NCR

Personal Computer

PC4i Technical Reference Manual

GW-BASIC is a trademark of Microsoft Corporation

January 1986

It is the policy of NCR Corporation to improve products as new technology, components, software, and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world.In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult your dealer of NCR office for information that is applicable and current.

INTRODUCTION

YOUR NCR PERSONAL COMPUTER MODEL 4i AND THIS MANUAL

Your NCR PERSONAL COMPUTER MODEL 4i is a powerful business/professional personal computer. It is designed to act not only as an independent, self-sufficient system, but also as a control center for systems fulfilling a wide range of business, scientific and technological requirements.

Essential electronics are contained on the controller board. Where possible, LSI components have been used, providing compactness and a high degree of reliability. The microprocessor is the 8088, which can be supplemented by the 8087 co-processor, if required. With the additional support of Direct Memory Access, high processing speeds can be achieved. 256 Kilobyte of random access memory are included in the standard version, over and above video RAM.

The NCR PERSONAL COMPUTER MODEL 4i is equipped with Centronics and RS-232-C interfaces as standard. These are accessed at the rear of the computer without any dismantling. Full access to the system bus is available at the internal expansion card assembly. This assembly can accomodate up to seven cards to expand the features of your system. One additional monitor can be connected via an expansion card, so that the computer can drive a monochrome and a color display simultaneously.

The NCR PERSONAL COMPUTER MODEL 4i is operationally compatible with a number of other widely used personal computers. This provides for portability of a large number of software applications and hardware extensions between your NCR PERSONAL COMPUTER 4i and other personal computers you may have occasion to use. In addition, your NCR PERSONAL COMPUTER 4i offers the advantages of single cabinet design, an extended keyboard, and an excellent high-resolution display in conjunction with a color CRT.

This Manual, consisting of two volumes, is intended for designers, system integrators, and programmers who require detailed information about the construction and operation of the NCR PERSONAL COMPUTER MODEL 4i. Volume 1 includes information about the I/O bus, signal levels and timing, power requirements, and pin assignments, not only of the NCR PERSONAL COMPUTER MODEL 4i itself, but also of the kits available from NCR. In addition, details are given about low-level software control of hardware functions. Volume 2 provides full schematics. Assembly language listings of the NCR ROM BIOS are also available as a separate document.

The NCR PERSONAL COMPUTER MODEL 4i is optionally available in a version for use with the Videotex information and communication system. Videotex is administered under the auspices of the post and telecommunications authorities of a number of countries. The Videotex version of the NCR PERSONAL COMPUTER MODEL 4i is described in a Chapter of its own in this Manual.

With the help of the information contained in this Manual, you may be intending to make hardware or software changes to the NCR PERSONAL COMPUTER 4i for specialized use. Because of the complex nature of computer technology, we must stress that NCR cannot accept responsibility for problems arising from such changes. At the same time, we are constantly endeavoring to maintain the quality of information provided in this Technical Reference Manual. Therefore, if you find that there is a particular aspect of information not already provided in this Manual, or if you want to let us know how useful you find it, please ask your supplier to forward your comments to us.

NCR PERSONAL COMPUTER MODEL 4i

TECHNICAL REFERENCE MANUAL

CONTENTS

CHAPTER 3: The Operating System

CHAPTER 4: Printers and Communications

CHAPTER 5: Inhouse DLC Interface

CHAPTER 6: Disk Storage

CHAPTER 7: The Screen Display

CHAPTER 8: The Keyboard

CHAPTER 9: The Loudspeaker

CHAPTER 10: Videotex

System Overview

MODELS AND OPTIONS

The NCR PERSONAL COMPUTER consists of a main display unit and a keyboard. The computer is available in different models and has a variety of options that can be added to each model. Expansion boards from non-NCR ("third party") manufacturers can be used with the NCR PERSONAL COMPUTER, as well as expansion boards offered by NCR.

The standard display unit is made up of the following parts:

* Cabinet
* Power supply
* One or two flexible disk drive(s) or one flexible disk drive and one hard disk drive
* 12-inch monochrome CRT display or 12-inch color CRT display
* Main Processor Board with:
    * 8088 Microprocessor
    * 256 KB Random Access Memory (RAM)
    * 16 KB Read Only Memory (ROM)
    * RS-232-C Serial Interface
    * Centronics Parallel Interface (for printer)
    * Plug-in board option expansion slots

* Fixed disk drive controller plug-in board for units with a hard disk drive.

* Video controller plug-in board.

A block diagram of the NCR PERSONAL COMPUTER hardware is shown in Figure 1.1.

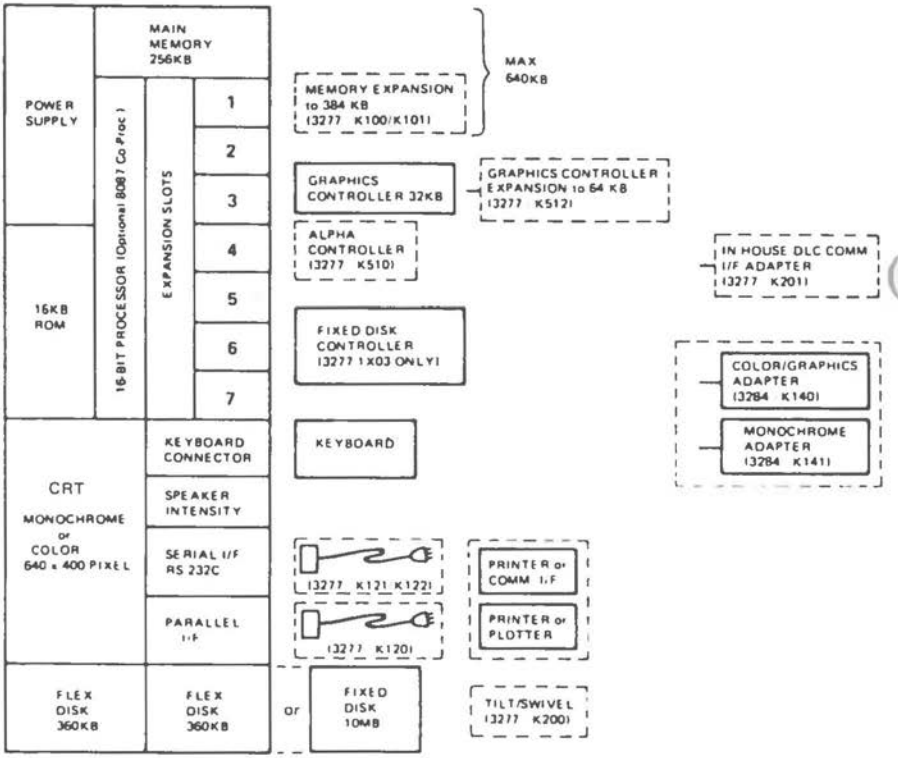


Figure 1.1 Hardware Block Diagram

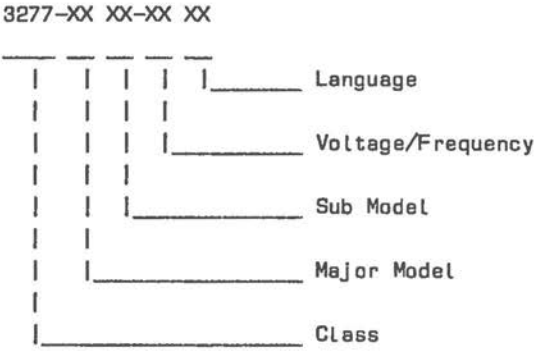
The NCR PERSONAL COMPUTER is available in various models. The standard internal RAM memory size for all models is 256 KB. The models currently available are:

* A 12-inch monochrome display, one flexible disk drive and 256K memory
* A 12-inch monochrome display, two flexible disk drives and 256K memory
* A 12-inch monochrome display, one flexible disk drive and one hard disk drive, and 256K memory
* A 12-inch color display, one flexible disk drive and 256K memory
* A 12-inch color display with two flexible disk drives and 256K memory
* A 12-inch color display with one flexible disk drive and one hard disk drive, and 256K memory
* A 12-inch color display with Videotex, one flexible disk drive and 256K memory
* A 12-inch color display with Videotex, two flexible disk drives and 256K memory
* A 12-inch color display with Videotex, one flexible disk drive and one hard disk drive, and 256K memory

The Videotex systems are dual purpose systems to be used either as a Videotex system, or as a normal personal computer. These systems are tailored to meet individual country requirements, for example, Bildschirmtext for Germany, Teletel for France, and Prestel for Great Britain.

MODEL IDENTIFICATION

The model number of the NCR PERSONAL COMPUTER is stated on the plate at the back of the unit. The meaning of the model number is shown below:



The NCR 3277 model numbers are defined in Figure 1.2.

NCR 3277				
Major Model	11	Monochrome CRT		
	12	Color CRT		
	13	Videotex CRT (Color)		
Sub Model	01	Single Flexible Disk Drive		
	02	Dual Flexible Disk Drives		
	03	Hard and Flexible Disk Drives		
Voltage/ Frequency	71	120 Volts, 50/60 Hz.		
	73	220/230/240 Volts, 50/60 Hz.		
Language	00	U.S. English	09	Swedish
	01	Int. English	10	Danish
	02	U.K. English	11	Norwegian
	03	Spanish	12	Italian
	05	French	20	Finnish
	06	German		

Figure 1.2 Model Number Definition

SYSTEM COMPONENTS

This section introduces elements of the NCR PERSONAL COMPUTER shared by all models. Figure 1.3 provides an easy-to-follow diagram illustrating the most important components on the main processor board and those external functions with which the main processor board interfaces.

The CLOCK is the key to co-ordination of the entire system. This unit consists of a crystal supplying a signal of frequency 14.31 MHz to a Clock Controller. The 8284A Clock Controller divides the frequency of this input signal, in order to provide a suitable timing signal (4.77 MHz) for the main system microprocessor (CPU), as well as timing signals for all the other system components.

The TIMER is an integrated circuit which can be programmmed to produce a variety of signals at various intervals. Accuracy is ensured by the Timer deriving its synchronizing signals, by way of suitable frequency division, from the system clock. You can use the Timer for running a system clock of your own, that is, a clock you can watch on the screen. In addition, the Timer can be used to provide the frequencies required if you wish to play music on the computer. The Timer is also used for refreshing random access memory (see below).

The CENTRAL PROCESSING UNIT (CPU), often simply called "microprocessor", is widely acknowledged as the most "influential" of the integrated circuits. It is the CPU which passes on instructions supplied by the programmer to the other units in the system. Optionally, the CPU can be supported by the 8087 Co-Processor. The Co-Processor is capable of performing, in response to simple instructions, floating-point calculations which would otherwise require extensive programming of the CPU.

The NCR PERSONAL COMPUTER makes use of two types of memory: read only memory (ROM), and a large random access memory (RAM) of at least 256 KB.

READ ONLY MEMORY contains instructions for the CPU which enable the latter to get the system started, as well as a "library" of short, but useful, programs known under the collective term BIOS (Basic Input/Output System). It is the task of the BIOS to constantly maintain communication between the CPU and the other system units.

RANDOM ACCESS MEMORY can likewise contain program instructions for the CPU, but it is also used for storing data being processed by the the program. The random access memory of your NCR PERSONAL COMPUTER is "dynamic", which means that its contents can be changed very quickly, thus providing for fast processing. This also means that memory must be "refreshed" at fixed intervals, to prevent its contents from disappearing. Fortunately, you need not normally be concerned about providing refresh cycles, as this is taken care of by the Timer.

The DIRECT MEMORY ACCESS (DMA) CONTROLLER is used for the transfer of large blocks of data to and from memory. DMA is particularly useful for reading data from or writing data to flexible disk. The CPU does not then have to specify a memory address for each byte read or written. Instead, all the CPU has to tell the DMA Controller is 1) that data is to be transferred between memory and flexible disk, 2) the address of the first byte in memory to be accessed, and 3) the number of bytes to be transferred.

The I/O CONTROLLER is used for reading the keyboard and turning the loudspeaker on and off. In addition, it keeps a note of the settings of system configuration switches, so that programs can detect the NCR PERSONAL COMPUTER model being used.

The INTERRUPT CONTROLLER provides fast notification of the CPU in the event that a system unit requires urgent attention. This is especially important for the Timer and keyboard: the Timer, because an accurate clock (whether your own or provided by the operating system) must not miscount the number of ticks; the keyboard, because this is the most expedient way for the user to intervene during execution of a program.

Additional equipment can be interfaced via one or more of the seven 62-pin I/O connectors inside the computer cabinet. One of these connectors is already occupied by the board containing the controller circuitry for the screen display.

With some personal computers, the three interfaces marked with asterisks in Figure 1.3 (flexible disk, parallel printer, RS-232-C) each require the use of one of the six remaining connectors. The NCR PERSONAL COMPUTER has the advantage that these three functions are interfaced separately to the main processor board, thus making these connectors available to other options.

The system BUS provides the lines by which communication between the units of the system can take place. The bus really consists of three busses. The address bus conveys memory or I/O map addresses by which memory bytes or external devices can be accessed. Data is passed as parallel signals on the data bus. Control bus is a collective term for those signals which otherwise influence data flow within the system, for example, signals which determine whether the direction of data flow is read or write, and whether transfer is to/from memory or an I/O device. Your NCR PERSONAL COMPUTER includes a BUS CONTROLLER which relieves the CPU of such bus arbitration duties, thus enabling the 8088 microprocessor to operate in its "maximum mode".

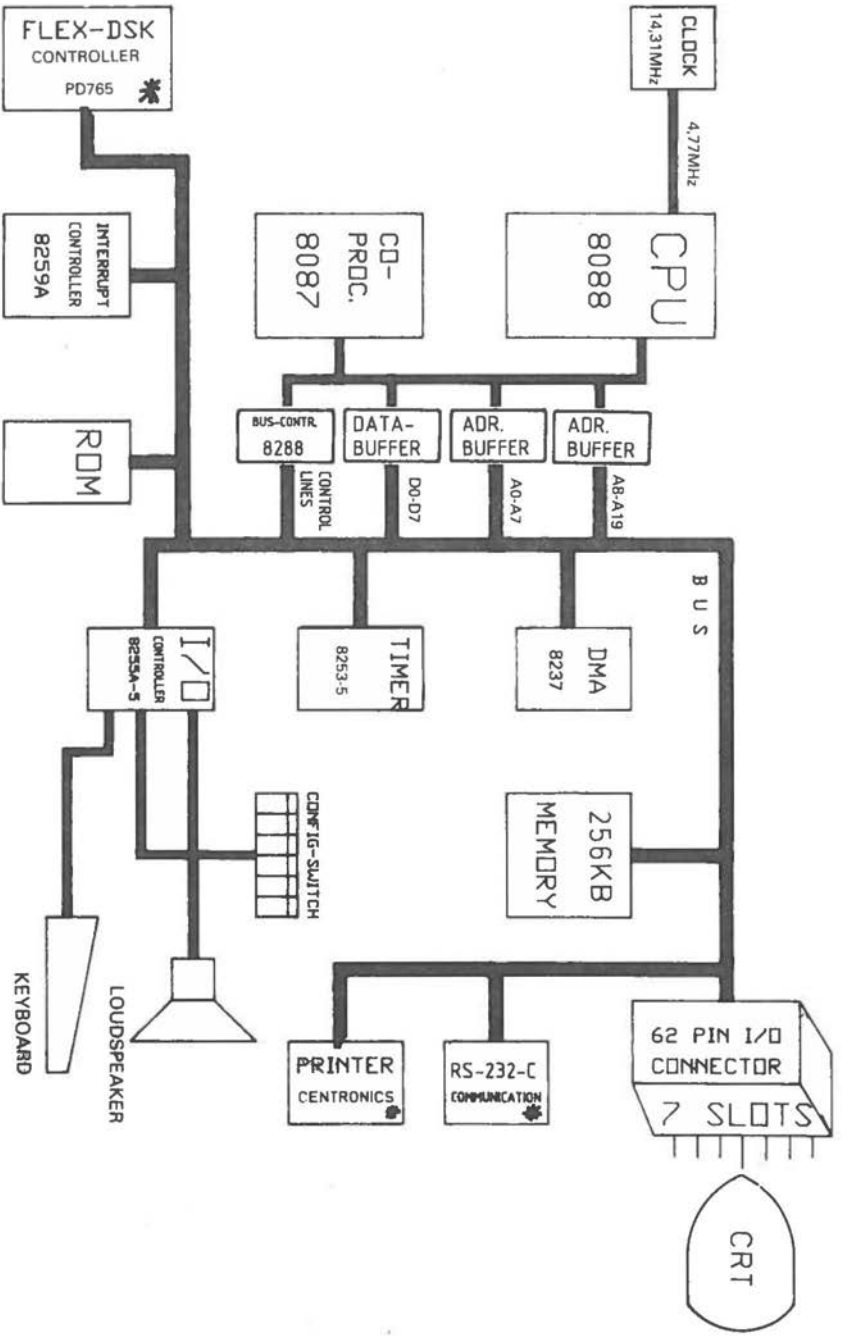


Figure 1.3 System Overview (1)

Figure 1.4 provides a more detailed view of the system. Busses are shown separately. Furthermore, a number of additional aspects are introduced.

NMI LOGIC determines what source of interrupt may be allowed to issue the CPU non-maskable interrupt (NMI). Possible sources are errors on I/O and memory transfers, and co-processor errors.

Bus transfer over a period of more than one clock unit is enabled by the WAIT STATE LOGIC.

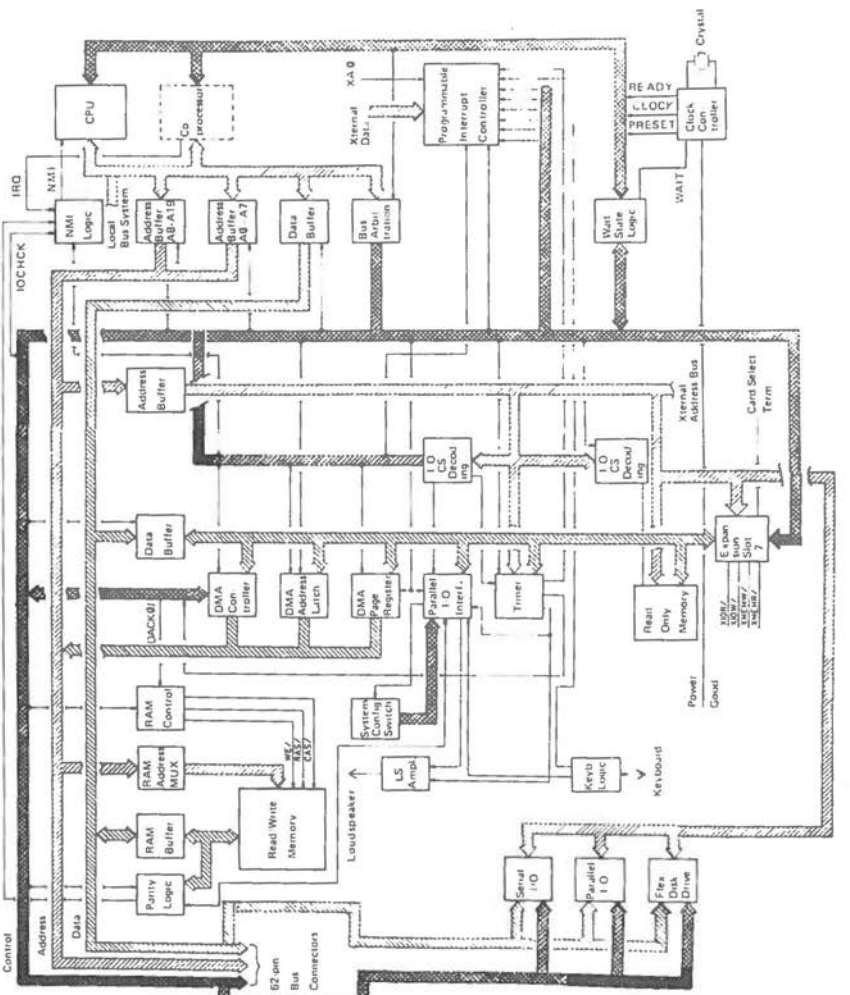
CHIP SELECT (CS) LOGIC converts address information into CS signals for memory and I/O interface integrated circuits.

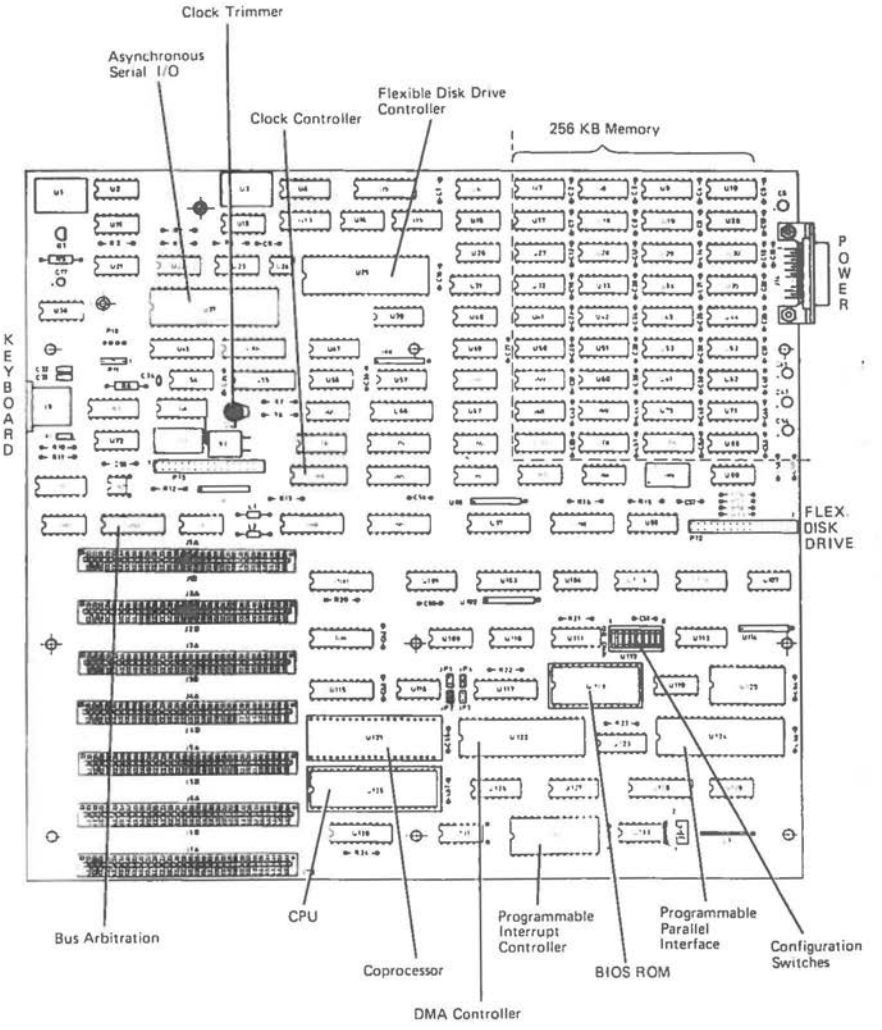
A DMA PAGE REGISTER allows DMA to address more than 64 KB of the memory address area.

PARITY CHECKING is performed during RAM access.

Finally, Figure 1.5 shows exactly where important integrated circuits are situated on the main processor board.

NOTE: Your NCR PERSONAL COMPUTER may include the Alternate Main Processor Board which is described in a separate section at the end of this Chapter.


CPU
Co Processor
XAO
External Data
Programmable Interrupt Controller
Crystal
READY
CLOCK
PRESET
Clock Controller
WAIT
Wait State Logic
IRQ
NMI
NMI Logic
Local Bus System
Address Buffer AB-A19
Address Buffer A0-A7
Data Buffer
Bus Arbitration
PCK
IOCHCK
Address Buffer
I/O CS Decoding
I/O CS Decoding
External Address Bus
Card Select Term
Data Buffer
DMA Controller
DMA Address Latch
DMA Page Register
Parallel I/O Interf.
Timer
Read Only Memory
Expansion Slot 7
I/O R/W
I/O R/W
MEM R/W
MEM R/W
DACK#
RAM Control
RAM Address MUX
RAM Buffer
Parity Logic
Read Write Memory
WE/
RAS/
CAS/
System Config Switch
LS Amp
Loudspeaker
Keyb Logic
Keyboard
Power Good
Serial I/O
Parallel I/O
Flex Disk Drive
Control
Address
Data
62-pin Bus Connectors


Clock Trimmer
Asynchronous Serial I/O
Clock Controller
Flexible Disk Drive Controller
256 KB Memory
POWER
KEYBOARD
FLEX DISK DRIVE
Bus Arbitration
CPU
Coprocessor
DMA Controller
Programmable Interrupt Controller
BIOS ROM
Programmable Parallel Interface
Configuration Switches

SYSTEM CONNECTION

The NCR PERSONAL COMPUTER can accomodate up to seven expansion boards (including the CRT Controller Board). For this purpose the main processor board, accessed after removal of the rack assembly from the back of the computer, is provided with seven expansion slots.

I/O SIGNALS

Each expansion slot consists of a 62-pin connector providing access to TTL-compatible address, data, and a number of control lines. The system connection is identical for each slot, with the exception of the lowermost slot (7), which reads a Card Select signal from the board Figure 1.6 summarizes the connector pin designations. The following descriptions provide explanatory notes to the control lines.

