN%, "sr",@E%,@HS%,@DS%,@BC%)        'get status & buffer
110 IF E%=0 THEN GOTO 130                            'count
120 PRINT "status read error = ";E% : STOP
130 IF BC%=0  THEN GOTO 100
140 CALL "ci"  (PN%, "r",@E%,@C$, 1)                'get one character
150 IF E%=0  THEN GOTO 270
160 PRINT "read error = ",E%
170 IF E%<>4  THEN STOP
180 SM% = HS% AND &H8
190 IF SM%=&H8 THEN PRINT "parity error"
200 SM% = HS% AND &H10
210 IF SM%=&H10  THEN PRINT "overrun error"
220 SM% = HS% AND &H20
230 IF SM%=&H20 THEN PRINT "framing error"
240 SM% = DS% AND &H100
250 IF SM%=&H100 THEN PRINT "buffer overflow error"
260 STOP
270 IF ASC(C$)<>13  THEN GOTO 300
280 IF A$="END"  THEN GOTO 320
290 PRINT A$ : A$ = "" : GOTO 90
300 A$ = A$ + C$
310 GOTO 100
320 CALL "ci" (PN%, "c",@E%)                         'close port
330 IF E%<>0  THEN PRINT "port close error = ";E% :STOP
340 PRINT :PRINT "end of receive program"
350 END
```

Fig. 3-13 Transmission of Data to the M20 with Status Checks

This example is similar to the above except for the status check, lines 180 to 260, and the use of the composite status mask variable SM.

The description of the part common to the two programs is not repeated here.

Status Check

Lines 180 to 260 make use of the hardware status (HS) byte and driver status word (DS). The returned values HS and DS read over the interface at line 100 are used to detect status errors. The nature of any errors detected, parity, overrun, framing and buffer overflow, are displayed at lines 190, 210, 230 and 250 respectively.

3.4.3 SYSTEM BUS INTERFACING

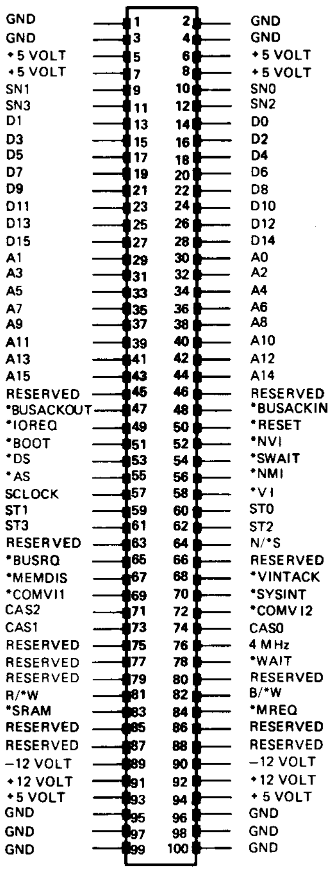


Fig. 3-14 System Interface Bus Connectors - J3/J4

3.4.3.1 SIGNAL NAMES & FUNCTIONS

GND	System Ground.
+5V	+5V Power Supply.
+12V	+12V Power Supply.
-12V	-12V Power Supply.
D0-D15	Bidirectional 16 bit Data Bus - The CPU relinquishes this bus when *BUSACKOUT is active low.
A0-A15	Bidirectional 16 bit Address Bus - The CPU relinquishes this bus when *BUSACKOUT is active low.
R/*W	Read/Write - Bidirectional control line used to determine the direction of data transfer for memory and input/output transactions. For memory read, R/*W is high; for memory write, R/*W is low. For input transactions, R/*W is high; for output transactions, R/*W is low. The CPU relinquishes this line when *BUSACKOUT is active low.
*DS	Data Strobe - Bidirectional control line provides timing information for data movement to and from the CPU. The CPU relinquishes this line when *BUSACKOUT is active low.
*AS	Address Strobe - Bidirectional control line. A high to low transition on this line indicates the begining of a bus transaction. The CPU relinquishes this line when *BUSACKOUT is active low.
*MREQ	Memory Request - Bidirectional control line. Indicates that the address/data bus is holding a memory request. The CPU relinquishes this line when *BUSACKOUT is active low.
B/*W	Byte/Word - Bidirectional control line used to determine if a byte or word is to be transmitted during a bus transaction. For byte transmission, B/*W is high; for word transmission, B/*W is low. The CPU relinquishes this line when *BUSACKOUT is active low.
N/*S	Normal/System Mode - Bidirectional control line. Indicates whether the CPU is operating in the normal mode or system mode. For normal mode, N/*S is high; for system mode, N/*S is low. The CPU relinquishes this line when *BUSACKOUT is active low.
ST0-ST3	Bidirectional Status lines - Used to determine the type of transaction occuring on the bus. The CPU relinquishes these lines when *BUSACKOUT is active low.
SN0-SN3	Bidirectional Segment Select lines - contain the segment number portion of the memory address. The CPU relinquishes these lines when *BUSACKOUT is active low.

BUSACKIN	Bus Acknowledge Input - Bus request input to assume bus control.
BUSACKOUT	Bus Acknowledge Output - When active low indicates that the CPU has relinquished control of the bus in response to a bus request. The Address Bus, Data Bus, Control Bus, Status Lines, Segment Select Lines and Mapping PROM are relinquished so that the J3/J4 Interface board can assume control of these buses and lines.
*I/OREQ	Input/Output Request - Indicates that the CPU is performing input/output operations.
*RESET	Reset - This line indicates that the initialization sequence has commenced. A low is generated on this line during power-up of the system or by pressing the reset button.
*BOOT	Bootstrap - The signal on this line indicates that the bootstrap operation has commenced.
*NVI	Non-Vectored Interrupt - A low on this line indicates that Timer 3 of the 8253 Programmable Interval Timer has made a request to the CPU for a non-vectored interrupt.
*NMI	Non-Maskable Interrupt - A high to low transition on this line requests the CPU for non-maskable interrupt. The J3/J4 Interface board will need to decode the Status lines in order to obtain *NMIACK.
*VI	Vectored Interrupt - A low on this line indicates that a request has been made to the CPU for a vectored interrupt. The request may originate from the J3/J4 Interface board or from the INT output of the 8259 Programmable Interrupt Controller.
*BUSRQ	Bus Request - Active low in order to assume control of the buses after the CPU has generated *BUSACKOUT.
*MEMDIS	Memory Disable - After *BUSACKOUT has been generated and the J3/J4 Interface board assumes control of the buses and Mapping PROM, *MEMDIS can be activated to disable the Mapping PROM.
*SYSINT	System Interrupt - A low on this line initiates a system interrupt via the 8259 Programmable Interrupt Controller.
*COMVI1 *COMVI2	Communication Vectored Interrupts - Interrupt request lines reserved for communication channels. These lines have the same function as the *SYSINT line.

*VINTACK	Vectored Interrupt Acknowledge - Two low going pulses on this line indicate to the 8259 Programmable Interrupt Controller that the CPU has acknowledged a request for a vectored interrupt. If the J3/J4 Interface board requires one low going pulse, then conversion will be required.
CAS0-CAS2	Cascade Lines - If the activated interrupt line is programmed as a slave interrupt controller, the first low pulse of *VINTACK generates the CAS0-CAS2 identification code to enable the slave interrupt controller on the J3/J4 Interface board.
4MHz	4MHz Clock - 50% duty cycle clock derived from the 16MHz Crystal Oscillator.
SCLOCK	System Clock - 50% duty cycle clock, is stretched when the *DRAM is selected or can be stretched by pulling *SWAIT low.
*SRAM	Select J3/J4 RAM - Active when Segment Number = 5 and A15, A14 = 11.
*SWAIT	System Wait - SCLOCK can be stretched by activating *SWAIT.