RESET DRIVE Output
This signal is issued in response to a system reset. It is synchronized to the falling clock edge.

IRQ2-IRQ7 Input
Input signals to the Interrupt Request Register of the 8259A Programmable Interrupt Controller. The signal must remain high until the interrupt is acknowledged by the microprocessor.

DRQ1-DRQ3 Input
DMA request lines to the 8237 DMA Controller. These lines can be programmed at the DMA Controller to active low or active high (the BIOS software specifies active low).

DACK0-DACK3 Output
DMA request acknowledge lines. These lines can be programmed at the DMA Controller to active low or active high (the BIOS software specifies active low).

Note that DACK0 is available at the connector. This is because RAM refresh (serviced by DMA Channel 0), must also be available to memory extensions not accomodated on the main processor board. The corresponding data request signal is supplied by Timer 1 on the main processor board.

MEMW/ Output
Determines that data on the data bus is for writing to a selected address in random access memory. This signal can be asserted by microprocessor or DMA Controller.

MEMR/ Output
Determines that RAM data from a selected address is to be placed on the data bus. This signal can be asserted by microprocessor or DMA Controller.

IOW/ Output
As for MEMW/, except that the selected address is to be understood as an address in the I/O map.

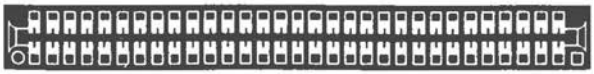
IOR/ Output
As for MEMR/, except that the selected address is to be understood as an address in the I/O map.

AEN Output
This signal is used by the DMA Controller to indicate that it is in control of the address and data busses, as well as the MEMW/, MEMR/, IOW/, and IOR/ lines. This bus reservation is especially important, as the DMA Controller needs the data bus for the 8 most significant bits of the RAM address to be accessed.

TC Output
The Terminal Count signal supplied by the DMA Controller when he programmed number of bytes has been transferred.

IO CH RDY Input
This signal is normally high. It can be pulled low (for a maximum of 10 clocks) by a device which requires more than one machine cycle for the transfer of data.

A1 (seen from expansion board)



	B	A
1	GND	IO CHECK/
2	RESET DRIVE	D7
3	+ 5V	D6
4	IRQ2	D5
5	- 5V	D4
6	DRQ2	D3
7	- 12V	D2
8	CARD SELECT/	D1
9	+ 12V	D0
10	GND	IO CH RDY
11	MEMW/	AEN
12	MEMR/	A19
13	IOW/	A18
14	IOR/	A17
15	DACK3	A16
16	DRQ3	A15
17	DACK1	A14
18	DRQ1	A13
19	DACK0	A12
20	CLK	A11
21	IRQ7	A10
22	IRQ6	A9
23	IRQ5	A8
24	IRQ4	A7
25	IRQ3	A6
26	DACK2	A5
27	TC	A4
28	ALE	A3
29	+ 5V	A2
30	OSC	A1
31	GND	A0

Figure 1.6 System Connector Pin Designations

IO CHECK/ Input
This signal active indicates an I/O or RAM parity error.

ALE Input
Address bus lock signal, whose falling edge cuts off the microprocessor from the address bus. This signal is normally used by the 8288 Bus Controller.

OSC
The system crystal oscillator signal of frequency 14.31818 MHz. This signal has equal high and low periods.

CLK
The divide-by-three output from the Clock Controller of frequency 4.77 Mhz, as supplied to the microprocessor and other system components. The high portion of the signal is 33 %.

CARD SELECT/
Connected to slot 7, this signal must be supplied by any adapter connected to expansion slot 7, otherwise the adapter cannot be addressed properly. This connection is disregarded at the other six expansion slots.

RACK ASSEMBLY
The shape of the Card Rack Assembly imposes certain restrictions on the size of the option boards that may be installed.

The option boards must conform to the following dimensions:

3 Boards 334 mm x 107 mm (13.1 in. x 4.2 in.)

3 Boards 296 mm x 107 mm (11.6 in. x 4.2 in.)

1 Board 254 mm x 107 mm (10.0 in. x 4.2 in.)

CONFIGURATION SWITCHES

The main processor board includes eight configuration switches which require setting as illustrated in Figure 1.8, before the computer is used for the first time, and after any of the hardware features concerned has been changed.

These switches can be read by software. This is explained in the description of the Parallel Input/Output Interface in Chapter 2.

One or more jumper switches are present on the main processor board (see Figure 1.7). The jumper switch block consists of either J5 or the three jumpers J1, J2, and J3. The significance of the jumper settings:

JP1 closed —> Enable flexible disk drives
JP2 closed —> Enable standard serial I/O
JP3 closed —> Enable parallel interface

JP5 closed —> Enable standard serial I/O

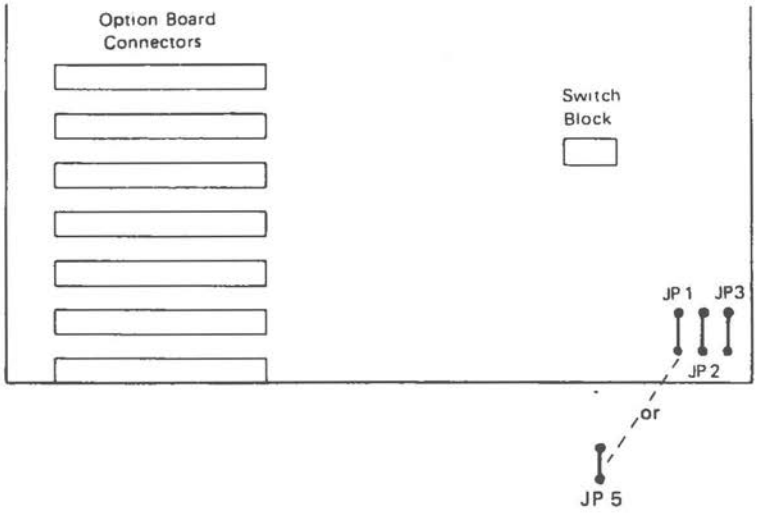


Figure 1.7 System Configuration Switch Locations

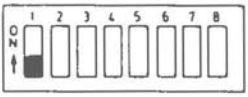
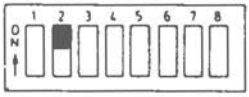
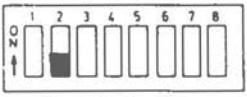
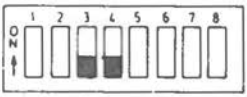
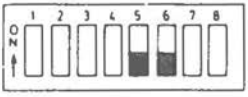
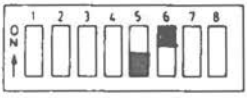
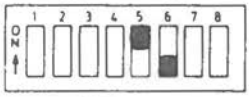
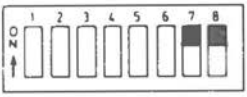
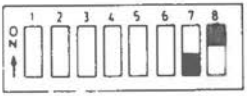
Normal Operation	Switch 1 Off	
Coprocessor Not Installed	Switch 2 On	
Coprocessor Installed	Switch 2 Off	
Memory Capacity 256 KB	Switch 3 Off Switch 4 Off	
Alpha Controller	Switch 5 Off Switch 6 Off	
Graphics Controller (40 x 25)	Switch 5 Off Switch 6 On	
Graphics Controller (80 x 25)	Switch 5 On Switch 6 Off	
1 Flexible Disk Drive	Switch 7 On Switch 8 On	
2 Flexible Disk Drives	Switch 7 Off Switch 8 On	

Figure 1.8 System Configuration Switch Settings

MEMORY EXPANSION

The NCR PERSONAL COMPUTER is supplied with 256 KB random access memory on the main processor board. To upgrade memory capacity, two additional kits are available:

* K101
  A memory expansion board with 64 KB of memory, starting at machine location 40000H. In addition, this kit contains sockets for 5 banks each of 9 integrated circuits, so that memory can be further expanded using up to 5 kits K100.

* K100
  A set of integrated circuits for insertion on the memory expansion board K101. K100 upgrades the system memory capacity by 64 KB. Installing the maximum 5 kits K100 creates a total system memory capacity of 640 KB, that is up to, but not including, the machine location 0A0000H. (Address area from this location upwards is reserved for video display memory and the NCR PERSONAL COMPUTER ROM BIOS.)

Memory banks should be populated in ascending address sequence. Configuration switches on the memory expansion board must be set to reflect the amount of expansion memory (see Figure 1.10).

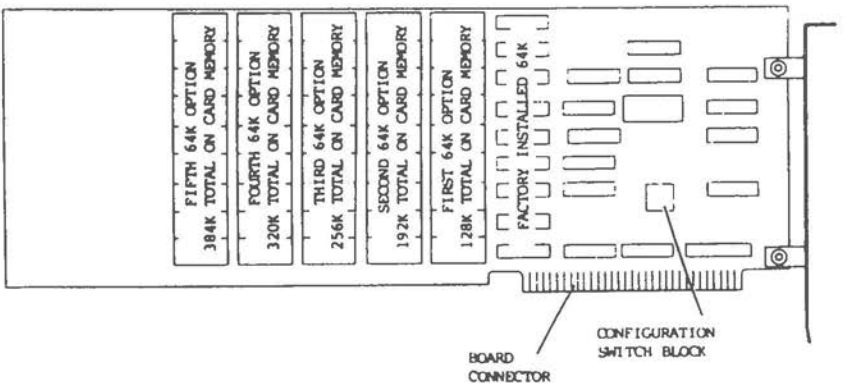


Figure 1.9 Memory Expansion Board (K101)

Memory:		Memory configuration switches:			
Expansion board	Total in system	SW1-1	SW1-2	SW1-3	SW1-4
64 KB	320 KB	ON	ON	ON	ON
128 KB	384 KB	OFF	ON	ON	ON
192 KB	448 KB	ON	OFF	ON	ON
256 KB	512 KB	OFF	OFF	ON	ON
320 KB	576 KB	ON	ON	OFF	ON
384 KB	640 KB	OFF	ON	OFF	ON

Figure 1.10 Memory Expansion Configuration Switches

NOTE: Systems with the Alternate Main Processor Board allow memory expansion up to the maximum 640 KB on the main processor board itself. A separate expansion board is not required.

MEMORY MAP

Figure 1.11 sets out the memory map of the NCR PERSONAL COMPUTER. The following memory areas are discussed in detail in further Chapters of this Manual:

Interrupt Vector	- Chapter 3
BIOS Data Areas	- Chapter 3
Screen display memory	- Chapter 7
ROM BIOS	- Chapter 3

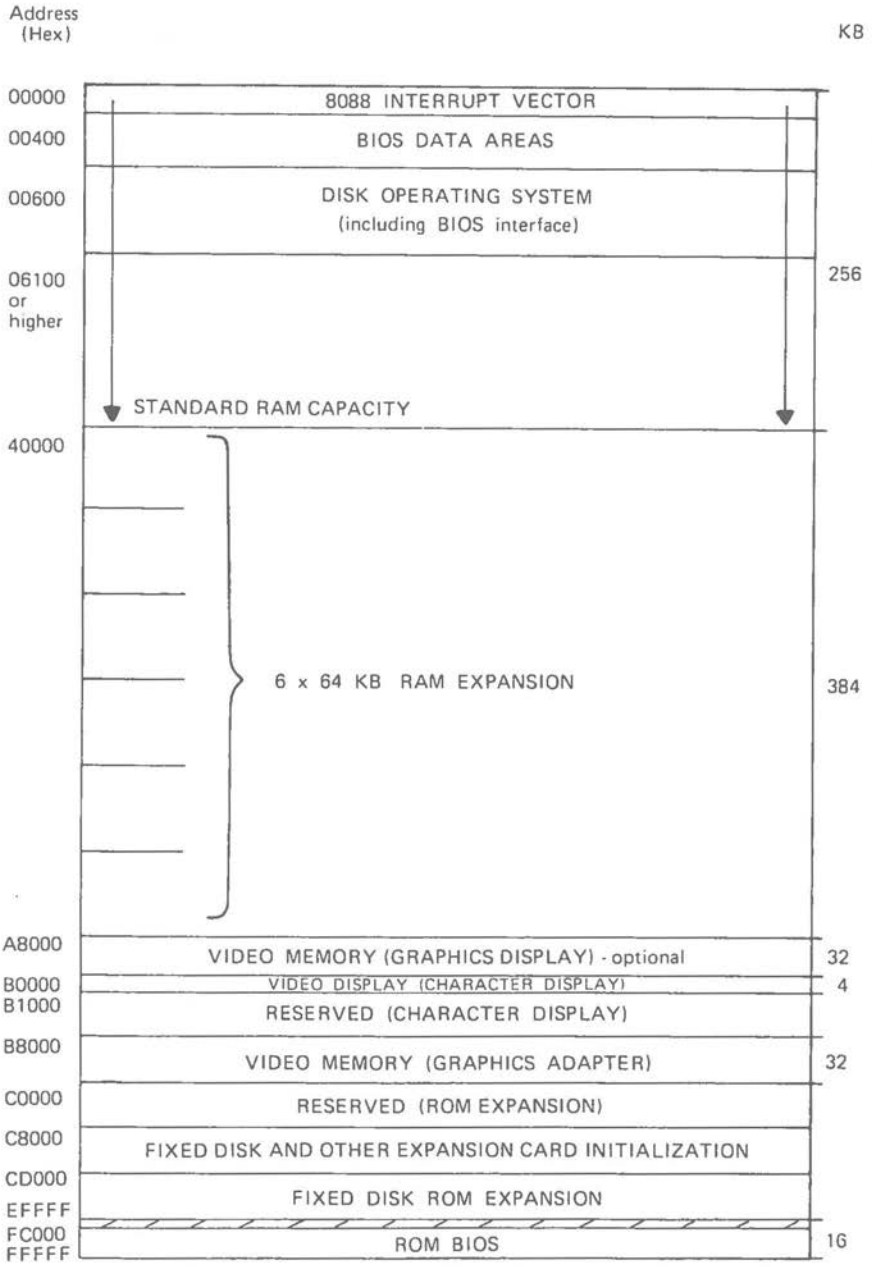


Figure 1.11 Memory Map

I/O MAP

Figure 1.12 illustrates the use of port addresses in the I/O map of the NCR PERSONAL COMPUTER. Details of system use are to be found in the appropriate interface and system function descriptions in this Manual.

Port (hex)	System Use
00-0F	DMA Controller (8237)
20-21	Programmable Interrupt Controller (8259)
40-43	Programmable Interval Timer (8253)
60-63	Parallel Input/Output Interface (8255)
80-83	DMA Page Select Register
A0	Non-maskable Interrupt (NMI) Register
C0-CF	Reserved
E0-EF	Reserved
100-1FF	User available
200-20F	Reserved (games adapter)
210-217	Reserved (expansion adapter)
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous Serial I/O (2)
300-31F	Reserved
320-32F	Fixed Disk Controller
378-37F	Parallel Printer
380-38F	DLC Interface
3A0-3AF	Reserved
3B0-3BF	Character Video Controller
3C0-3CF	Reserved
3D0-3DF	Graphics Video Controller
3E0-3E7	Reserved
3F0-3F7	Flexible Disk Controller
3F8-3FF	Asynchronous Serial I/O (1)

Figure 1.12 I/O Map

POWER SUPPLY

Two types of power supply are installed in the NCR PERSONAl COMPUTER, according to whether the internal CRT is monochrome or color.

MONOCHROME SYSTEMS

The power supply used in conjunction with a monochrome CRT has the following characteristics:

* Primary switched
* Fuse in primary
* Strappable for the nominal mains supplies of 120 V and 240 V (total permissible range 107 V to 257 V)
* DC outputs (max. power output - 133 W):

  + 5 V 15 A max
  + 12 V 5.5 A max
  - 5 V 0.3 A max
  - 12 V 0.25 A max

* Current limiting and overload protection on DC outputs. Maximum short circuit current is 3.0 A on all DC outputs.
* Power is good after 500 ms at latest
* Maximum inconsequential supply deviation:

  +/- 50 % for 1/2 cycle once every 10 seconds

Figure 1.13 illustrates connector designations.

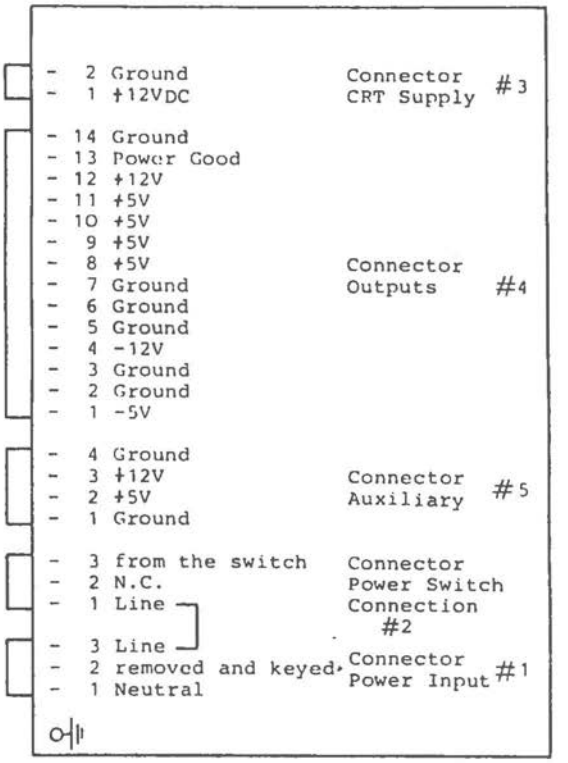


Figure 1.13 Power Supply Connections (Monochrome Systems)

COLOR SYSTEMS

The power supply used in conjunction with a color CRT has the following characteristics:

* Primary switched
* Fuse in primary
* No strapping necessary.
* Input range: 90 V - 264 V

* DC outputs (max. power output - 115.5 W):

  + 5 V 15 A max
  + 12 V 4.0 A max
  - 5 V 0.3 A max
  - 12 V 0.25 A max
  + 85 V 0.82 A max

* Current limiting and overload protection on DC outputs. Maximum short circuit current is 3.0 A on all DC outputs.

* Power is good after 500 ms at latest

* Maximum inconsequential supply deviation:

  +/- 50 % for 1/2 cycle once every 10 seconds

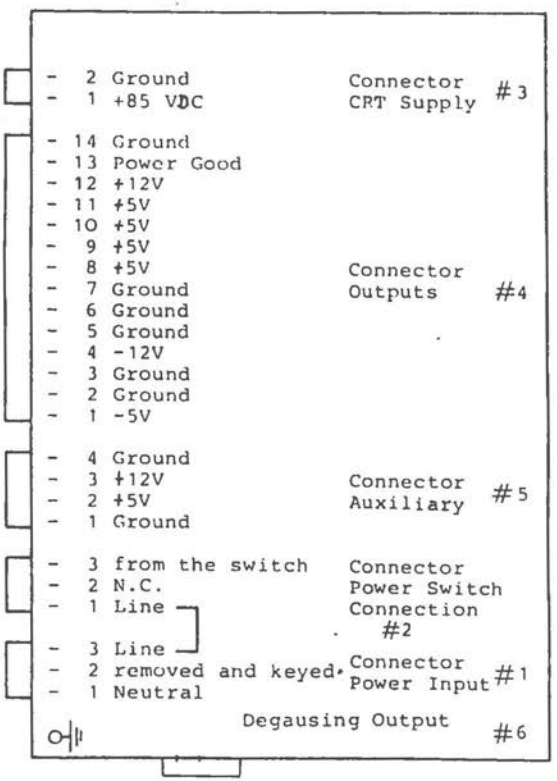


Figure 1.14 Power Supply Connections (Color Systems)

SPECIFICATIONS

Technical specifications for the NCR PERSONAL COMPUTER are given in Figure 1.15.

SIZE	Height		Width	Depth
Processor	368 mm		460 mm	440 mm
Keyboard	19 mm front	34 mm back	520 mm	200 mm
WEIGHT				
Processor	22.68 kg			
Keyboard	2.04 kg			
VOLTAGE (NOMINAL)	Strappable: 120/220/230/240 range 107-257			
FREQUENCY	60 Hz (49-61 Hz)			
POWER REQUIREMENT	Monochrome: 150 W Color: 200 W			
CABLE LENGTHS				
Power	3 m			
Keyboard	1 m (coiled, extendable)			
ENVIRONMENT	Operating		Power Off	Transit
Temperature (C)	10-32		10-43	-40 to +60
Humidity (%)	20-80		20-80	20-80
PRODUCT SAFETY				
USA	UL 478			
Canada	CSA 22.2			
Europe	IEC 380			
Germany	VDE 0806, GS label granted by TÜV			
RADIO INTERFERENCE				
USA	FCC Docket No. 20780, Class B			
Germany	VDE 0871, Class A			

Figure 1.15 Technical Specifications

ALTERNATE MAIN PROCESSOR BOARD

It is possible that your NCR PERSONAL COMPUTER is supplied with an Alternate Main Processor Board. This board makes more extended use of VLSI components. The layout of this board is shown in Figure 1.16.

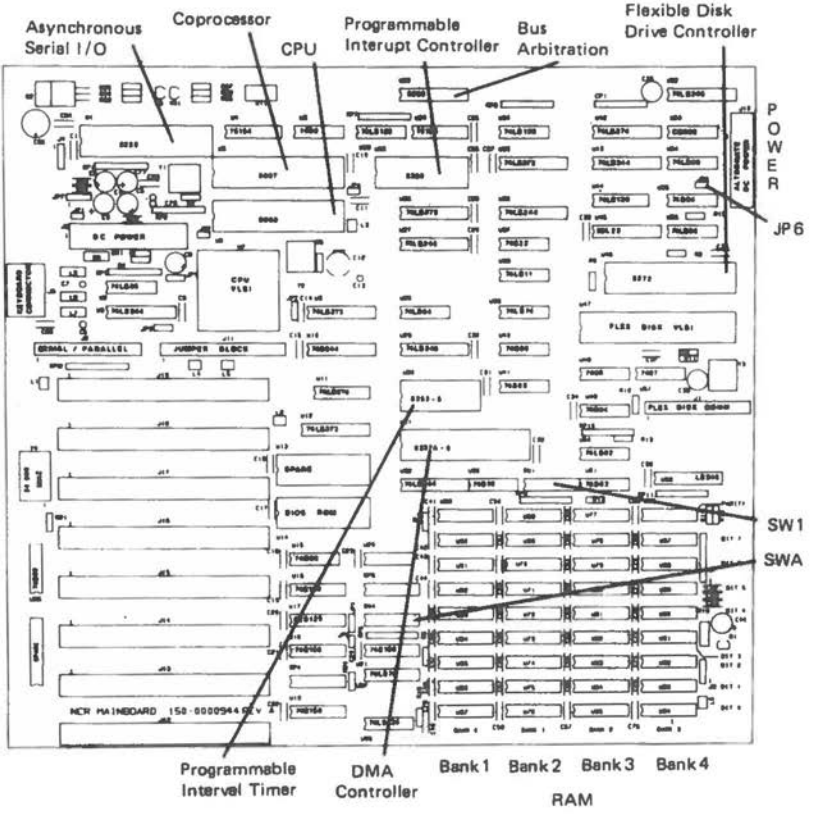


Figure 1.16 Alternate Main Processor Board

The two blocks of configuration switches are explained in Figures 1.17 and 1.18.

Jumper JP6 is normally installed, but must be removed if the optional controller for a 1.2 MB flexible disk drive is installed. This disables the flexible disk controller IC on the main processor board.

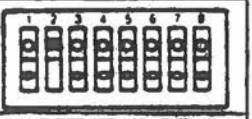
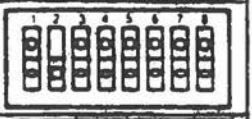
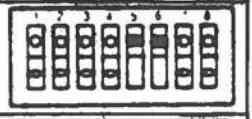
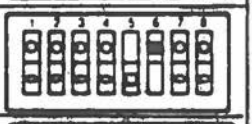
Coprocessor Not Installed	Switch 2 On	
Coprocessor Installed	Switch 2 Off	
Graphic Adapter 40 x 25	Switch 5 On Switch 6 On	
Graphic Adapter 80 x 25	Switch 5 Off Switch 6 On	
Alpha Adapter	Switch 5 On Switch 6 Off	

Figure 1.17 Configuration Switch SW1

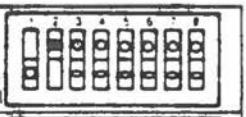
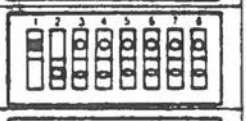
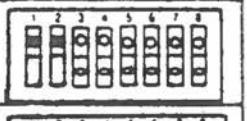
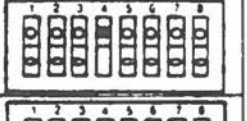
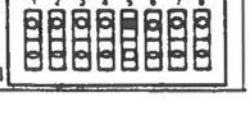
Memory Capacity 256KB Bank 0 256K, Banks 1, 2, 3, Empty	Switch 1 Off Switch 2 On	
Memory Capacity 640KB Bank 0 64K, Bank 1 64K Bank 2 256K, Bank 3 256K	Switch 1 On Switch 2 Off	
Memory Capacity 640KB Bank 0 256K, Bank 1 256K Bank 2 64K, Bank 3 64K	Switch 1 On Switch 2 On	
Serial Port Enabled	Switch 3 On	
Parallel Port Enabled	Switch 4 On	
	Switch 5 On	

Figure 1.18 Configuration Switch SWA

The CPU is included in a special 68-pin VLSI unit which also provides

- parity generation
- loudspeaker control
- keyboard interface
- configuration switch registers
- parallel printer control
- wait state generation
- clock control

Figure 1.19 summarizes system integration of the VLSI unit.

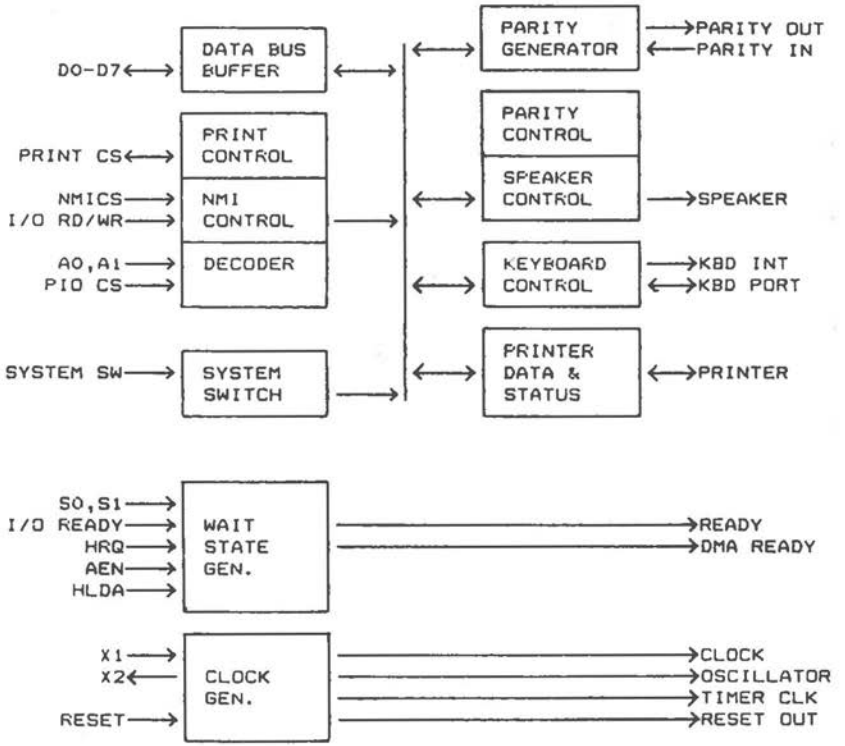


Figure 1.19 CPU VLSI

The VLSI unit registers fulfil the functions provided by the 8255A parallel I/O controller on other types of main processor board:

Port (Hex)	R/W	Bit	Function
61	R/W	0	Gate speaker data (1)
		1	Speaker data
		2	Not used
		3	Select group of registers for Port 62H
		4	Enable RAM parity (0)
		5	Enable I/O check (0)
		6	Enable keyboard clock (1)
		7	Clear keyboard (1)
62	R	0	Display configuration switch SW1-5
		1	" " " SW1-6
		2	Not used
		3	Not used
			When Port 61H, bit 2 = 0
		0	Not used
		1	Coprocessor installed
		2	System memory SW1-3
		3	" " SW1-4
			When Port 61H, bit 2 = 1
		4	Output Timer Counter 2
		5	" " " "
		6	I/O check
		7	Parity check
60	R		Keyboard data

Port (Hex)	R/W	Bit	Function
62	W	0	No. of flexible disk drives
		1	Coprocessor installed (1)
		2	} System memory
		3	}
		4	Display configuration switch SW1-4
		5	" " " SW1-5
		6	} No. of flexible disk drives
		7	}
63	W	0	Enable parity (0)
		1	Coprocessor installed (1)
		2-4	Not used
		5	Lock system (1)
		6-7	Not used

The VLSI unit also controls the parallel I/O interface via Ports 378H, 379H, and 37AH. Control of this interface is discussed in detail in the "Centronics Interface" section of Chapter 4.

Programmable Intelligence

This Chapter introduces a number of highly versatile integrated circuits present on the main processor board. The ICs described perform functions which to many users are not always evident at first sight, but which are, however, integral to the efficient working of your NCR PERSONAL COMPUTER. These ICs are the Timer, the Programmable Interrupt Controller, the Parallel Input/Output Interface, and the DMA controller.

The Timer, as its name suggests, provides a useful tool for real-time applications. It can be set to issue an output signal after a programmed delay on a one-off basis, or it can supply repeated signals at programmed intervals. It can even be used to supply oscillator frequencies for the small loudspeaker (this is described in a separate Chapter).

The Programmable Interrupt Controller detects and manages service requests not only from selected peripheral devices but also from other internal system functions, such as the Timer. The obvious advantage of the Programmable Interrupt Controller is that the system microprocessor (Central Processing Unit) does not have to continuously check the other units ("polling") to see if they are requiring attention: the CPU is interrupted only if a unit has expressly requested attention and the Programmable Interrupt Controller has decided, on the basis of its programmed priority logic, that the request deserves attention.

The Parallel Input/Output Interface relates to a number of system functions: it reads keyboard data as well as the system hardware configuration (memory, disk, co-processor, display), and has certain enabling functions (serial I/O, RAM parity check).

PROGRAMMABLE INTELLIGENCE

The Direct Memory Access (DMA) controller offloads the CPU where blocks of data have to be transferred to and from random access memory. Although the CPU cannot use the system bus during DMA cycles, it can perform internal register arithmetic. Bus arbitration circuitry ensures that CPU and DMA controller do not clash over the system bus.

Your NCR PERSONAL COMPUTER includes other programmable integrated circuits dedicated to more readily recognized peripheral control functions, for example the serial I/O, CRT, flexible and fixed disk controller interfaces. These are described in the relevant Chapters of this Manual.

For information on the 8088 system microprocessor with the possible addition of an 8087 mathematics co-processor, there is a wealth of generally available literature on the 8086 family of microprocessors (which includes the 8088 and 8087) to which you can refer.

8088/8087 CO-ORDINATION

The co-processor, which can be installed as an option, uses the same clock as the CPU. Both processors are connected to the bus without any intervening arbitration, so that, with the assistance of the status lines QS0 and QS1, the co-processor can read and decode instructions while the CPU is processing instructions from its internal queue. The co-processor uses its BUSY signal to place the CPU in a WAIT state, where it remains until the co-processor busy signal is no longer active.

If the co-processor detects an error (e.g. division by zero), it can interrupt the CPU. This is done by means of the non-maskable interrupt (writing to port 0A0H with data MSB set enables the NMI, resetting this bit disables the NMI). The presence of the co-processor, must be indicated by the setting of the appropriate switch on the main processor board.

An NMI can also be caused by a parity failure. For this reason, the corresponding bit (7) of the PC port of the Parallel Input/Output Interface should be checked, in the event of an NMI being issued. If the NMI was issued by the co-processor, the error can be localised by inspection of the co-processor's status word. If processing is then to be continued, clearing the BUSY flag in the status word (8087 mnemonic: FCLEX) will de-activate the BUSY line and terminate the CPU's WAIT state.

PROGRAMMABLE INTERVAL TIMER

Interval timing in the NCR PERSONAL COMPUTER is provided by an 8253 Programmable Interval Timer. This integrated circuit can be used as three independent 16-bit counters. The Timer is interfaced to the data bus, so that Timer values can be transmitted and read by the microprocessor. In addition, the Timer can be used to generate interrupts at programmed intervals or a single interrupt after a specific interval. Counting is carried out internally by the Timer, either as a binary or a Binary Coded Decimal (BCD) operation. Counting speed is determined by an external clock signal. Counting is achieved by decrementation of a Counter from the value loaded down to zero.

HARDWARE CONFIGURATION

Communication between Timer and microprocessor is via the Port addresses 40H-43H (see Figure 2.1).

Instruction	Function
OUT 40H	Load Counter 0
OUT 41H	Load Counter 1
OUT 42H	Load Counter 2
OUT 43H	Specify Timer operation
IN 40H	Read Counter 0
IN 41H	Read Counter 1
IN 42H	Read Counter 2
IN 43H	No operation

Figure 2.1 Timer Ports

The signal frequency of 1.19318 MHz common to all three clock inputs of the Timer (CLK0, CLK1, CLK2) is derived from the system crystal (14.318 MHz with

capacitive trimmer adjustment), the 8284A Clock Controller, and one further division (by 2).

The three output signals from the Timer (OUT0, OUT1, OUT2) are issued to the following system components:

OUT0 - Timer Interrupt (type 8) to the 8259A Programmable Interrupt Controller. This Timer output is set to an effective frequency of approximately 18.2 ticks per second (more accurately: 18.2065) by loading the 16-bit Counter with the maximum decrement value of zero (not OFFFFH: terminal count is not attained until the decrementing counter **passes** to zero).

OUT1 - Memory Refresh line. Initialization firmware requires only the lower 8 bits of the decrement Counter. The value set is 12H, which yields a signal period of approximately 15.1 microseconds.

OUT2 - Signal to the 8255A PIO with unrestricted user availability.

TIMER PROGRAMMING

Counters can be programmed independently of one another. Before a counter is initialized it is in an undefined state. The following programming steps are required to set up a Timer Counter.

One byte must be transmitted to the Timer's Control Register via Port 43H. The value of this byte is made up as shown in Figure 2.2.

D7	D6	D5	D4	D3	D2	D1	D0	via Port
COUNTER		R/W SELECT		MODE			BCD	43H

Figure 2.2 Timer Control Register

COUNTER is a two-bit binary value 0-2, denoting the number of the Counter to be accessed. Therefore, to access Counter 2, D7 should be set and D6 zero.
R/W SELECT determines the way in which the specified Counter is to be loaded or read. The type of operation to be carried out (read or load) depends on whether an IN or OUT instruction is being used (the Timer has pin connections for /RD and /WR signals). The significance of the binary value contained in these two bits is as follows:

0 Counter Latching (see below)
1 Read/load more significant byte of Counter
2 Read/load less significant byte of Counter
3 Read/load both Counter bytes (less significant first)

BCD: if this bit is set, the 16 bits of the selected Counter are used as a 4-digit BCD counter. If this bit is zero, the Counter represents a 16-bit binary value.
MODE may be a binary value 0-5 in three bits. The following modes can be implemented in the NCR PERSONAL COMPUTER hardware (applications requiring a rising edge at the GATE can be implemented only with Timer 2):

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

These Timer modes are described in detail in the following sections. Timing characteristics for the Timer modes are illustrated in Figure 2.3.

Output on Terminal Count (Mode 0)
Following the loading of the Counter, the signal OUTput pin for that Counter goes low, and remains low until the Counter has decremented to zero. The OUT signal then goes high, and remains high until the Counter is next programmed. If you write a new value

to the Counter before the old count has expired, decrementing resumes from the new value. From the hardware point of view, mode 0 is enabled by a high signal at the Gate for the specified Counter. This signal is permanently present at the Gate for Counters 0 and 1.

Programmable One-Shot (Mode 1)
OUTput goes low following a rising edge at the GATE, and goes high at terminal count. Resetting the counter value while OUTput is low does not affect the current terminal count, but comes into effect the next time the OUTput signal goes from high to low. The one-shot can be re-triggered without any need to reload the Counter.

Rate Generator (Mode 2)
An OUTput low pulse of one clock is issued upon terminal count. This cycle is initiated by the GATE going from low to high, and continues until the GATE goes low. A newly set Counter value comes into effect after the next OUTput pulse.

Square Wave Generator (Mode 3)
As Mode 2, except that OUTput remains high for the first half of the count and goes low for the second half (achieved by decrementing by 2 at each clock). If the Counter specifies an odd number, the first decrement in the first countdown is by 1, the next countdown starts with a decrement of three. This sequence is repeated until the GATE goes low.

Software Triggered Strobe (Mode 4)
As soon as this mode has been loaded, the OUTput for the selected Counter on the Timer goes high. When the Counter has been subsequently loaded, counting begins. As soon as the count has decremented to zero, the OUTput goes low for one clock period, and then high again. If your software reloads the Counter during decrementing, the new Counter value takes effect at the next clock signal. As in mode 0, operation of mode 4 is dependent on the presence of a high signal at the Gate (decrementing would be suspended if this signal were low).

Hardware Triggered Strobe (Mode 5)
Following a rising edge at the GATE, counting starts. The count continues until terminal count, whereupon the output goes low for one clock period. The count can be re-triggered at the GATE without reloading the Counter.

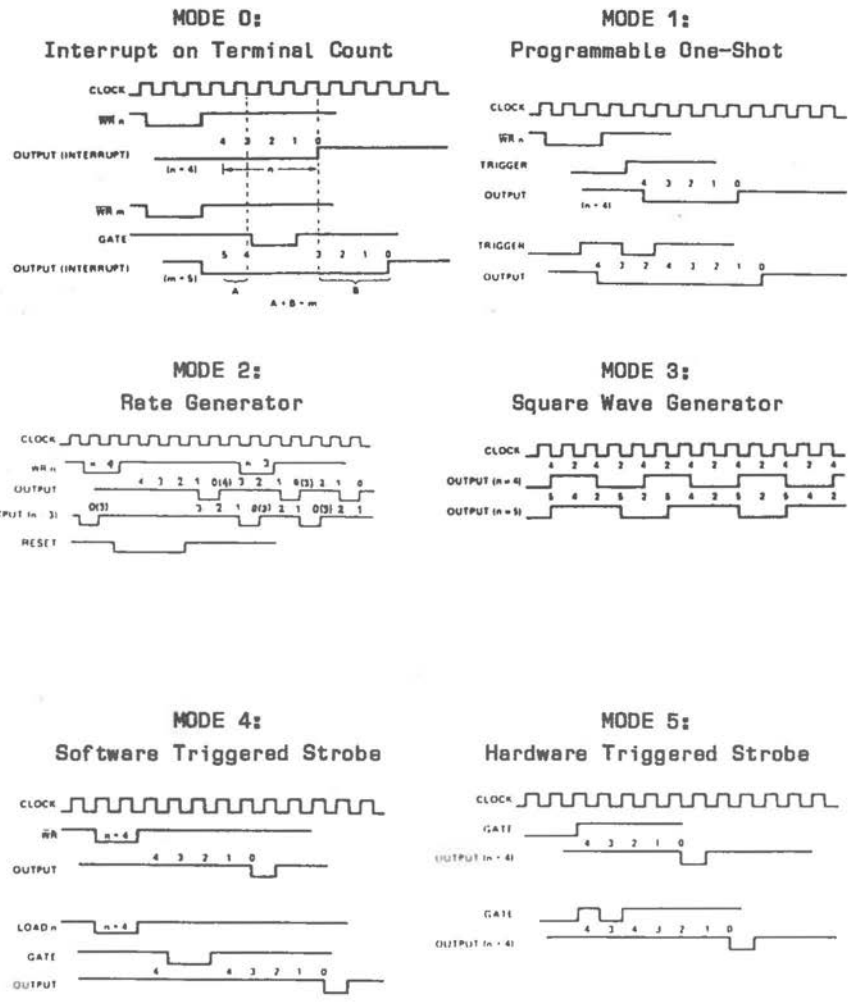


Figure 2.3 Timer Modes

Timer Parameters

Following the Timer Control Register byte detailed above, the Timer expects the number of bytes specified in the two bits R/W to be transmitted by the microprocessor (load operation), or the specified number of bytes to be read. The one or two-byte value is then read from or written to the data bus. Loading all zeros into a Counter results in a maximum count (0FFFFH in binary, 9999 in BCD counting).

Note that it is not necessary to read or write immediately after setting the Timer Control Register. However, the specified number of bytes must be read or written. In mode 0, as soon as the Timer recognizes that the first (or only) byte is being transmitted, the decrementing process is suspended until the new Counter contents have been read.

Reading the Counters

Reading Counter registers requires some care in order not to disturb the counting process. A Counter can be read directly or via a Counter Latch. The former method requires counting to be inhibited during the reading process. This can be achieved only by controlling the Gate or suspending the clock signal to the Counter which is to be read. For this reason, you should use the Counter Latch method.

To read a Counter Latch, a byte must be written to the Control Register, specifying the Counter and with D5 and D4 zero (this command has no effect on the MODE and BCD settings). Then issue a read Counter byte to the Control Register and read the one or two bytes specified.

An example of Timer programming in a real-time application is included in the section of this Chapter dealing with the Programmable Interrupt Controller. The programming example given in the Chapter describing the loudspeaker also makes of the Timer.

PROGRAMMABLE INTERRUPT CONTROLLER (PIC)

INTRODUCTION

Why use Interrupts?
There are two methods by which a call for attention from a device can be recognized by a program. One method is for the microprocessor to sample the status of the device at periodic intervals (polling), whereupon the program decides whether that status represents a call for attention. The other method is for the processor to be "interrupted" by a signal from the device, and then to proceed to a program routine which deals with the special situation. Afterwards, execution of the interrupted program may continue, if desired.

The advantage of the first method is that a large number of devices can be read via the computer port addresses, without additional hardware. This method can be most effectively applied in situations where information from a device is constantly required at regular intervals. Disadvantages arise when this need for information, or the device requests for attention, are only occasional and irregular. Polling then means in many cases ineffective use of microprocessor time. Furthermore, the time which elapses before the device request for attention is recognized can vary, depending on conditional branches taken within the current program.

For these reasons, it is often more practical to make use of the ability of the microprocessor to recognize an external interrupt in the form of a signal at the INTR pin. Assuming that interrupts at the microprocessor are currently enabled, the interrupt signal is noted by the microprocessor in an internal flip-flop, and another microprocessor pin issues a signal acknowledging the presence of the interrupt signal. Although the microprocessor does not proceed to deal with the interrupt until completing the current instruction cycle (assembler instruction), it is not necessary to synchronize the issuing of an interrupt with the microprocessor.

Interrupts and the 8088 Microprocessor

Following acknowledgement of the interrupt by the microprocessor (8088), the interrupting device must place an 8-bit value on the data bus. This value represents one of 256 different interrupt "types". The first five types of interrupt are reserved by the microprocessor for special, internal purposes:

Type 0 Division by Zero.
Type 1 Single Step.
Type 2 Non-maskable Interrupt: this interrupt type is recognized when a signal (active low) is present at the NMI pin of the microprocessor.
Type 3 This interrupt type is recognized when a software interrupt is issued by means of the one-byte INT instruction.
Type 4 Integer Overflow.

The NCR PERSONAL COMPUTER hardware makes use of the eight interrupt types in the range 8-15. The Basic Input/Output System software (BIOS) uses a number of other interrupt types for "software" interrupts (INT instruction). Both uses of microprocessor interrupts are summarized in Chapter 3, and can be investigated further in the ROM BIOS assembly language listings available as a separate document.

Starting with the next instruction cycle, the microprocessor automatically PUSHes the flags onto the stack and clears the interrupt flag, thus disabling further interrupts. Following this, the current values of the Code Segment and Instruction Pointer are PUSHed onto the stack, and new values, determining the starting address of the interrupt service routine, are fetched from the "interrupt vector" which occupies the first 1024 bytes of machine memory. After this routine has dealt with the interrupt situation, and provided that the interrupt service routine is concluded by an IRET instruction, the Code Segment and Instruction Pointer are restored to their former values. The flags are likewise restored, with the effect that the interrupt flag is enabled for further interrupts. (It is admissible for

the interrupt service routine to enable the interrupt flag, thus allowing interrupts to be nested.)

The "interrupt vector" is the 1024-byte area at the beginning of machine memory, where the CS and IP values for up to 256 interrupt types are held. Therefore, addresses 0-3 are concerned with interrupt type 0, addresses 4-7 with type 1, and so on. In each case, the paragraph value (CS) of the address of the interrupt handling routine must be present in the two uppermost bytes of the four byte block, the Instruction Pointer value in the two lowermost bytes. For example, the CS part of the starting address of the interrupt service routine for interrupt type 10 (0AH) is fetched from bytes 2AH and 2BH, the IP value from bytes 28H and 29H.

INTERRUPT CONTROLLER HARDWARE AND LOGIC

The NCR PERSONAL COMPUTER makes use of a single 8259A Programmable Interrupt Controller (PIC) integrated circuit. The PIC can control up to eight interrupts from different sources, tell the microprocessor which device has requested attention, and allow the programmer to set priorities among the eight possible interrupts.

Refer to the schematics in the Appendices regarding the integration of the PIC into the NCR PERSONAL COMPUTER hardware. The three pins of the integrated circuit which are not connected (12, 13, 15) are the cascade lines.

Use of the eight interrupt lines which can be controlled by the PIC is shown in Figure 2.4.

Interrupt Request No.	Microprocessor Int. Type No. (CS+IP in int. vector)	IRQ line accessible at bus connection	Hardware Function
0	8 (20H-23H)	not accessible	Timer 0
1	9 (24H-27H)	not accessible	Keyboard
2	0AH (28H-2BH)	B4	not dedicated
3	0BH (2CH-2FH)	B25	COM2
4	0CH (30H-33H)	B24	COM1
5	0DH (34H-37H)	B23	Fixed Disk
6	0EH (38H-3BH)	B22	Flex. Disk
7	0FH (3CH-3FH)	B21	Printer

Figure 2.4 PIC Interrupt Requests

PROGRAMMING THE PIC

The 8259A Programmable Interrupt Controller can be regarded as consisting of four logical aspects, as shown in Figure 2.5. The cascade logic need not concern us in this description, as the three cascade lines of the PIC are not connected. This programming description deals with the most important general aspects of programming the PIC, drawing special attention to those programming elements which are essential to it's functioning in the NCR PERSONAL COMPUTER environment. For a complete description of the PIC hardware and software interfacing possibilities, you may wish to refer to the publications of the integrated circuit manufacturer.

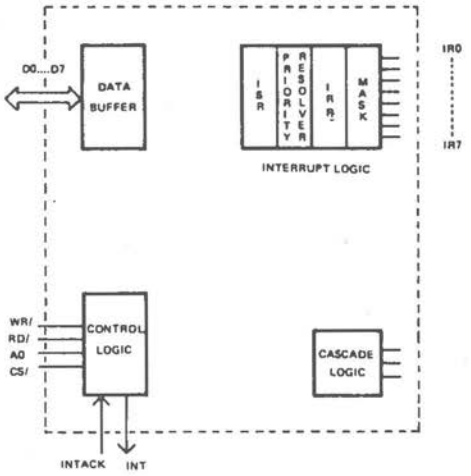


Figure 2.5 PIC Architecture

The data buffer provides a three-state interface to the system bus.

The Control Logic consists of

* Selection of the PIC by means of an (active low) signal to the CS/ pin. If this signal is not present, the PIC cannot be read or written to.

* Issuing an interrupt signal (active high) to the microprocessor and awaiting acknowledgement (active low) from the microprocessor. In 8088 mode, the first of these acknowledgement signals places the identity of an interrupt in the In-Service Register (see below). A second signal causes the PIC to place an interrupt number on the data bus, with which the processor will calculate the four bytes (CS+IP) to be read from the interrupt vector in machine memory.

* The Read/Write logic. Following an active low signal at the WR/ pin, the PIC can receive commands from the microprocessor. If there is an (active low) signal at the RD/ pin, the PIC can be instructed to place information on the data bus. If active signals are present at both these

pins concurrently, the PIC can be neither read nor written to.

* Additional selection of Read/Write functions by means of data lines D3 and D4 and the address line 0. Addressing the PIC via port 21H of the NCR PERSONAL COMPUTER sets this address line to high, addressing the PIC via port 20H sets it to low.

The Interrupt Logic comprises

* The Interrupt Request Register (IRR). This register notes for each of the eight interrupt lines whether an interrupt is waiting to be handled. A bit in the IRR is set by a signal (active high) on the corresponding interrupt line. This signal must remain high until the processor is ready to deal with that interrupt.

* The In-Service Register (ISR). This register stores interrupts which are currently being handled. The bit for a particular interrupt is set as soon as the acknowledgement signal arrives from the microprocessor. The corresponding bit in the IRR is then reset.

* The Priority Resolver. This decides the priority of the interrupts. An interrupt acknowledgement signal transfers the interrupt with the highest priority from IRR to ISR.

* Interrupt Mask Register. This register disables individual interrupt request lines by means of a bit mask. Masking an interrupt request (IR) line does not affect the other IR lines.

Communication between microprocessor and PIC consists of Initialization Command Words (ICW) and Operation Command Words (OCW) transmitted to the PIC, and PIC register status (IRR, ISR, IMR, or interrupt type) read by the processor. Figure 2.6 summarizes the signals required in order to switch between these various modes of communication. The software means by

which these initialization and operational functions are achieved is explained in the remainder of this description of interrupts.

A0	RD/	WR/	D3	D4	
0	1	0	x	1	Data bus is to be read by PIC as ICW1.
0	1	0	0	0	..... OCW2.
0	1	0	1	0	..... OCW3.
1	1	0	x	x	Data bus is read by PIC as OCW1, ICW2, or ICW3, according to logic sequence of commands [see below].
0	0	1			The PIC is to place IRR, or ISR, or interrupt type on data bus, according to contents of previous OCW2.
1	0	1			The PIC is to place IMR on data bus.

Figure 2.6 PIC Addressing

Note: These commands are valid only if the CS line is active [low]. If this line is high, or if both RD/ and WR/ are active [low], PIC access to the data bus is disabled.

Initializing the PIC

The microprocessor must place up to four Initialization Command Words (ICWs) on the data bus in order to initialize the PIC. ICW1 must be transmitted to the PIC with address line A0 at logical 0 (Port 20H of the NCR PERSONAL COMPUTER). Transmission of the other ICW's requires that A0 is set to 1 (Port 21H). The format of the first two of these data bytes (for the 8088) is shown in Figure 2.7.