*WAIT	Wait - A low on this line indicates to the CPU that data transfer is not yet complete.

3.4.3.2 LEVELS & LOADING

LEVELS

All Inputs and Outputs are TTL compatible.

LOADING

Outputs - The Address Bus (A0-A15), Data Bus (D0-D15), Segment Lines (SN0-SN3), Status Lines (ST0-ST3), R/*W, B/*W, N/*S, *DS, *AS and *MREQ are driven from high driving capability chips, LS373 D-Type Transparent Latches, LS245 Bus Transceivers or LS244 Buffers & Line Drivers (IOH = -15 mA, IOL = 24 mA) and SCLOCK is driven from a S163 4-Bit Counter (IOH = -1 mA, IOL = 20 mA). Each of these outputs can drive at least 10 TTL loads on a J3/J4 Interface Board without buffering. The maximum number of loads is not stated as this will depend on the system layout and timing requirements.

The remainder of the outputs are driven from moderate driving capability chips (IOH = -400 uA, IOL = 8 mA). It is suggested, that if more than 3 to 4 TTL loads are to be driven on a J3/J4 Interface Board, these outputs are buffered.

Inputs - The load presented to a J3/J4 Interface Board by inputs like the Address Bus (A0-A15) and Data Bus (D0-D15) are high (250 uA chip loads + I/O port loads + bus line loads) and will require high driving capability buffers or drivers such as the LS244 Buffer and Line Driver or the LS245

If the J3/J4 Interface Board does not need to return an address, the Address Bus (A0-A15) will not require buffering.