	D7	D6	D5	D4	D3	D2	D1	D0	via Port
ICW1	0	0	0	1	DETECT	0	CASC	ICW4	20H
ICW2	<—— PIC base int. type ——>					0	0	0	21H

Figure 2.7 Initialization Command Words 1 and 2

D7-D5 in ICW1 are really "don't care". (They are only required in the PIC 8080 mode, where they state an interrupt vector address.)
DETECT indicates whether interrupts are level or edge triggered. The NCR PERSONAL COMPUTER firmware resets this bit (0), thus determining edge triggering.
If edge triggering is set, an interrupt is recognized upon a low to high transition on an interrupt request line. Level triggering is achieved by the presence of a high signal. Note that the interrupt signal (sequence of trigger pulses or constant high, according to DETECT mode selected) must not be removed from the interrupt request line until the first acknowledgement signal for that interrupt has been received from the microprocessor. In practical terms, this means that the IR signal should be reset at the issuing device by the interrupt service routine, otherwise the PIC may recognize the continued signal as a second interrupt request (see also "Signal Integrity", below).
CASC is set (1) to indicate that the PIC is single, and is not operating in a master/slave configuration (cascading).
ICW4 is set (1) to indicate that ICW4 is included as the last Initialization Command Word.

D7-D3 in ICW2 are to contain a binary value in five bits corresponding to the microprocessor interrupt type to be issued in response to an interrupt signal at PIC pin IR0. For example, the value 8 (01000B) determines that a valid interrupt request at PIC pin IR0 will cause the PIC to notify interrupt type 40H to the microprocessor. IR1 then corresponds to interrupt type 41H, and so on. The actual five-bit

value for D7-D3 used by your NCR PERSONAL COMPUTER is 1, so that IR0 results in microprocessor interrupt type 8 (IR1: interrupt type 9, etc.).

ICW3 is used by the PIC for master/slave identification. However, the setting of the CASC bit in ICW1 has already indicated that the PIC is not part of a cascade configuration. Therefore, the PIC expects that ICW3 is being omitted. ICW4 must then directly follow ICW2.

The format of ICW4 is shown in Figure 2.8.

	D7	D6	D5	D4	D3	D2	D1	D0	via Port
ICW4	0	0	0	SM	0	0	AEOI	P	21H

Figure 2.8 Initialization Command Word 4

D3 and D2 are significant only for cascaded PICs (buffered mode). Accordingly, these bits should be zero.
SM, if set, instructs the PIC to apply a special fully nested mode when PIC's are cascaded in a master/slave configuration. This ensures that servicing an interrupt from a slave does not preclude higher priority interrupts from the same slave. This is of no significance for the NCR PERSONAL COMPUTER, therefore this bit should be zero.
AEOI can be set or zero. When zero, this has the effect that the ISR bit is not reset until an End-of-Interrupt (EOI) command is encountered at the end of the interrupt service routine (see Operation Command Words, below). If AEOI is set, the ISR bit is reset when the PIC detects the trailing edge of the second interrupt acknowledgement signal from the microprocessor. In order to clear this "automatic" End-of-Interrupt mode, it is necessary to repeat the PIC initialization, this time with AEOI zero.

P denotes the type of microprocessor being used in conjunction with the PIC. This bit is set (1) to denote the presence of the 8088 processor.

Operation Command Words

Following the initialization described above, the PIC is ready to receive interrupts on all eight interrupt lines (IR0-IR7). It is good practice initially to mask (disable) all the interrupt lines, either by means of the microprocessor CLI instruction or the Operation Commands Words described below, or both. An interrupt request must not be allowed to occur until its interrupt vector entry is written and its interrupt service routine is available. Furthermore, interrupt requests which are not in use should be masked by OCW, so that spurious line signals do not pass control to a possibly non-existent interrupt service routine. Fortunately, this set-up procedure is taken care of by the initialization firmware. However, you should bear these considerations in mind, if you wish to use a particular interrupt request for a purpose other than its standard hardware function. If this is the case, you should be wary of re-assigning these hardware interrupts, as this means depriving the BIOS of its hardware interface. More often than not, such changes will entail writing alternative initialization procedures and BIOS routines.

The Operation Command Words (OCWs) make it possible for you to mask and unmask (disable and enable) individual interrupt request lines to the PIC, and to set priority schemes for dealing with interrupts. These facilities result in considerable savings of microprocessor time for your application, as an interrupt is passed to the microprocessor only if it fulfills the conditions determined by the current masking and priority scheme.

Even without Operation Command Words the PIC is capable of receiving interrupt requests. The PIC is then in the "Fully Nested Mode". Interrupt requests are regarded by the PIC as having different

priorities: IR0 has the highest priority, IR7 the lowest.

As soon as the interrupt acknowledgement is received from the microprocessor, the PIC places on the data bus the interrupt type of the interrupt request (see ICW2) with the highest priority. This means that if more than one interrupt request is present at a time, the one with the highest priority is dealt with. The PIC recognizes which IR lines have requested an interrupt by inspecting its Interrupt Request Register (IRR).

The PIC then checks its In-Service Register (ISR) to see if an interrupt is currently being processed. If none is being serviced, the interrupt type is transmitted to the processor and the interrupt service routine is activated as outlined in the Introduction to this description. The ISR bit for the interrupt is now set, and the corresponding IRR bit zero.

If an interrupt is currently being serviced, the PIC checks whether the new interrupt request is of higher priority. If this is the case, the ISR bit for the new interrupt is set, and the corresponding IRR bit is reset. The new interrupt request is then allowed to interrupt the currently active interrupt service routine: flags, CS and IP of the interrupted routine are PUSHed onto the stack in the usual way, so that the microprocessor can later find its way back to this routine, without assistance from the PIC. In the event that the new interrupt request is of priority equal to or lower than that of the interrupt currently being serviced, the new interrupt request must wait (IRR bit remains set) until higher priority interrupts have been serviced.

Your software can, of course, set and clear the interrupt flag of the 8088 microprocessor (assembler instructions: STI, CLI). By clearing the interrupt flag, all interrupts to the processor, including those of higher priority, are disabled. However, interrupt requests occurring while the processor

interrupt flag is cleared are not lost, as an IRR bit in the PIC is reset only when the interrupt is about to be serviced.

Once an interrupt from the PIC has been serviced, its ISR bit must be reset, so that waiting interrupts can be dealt with. Assuming that the AEOI bit in ICW4 was zero, interrupt service routines should conclude with an EOI Operation Command Word. This clears the appropriate bit in the ISR, thus informing the PIC that a particular interrupt has been serviced. Only one EOI need be issued in the NCR PERSONAL COMPUTER (two are necessary only in a cascade configuration, where both master and slave must be informed).

There are two types of EOI commands. As long as the priority scheme is the default Fully Nested Mode, you do not have to specify which interrupt has been serviced. The PIC will automatically reset the highest priority ISR bit currently set, this representing the most recently acknowledged and serviced interrupt. Therefore, a "non-specific" EOI is all that is required. A "specific" EOI is required in situations where your software has determined a priority scheme deviating from that of the Fully Nested Mode (see OCW2, below). This is because the PIC cannot ascertain the source of the interrupt most recently acknowledged and serviced. A "specific" EOI differs from a "non-specific" EOI, in that the former must specify one of the eight interrupts.

If automatic End-of-Interrupt is active (AEOI bit in ICW4 was set), the EOI automatically issued is non-specific. You will normally use the automatic End-of-Interrupt facility only when a nested priority structure is not required, for example, when all IR lines except one are masked. To alter the value of the AEOI bit, your software must go through the complete ICW sequence again.

The masking and priority scheme of the PIC can be programmed by means of up to three Operation Command Words. OCW1 can be transmitted only when address line A0 is at logical 1 (Port 21H). For the other OCWs, A0

must be 0 (Port 20H). Operation Command Words do not have to be issued in a particular sequence.

OCW1 masks all or selected interrupt requests to the PIC. If a bit in the Interrupt Mask register is set, the corresponding ISR bit is masked. However, once an interrupt has been acknowledged by the PIC, it will inhibit lower priority interrupt requests irrespective of subsequent masking.

	D7	D6	D5	D4	D3	D2	D1	D0	via Port
OCW1	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0	21H

Figure 2.9 Operation Command Word 1

	D7	D6	D5	D4	D3	D2	D1	D0	via Port
OCW2	R	SEOI	EOI	0	0	Int. Request			20H

Figure 2.10 Operation Command Word 2

The significance of bits D7-D5 in OCW2 is as follows:

R: when set, this bit instructs the PIC to rotate priority.
SEOI: when set, this bit denotes a specific End-of-Interrupt or a new priority setting. The binary value in bits D2-D0 must denote which interrupt is to be affected.
EOI: when set, this bit tells the PIC to recognize an End-of-Interrupt.

The principle of priority rotation requires a few words of explanation. Priority rotation can be applied when the default Fully Nested Mode is not required. Then the IR number of an interrupt request does not necessarily indicate its current priority.

When the ISR bit of the most recently serviced interrupt has been reset, this interrupt should be given the lowest priority. This same interrupt request line must then wait a maximum of 7 interrupts before its ISR bit can be set again. This equal priority rotation can be achieved by OCW2. If both bit R and bit EOI are set (OCW2=0A0H), this is interpreted by the PIC as a "rotate on non-specific EOI". This command has the effect illustrated by the following example.

Let us assume that IR5 is waiting for attention, and that IR3 is currently being serviced. As soon as the IR3 bit in the ISR has been reset, that interrupt request line is given lowest priority. The line with the highest priority is the IR4. As the IR4 bit in the Interrupt Request Register (IRR) is not set, the PIC proceeds to deal with IR5. If a new, unmasked interrupt request appears at IR3, and assuming no further unmasked interrupts appear at IR6, IR7, IR0, IR1, and IR2, the new interrupt at IR3 will be dealt with after IR6 has been serviced.

Non-specific EOI with rotation in automatic End-of-Interrupt mode (see ICW4) can be achieved by issuing OCW2 with only bit R set (OCW2=80H). This is interpreted by the PIC as "rotate on automatic EOI", which remains in force until cleared by issuing OCW2 with all bits zero.

OCW2 can be used to set a specific priority. In this case bits R and SEOI must be set, and the binary value of the interrupt request which is to have the lowest priority must be encoded in D2-D0. For example, if D2-D0 contain the value 4, the interrupt request with the highest priority is now 5, the interrupt request with the second highest priority is 6, and so on. This command does not actually issue an End-of-Interrupt. If an EOI is required at the same time, all three bits R, SEOI, and EOI must be set. The IR line specified in D2-D0 is then set to lowest priority, and its ISR bit is reset.

Figure 2.11 gives a summary of the various commands effected by the bits R, SEOI, and EOI.

D7	D6	D5	
..	..	..	
R	SEOI	EOI	
0	1	0	No operation.
0	0	0	Clear "rotate on automatic EOI" mode.
1	0	0	Rotate on automatic EOI.
0	0	1	Non-specific EOI.
0	1	1	Specific EOI.
1	0	1	Rotate on non-specific EOI.
1	1	0	Set priority.
1	1	1	Rotate on specific EOI.

Figure 2.11 Priority And Termination

OCW3 is available for setting a special mask mode and reading PIC registers. OCW3 is illustrated in Figure 2.12.

	D7	D6	D5	D4	D3	D2	D1	D0	via Port
OCW3	0	SME	SM	0	1	P0	RREG	REG	20H

Figure 2.12 Operation Command Word 3

SME, if set, tells the PIC to set or cancel a special mask: bit SM set then means set, bit SM means cancel the special mask. If SME is zero, SM is "don't care".
The setting of a special mask gives your software the possibility of temporarily enabling interrupts of lower priority which are currently disabled. To set a special mask, issue OCW3 with both SME and SM set. From then on **every** interrupt request whose bit in the Interrupt Mask Register is not set, is enabled, irrespective of nesting priority. The special mask

can be changed as often as you wish by means of OCW1. The former priority scheme is restored by means of a further OCW with SME set and SM zero.

PO, if set, acts as a Polling command. Your software can use this command to detect interrupts while the microprocessor interrupt flag is cleared. The next RD/ signal to the PIC with address line A0 at logical 0 (IN from Port 20H) causes the PIC to place the following information on the data bus:

D7	D6	D5	D4	D3	D2	D1	D0
INT	-	-	-	-	Int. Request		

Figure 2.13 Interrupt Polling

If INT is zero, there is no interrupt waiting to be serviced. If INT is set, at least one interrupt is waiting to be serviced. The number of the highest priority interrupt waiting for service is represented as a 3-bit binary value in D2-D0.

RREG and REG enable your software to read the PIC registers IRR and ISR. To read one of these registers, RREG must be set. Which of these two registers is to be read is determined by REG: if REG is set, ISR is to be read, otherwise IRR. The next RD/ signal with address line A0 at logical 0 (IN from Port 20H) causes the PIC to place the contents of the selected register on the data bus.

It is not necessary to repeat OCW3 for successive reading of the same register. The PIC remembers which of the two registers ISR and IRR was last read. However, after placing interrupt information on the data bus by means of the PO command in OCW3, a new OCW3 is required before ISR or IRR can be read. Following initialization of the PIC, IRR is the register automatically selected for reading.

In the event that OCW3 contains both a Polling command and a read PIC register command, the Polling command overrides.

OCW3 is not used for reading the interrupt mask register (IMR). Instead, the contents of IMR are placed on the data bus when the PIC receives a RD/ signal and address line A0 is at logical 1 (IN from Port 21H).

Signal Integrity
As already stated, an IR input signal must be present until the first interrupt acknowledgement from the microprocessor, irrespective of whether edge or level triggering is being used.

If the IR signal goes low before microprocessor acknowledgement, the PIC does not recognize a valid interrupt, but a signal is nevertheless recognized. This is particularly useful for detecting spurious noise on the IR lines.

If a non-valid interrupt request occurs, the PIC recognizes it as an IR7 signal. Your software for this interrupt type need only execute a microprocessor IRET instruction. In this way, the interrupt is ignored. If IR7 is being used for other purposes, your software can still distinguish between a genuine IR7 and a non-valid interrupt request: the non-valid interrupt request does not set ISR bit 7. Therefore, it is possible to detect a non-valid interrupt by examining ISR contents (read by means of OCW3) at the beginning of the interrupt service routine.

SYSTEM INTERRUPTS
Because the hardware interrupts, as well as a number of software interrupts, are already dedicated to specific functions of the NCR PERSONAL COMPUTER, a few words are necessary concerning your own application of these interrupts. Even the 8259A interrupt request 2 (that is, interrupt type 0AH) should be regarded as reserved for further system expansion.

There are two possible ways of using the 8259A interrupt requests for your own applications. One possibility is to write the address of your own interrupt service routine directly into the four vector bytes from which the 8259A interrupt fetches its interrupt service routine address. This has the disadvantage of denying the ROM BIOS access to its important system-maintaining routines. If you are going to do this, you will need to include routines to replace those which have now been by-passed, and to return to the interrupting level by means of a PIC non-specific End-of-Interrupt command.

To make life easier, the ROM BIOS has provided the possibility of adding your own interrupt service routines to those already implemented by the system for the timer and keyboard hardware. During the ROM BIOS service routines for timer (type 8) and keyboard (type 9), a software interrupt is issued: type 1CH for the timer, type 1BH for the keyboard. The addresses placed by the ROM BIOS in the interrupt vectors for these two interrupts (at 70H and 6CH, respectively) simply refer to a dummy IRET instruction at a ROM location. To address an interrupt service routine of your own for one of these interrupts in read/write memory, you need only write the CS+IP address into the interrupt vector at the appropriate location. Note that while the interrupt vector is being written, hardware interrupts must be disabled (CLI instruction). This prevents the interrupt vector entry from being read by the CPU until both words are written and the address is therefore good. Routines accessed in this way need only be concluded by an IRET instruction. The 8259A End-of-Interrupt is supplied upon return to the ROM BIOS routine.

Remember that disk operating system software may write these interrupt vector positions for interrupt service routines loaded from disk.

INTERRUPT HANDLING: EXAMPLES

The following sample program shows just one of the many possibilities or real-time applications making use of system interrupts.

The time of day in the form of a 24-hour clock is constantly displayed in the top left corner of the screen. Once the interrupt handler for the timer software interrupt 1CH has been set up, the CPU is no longer required, except during interrupt handling triggered by Timer CLOCK 0 19 times per second. This means that other applications can be run while the clock continues to function. If you are using NCR-DOS, you should exit the setting up routine by means of Function Call 31H. This terminates the process, but preserves a specified size of memory in the program segment, so that the interrupt service routine is not overwritten by a subsequently loaded program.

The program consists of the following routines:

SETUP reads the initial clock time from the keyboard buffer (NCR-DOS system call) in the form hh:mm:ss. Specify two digits each for hours, minutes, and seconds, with intervening separators of your choice. Here you are required to include redundant zeros, and no syntax checking is performed. You can add user interface elements of your choice.

The Timer Counter 0 is initialized to hardware triggered strobe (Mode 5) and a count value of 62799 (0F54FH), with the effect that the interrupt request line 0 of the PIC is strobed 19 times every second. Using the maximum count value (setting Counter to zero) would mean that the interrupt request frequency (18.2065) could not be related to a whole number. The result would be a considerable overhead of clock idling software. There is only one side-effect to writing this Timer Counter, namely that it invalidates the system clock otherwise maintained by NCR-DOS.

The interrupt flag is cleared to prevent disturbance of Timer Counter writing, and especially to prevent interrupts while the interrupt vector is being written.

The setup routine concludes with a terminate but Keep Process function call to NCR-DOS. This means that the interrupt service routine is retained and can be activated during execution of subsequently loaded applications, as long as the NCR-DOS structure of program segments is not disturbed. It is not really necessary to retain the setup and interrupt vector writing routines. Only ISR1C, its dependent routine INCTIM, the display routine, and the data areas (except the keyboard data areas) are required for interrupts service. Therefore, if memory is at a premium, you could place the setup routines at the end of your code and provide an appropriate jump instruction at the beginning of your code so that the setup will still execute as a normal EXE file. You can then retain a reduced nuber of paragraphs at the end of the setup process.

NEWIVEC exchanges the current vector entry for interrupt type 1CH with the CS value for the setup routine (which is the same as that for the interrupt service routine), and the IP value of ISR1C. (Strictly speaking, it is not necessary to retain the former vector entry in this example.)

OLDIVEC is included simply in order to show how to restore a former vector entry. Note that interrupts should also be inhibited when using such a routine.

ISR1C services the interrupt type 1CH, which itself is issued by the hardware interrupt type 8 (IRQ0 at the PIC). The Code Segment value is as in the setup routine. The Data Segment is also set to this value, so that the interrupt service routine can address those data areas initialized in the setup routine. A completely new stack is set up within the program segment, so that stack space does not have to be accounted for in the programs which are, or might be, interrupted. As the data and stack areas are local to

the interrupt service routine, the ISR inhibits interrupts.

The interrupt service routine increments the 1/19 second TICK counter, and updates the time display every 19 ticks using BCD arithmetic.

SEETIM displays the time, blinking at a rate of once per second, in the top left corner of the screen. The display memory base paragraph value read from DBASE applies if your CRT has a graphic controller (0B800H). If your CRT has a character (alpha) controller, replace this value by 0B000H.

```
        CSEG   SEGMENT
        ASSUME CS:CSEG,DS:CSEG,SS:CSEG
;
        ORG    100H
        PUSH   CS
        POP    DS
;
        XOR    AX,AX
        MOV    ES,AX           ;For use as segment override prefix
                               ;when addressing absolute memory
                               ;(int. vector) locations.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                  ;
;          Main program to set initial time                        ;
;          and user interrupt vector entries.                      ;
;                                                                  ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
SETUP:                         ;Read initial values into clock.
        MOV    DX,OFFSET KEYIN
        MOV    AH,0AH
        INT    21H             ;MCR-DOS Read Keyboard Buffer.
        MOV    BX,OFFSET KEYINDAT
        MOV    DI,OFFSET HOUR
        MOV    DL,3            ;Read hour, minute, second.
                               ;Note: here no syntax check.
KEY2CLK: MOV   AX,WORD PTR [BX]
        AND    AX,0F0FH        ;Mask ASCII to BCD.
```

```
        MOV   CL,4
        SHL   AL,CL
        OR    AL,AH          ;Two BCD digits in AL.
        MOV   BYTE PTR [DI],AL
        INC   DI
        ADD   BX,3           ;Address byte following separator.
        DEC   DL             ;Until all 3 time units read.
        JNZ   KEY2CLK
;
        MOV   [TICK],0       ;Counts fractions of a second.
;
        CLI                  ;Inhibit interrupts while int.
                             ;vector is being written.
;
                             ;Set timer counter 0 for 19 ticks
                             ;per second.
;
        MOV   DX,43H         ;Timer operation port.
        MOV   AL,3CH         ;Initialize Timer 0, both bytes,
                             ;not BCD, hardware triggered strobe.
        OUT   DX,AL
        MOV   DX,40H
        MOV   AL,4FH         ;Timer low byte.
        OUT   DX,AL
        MOV   AL,0F5H        ;Timer high byte.
        OUT   DX,AL
;
        CALL  NEWIVEC        ;Set new (user) interrupt service
                             ;routine address.
        STI                  ;User ISR counting starts at next
                             ;pulse from 8253 Timer CLOCK 0.
;
XSETUP: MOV   AL,0           ;optional return code.
        MOV   DX,51H         ;paragraphs.
        MOV   AH,31H
        INT   21H            ;NCR-DOS Terminate but
                             ;Keep Current Process.
        NOP                  ;[for DEBUG]
;
```

```
          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
          ;                                             ;
          ;  Routine to set time in top left corner of  ;
          ;  screen in accordance with BCD values in    ;
          ;  HOUR, MINUTE, and SECOND.                  ;
          ;                                             ;
          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
SEETIM:   PUSH  AX
          PUSH  BX
          PUSH  CX
          PUSH  DI
          PUSH  ES               ;Save, as required for zero segment
                                 ;override prefix.
;
          MOV   BH,[HOUR]
          MOV   BL,[MINUTE]
          MOV   CH,[SECOND]
          MOV   DI,BBASE
          MOV   ES,DI            ;Extra segment is now video memory.
          XOR   DI,DI            ;Offset zero in video memory.
          MOV   AL,BH            ;BCD hour.
          MOV   CL,4             ;Shift factor.
          SHR   AL,CL            ;Upper digit of hour now in 4 LSBs.
          CALL  SCRN
          MOV   AL,BH            ;BCD hour, lower digit.
          CALL  SCRN             ;Hour now displayed.
          MOV   AL,0AH           ;Produces colon separator in SCRN,
                                 ;(ORed with 30H).
          CALL  SCRN
          MOV   AL,BL            ;BCD minute.
          SHR   AL,CL            ;Upper digit of min. now in 4 LSBs.
          CALL  SCRN
          MOV   AL,BL            ;BCD minute, lower digit.
          CALL  SCRN
          MOV   AL,0AH           ;Colon, as after hour.
          CALL  SCRN
          MOV   AL,CH            ;BCD second.
          SHR   AL,CL            ;As with hour and minute ...
          CALL  SCRN
          MOV   AL,CH
          CALL  SCRN
          MOV   AL,20H           ;Trailing space on screen
          MOV   ES:[DI],AL       ;(DI already incremented in SCRN).
```

```
;
          POP   ES
          POP   DI
          POP   CX
          POP   BX
          POP   AX
          RET
;
         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         ;                                            ;
         ; Convert binary value 0-9 to ASCII digit    ;
         ; and write to screen memory.                ;
         ;                                            ;
         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
SCRN:     AND   AL,0FH         ;Blank upper 4 bits.
          OR    AL,30H         ;Makes ASCII digit.
          MOV   ES:[DI],AL     ;Write to screen.
          INC   DI
          MOV   AL,ES:[DI]     ;Get attribute byte.
          NOT   AL
          XOR   AL,88H         ;And invert displ. mode (monochrome).
          MOV   ES:[DI],AL     ;Write attribute byte to screen.
          INC   DI
          RET
;
;
         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         ;                                            ;
         ; Write address of user interrupt service    ;
         ; routine into vector for int. type 1CH.     ;
         ; Store old vector entry (here, not necessary);
         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
NEWIVEC:                       ;(CLI prior to call).
;
          MOV   BX,[VECT1C]    ;Location in interrupt vector
                               ;of ISR CS:IP.
          MOV   CX,ES:[BX]     ;IP of present ISR.
          INC   BX
          INC   BX             ;Point to CS in int. vector.
          MOV   DX,ES:[BX]     ;CS of present ISR.
          MOV   ES:[BX],CS     ;CS of new ISR.
          DEC   BX
```

```
          DEC    BX
          MOV    AX,OFFSET ISR1C  ;IP of new ISR
          MOV    ES:[BX],AX     ;now written to int. vector.
          MOV    [IVECIP],CX    ;store IP of old ISR.
          MOV    [IVECCS],DX    ;store CS of old ISR.
;
          RET
;
;
;          Routine to restore former int. vector values.
;            (not required in this example)
;
OLDIVEC:
;
          MOV    CX,[IVECIP]    ;Old ISR IP from storage.
          MOV    DX,[IVECCS]    ;Old ISR CS from storage.
          MOV    BX,[VECT1C]    ;Lowest address in int. vector of
                                ;entry for int. type.
          MOV    ES:[BX],CX     ;Restore former IP.
          INC    BX
          INC    BX
          MOV    ES:[BX],DX     ;Restore former CS.
;
          RET
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                              ;
; Interrupt Service Routine maintaining and displaying         ;
; a user clock, using BCD arithmetic.                          ;
;                                                              ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
ISR1C:    CLI                   ;Procedure must not be interrupted
                                ;while DS/SS are local to ISR.
          MOV    CS:[SSSTORE],SS ;Store stack segment of calling
                                ;procedure.
                                ;CS segment override prefix, as DS
                                ;still as set for interrupted
                                ;procedure (CS equiv. to DS when ISR
                                ;was set up).
          PUSH   CS             ;Still on stack of interrupted
                                ;procedure.
          POP    SS             ;Set stack to this prog. segment.
          MOV    CS:[SPSTORE],SP ;Store stack pointer of calling
```

```
                        ;procedure.
        MOV   SP,OFFSET STACKTOP
                        ;Local stack pointer for ISR.
;
        PUSH  DS    ;To account for NCR-DOS changed data segment
                    ;this routine must be accessible from other
                    ;program segments, as, obviously, interrupts
                    ;continue to occur after procedure
                    ;termination (procedure must be retained).
        PUSH  AX    ;PUSH registers so that calling program
        PUSH  BX    ;status can be restored at end of ISR.
        PUSH  CX
        PUSH  DX
        PUSH  DI
        PUSH  SI
;
        PUSH  CS
        POP   DS    ;Ensures data segment locations relate
                    ;to local data after intersegment call.
;
        MOV   AL,[TICK]
        INC   AL            ;Counts up to 19 ticks, after which
                            ;seconds display must be updated.
        CMP   AL,19
        JZ    SECUP         ;Jump if 19th tick counted,
        MOV   [TICK],AL     ;otherwise increment tick counter
        JMP   ISREND        ;and IRET without altering display.
SECUP:                      ;Arrive here only if display to
                            ;be updated.
;
        MOV   [TICK],0      ;Zero 19 tick counter.
        MOV   AL,[SECOND]   ;Fetch current seconds value.
        CALL  INCTIM        ;Add one second to time in AL.
        CMP   AL,60H
        JZ    MINUP         ;Jump if minute reached, otherwise
        MOV   [SECOND],AL   ;write new sec. value back to storage
        CALL  SEETIM        ;and display new time.
        JMP   ISREND
;
MINUP:  MOV   [SECOND],0    ;Set seconds to zero.
        MOV   AL,[MINUTE]   ;Fetch current minutes value.
        CALL  INCTIM        ;Add one minute to time in AL.
        CMP   AL,60H        ;Check if hour reached.
        JZ    HOURUP        ;Jump if hour to be incremented, else
```

```
          MOV   [MINUTE],AL  ;write new min. value back to storage
          CALL  SEETIM       ;and display new time.
          JMP   ISREND
;
HOURUP:   MOV   [MINUTE],0   ;As for seconds, minutes ...
          MOV   AL,[HOUR]
          CALL  INCTIM
          CMP   AL,24H       ;Check for total clock rollover.
          JZ    NEWDAY
          MOV   [HOUR],AL
          CALL  SEETIM
          JMP   ISREND
;
NEWDAY:   MOV   [HOUR],0     ;Seconds and minutes already zeroed.
          CALL  SEETIM
;
;
ISREND:   POP   SI           ;Restore status of
          POP   DI           ;interrupted program.
          POP   DX
          POP   CX
          POP   BX
          POP   AX
          POP   DS           ;Data segment now refers to NCR-DOS
                             ;program sement of calling program.
;
          MOV   SS,CS:[SSSTORE]
          MOV   SP,CS:[SPSTORE]
                             ;Restore old stack.
          STI                ;Re-enable interrupts.
          IRET
;
          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
          ;                                                               ;
          ; Accept one or two digit BCD value in AL,                      ;
          ; add BCD 1 to that value, return result in AL;
          ;                                                               ;
          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
INCTIM:   MOV   CL,4         ;Shift factor.
          SHL   AX,CL        ;Upper digit in AH low 4 bytes,
          SHR   AL,CL        ;lower digit in AL low 4 bytes.
          INC   AL           ;Add 1 to lower digit.
          AAA                ;Increment upper digit [AH],
```

```
                                    ;if lower digit > 9.
            SHL     AL,CL
            SHR     AX,CL           ;New, incremented BCD value in AL.
            RET
;
;
;
KEYIN       DB      9               ;Data area for NCR-DOS buffered
            DB      0               ;keyboard input (Function Call 0AH).
KEYINDAT    DB      10 DUP (?)
;
VECT1C      DW      0070H           ;Lowest int. vector address used by
                                    ;int. type 1CH (user timer).
IVECIP      DW      0               ;Storage for int.vect. IP.
IVECCS      DW      0               ;  "      "     "      CS.
HOUR        DB      0               ;Byte storage for BCD values
MINUTE      DB      0               ;  "     "     "   "    "
SECOND      DB      0               ;  "     "     "   "    "
                                    ;(values read from keyboard).
TICK        DB      0               ;Single CLOCK pulse counter.
;
DBASE       DW      0B800H          ;Beginning of graphic display memory,
                                    ;use 0B000H if character display card
                                    ;(values are paragraph).
;
SSSTORE     DW      0               ;Store stack seg. of calling proc.
SPSTORE     DW      0               ;            pointer            .
;
;
OWNSTACK    DW      16 DUP (?)
STACKTOP    DW
;
            CSEG    ENDS
            END
;
```

The following example uses the software interrupt (1BH) of the keyboard hardware interrupt service routine. This software interrupt is issued only when the hardware interrupt service routine was activated because of the Ctrl-Break key combination.

This program sets the interrupt vector for interrupt 1BH to address a routine which inverts the monochrome display contrast. Normally, upon returning from INT 1BH, the Ctrl-Break code is placed in the system keyboard buffer. To suppress this buffering, and its inherent display of ^C on the screen, the Interrupt RETurn and POPping of registers for the hardware interrupt (type 9) is provided by the software interrupt service routine. The stack is adjusted so that the interrupt return is to the routine from which the ISR for INT 9 (not 1BH) was activated.

The considerations regarding CPU segment registers and NCR-DOS program segment protection in the first example can also be applied to the following program.

```
CSEG    SEGMENT
ASSUME CS:CSEG,DS:CSEG,ES:CSEG,SS:CSEG
ORG     100H
PUSH    CS
POP     DS

;
        XOR     AX,AX
        MOV     ES,AX           ;for use as segment override prefix
                                ;when addressing absolute memory
                                ;locations [int. vector].
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                       ;
;       Main program to set keyboard user int. vector entry.            ;
;                                                                       ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
        CLI                     ;inhibit interrupts while vector is
                                ;being written.
        CALL    NEWIVEC
        STI
;
        MOV     AL,0            ;optional return code.
        MOV     DX,30H
        MOV     AH,31H          ;NCR-DOS Terminate but Keep Process.
        INT     21H
        NOP
```

```
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ;                                                                       ;
        ; Write address of user ISR into int. vector                            ;
        ; for ^-Break keyboard interrupt [type 1BH].                            ;
        ; Store old vector entry [not used].                                    ;
        ;                                                                       ;
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
NEWIVEC:                        ;CLI prior to call.
          MOV   BX,[VECT1B] ;address in interrupt vector.
          MOV   CX,ES:[BX]  ;IP of present ISR.
          INC   BX
          INC   BX              ;point to CS in int. vector.
          MOV   DX,ES:[BX]  ;CS of present ISR.
          MOV   ES:[BX],CS  ;CS of new ISR.
          DEC   BX
          DEC   BX
          MOV   AX,OFFSET ISR1B  ;IP of new ISR
          MOV   ES:[BX],AX       ;now written to int. vector.
          MOV   [IVECIP],CX
          MOV   [IVECCS],DX
;
          RET
;
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
        ;                                                                       ;
        ; Interrupt Service Routine to invert video                             ;
        ; display [monochrome] when ^-Break pressed,                            ;
        ; suppressing acceptance of ^C into keyboard                            ;
        ; buffer.                                                               ;
        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
ISR1B:    CLI
          PUSH  CS              ;Still on stack of interrupted proc.
          POP   DS              ;     from                              .
          MOV   [SSSTORE],SS
          PUSH  CS
          POP   SS
          MOV   [SPSTORE],SP  ;SS:SP of interrupted proc. saved
                              ;in local Data Segment.
          MOV   SP,OFFSET STACKTOP
                              ;Local stack now in force.
;
          PUSH  ES              ;Needed as segment override prefix
```

```
                                ;to address video memory.
          MOV   AX,[DBASE]
          MOV   ES,AX
          MOV   BX,1            ;Offset of first attribute byte.
INVERT:   MOV   AL,ES:[BX]
          NOT   AL              ;Invert it ...
          XOR   AL,88H
          MOV   ES:[BX],AL      ;... and write it back.
          INC   BX
          INC   BX              ;Point to next attribute byte.
          CMP   BX,0FA1H
          JNE   INVERT          ;Jump if last attribute byte
                                ;not yet written.
;
          POP   ES
;
          MOV   SP,[SPSTORE]
          MOV   SS,[SSSTORE]    ;Restore former stack.
;
          ADD   SP,6            ;So that IRET CS:IP and Flags effect
                                ;return as if from type 9
                                ;(instead of type 18H) ISR.
;
          MOV   AL,20H
          OUT   20H,AL          ;Non-specific EOI to PIC.
;
          POP   SI              ;POPs required for orderly completion
          POP   DX              ;of type 9 ISR
          POP   DS              ;(see ROM BIOS listings).
          POP   CX
          POP   BX
          POP   AX
;
          STI
;
          IRET
;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                  ;
; Data area must be accessible to both initialization and          ;
; interrupt service routines.                                      ;
;                                                                  ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
VECT1B      DW     006CH          ;lowest int.vector address used by
                                  ;int. type 1BH (Ctrl-Break keyboard).
IVECIP      DW     0              ;storage for old int. vect. IP.
IVECCS      DW     0              ;   "     "  "  "    "  CS.
;
DBASE       DW     0B800H         ;begining of graphic display memory,
                                  ;use 0B000H if character display
                                  ;(values are paragraph).
;
SSSTORE     DW     0              ;Store stack seg. of calling proc.
SPSTORE     DW     0              ;          pointer          .
OWNSTACK    DW     16 DUP (?)
STACKTOP    DW     0
;
            CSEG   ENDS
            END
;
```

PARALLEL INPUT/OUTPUT INTERFACE

A number of essential hardware functions of the NCR PERSONAL COMPUTER are managed by the 8255A parallel interface integrated circuit. These include recognition of main processor board switch settings, keyboard data reading, disk configuration and display type detection, loudspeaker control, as well as I/O, parity, and timer functions.

PROGRAMMING THE 8255A

The 8255A governs four port addresses 60H to 63H. Port 63H is used for accessing the command register, the other three for 8-bit parallel data I/O to and from the system data bus. These data ports are designated PA, PB, and PC. They can be regarded as 3 separate groups, each with 8 data lines, or as 2 groups, each with 8 data and four control lines. In the latter configuration, the ports PA and PB represent data, the upper 4 bits of PC are assigned as control lines to PA, the lower 4 bits to PB.

The 8255A can operate in one of three modes:

* Mode 1: I/O with no handshaking.
* Mode 2: I/O with handshaking.
* Mode 3: bi-directional bus with handshaking.

The mode of operation is set by means of a command byte from the CPU to port 63H. This byte is illustrated in Figure 2.14.

Bit	Significance	
7	Must be set to indicate that the Mode is being selected.	
6	00 = Mode 1	PA Mode Select
5	01 = Mode 2	
	1x = Mode 3 (lower bit "don't care")	
4	0 = PA is output port; 1 = PA is input port.	
3	PC bits 4-7: 0 = output; 1 = input.	
2	0 = select Mode 1 for PB; 1 = select Mode 2 for PB. (Mode 3 not possible for PB.)	
1	0 = PB is output port; 1 = PB is input port.	
0	PC bits 0-3: 0 = input; 1 = output.	

Figure 2.14 8255A Command Byte

Mode 1

16 I/O combinations are possible, according to the settings of bits 4, 3, 1, and 0 in the command word (see Figure 2.15). Input to the 8255A is unlatched, but output is latched.

Bit				Port Function			
4	3	1	0	PA	PB	PC bits 4-7	PC bits 0-3
0	0	0	0	0	0	0	0
1	0	0	0	I	0	0	0
0	1	0	0	0	0	I	0
1	1	0	0	I	0	I	0
0	0	1	0	0	I	0	0
1	0	1	0	I	I	0	0
0	1	1	0	0	I	I	0
1	1	1	0	I	I	I	0
0	0	0	1	0	0	0	I
1	0	0	1	I	0	0	I
0	1	0	1	0	0	I	I
1	1	0	1	I	0	I	I
0	0	1	1	I	0	I	0
1	0	1	1	I	I	0	I
0	1	1	1	0	I	I	I
1	1	1	1	I	I	I	I

Figure 2.15 Mode 1 Combinations

Mode 2

In this mode of operation only two I/O groups are possible as the PC lines are used for control purposes. PA and PB can be used independent of one another for input or output. Figure 2.16 shows the signal directions and command words needed to initiate PA for input and PB for output in Mode 2.

The control lines (PC0 - PC7) are used as follows (0 and 1 represent logic signals):

Input

PC4 = STB/ for PA PC2 = STB/ for PB	At 0, data is read into into the 8255A internal data register. This strobe also sets the INTE signal, so that one condition for INTR is fulfilled.
PC5 = IBF for PA PC1 = IBF for PB	Input Buffer Full: 1 confirms that data has been held in the internal data register.
PC3 = INTR for PA PC0 = INTR for PB	If INTE is set, this signal is triggered by the STB/ rising edge, and is usually used to request a CPU interrupt.

Output

PC7 = OBF/ for PA PC1 = OBF/ for PB	Output Buffer Full: 0 confirms that the data bus has been read into the 8255A internal data register. This signal is triggered by the WR/ rising edge.
PC6 = ACK/ for PA PC2 = ACK/ for PB	With 0, the receipt of data is confirmed to the 8255A.
PC3 = INTR for PA PC0 = INTR for PB	As for Input.

Unused PC lines can be used for data purposes (see Mode 3).

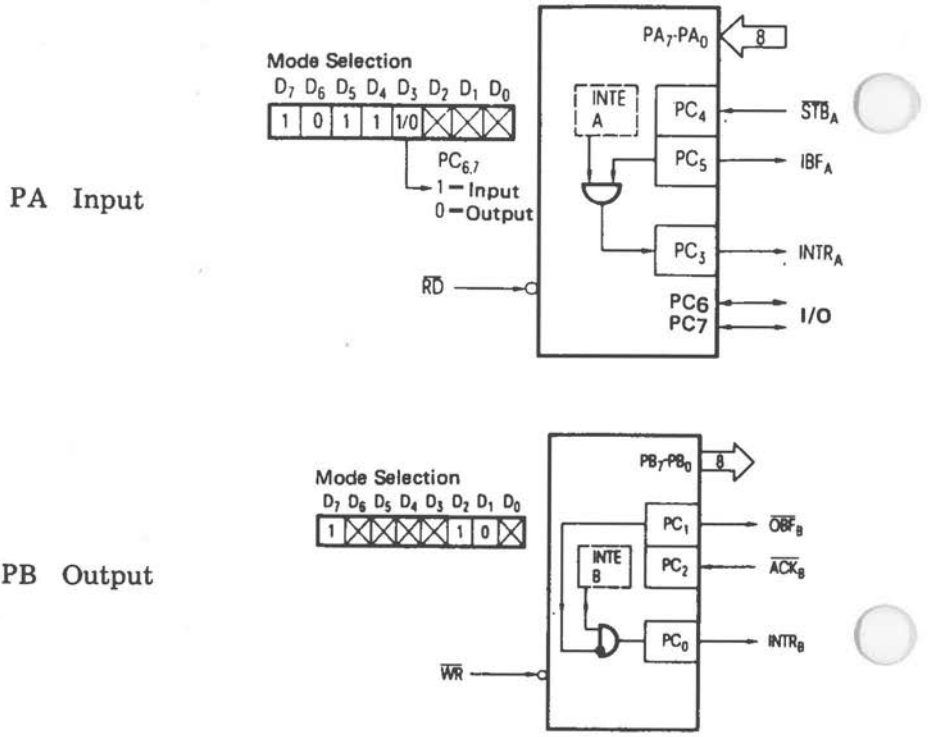


Figure 2.16 Mode 2 Control Signals

Mode 3

PA can be used as an 8-bit bi-directional data bus, using the control signals shown in Figure 2.18.

In Modes 2 and 3, PC lines not dedicated to control functions can be used as normal data lines. Any PC line can be set or reset by means of a command from the CPU (port 63H):

Bit	Significance
7	Must be zero to denote PC command (1 denotes Mode selection).
6 5 4	} Not used.
3 2 1	} 3-bit binary value (0-7) indicating which PC line is to be set/reset.
0	0 = reset line (0); 1 = set line (1).

Figure 2.17 Mode 3: PC line control

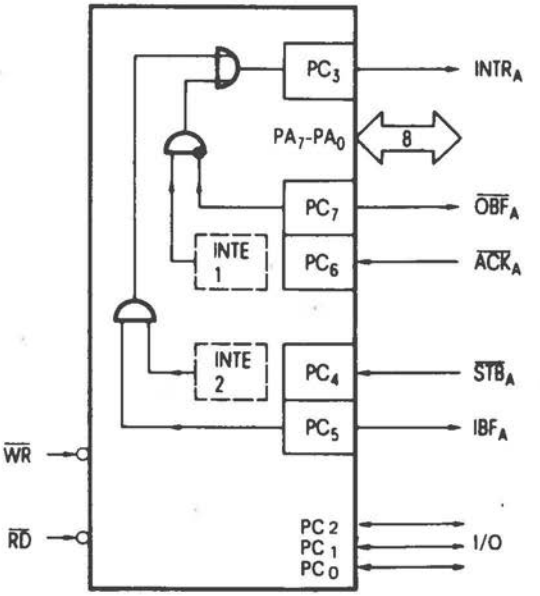


Figure 2.18 Mode 3 Control Signals

8255A: SYSTEM USE
Ports 60H-63H are dedicated to the 8255A as follows:

OUT 63H - CPU writes command byte (Mode or PC line selection) to 8255A.

IN/OUT 60H - CPU reads/writes PA.
IN/OUT 61H - CPU reads/writes PB.
IN/OUT 62H - CPU reads/writes PC.

The initialization firmware of your NCR PERSONAL COMPUTER selects Mode 1 with PA and PC as input ports and PB as an output port. Figure 2.19 shows the system use of PA, PB, and PC. Note that PC has two sets of input functions, according to the status of PB bits 7 and 3, respectively.

The following 2-bit binary values concerning the flexible disk drive, mainboard memory, and display configurations can be read from port PC:

* Display controller type at switching on - (PB bit 3 = 0)

1 = graphic display (40 x 25)	x x x x x x 0 1
2 = graphic display (80 x 25)	x x x x x x 1 0
3 = character display (80 x 25)	x x x x x x 1 1

* Number of flexible disk drives in system - (PB bit 3 = 0)

0 = 1 flexible disk drive	x x x x 0 0 x x
1 = 2 flexible disk drives	x x x x 0 1 x x
2 = 3 ...	x x x x 1 0 x x
3 = 4 ...	x x x x 1 1 x x

* Amount of memory on main processor board - (PB bit 3 = 1)

0 = 64 KB	x x x x 0 0 x x
1 = 128 KB	x x x x 0 1 x x
2 = 192 KB	x x x x 1 0 x x
3 = 256 KB	x x x x 1 1 x x

Keyboard and loudspeaker use of the 8255A is described in the Chapters dealing with these devices.

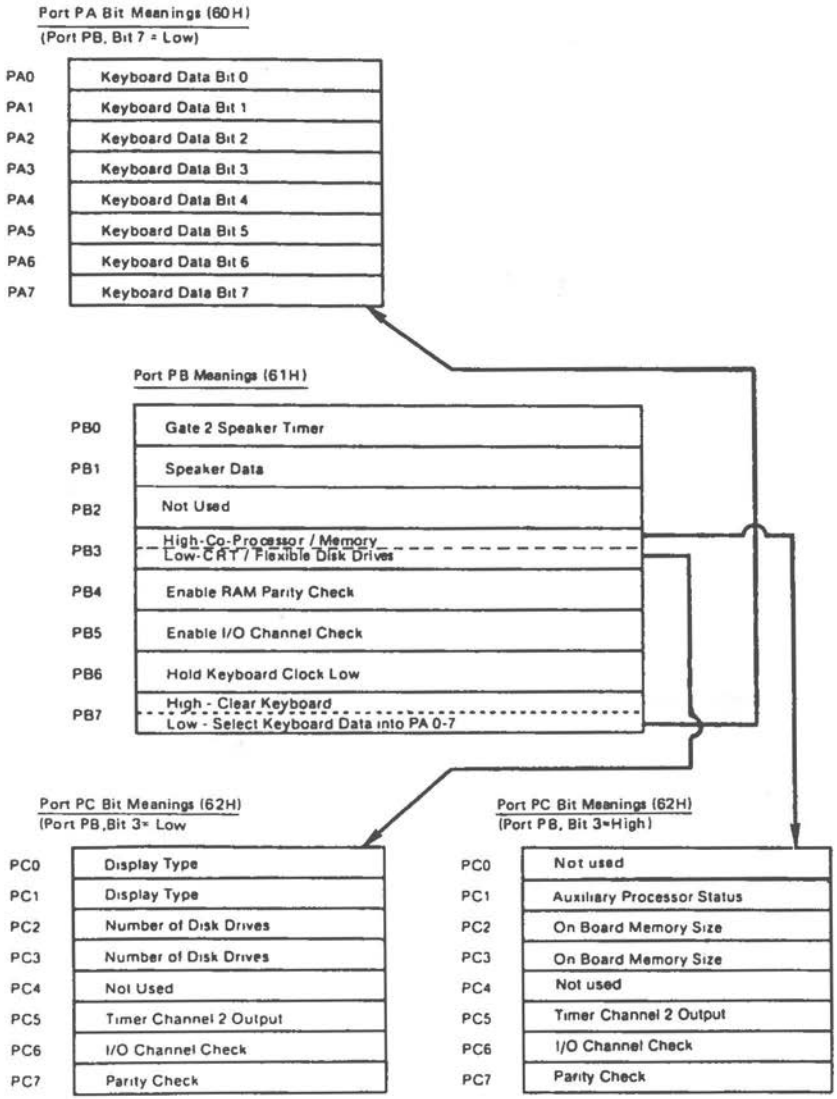


Figure 2.19 8255A: System Use

DIRECT MEMORY ACCESS

Data transfer between disk drives and random access memory is performed with the assistance of the 8237 DMA controller. The advantage of DMA transfer is that the system microprocessor (CPU) need specify only the initial RAM address, the number of bytes to be transferred, and certain items of control information, whereupon the transfer can take place without the CPU having to manage the transfer on a byte-by-byte basis. Bus arbitration prevents the data and addresses on the system busses being read by the CPU as part of its program. Obviously, the CPU cannot make use of the busses for read/write operations while the DMA transfer is in progress, but it can still perform internal arithmetic as well as processing instructions already waiting in its internal instruction queue.

The DMA controller can manage up to four channels, for which two types of priority logic can be set. In principle, any suitable peripheral device can be serviced by DMA. It is even possible to perform memory to memory transfers.

SYSTEM INTEGRATION

The pin configuration and internal logic of the 8237 are shown in Figure 2.20. The 8237 pin connections have the following significance:

Vcc
Power supply + 5V.

Vss
Ground.

CLK Input
Clock signal controlling internal functions and the data transfer rate.

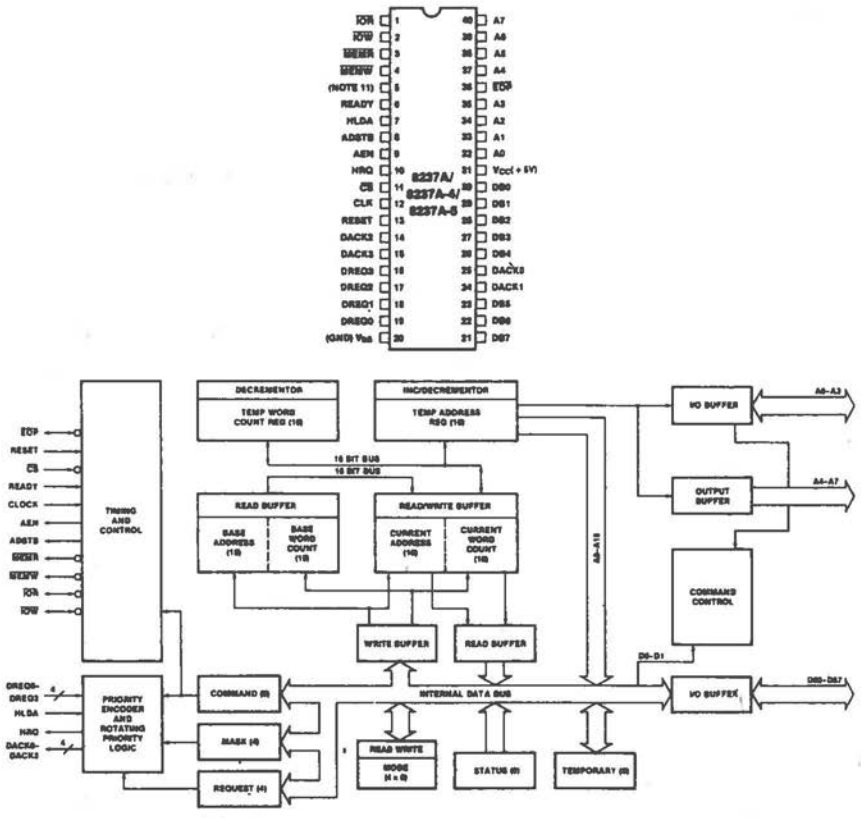


Figure 2.20 DMA Controller Overview

CS/ Input
Selects the DMA controller as a normal I/O device during its idle cycle. This allows the CPU to use the bus system (and program the DMA controller).