The loads presented to the J3/J4 Interface Board by the remainder of the inputs are low and may be driven by standard TTL gates.

3.4.3.3 TIMING

MEMORY READ/WRITE TIMING

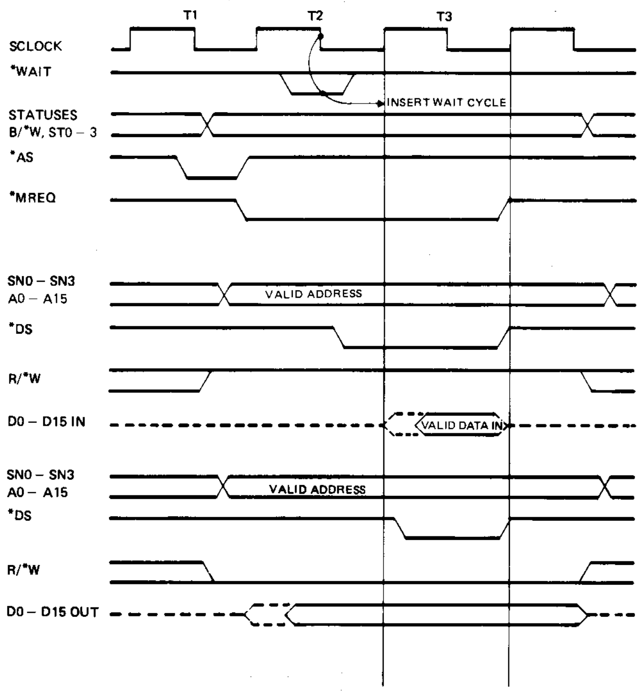


Fig. 3-15 Memory Read/Write Timing

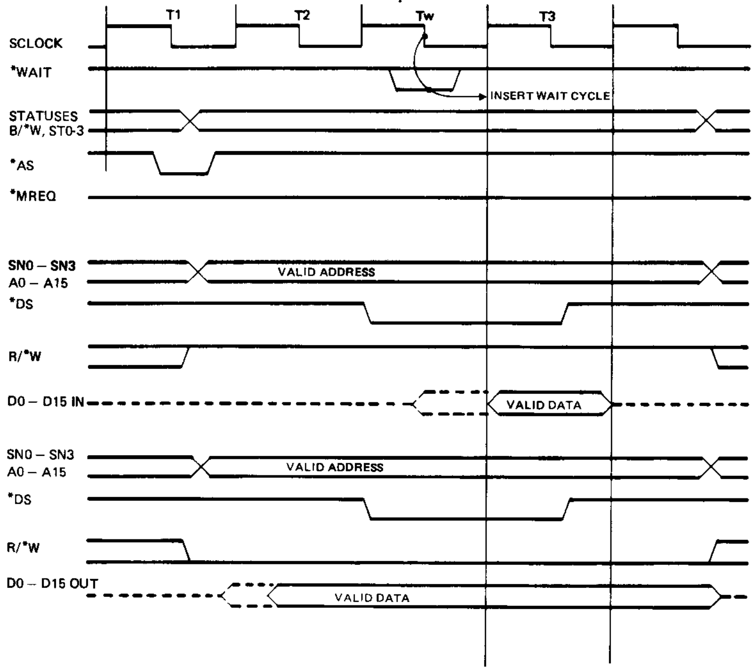
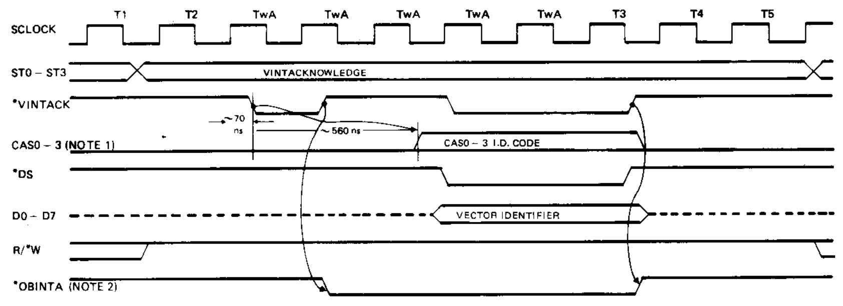


Fig. 3-16 Input/Output Timing

INTERRUPT ACKNOWLEDGE CYCLE TIMING



NOTE: 1 – IF THE ACTIVATED *INT CHANNEL IN 8259 IS PROGRAMMED AS SLAVE *INT CONTROLLER
2 – IF OPTION BOARD NEEDS ONLY ONE LOW PULSE FOR *INTA THEN 74LS74 D FLIP-FLOP IS NEEDED TO OBTAIN *OBINTA

Fig. 3-17 *INTACK Cycle Timing

A.C. ELECTRICAL CHARACTERISTICS

NO	SYMBOL	PARAMETER	DELAY		
			MAX	TYP	MIN
1	Tcc	CLOCK CYCLE TIME		250ns	
2	Tpd	CLOCK PHASE DIFFERENCE	25ns		
3	Tdc	CLOCK ↑ TO SEGMENT NUMBER VADID	175ns		
4	Tdc (ASf)	CLOCK ↑ TO *AS ↓ DELAY	125ns		
5	Tdc (A)	CLOCK ↑ TO ADDRESS VALID	170ns		
6	Tdc (S)	CLOCK ↑ TO STATUS VALID	155ns		
7	Tdc (DW)	CLOCK ↑ TO WRITE DATA VALID	135ns		
8	Tdc (DSW)	CLOCK ↓ TO *DS (WRITE)↓DELAY	140ns		
9	TwDSW	*DS (WRITE) WIDTH (LOW)			185ns
10	Tdc (DSf)	CLOCK ↓ TO *DS (I/O) ↓ DELAY	165ns		
11	TwDS	*DS (I/O) WIDTH (LOW)			410ns
12	Tdc (DSR)	CLOCK ↑ TO *DS (READ) ↓ DELAY	165ns		
13	TwDSR	*DS (READ) WIDTH (LOW)			275ns
14	TsDR (C)	READ DATA TO CLOCK ↓ SET UP			20ns
15	Tds (AS)	STATUS VALID TO *AS ↑ DELAY			50ns
16	TdDSi	*DS (I/O) ↓ TO READ DATA VALID			300ns