RESET Input
Clears the Command, Status, Mask, Request and Temporary registers (described in later section), as well as the first/last flip-flop. Following a reset, the DMA controller is in the idle cycle.

READY
Memory read/write puses are extended for the duration of this signal.

HLDA Input
The hold acknowledge signal from the CPU, indicating that it has relinquished control of the bus system.

DREQ0, DREQ1, DREQ2, DREQ3 Input
DMA service request line for each of the 4 DMA channels. The polarity of these lines is programmable, the RESET default active high. The request is valid only if the DREQ line remains active until the corresponding acknowledge (DACK) signal is asserted by the DMA controller.
Default priority is "fixed", that is, DREQ0 has the highest priority, DREQ3 the lowest.

DB0-DB7 Input/Output
During idle, the data bus is available for normal CPU operations, including the reading and writing of DMA controller registers. During DMA cycles (except memory to memory), the data bus transmits the 8 MSBs of the DMA **address**, which are strobed into an external latch by means of the ADSTB signal.

IOR/ Input/Output
In the idle cycle, this is an input control signal used by the CPU to read the control registers. In the DMA active cycle, it is an output control signal used to accept data from the peripheral device during a DMA read.

IOW/ Input/Output
In the idle cycle, this is an input control signal used by the CPU to write the 8237 registers. In the DMA active cycle, it is an output control signal used to load data to the peripheral device.

EOP (End of Process or Terminal Count)/ Input/Output
This is a bi-directional signal which terminates a DMA transfer. The signal may be external, or it is asserted by the DMA controller itself to indicate that the terminal count for a DMA channel has been reached. When this signal goes active through internal or external cause, the DMA request is considered to have been serviced. If Auto-Initialize is enabled, the Base registers are written to the

current registers of the channel concerned. The Mask and TC bits of the channel status are set (with Auto-Initialize, the Mask bit is unaffected).

A0-A3 Input/Output
In the idle cycle, these system bus address lines select 8237 registers. During DMA activity they function as the 4 LSBs of the address of the RAM location to be accessed.

A4-A7 Output
In conjunction with A0-A3, these address lines provide the 8 LSBs of the RAM address to be accessed (the 8 MSBs are placed on the data bus).

HRQ Output
With this signal, the DMA controller requests bus control from the CPU. Assuming the request line is not masked and is of adequate priority, a DREQ input to the DMA controller results in assertion of the HRQ signal. HLDA is the acknowledgement signal from the CPU.

DACK0, DACK1, DACK2, DACK3 Output
The DMA controller uses one of these signals to grant a DMA request to a peripheral device. Like the DREQ lines, they can be programmed as active high or low. Reset default is active low.

AEN Output
The Address Enable signal enables the 8-bit latch containing the 8 MSBs of the RAM address to be placed on the data bus.

ADSTB Output
Strobes the 8 MSBs of the RAM address available on the data bus into an external latch.

MEMR/ Output
Active during DMA read or memory to memory transfer.

MEMW/ Output
Active during DMA write or memory to memory transfer.

8237 REGISTERS

The I/O address area between 0 and 0FH is dedicated to the DMA controller. Figure 2.21 illustrates the registers which are written or read in the form of a single byte. Other registers require or return a 16-bit word representing an address or counter value. In each case, two CPU input or output instructions are required in order to specify or read the complete value. For each register an internal first/last flip-flop keeps note of which half of the word is about to be read or written: if this flip-flop is in a zero condition, the byte on the data bus represents the 8 LSBs of the word, otherwise the data byte is the 8 MSBs. The 16-bit registers are shown in Figure 2.22.

An additional I/O address area (80H-83H) is used for the selection of DMA pages. This is described in a separate section.

Direction	Port	Operation
IN	8	Read Status Register
OUT	8	Write Command Register
OUT	9	Write Request Register
OUT	0AH	Write Single Mask Register Bit
OUT	0BH	Write Mode Register
OUT	0CH	Clear First/Last Flip-Flop
IN	0DH	Read Temporary Register
OUT	0DH	Master Clear (equivalent to hardware reset)
OUT	0EH	Clear Mask register
OUT	0FH	Write All Mask Register Bits

Figure 2.21 8237 8-Bit Programmable Registers

Direction	Port	Address/Counter Operation	Channel
IN	0	Read Current Address	0
OUT	0	Write Base and Current Address	0
IN	1	Read Current Count	0
OUT	1	Write Base and Current Count	0
IN	2	Read Current Address	1
OUT	2	Write Base and Current Address	1
IN	3	Read Current Count	1
OUT	3	Write Base and Current Count	1
IN	4	Read Current Address	2
OUT	4	Write Base and Current Address	2
IN	5	Read Current Count	2
OUT	5	Write Base and Current Count	2
IN	6	Read Current Address	3
OUT	6	Write Base and Current Address	3
IN	7	Read Current Count	3
OUT	7	Write Base and Current Count	3

Figure 2.22 8237 16-Bit Programmable Registers

It is the responsibility of software initiating DMA to ensure that a DMA request is not granted while the DMA controller is being programmed. For example, it is important to prevent DMA from being performed on a channel for which half of a 16-bit address or counter value has not yet been appropriately written. This can be achieved by one of two methods: either mask the channel concerned, or disable the controller by setting bit 2 in the Command Register.

Command Register
This register is illustrated in Figure 2.23. It can be cleared by the Reset or Master Clear instructions.

Bit:							
7	6	5	4	3	2	1	0
DACK +/-	DREQ +/-	WS	PRTY	TIMING	ENABLE	CH 0 ADDR. H	MEM-MEM

Figure 2.23 DMA Command Register

Bits 7 and 6 determine the polarity of the DACK and DREQ signals (active low for DREQ is determined by setting this bit, active low for DACK requires a zero bit).

TIMING determines whether normal (0) or compressed (1) timing is active. The latter compresses the transfer time to 2 clock cycles (this is illustrated in the section "DMA Timing" in this Chapter). If compressed timing is selected, WS selects late (0) or extended (1) write selection.

PRTY is the priority scheme selection bit. Fixed priority (bit zero) means that DMA request priority is equivalent to the sequence of request lines, DREQ0 having the highest priority. Rotating priority has the effect that the DMA request most recently serviced assumes lowest priority and priority rotates. Therefore, no DMA request must ever wait more that 3 request services before itself being serviced.

ENABLE, when set, prevents the controller from entering DMA active mode. This command can be used to hold the controller in the idle state while the address and counter registers are being programmed. To re-enable the controller, write this bit as zero.

Bit 1 is the Channel 0 Address Hold bit. Setting this bit means that the same address is used for all transfers on this channel.

MEM-MEM set determines that a memory to memory transfer is to take place from the address specified for Channel 0 via the Temporary Register to the address specified for Channel 1 until the Current Count Register goes to 0FFFFH.

Mode Register

Mode selection information can be specified for any one of the 4 DMA channels. The actual channel selected is encoded in two bits of this register.

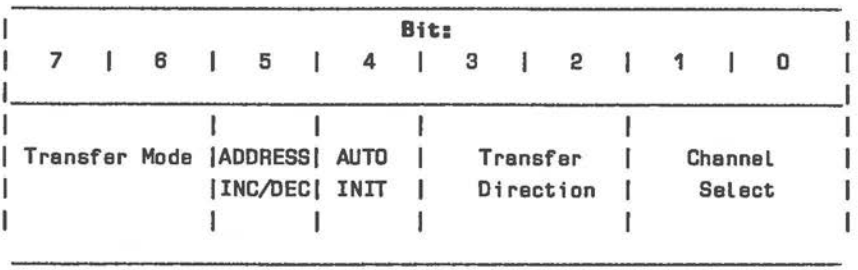


Figure 2.24 DMA Mode Register

Transfer Mode is a value in 2 bytes specifying Single Transfer (1), Block Transfer (2), or Demand Transfer (0). (A value of 3 would denote an 8237 cascade configuration, which is not present in the NCR PERSONAL COMPUTER.)

In Single Transfer mode, only one transfer is made. The Current Address Register is then altered by 1 and the Current Count Register is decremented. Even if the DREQ line is still active at completion of the transfer, bus control is released to the CPU. If DREQ is then still active, a further transfer is performed, and so on. Therefore, there is at least one machine cycle between DMA transfers. If the Current Count Register decrements from 0 to 0FFFFH (Terminal Count) and Auto-Initialize is programmed (bit 4), an Auto-Initialize then occurs.

Block Transfer mode transfers continuously until

Terminal Count or an external End of Process signal, whereupon Auto-Initialize occurs, if programmed. For this transfer mode, the DREQ signal need remain active only until the corresponding acknowledgement (DACK) is received.
In Demand Transfer mode the transfer runs continuously until Terminal Count or an external End of Process signal occurs. If programmed, Auto-Initialize then occurs. During the transfer, the DREQ signal must be held active. If DREQ goes inactive, the transfer is suspended, but not terminated. Re-activating DREQ rseumes the transfer process.

The ADDRESS INC/DEC bit determines whether the value in the Current Address Register is incremented (0) or decremented (1) after each transfer.

AUTO INIT, if set, enables the Auto-Initialize mode of operation. Auto Initialize can then take place following an End of Process or Terminal Count condition. During Auto-Initialize the original values of the Current Address and Current Count Registers are restored using the values held in the Base Address and Base Count Registers (the Base registers are not affected by DMA transfer operations).

The Transfer Direction value determines whether data is to be transferred to (1) or from (2) memory, or whether data is only to be verified (0). In the verify mode, memory is not actually affected.

The Channel Select value selects one of the DMA channels 0-3.

Request Register
Assuming that Block Transfer is selected in the Mode Register, the Request Register can be used to initiate a DMA transfer, as if a hardware transfer request had been received on DREQ line. The channel to which this "software" DMA request applies is selected by the binary value of the two LSBs written to this register, with Bit 2 set. If Bit 2 is zero when this register is written, the request is cleared (other bits are "don't care"). This type of

DMA request is not subject to the Mask Register (see below), but it is subject to priority control. A Terminal Count or external End of Process signal terminates the transfer in the normal way.

Mask Register
This register determines which DMA requests are inhibited (masked). One of two registers can be used to mask DMA requests, namely, Write Single Mask Register Bit (port 0AH) and Write All Mask Register Bits (port 0FH).

Write Single Mask Register Bit selects a channel by means of the binary value in the 2 LSBs of that register; Bit 2 sets or clears the mask bit for the selected channel, bits 3-7 are "don't care".

Write All Mask Register Bits sets or clears the mask bits for Channels 0-3 according to the status of bits 0-3, respectively. Bits 4-7 are "don't care".

A Clear Mask Register command allows all four DMA requests.

In addition to these posssibilities of software control, the following signal conditions affect the Mask Register:

* A mask bit is set when its channel produces an End of Process condition, if that channel is not programmed for Auto-Initialize.

* A Reset inhibits all DMA requests.

Status Register
The Status Register is illustrated in Figure 2.25. This register states which channels, if any, have received DMA requests, and whether on individual channels Terminal Count conditions or external End of Process signals have occurred. Bits 0-3 are cleared immediately upon the register being read and at Reset. Bits 4-7 are set whenever service is being requested on the corresponding channel.

Bit:							
7	6	5	4	3	2	1	0
DMA Request on Channel:				TC/EOP on Channel:			
3	2	1	0	3	2	1	0

Figure 2.25 DMA Status Register

Temporary Register
The Temporary Register always contains the last byte transferred in a memory to memory transfer operation. This register can be read with the Read Temporary Register Command (port 0DH) and is cleared at Reset.

Current Address Register
This register is a running record of the address used for the byte currently being transferred. This address value is incremented or decremented (in accordance with bit 5 of the Mode Register) after each byte transfer. Auto-Initialize restores the programmed value.

Current Count Register
This register functions as a counter to limit the number of bytes transferred. It should be programmed with a binary value which is one less than the number of transfer bytes. Terminal Count occurs as soon as this register has decremented from zero to 0FFFFH. Auto-Initialize restores the programmed value.

Base Address and Base Count Registers
These are the registers used by the DMA controller to store the values programmed for the Current Address and Current Count registers. These values are required by the controller in the event of an Auto-Initialize. It is not required to program the Base registers, as they are automatically set to the values of the Current Address and Current Count Registers when the latter are programmed by the CPU. It is not possible to read the Base registers under CPU control.

DMA TIMING
The Figures in this section illustrate the time-base co-ordination between the various signals involved in DMA transfers.

Note that memory to memory transfers require a read and write phase.

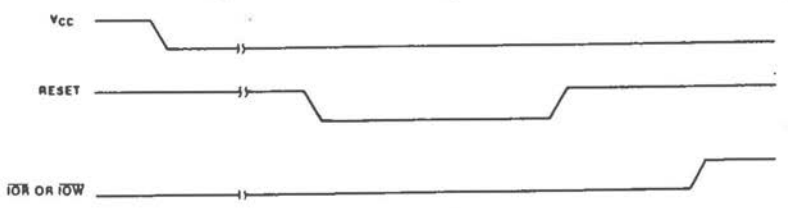


Figure 2.26 DMA Reset

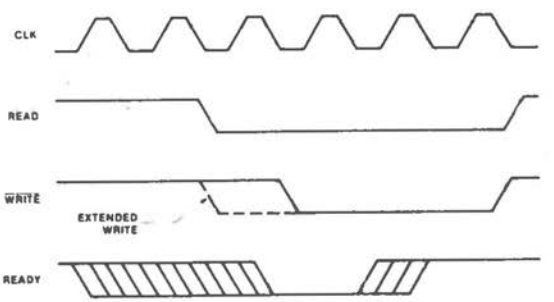


Figure 2.27 DMA Ready

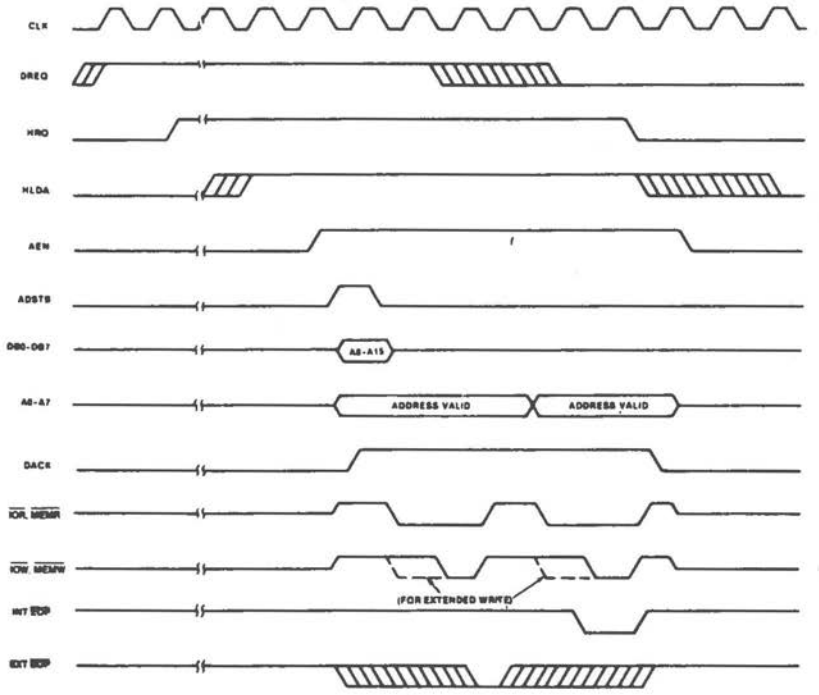


Figure 2.28 DMA Transfer

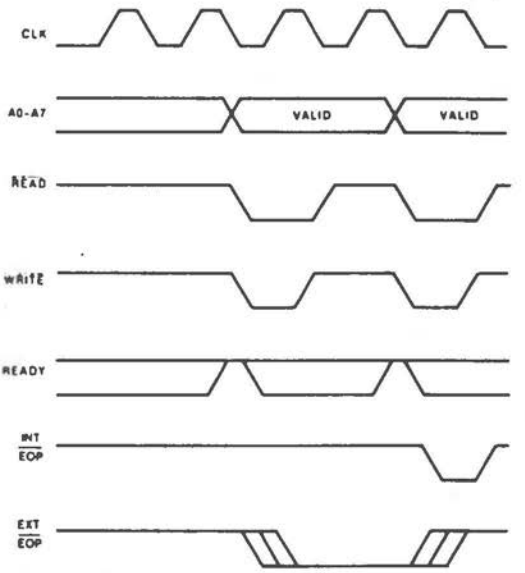


Figure 2.29 DMA Compressed Transfer

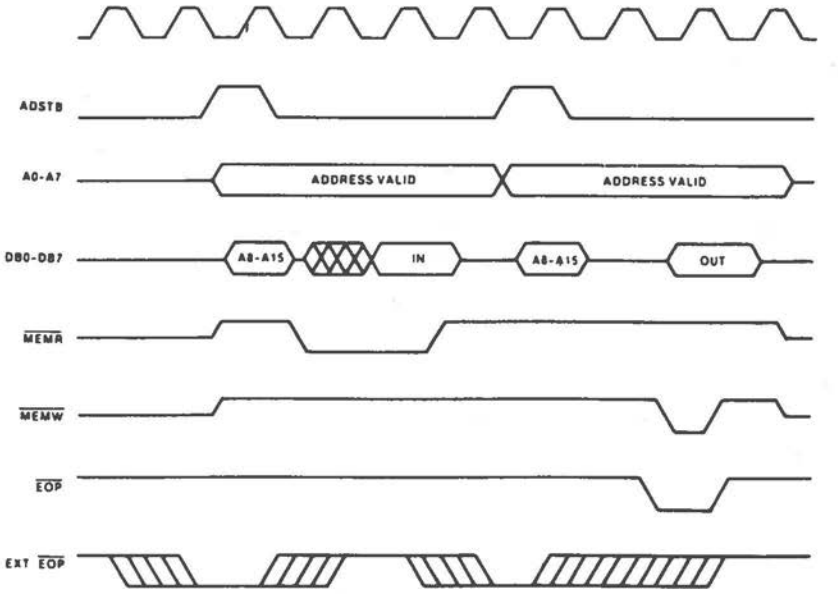


Figure 2.30 DMA Memory To Memory Transfer

8237: SYSTEM USE
Two of the four channels of the DMA controller are dedicated to system functions of the NCR PERSONAL COMPUTER:

* Channel 0 is used for RAM refresh.

* Channel 2 provides for data transfer to and from the flexible disk controller.

You will have noticed that the Address registers of the DMA controller are capable of addressing 64 KB of random access memory, while your NCR PERSONAL COMPUTER contains at least 256 KB. It is, however, possible to access all RAM areas by means of page switching. This is discussed in the section "DMA Page Selection".

RAM Refresh
It is a characteristic of dynamic random access memory that it requires power refreshing at regular intervals, in order to ensure data integrity.

It is the task of DMA controller to address random access memory during the refresh cycles. Because of the importance of these refresh cycles, they are accorded the highest priority (DMA Channel 0) in the "fixed" priority logic. The regularity of this process is ensured by the DMA request for this channel being derived from output 1 of the 8253 Programmable Interval Timer (output every 15.1 microseconds). Due to way in which RAM integrated circuits are selected (see Schematics in Appendix A), the DMA controller does not have to provide fully decoded addresses. Therefore, there is no need for page switching during RAM refresh cycles.

After initialization by the system, RAM refresh requires no further CPU attention. For your information, the initialization steps are summarized in the remainder of this section discussing RAM refresh. However, you should take care not to disturb Timer 1 and DMA Channel 0 settings, as alterations might lead to loss of system control. It is also

inadvisable to change the priority logic of the DMA controller, as this too could seriously impair the reliability of refresh cycle intervals.

1. Initialize the Programmable Interval Timer. The following instruction selects Timer 1, determines that only the lower byte of the clock is to be written, sets output Mode 2 (Rate Generator) and the binary (not BCD) mode of counting:

```
MOV  AL,01010100B
OUT  43H,AL
```

   The next instruction loads the refresh divisor into Timer 1, so that an output signal occurs every 15.1 microseconds:

```
MOV  AL,12H
OUT  41H,AL
```

2. Perform a DMA controller Master Clear (the value in AL is insignificant):

```
OUT  0DH,AL
```

3. Set the Base and Current Counter for Channel 0 to 64 KB (the first/last flip-flop was cleared by the Master Clear and toggles automatically after the DMA has received the lower byte):

```
MOV  AL,0FFH
OUT  1,AL
OUT  1,AL
```

4. Write the Mode Register - single transfer, address is incremented, Auto-Initialize is enabled, transfer direction is read, channel is 0:

```
MOV  AL,01011000B
OUT  0BH,AL
```

5. Write the Command Register - DACK and DREQ lines are active low,no extended write, priority is "fixed", timing is normal, controller is enabled, no Channel 0 address hold, transfer is not memory to memory:

```
MOV  AL,0
OUT  8,AL
```

6. Finally clear the mask register bit for Channel 0, so that DMA requests for that Channel can be serviced. This is best achieved by writing a Single Mask Bit (value zero in 2 LSBs specifies Channel 0, zero in Bit 2 clears mask for that channel only):

```
MOV  AL,0
OUT  0AH,AL
```

Flexible Disk DMA Service

Channel 2 of the DMA controller is dedicated to servicing I/O operations requested by the flexible disk controller on the main processor board.

This section illustrates DMA controller programming for the flexible disk I/O operations read, write, and verify. The Command Register need not be written explicitly, as the setting required for RAM refresh still applies.

1. Write the Mode Register to port 0BH. The value written depends on whether a disk read, write, or verify operation is to take place:

   Disk Read - 46H. This specifies single transfer, address increment (decrement may be specified if required), no Auto-Initialize (otherwise Base Address and Counter would be restored to programmed values after each transfer), transfer direction is write (i.e. to memory), Channel is 2.

Disk Write or Format - 4AH. This specifies the same mode of operation as for Disk Read, except that the transfer direction is now read (i.e. **from** memory).

Disk Verify - 42H. This specifies the same mode of operation as for Disk Read, except that transfer direction is set to verify.

2. The DMA page number must now be specified. This binary value species in which 64 KB block of memory the data is to be read or written. For example,

```
MOV  AL,1
OUT  81H,AL
```

specifies that the Base and Current Address registers refer to random access memory at locations offset to the beginning of the first 64 KB boundary from absolute address 10000H onwards. (DMA pages are described in the next section.)

3. You should now ensure that the first/last flip-flop is cleared, so that subsequent Address and Counter register write operations address first the lower, then the upper byte (the value in AL is insignificant):

```
OUT OCH,AL
```

4. Transmit low and high bytes to the Current Address and Counter registers (this automatically sets the Base registers). Remember that the Address Register value is the offset to the DMA page boundary calculated at step 2. Furthermore, the entire transfer specified for the Current Count Register must be able to take place within the confines of that page.

```
        MOV  DX,4
        MOV  AL,ADDR_LO
        OUT  DX,AL
        MOV  AL,ADDR-HI
        OUT  DX,AL
;
        INC  DX
        MOV  AL,COUNT_LO
        OUT  DX,AL
        MOV  AL,COUNT_HI
        OUT  DX,AL
```

5. Finally, clear the mask bit for Channel 2 (the mask bit will have been set at End of Process/Terminal Count of the previous pass through these procedures, because Auto-Initialize is disabled):

```
MOV  AL,2
OUT  0AH,AL
```

DMA Page Selection

At first sight, it would seem that DMA controller activity is confined to a 64 KB block of memory, by virtue of the fact that programmable Address values are 16-bit. To overcome this apparent limitation, your NCR PERSONAL COMPUTER includes an LS670 4-by-4 register file, of which the RA and RB input lines are connected to the DMA controller (see Schematics in Appendix A).

The LS670 RA and RB lines derive their input from the 8237 DMA request acknowledgement lines DACK3 and DACK2, respectively. These lines can be both inactive at the same time, or one can be inactive while the other is active. Selected by means of the system port address 82H for Channel 3 and 81H for Channel 2, a DMA page number can be written to the LS670 via the four least significant data bus lines. This data value drives the four most significant lines on the system address bus (A16-A19), so that, in principle, the entire 8088 memory address area can be accessed

by DMA. This is a particularly useful feature if you wish to perform DMA on video memory.

The 4-bit value for output via the DMA page selection port for flexible disk I/O (81H) is as follows:

0	Absolute address range	00000H-0FFFFH
1		10000H-1FFFFH
2		20000H-2FFFFH
3		30000H-3FFFFH

and so on.

Because the address lines are driven without any intervening logic, it is important to note that the 64 KB memory blocks thus selected are separated from one another at multiples of 64 KB, starting from the lowest absolute machine location. It is not possible to define CPU-like "segments" at convenient paragraph boundaries. If the memory area you wish to access by DMA straddles one of these page boundaries, you will have to execute two DMA operations, of which the second specifies an incremented (or decremented) DMA page value (do not forget to ensure that suitable values are written to the DMA controller Address and Count registers).

The Operating System

INTRODUCTION

This Chapter provides a quick-refernce summary of Basic Input/Output System system functions. The functions described in the first section, "The ROM BIOS", are those most immediately concerned with handling the hardware of your NCR PERSONAL COMPUTER. The corresponding BIOS routines are contained in read only memory from machine address 0F0000H upwards. There is no ROM switching, therefore the BIOS routines are always accessible in the NCR PERSONAL COMPUTER memory map.

The second section, "BIOS Data Areas", provides a list of those random access memory locations used by the BIOS for the maintenance of variable system parameters.

Other routines, largely dedicated to directory and file management, are supplied when the disk operating system is loaded. Access to these routines is fully described in the NCR-DOS programmer's information included in the Macro-Assembler Manual.

The BIOS routines are activated by means of the hardware and software interrupts in the 8088 interrupt vector. Eight hardware interrupts are managed by the Programmable Interrupt Controller. These are the interrupt types 8-0FH. The other interrupts are issued by means of the INT instruction. ROM BIOS does not write the interrupt vector for all interrupt types, but provides a number of "dummy" interrupt returns. Those vector positions not written can, however, be written by code on read only memory on additional adapters (e.g. fixed disk controller), or subsequently by disk operating system software or individual applications.

The disk operating system NCR-DOS provides system function calls by which the interrupt vector can be read (Function Call 35H) and written (Function Call 25H). If you are writing applications which do not require the disk operating system, there is no reason why you should not read and write the interrupt vector directly. However, you should then ensure that a service request cannot be granted to the Programmable Interrupt Controller while the vector is being accessed, either by clearing the microprocessor Interrupt Flag and/or masking the 8 interrupt lines to the Programmable Interrupt Controller. You should observe this precaution, even if you are accessing only a software interrupt vector entry, as software interrupts can themselves be activated during the interrupt service routine of a hardware interrupt. This is the case, for example, with the interrupt types 1BH and 1CH, which are activated within the ROM BIOS ISRs for the Keyboard (type 9) and Timer 0 (type 8) interrupts, respectively.

For the most part, the ROM BIOS interrupts provide routines essential to the basic hardware functions of the computer. If you are considering writing the interrupt vector in order to substitute an interrupt service routine of your own for one available in the ROM BIOS or supplied by the disk operating system, you should ensure adequate replacement for vital system functions, except where you are sure that a function is not required for your particular application.

Finally, a note on "compatibility". Your NCR PERSONAL COMPUTER is operationally compatible with a number of other computers in the NCR PERSONAL COMPUTER range, as well as a number of other widely used professional computers. Accordingly, you may wish your applications, written principally for your NCR PERSONAL COMPUTER, to be capable of executing on a compatible machine. For this reason, you should always use the system interrupts as the means of accessing system functions. With the following exceptions, you should not refer directly to machine addresses:

* The 8088 interrupt vector in the machine address range 0000-03FFH
* System parameters described in the section "BIOS Data Areas" in this Chapter
* Offset to address in interrupt vector entry 1F of the character set bit patterns
* The power-on address: F000:FFF0
* The first address of power-on tests: F000:E000
* The BIOS release date at F000:FFF5

ROM BIOS

This section provides a guide to the 8088 interrupt vector, as used by the NCR PERSONAL COMPUTER. Details of entry and return parameters are included for ROM BIOS interrupts. Interrupt types 0-4 are dedicated to CPU functions. Apart from type 2 (NMI), the sources of these interrupts are common to all 8088 systems and cannot be influenced by the hardware or software configuration.

Memory addresses given in this section are absolute machine addresses, unless otherwise stated. Byte, word, and double word values are indicated by the abbreviations DB, DW, and DD, respectively. Registers (except those used to return results) are preserved by interrupt service routines, unless otherwise stated.

Interrupts initialized and used by the disk operating system routines resident in random access memory are also indicated.

Interrupt (Hex)	Description
0	CPU Divide by Zero. This error condition is most likely the result of a defective arithmetic operation executed by an application program. This interrupt is not serviced in the ROM BIOS. Normally, there is no need to service this interrupt, beyond a courtesy message and CPU Halt, as such an error is to be considered unrecoverable
1	Single Step. This interrupt, normally used in software development, is issued at the end of an instruction cycle, if the CPU Trap Flag is set
2	Non-maskable Interrupt. The ROM BIOS ISR checks whether the cause is RAM parity or external. The ISR is concluded by a courtesy message and CPU Halt
3	Breakpoint. This interrupt is issued by the one-byte CPU interrupt instruction (not serviced by ROM BIOS)
4	Overflow. CPU integer overflow (not serviced by ROM BIOS)
5	Send current contents of character screen to printer Return: DB at 500H - 0 = not busy 1 = busy 0FFH = error (as ret. by INT 17H)
6	Reserved
7	Reserved
8	Service Programmable Interval Timer output 0 Execute secondary ISR int. type 1CH Return: DD at 46CH - counter incremented DB at 470H - set to value 1 at counter overflow DB at 440H - decrement motor time-out counter

Interrupt (Hex)	Description
9	Service keyboard (key has been depressed) Execute secondary ISR int.type 1BH Return: DB at 417H - bit flags: 0 right shift depressed 1 left shift depressed 2 ctrl depressed 3 alt depressed 4 scroll toggled 5 num lock toggled 6 caps toggled 7 ins toggled DB at 418H - bit flags: 4 wait toggled 5 num depressed 6 caps depressed 7 ins depressed DB at 419H DW at 41AH DW at 41CH Keyboard Buffer at 41EH - see "BIOS Data Areas"
A	Reserved
B	Hardware interrupt for serial I/O (COM2).
C	Hardware interrupt for serial I/O (COM1).
D	Used by disk operating system for fixed disk.
E	Service interrupt from flexible disk controller. This consists of flagging the interrupt to indicate that drive requires re-calibration: Return: DB at 43EH - bit flag: 7 interrupt noted
F	Hardware interrupt for printer.

Interrupt (Hex)	Description

10 Video I/O.

Entry:

AH = 0, then set video mode according to value
AL = 0 40 x 25 character monochrome
1 40 x 25 character color
2 80 x 25 character monochrome
3 80 x 25 character color
4,5 320 x 200 graphics color
6 640 x 200 graphics monochrome
[7 internal use]
8 640 x 400 graphics monochrome
9 640 x 400 graphics color

Return: screen cleared
screen data areas initialized, see next section

AH = 1, then set cursor style according to value
CH = top scan line
CL = bottom scan line

Return: DW at 460H - CX as at entry
cursor style set

AH = 2, then set char. cursor pos. according to value
DH = row
DL = column
BH = display page

Return: cursor position set if BH was current page
DX as at entry stored in screen data area,
see next section

AH = 3, then get char. cursor pos. according to value
BH = display page

Return: row in DH
column in DL
cursor style in CX

Interrupt (Hex) | **Description**

10 (cont.)

AH = 4, then get light pen position

Return: AH = 0 position not available
1 position available -
char. row in DH
char. column in DL

or

raster pos. (modes 0-6) in CH
(value 0-199)
(raster pos. (modes 8,9) in CX
(value 0-39))
horiz. pixel pos. in BX (value
0-319/639 approx.)

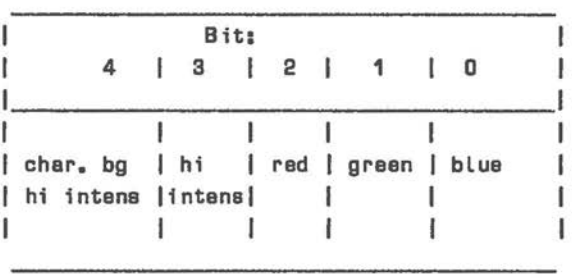
AH = 5, then select active display page according to
AL = new active display page
minimum value = 0 (page 1)
maximum no. of pages:

Vid. Mode	Video memory: NCR 64KB	NCR 32 KB	16 KB
0-1	8	8	8
2-3	8	8	4
4-6	4	2	1
8	2	1	—
9	1	—	—

Return: new page active
cursor at position stored for that page
in screen data area

Interrupt (Hex)	Description
10 (cont.)	AH = 6, then scroll block of active page up a number of character rows: AL = no. of rows, or 0 (blank block) CH = top left row co-ordinate of block CL = column DH = bottom right row co-ordinate of block DL = column BH = attr. byte for blank char. used in vacated row AH = 7, then scroll block of active page down one character row. Parameters as for scroll up. AH = 8, then read character/attribute at cursor pos. BH = display page Return: AL = character AH = character attribute (char. modes only) AH = 9, then write character at cursor position, if character mode, also write attribute AL = character BL = attribute BH = display page CX = count of characters to write Return: character/attribute written AH = 0AH, then write character at cursor position AL = character BH = display page CX = count of characters to write Return: character written

Interrupt [Hex]	Description
10 [cont.]	AH = 0BH, then set color palette according to BH = 0 select background color BL = color selection in 5 LSBs:



or

BH = 1 select foreground palette
BL = 0 (green/red/yellow) or 1 (blue/cyan/magenta)

Return: DB at 466H - copy of CRT controller Color Select Register
color selected

AH = 0CH, then write graph. dot in active display page
AL = dot value (color info.), bit 7 set means XOR dot value with current display
DX = pixel row
CX = pixel column

Return: graphics dot written

AH = 0DH, then return dot value from active displ. page
DX = pixel row
CX = pixel column

Return: AL = dot value

Interrupt (Hex)	Description
10 (cont.)	AH = 0EH, then write teletype to active screen page, i.e., write at cursor pos., advance cursor pos., wrap round to new line if end of line, scroll up one line if end of page AL = character to write (7,8,0AH,0DH are interpreted as commands, not characters) BL = foreground color (graphics) Return: character written AH = 0FH, then get current video state Return: AL = video mode AH = no. of display columns on screen BH = active display page
11	Get equipment flag (see section "BIOS Data Areas") Return: AX = equipment flag
12	Read RAM size Return: AX = RAM size in KB

Interrupt (Hex)	Description

13 Disk I/O, refers to fixed disk I/O if DL bit 7 is set at entry, to flexible disk I/O if this bit is zero.

BIOS flexible disk I/O:

```
Entry: AH = 0  reset disk
            1  read status of previous operation
            2  read sectors into memory
            3  write sectors into memory
            4  verify sectors
            5  format track
       AL = no. of sectors          }
       ES = segment of buffer       }
       BX = offset of buffer        }
       DL = drive (0-3)             }     for
       DH = head (0-1)              }  read/write/
       CH = track (0-39/79)         } verify/format
       CL = sector (1-8/9/15)       }
       disk parameter table pointer }
            (see Interrupt 1EH)     }

Return: Carry Flag zero if successful
                 set  if error, and
              AH =   1  bad command
                     2  address mark not found
                     3  disk write-protected
                     4  sector not found
                     8  data overrun
                     9  DMA 64 KB boundary error
                    10H CRC error on reading
                    20H flex. disk controller failure
                    40H error on seek
                    80H flex. disk controller timeout
```

Interrupt (Hex) | **Description**

14 Serial I/O (CPU registers corrupted)

Entry:

AH = 0, then initialize interface according to bits in AL:

BAUD	PARITY	STOP	CHAR. LENGTH
7 6 5	4 3	2	1 0
0 0 0 = 110	0 0 = none	0 = 1	1 0 = 7
0 0 1 = 150	0 1 = odd	1 = 2	1 1 = 8
0 1 0 = 300	1 0 = none		
0 1 1 = 600	1 1 = even		
1 0 0 = 1200			
1 0 1 = 2400			
1 1 0 = 4800			
1 1 1 = 9600			

DX = serial COM1 or COM2 device (0 or 1)

Return: AH = Line Status bit flags
- 7 timeout
- 6 transm. shift reg. empty
- 5 transm. hold reg. empty
- 4 break detected
- 3 framing error
- 2 parity error
- 1 overrun error
- 0 operation successful

AL = Modem Status bit flags
- 7 RLSD
- 6 RI
- 5 DSR
- 4 CTS
- 3 DRLSD
- 2 TERI
- 1 DDSR
- 0 DCTS

Interrupt (Hex)	Description
14 (cont.)	AH = 1, then transmit character AL = character DX = serial COM device (0 or 1) Return: AH = bit flag 7 timeout AL = Modem Status (see above) AH = 2, then receive character DX = serial COM device Return: AL = character AH = bit flag 7 timeout AH = 3, then read status DX = serial COM device Return: AH = Line Status (see above) AL = Modem Status (see above)
15	Reserved
16	Read Keyboard (Flags not restored) Entry: AH = 0, then wait for and read character from keyboard Return: AL = ASCII character AH = scan code AH = 1, then read keyboard buffer (non-destructive) Return: Zero Flag = 1 - no character in buffer 0 - AL = ASCII character AH = scan code

Interrupt (Hex)	Description
16 (cont.)	AH = 2, then get keyboard shift status Return: AL = bit flags 0 right shift depressed 1 left shift depressed 2 ctrl depressed 3 alt depressed 4 scroll toggled 5 num toggled 6 caps toggled 7 ins toggled
17	Parallel printer interface Entry: AH = 0 print character AL = character 1 initialize interface 2 read status only DX = 0,1,2, or 3: 1 of 4 printer base ports, 0 takes base port from DW 408H 1 DW 40AH 2 DW 40CH 3 DW 40EH Return: AH = status bit flags 7 busy 6 acknowledge 5 out of paper 4 select 3 error 0 timeout
18	Reserved
19	Boot loader, reading disk drive 0, head 0, track 0, sector 1 to machine address 7C00H. The disk read operation (INT 13H) would normally be from flexible disk. However, if ROM BIOS initialization detected a fixed disk drive, that disk is read (see "System Initialization" in this Chapter).

Interrupt (Hex)	Description

1A Read or set binary clock counter.

Entry:

AH = 0, then read clock counter

Return: CX = upper word of counter
DX = lower word of counter
AL = 0 (no timer wrap) or 1 (timer wrap)
DB at 470H (timer wrap) zero.

AH = 1, then set clock counter according to values in
CX = upper word of counter
DX = lower word of counter

Return: DD at 46CH (counter) = new value
DB at 470H (timer wrap) = 0

1B Keyboard Control-Break interrupt service, issued during interrupt type 9 (key depressed) service. Used by disk operating system.

1C User timer interrupt, always issued during service of interrupt type 8 (Timer 0).

1D Contains segment:offset address of the beginning of the video parameter table. These are the parameters used for initialization of the 6845 controller. This vector entry is not used to refer to code, therefore an interrupt of this type must not be issued. The table has 3 entries each of 16 bytes for

Modes 0 and 1
Modes 2 and 3
Modes 4, 5, 6 and 8

in that order. Each entry consists of 16 bytes for the 16 display controller registers (lowest bytes is for register 0, lowest byte but one for register 1, etc.

Interrupt (Hex)	Description
1E	Contains segment:offset of the flexible disk parameter table. This vector entry is not used to refer to code, therefore an interrupt of this type must not be issued. The table provides parameters for the flexible disk controller and the corresponding ROM BIOS routine. The table is built up as follows: DB Step rate (SRT) and head unload time (HUT) DB Head load time (HLT) and DMA/non-DMA mode selection DB Motor wait time DB Bytes/sector selection DB Final sector number DB Gap length (GPL) DB Data length (DTL) DB Length of format Gap 3 DB Format filler byte DB Head settle time DB Motor start time
1F	Contains segment:offset of graphics character bit patterns (character codes 80H-0FFH) in random access memory (see Chapter 7). These patterns, and the interrupt vector entry, are loaded by the NCR-DOS GRAFTABL utility, but they can be supplied by individual applications. This vector entry is not used to refer to code, therfore an interrupt of this type must not be issued.
20	NCR-DOS program terminate.
21	NCR-DOS function call.
22	NCR-DOS terminate address.
23	NCR-DOS Control-Break exit address.
24	NCR-DOS fatal error address.

Interrupt (Hex)	Description
25	NCR-DOS absolute disk read.
26	NCR-DOS absolute disk write.
27	NCR-DOS terminate but stay resident.
28-3F	Reserved for NCR-DOS.
40	Reserved for re-direction of disk processing from flexible disk to fixed disk routines (see Int. 13H).
41	Contains segment:offset of the parameter table for a fixed disk drive. This vector entry is not used to refer to code, therefore an interrupt of this typemust not be issued. The table is built up as follows: DW Maximum number of cylinders DB " " " heads DW Cylinder where reduced write-current starts DW Cylinder where write pre-compensation starts DB Maximum ECC data burst length DB Control byte: bit 7 = 1 - disable access retries 6 = 1 - " ECC retries 5,4,3 = 0 2,1,0 = drive option DB Standard time-out DB Time-out for formatting drive DB " " checking drive DB } DB } Reserved DB } DB }
42-5F	Reserved.
60-7F	User available.
80-F0	Reserved.
F1-FF	User available.

BIOS DATA AREAS

Situated in memory immediately above the interrupt vector and below the disk operating system (when loaded) are a number of data areas used by system I/O and memory management functions.

This section presents a summary of system use of these areas. Where binary numeric data is stored, this is indicated by the appropriate data declaration type DB, DW, or DD.

Type	Location (Absolute Hex)	Description
DW	400	Base port addresses for up to 4 serial I/O
DW	402	interfaces.
DW	404	
DW	406	
DW	408	Base port addresses for up to 4 parallel
DW	40A	printer interfaces.
DW	40C	
DW	40E	
DW	410	Equipment Flag bits:

14,15	no. of installed parallel printer interfaces
13	1 = no NCR graphics adapter installed
12	1 = games card installed
9-11	no. of installed serial I/O channels
8	1 = char. generator installed for 640 x 400 modes (8 and 9)
6,7	no. of flexible disk drives - binary value 0 = 1 drive 1 = 2 drives 2 = 3 drives 3 = 4 drives

Type	Location (Absolute Hex)	Description
		4,5 type of monitor – binary value 0 = serial 1 = 40 x 25 2 = 80 x 25 color 3 = 80 x 25 mono 3 always set 2 always set 1 1 = serial monitor installed 0 always set
DB	412	Reserved.
DW	413	Memory size in KB.
DW	415	Reserved.

DB	417	Keyboard Flag 1 (see Interrupt 9).
DB	418	Keyboard Flag 2 (see Interrupt 9).
DB	419	Alternate keypad buffer.
DW	41A	Keyboard out buffer pointer.
DW	41C	Keyboard in buffer pointer.
	41E–43D	Keyboard buffer.

DB	43E	Flexible disk recalibration bits: 0 drive 0 1 drive 1 2 drive 2 3 drive 3 A bit is set to show that no recalibration is required for that drive.

Type	Location (Absolute Hex)	Description
DB	43F	Flexible disk motor status bits (only one bit set at a time): 0 set = drive 0 on 1 set = drive 1 on 2 set = drive 2 on 3 set = drive 3 on 7 set = write operation
DB	440	Timeout counter for motor, decremented by ISR for Timer Counter 0 (int. type 8).
DB	441	Flexible disk operation status bits: 0 command error 5 FDC failure 7 timeout 3 and 0 both set can indicate DMA error (straddled 64 KB boundary)
	442-448	Storage of up to 7 bytes from FDC Result Phase (see Chapter 6).

DB	449	CRT Mode (see interrupt 10H).
DW	44A	Number of display columns.
DW	44C	Size of display buffer (bytes).
DW	44E	Starting address in display buffer (see description of CRT controller in Chapter 7).
	450-45F	Cursor position information for up to 8 screen pages. First page cursor is at 450-451; column is lower byte, row is upper byte.
DW	460	Cursor style, one bit for each scan line.

Type	Location (Absolute Hex)	Description
DB	462	Current display page.
DW	463	Base port address or current display adapter.
DB	465	CRT controller Mode Control Register (see Chapter 7).
DB	466	Color palette setting, written during interrupt 10H (entry: AH=0BH) service.
DW DW	467 469	This is a segment:offset pointer to the first byte of code of initialization routines contained in ROMs on adapters not included on the main processor board (see section "System Initializaion"). Segment value is the higher word.
DB	46B	Interrupt type of hardware interrupt most recently serviced by the ROM BIOS dummy ISR. This byte is all bits set for a software interrupt.
DD	46C	Counter incremented by Timer Counter 0 ISR (interrupt type 8).
DB	470	Timer overflow flag. Value 1 denotes timer overflow. This byte is written zero when system time is read or written (interrupt type 1AH).

Type	Location (Absolute Hex)	Description
DB	471	Break indicator: bit 7 set = Break key pressed
DW	472	Ctrl-Alt-Del indicator: word value 1234H if reset in progress
	474-477	Fixed disk data area used during service of int 13H.
DB	478	Timeout counters for up to four parallel printer channels.
DB	479	
DB	47A	
DB	47B	
DB	47C	Timeout counters for up to four serial I/O channels.
DB	47D	
DB	47E	
DB	47F	
	480-483	Additional keyboard buffer pointers.
	490-4EF	Reserved.
	4F0-4FF	Available for passing data between program segments. [True segment:offset addresses cannot otherwise be used, as the loading segment value under the operating system can vary.]

Type	Location (Absolute Hex)	Description
DB	500	Printer status values: 0 not busy, no error 1 busy 0FFH error
DB	504	Single drive status (A or B).
	510-51D	Reserved
	C0000-C7FFF CD000-EFFFF	ROM expansion.
	C8000-CCFFF	Fixed disk control.
	F0000-FFFFF	NCR PERSONAL COMPUTER ROM BIOS.

SYSTEM INITIALIZATION

This section draws attention to one particular aspect of system initialization: the presence of additional adapters installed in the computer. This information is of particular interest if you are designing your own adapters for the NCR PERSONAL COMPUTER.

After initializing the programmable integrated circuits on the main processor board before booting the disk system, the initialization procedure looks for read only memory, starting at the machine address 0C8000H up as far as 0FE000H, which is the lowest address occupied by the ROM BIOS. Any adapter which wishes to execute code from its own ROM can identify itself to the main processor board initialization routine.

The main processor board initialization routine looks for the word identifier 0AA55H at machine address 0C8000H. If the identifier is not found (that is, if no ROM with this identifier is installed on an adapter), the next attempt to find this identifier is made at the machine address 0C8800H. This process continues with address increments of 2048 (800H) bytes, until the beginning of main processor board ROM BIOS.

If the identifier is found, a one-byte binary value representing the number of 512-byte blocks on the adapter is assumed at a positive offset of 2 to the above-mentioned 2048-byte boundary. Valid 8088 code is assumed at a positive offset of 3, and a far call is executed to that address. This code can provide, for example, additional initialization routines not provided by the main processor board ROM BIOS. The final instruction of the code on the adapter ROM is a far return. The search for further occurrences of the 0AA55H identifier then continues with the search address set to 2048 byte boundary following the last 512-byte block on the adapter most recently detected.

A fixed disk adapter, if present, provides just such an identifier at the machine address 0C8000H. If no flexible disk is installed in drive A, the fixed disk ROM code re-writes the interrupt vector for type 13H, so that it points to boot loading routines on the fixed disk adapter ROM. The most immediate effect of this is that the system boot subsequently executed by means of interrupt 19H reads the boot record from the fixed disk, instead of from the flexible disk.

Printers and Communications

INTRODUCTION

This Chapter describes the two standard types of interface provided by your NCR PERSONAL COMPUTER, namely RS-232-C and Centronics.

A number of printers, plotters and communications devices can be controlled by your computer by means of these interfaces. The RS-232-C serial input/output interface is especially versatile in communications applications. This Chapter provides information about the hardware integration of these interfaces as well as instructions for their low level software control. The inhouse data link control (DLC) interface is described in a separate Chapter.

The following cables relating to the RS-232-C serial and Centronics parallel interfaces are available:

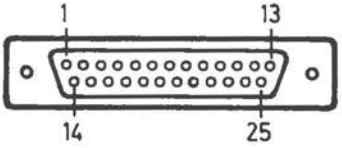
K120 Centronics
K121 RS-232-C (Printer)
K122 RS-232-C (Communications)

Your NCR PERSONAL COMPUTER can also drive a second RS-232-C interface installed as an option. An additional hardware interrupt type is made available by the BIOS in the system interrupt vector, so that two RS-232-C interrupt driven interfaces can be active simultaneously.

RS-232-C INPUT/OUTPUT

The electronics of the standard RS-232-C interface are included on the main processor board. The external connections are by means of the uppermost D-shaped connector at the back of the computer. The pin configuration of this connector is shown in Figure 4.1.

Pin	Signal	
1	Frame Ground	
2	Transmit Data (TxD)	
3	Receive Data (RxD)	
4	Request to Send (RTS)/	
5	Clear to Send (CTS)/	
6	Data Set Ready (DSR)/	
7	Signal Ground	
8	Carrier Detect (CD)/	
9	+XMIT CL RET	*
10	Not Used	
11	-XMIT CL RET	*
12 ..17	} Not } Used	
18	+RCV CL Data	*
19	Not Used	
20	Data Terminal Ready (DTR)/	
21	Not Used	
22	Ring Indicator/	
23	} Not	
24	} Used	
25	-RCV CL Data	*



* not provided with all main processor boards

Figure 4.1 RS-232-C Pin Configuration

The RS-232-C interface is programmable by means of an 8250 asynchronous receiver/transmitter. This integrated circuit includes a programmable baud rate generator and interrupt handling logic. The logic structure and pin assignments of this integrated circuit are illustrated in Figure 4.2.