17	TsW (C)	*WAIT TO CLOCK ↓ SET UP TIME			30ns
18	ThW (C)	*WAIT TO CLOCK ↓ HOLD TIME			30ns
19	TsBRQ (C)	*BUSRQ TO CLOCK ↑ SET UP TIME			70ns
20	ThBRQ (C)	*BUSRQ TO CLOCK ↑ HOLD TIME			30ns

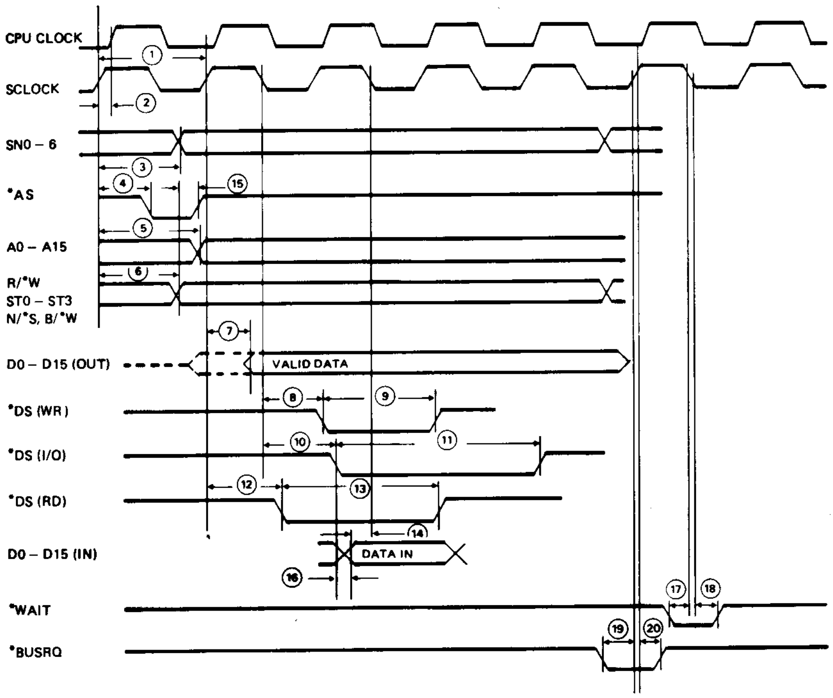


Fig. 3-18 A.C. Characteristics Waveforms

4. POWER SUPPLY DISTRIBUTION

ABOUT THIS CHAPTER

This chapter deals with the M20 power supply unit characteristics. It lists the maximum power available and the power requirements.

CONTENTS

4. POWER SUPPLY DISTRIBUTION

4.1 MAXIMUM POWER AVAILABLE

The following lists the maximum power available from the M20 Power Supply Unit.

VOLTAGE	TOLERANCE	CONTINUOUS CURRENT
+ 5V	5%	3.3A min, 9.0A max
+12V	3%	2.0A min, 5.6A max
-12V	5%	-- 0.7A max

4.2 M20 POWER REQUIREMENTS

The following lists the nominal power supply requirements for the various boards and units of the M20.

Board or Unit	+5V	+12V	-12V
Motherboard	3.51	0.585	0.030
Keyboard	1.0	-	-
Diskette Drive	0.75	1.15	-
OPE		1.85, 50ms start up	
Hard Disk Unit	1.2	1.8	-
OPE		4.20, 6-8s start up 2.80, when stepping	
Hard Disk Unit	1.0	1.50	
Seagate		3.30, 15s start up	
Hard Disk Controller	2.5	-	0.020
Display Monochrome	-	1.5	-
Memory Board Monochrome	0.045	0.60, active 0.06,standby	0.0032, active 0.0016,standby
Memory Board Colour	0.245	0.60, active 0.06,standby	0.0032, active 0.0016,standby
IEEE 488 Interface	1.4	-	-
TWIN RS-232-C Interface	1.0	0.075	0.075