The control, address, and data signals present at the 8250 are set out in Figure 4.3.

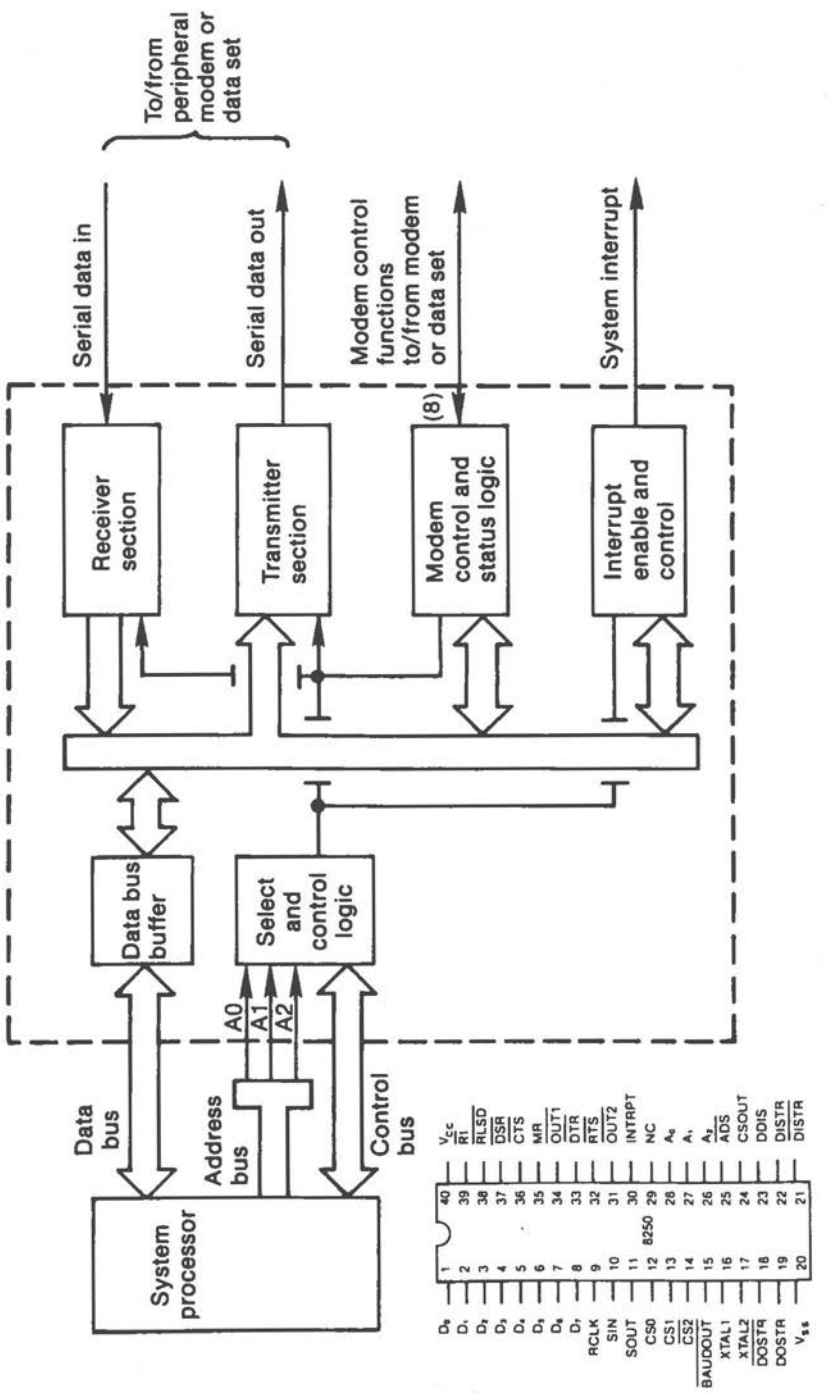


Figure 4.2 8250 Serial Receiver/Transmitter

Pin	(I/O)	Signal
1-8	D0-D7	The system data bus, used for transferring control information and data in parallel form prior to serialization for transmission or after received data has been de-serialized
9	RCLK (I)	16X baud rate clock input for the receiver
10	SIN (I)	Receives serial data from RxD input line
11	SOUT (O)	Transmits serialized data via the TxD output line. A Master reset sets SOUT to logic 1 marking condition
12	CS0 (I)	All 3 Chip Select lines must be active in
13	CS1 (I)	order to select the 8250
14	CS2/ (I)	
15	BAUDOUT/ (O)	16X clock signal for the transmitter. The clock rate = oscillator frequency divided by contents of divisor latches
16	XTAL1	Connect signal clock to the baud rate divider
17	XTAL2	circuit on the 8250
18	DOSTR/ (I)	Either one of these Data Strobe lines is used
19	DOSTR (I)	to latch data from the CPU into the selected 8250 register
20	Vss	System signal ground
21	DISTR/ (I)	Either one of these Data Strobe lines serves
22	DISTR (I)	to latch an 8250 register for reading by the CPU

Figure 4.3 8250 Signals (1 of 3)

Pin		[I/O]	Signal
23	DDIS	[O]	Driver Disable. Output goes low whenever data is being read from the 8250. Can be used to reverse data direction of external receiver
24	CSOUT	[O]	Confirms chip selection by means of a high output. A pre-requisite for data trasfer.
25	ADS/	[I]	Address Strobe providing latching for 8250 internal registers. An active ADS/ signal is required only if the 3 address lines selecting the register do not maintain stable signals during throughout an 8250 operation.
26	A2		System address bus lines selecting an 8250 internal register
27	A1		
28	A0		
29			No connection
30	INTRPT	[O]	Output goes high whenever an enabled interrupt is imminent
31	OUT2/	[O]	User-designated output programmable by means of bit 3 of the Modem Control Register: bit set makes signal active
32	RTS/	[O]	Signal active informs the external device that the 8250 is ready to transmit data
33	DTR/	[O]	Signal active informs the external device that the 8250 is ready to communicate
34	OUT1/	[O]	User-designated output programmable by means of bit 2 of the Modem Control Register: bit set makes signal active

Figure 4.3 8250 Signals [2 of 3]

Pin		(I/O)	Signal
35	MR	(I)	Signal active resets the 8250 internal registers
36	CTS/	(I)	Input signal to the 8250 from the external device, indicating that the latter is ready to transmit
37	DSR/	(I)	Input signal from the external device indicating its ready status
38	RSLD/	(I)	Received Line Signal Detect or Carrier Detect. Input signal from the external device confirming that adequate signal conditions are present
39	RI/	(I)	Input signal confirming that a ringing signal is being received by the external device
40	Vcc		+5 Vdc supply

Figure 4.3 8250 Signals (3 of 3)

PROGRAMMING THE SERIAL RECEIVER/TRANSMITTER

The 8250 converts parallel data to serial on the transmit side and serial to parallel on the receive side. The serial interface driven by the 8250 included on the system main processor board is often given the name COM1. The port addresses dedicated to this interface and the 8250 8-bit internal registers they access are:

3F8H Receiver Buffer/Transmit Holding Register. This port is also used to access the lower byte of the baud rate Divisor Latch (switched by means of the Line Control Register).

3F9H Interrupt Enable.
Also upper byte of baud rate Divisor Latch (switched by means of Line Control Register).

3FAH Interrupt Identification (read only).

3FBH Line Control.

3FCH Modem Control.

3FDH Line Status.

3FEH Modem Status.

3FFH Not used.

The serial I/O unit commonly given the title COM2 occupies corresponding port addresses in the range 2F8H to 2FFH. Concurrent use of COM1 and COM2 is possible after installation of the optional Second RS-232-C Interface (3299-K301), which must then be strapped to COM2. A separate hardware interrupt is dedicated to the second serial interface (see Chapter 3).

Figure 4.4 shows the structure of the registers which control the operation of the 8250. The Receiver Buffer/Transmitter Holding Register is no more than an 8-bit data register accessed via the processor IN or OUT instruction, according to the direction of data flow. Bit 0 is always the first bit transmitted or received. The Divisor Latch sets the baud rate generator by means of a 16-bit binary value. The Divisor Latch is selected/de-selected by bit 7 in the Line Control Register.

Interrupt Enable Register (port 3F9H)

Bit: 7	6	5	4	3	2	1	0
0	0	0	0	Modem	Receive	Transmit	Data

Interrupt ID Register (port 3FAH)

Bit: 7	6	5	4	3	2	1	0
0	0	0	0	0	< Interr. ID >		Pending

Line Control Register (port 3FBH)

Bit: 7	6	5	4	3	2	1	0
Divisor Latch	Set Break	Stick Parity	Even Parity	Parity Enable	Stop Bits	< Char. Length>	

Figure 4.4 8250 Control/Status Registers (1 of 2)

Modem Control Register (port 3FCH)

Bit: 7	6	5	4	3	2	1	0
0	0	0	Loop	Out 2	Out 1	RTS	DTR

Line Status Register (port 3FDH)

Bit: 7	6	5	4	3	2	1	0
0	TSRE	THRE	Break Interr.	Framing	Parity	Overrun	Data Ready

Modem Status Register (port 3FEH)

Bit: 7	6	5	4	3	2	1	0
RLSD	RI	DSR	CTS	DRLSD	TERI	DDSR	DCTS

Figure 4.4 8250 Control/Status Registers (2 of 2)

Interrupt Enable Register

If all bits are zero, no interrupts are issued by the 8250. Setting individual bits 0-3 determines which events result in the issue of an interrupt. These bits may be set in any combination:

0: Interrupt when received data available.
1: Transmitter Holding Reg. empty.
2: error or Break Interrupt.
3: change of state on CTS, DSR, RI, or RLSD.

Interrupt ID Register
If an interrupt is waiting to be acknowledged, bit 0 is zero. The binary value represented by bits 1 and 2 then denotes the pending interrupt with the highest priority (see Figure 4.5).

Priority/ ID value	Int. Type	Int. Source	Acknowledge/reset:
Highest/3	Receiver	Overrun error or Parity error or Framing error or Break Interrupt	Read Line Status
Second /2	Data	Receiver data available	Read Receive Buffer Register
Third /1	Transmit	Transmit Holding Register empty	This register read/ next write to THR
Lowest /0	Modem	Clear to Send or Data Set Ready or Ring Indicator or Received Line Signal Detect	Read Modem Register

Figure 4.5 8250 Interrupt Priority

Line Control Register
This register can be read as well as written, so that there is no need for applications to store the current settings.

Bits 0 and 1 contain a binary value specifying length of the serial character: 0 = 5 bits, 1 = 6 bits, 2 = 7 bits, and 3 = 8 bits.

Bit 2 specifies the number of stop bits: 0 = 1 stop bit; 1 sets 1.5 stop bits (for 5-bit character), or 2 stop bits (for other character lengths).
Bit 3 set determines that parity checking takes place. The type of parity checking is controlled by bits 4 and 5: bit 4 set means that an even number of bits are transmitted or checked; if bit 5 is set, parity is transmitted and then detected by the receiver as logic 0 (if bit 4 = 1) or as logic 1 (if bit 4 = 0).
Bit 6 forces the serial output line to a low (Break) state, where it remains until this bit is reset. The Break condition is often used to alert the external device.
Bit 7 set selects the Divisor Latch at ports 3F8H and 3F9H, in order to program the baud rate generator. As long as this bit is zero, the Receive Buffer/Transmit Holding Register (3F8H) and Interrupt Enable Register (3F9H) are selected.

Modem Control Register
Bits 0-3 set force high the 8250 output lines DTR/, RTS/, OUT1/, and OUT2/, respectively.
Bit 4 set activates 8250 internal diagnostics (see later section).

Line Status Register
Bit 0 set indicates that a serial character from the receiver has been converted to parallel form and is waiting to be read from the Receiver Buffer Register (port 3F8H). This status bit is automatically reset when the data byte is read, or if that register is filled with zeros by means of an OUT instruction from the microprocessor.
Bit 1 set means that an overrun error occurred, that is, the Receiver Buffer Register was not read before a new character was introduced to it. Reading the Line Status Register resets this bit.

Bit 2 set indicates a parity error. This bit is reset by reading the Line Status Register.
Bit 3 set indicates a framing error, that is, the stop bit is not valid.

Bit 4, the Break Interrupt detector, is set if a break condition is recognized by the receiver. Such a condition exists when the serial input line is at zero beyond the length of one character consisting of Start + Data + Parity + Stop bits.
Bit 5 indicates when the Transmit Holding Register can accept a new character for serialization and transmission (bit set), as the most recent character has been forwarded to the 8250's Transmit Shift Register. This bit is reset when this register is next written to.
Bit 6 reflects the status of the Transmit Shift Register: this bit set indicates that the register is idle.

Modem Status Register
Whenever bit 0, 1, 2, or 3 is set, a Modem Status interrupt condition is asserted (see Figure 4.5).

Bits 0 set indicates that the CTS/ line has changed state since the Modem Status Register was last read.
Bit 1 applies analogously to the DSR/ line.
Bit 2 is set when the trailing edge of the Ring Indicator is detected.
Bit 3 set indicates that the RLSD/ input has changed state.
Bits 4-7 represent the complements of the CTS/, DSR/, RI/, and RLSD/ inputs, respectively, provided that bit 4 in the Modem Control Register is zero. If this bit is set, the four Modem Status Register bits represent RTS/, DTR/, OUT1/ and OUT2/.

INTERNAL DIAGNOSTICS
The 8250 includes its own loopback diagnostic feature, activated by writing the Modem Control Register with bit 4 set.

Data Integrity

The serial transmitter output (SOUT)is set to a state of marking (logic 1) and the serial receiver line (SIN) is disregarded. The output from the Transmit Shift Register is looped back into the Receiver Shift Register. The Modem control inputs CTS/, DSR/, RLSD/, and RI/ are disregarded. Instead, these signals are supplied internally by the Modem control outputs DTR/, RTS/, OUT1/, and OUT2/. This configuration allows the 8250 to verify its internal transmit and receive data paths.

The Interrupt System

In the diagnostic mode, receiver and transmitter interrupts are fully operational. The Modem Control interrupts are also operational, but the interrupts are now derived from the four least significant bits of the Modem Status Register, not from the Modem Control input lines. The interrupts can still be enabled and disabled by means of the Interrupt Enable Register.

A test of the interrupt system requires writing the lower 6 bits of the Line Status Register and the lower 4 bits of the Modem Status Register: a bit set results in the assertion of the corresponding interrupt.

To conclude diagnostics, restore the 8250 registers where appropriate and write the Modem Control Register with bit 4 set.

RECEIVER/TRANSMITTER TIMING

Figures 4.6 - 4.11 illustrate 8250 signal timing.

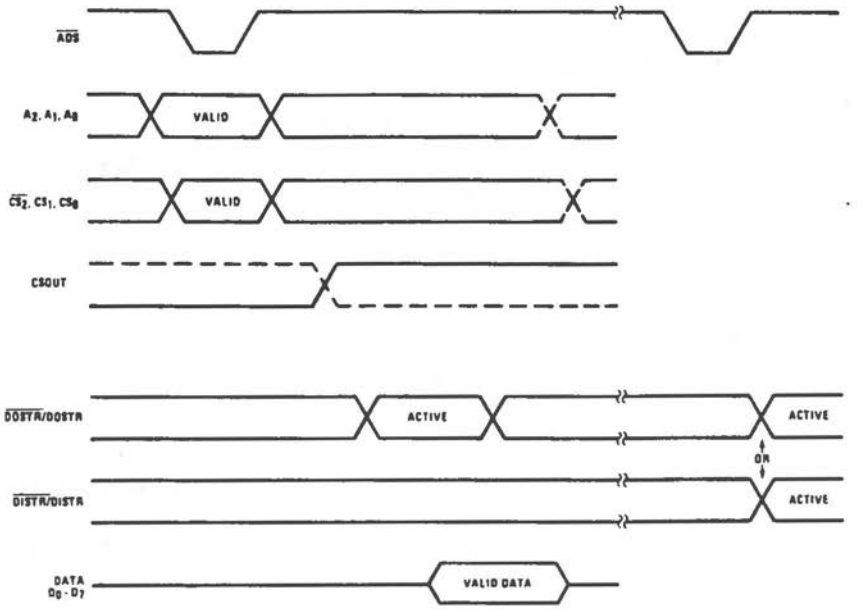


Figure 4.6 Read Cycle

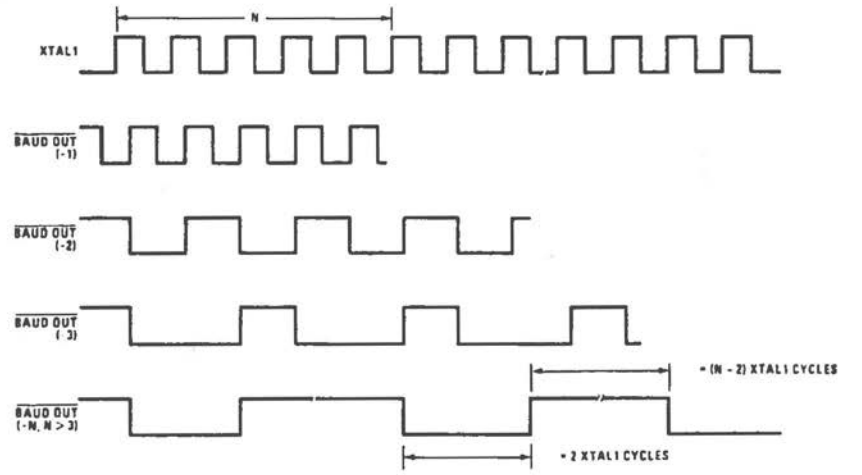


Figure 4.7 BAUDOUT/ Timing

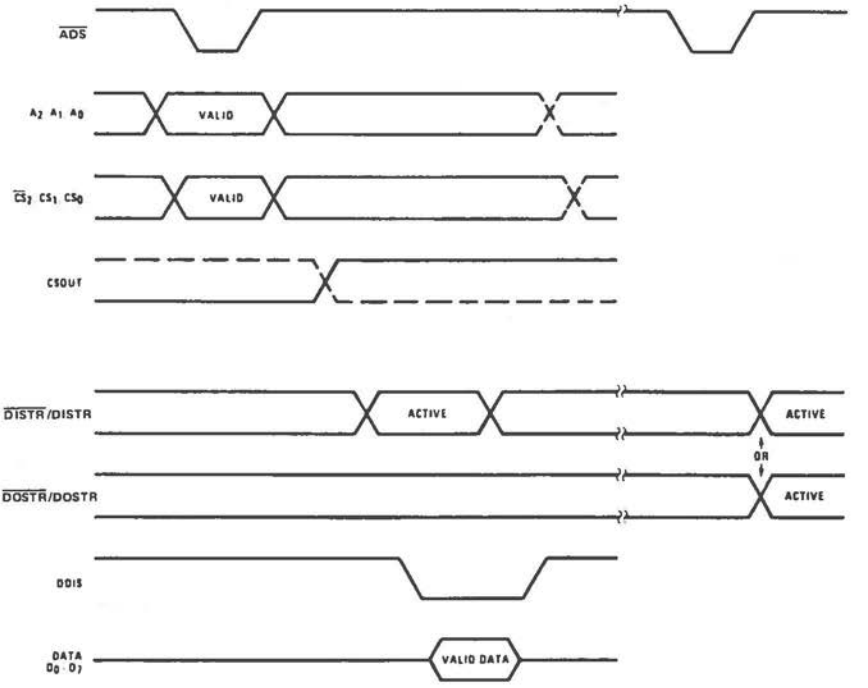


Figure 4.8 Write cycle

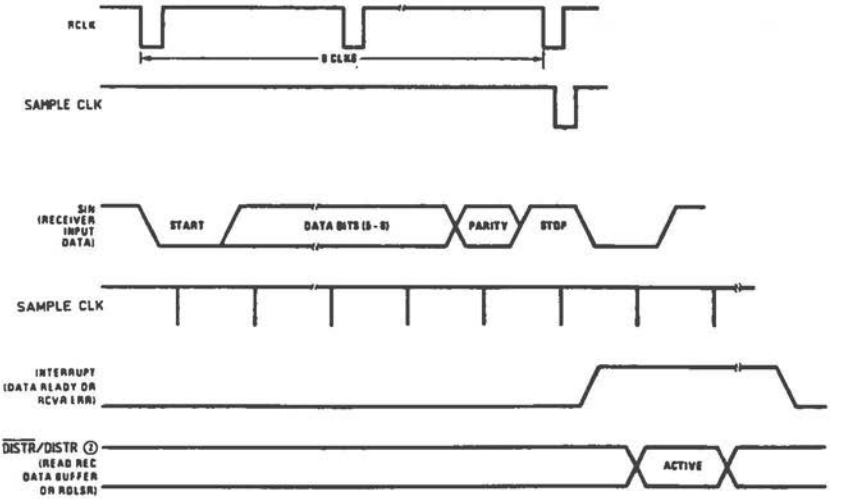


Figure 4.9 Receiver Timing

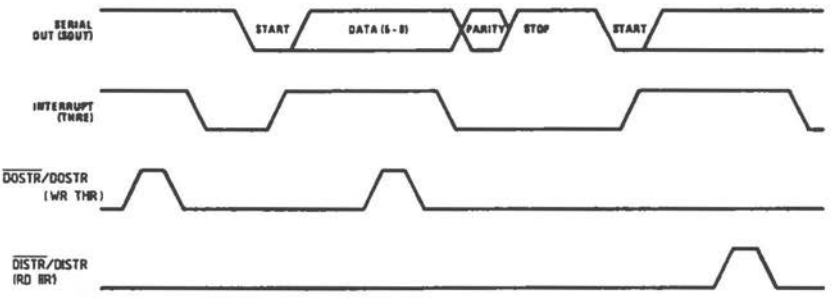


Figure 4.10 Transmitter Timing

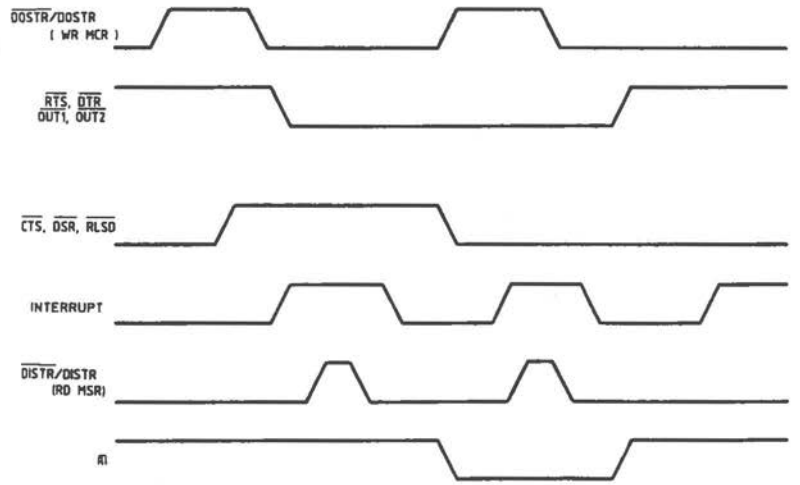


Figure 4.11 Modem Control Timing

BAUD RATE SELECTION

The clock input governing the 8250 internal baud rate generator in your NCR PERSONAL COMPUTER is set at 1.8432 MHz. Accordingly the 16-bit Divisor Latch values for the standard baud rates are as set out in Figure 4.12. Where the integer nature of 16-bit binary counting results in a negligible deviation from the nominal baud rate, this is indicated in the rightmost column.

Nominal baud rate	Divisor Latch (hexadecimal)	Actual deviation (%)
50	900	
75	600	0.026
110	417	
134.5	359	0.058
150	300	
300	180	
600	C0	
1200	60	
1800	40	
2000	3A	0.69
2400	30	
3600	20	
4800	18	
7200	10	
9600	C	
19200	6	

Figure 4.12 Baud Rate Divisor Latch

PROGRAMMING HINTS

This section presents a starting point for the creation of your own serial input/output protocols.

Constants used in these routines:

```
COM         EQU  3F8H          ;otherwise 2F8H.
;
                               ;Line Control Register -
PARITY      EQU  18H           ;Example: even parity enabled.
DATBIT      EQU  2             ;         7 data bits.
STOPBIT     EQU  0             ;         1 stop bit.
BAUDDIV     EQU  80H           ;         select Divisor Latch.
;
BAUD        EQU  30H           ;         baud rate 2400.
;
TIMEOUT     EQU  0FFFFH        ;Timeout for external device
                               ;in terms of 8088 CX LOOPs taken.
;
```

Interface Availability

This examination of your system configuration checks for the availability of a serial I/O interface as COM1 (base port 3F8H) or COM2 (base port 2F8H), according to the base port represented by COM:

```
            MOV  DX,COM+2      ;Refer to Interrupt Identification
                               ;Register.
            IN   AL,DX
            TEST AL,00000111B  ;Will not yield zero if Modem
                               ;signals not present.
;           JNZ  NOCOM         ;Jump if COM not available.
```

Initialization

The following routines initialize the 8250 for the transmission and receiving of serialized data.
Set the Line Control Register to the required parity, data, and stop bits. At the same time, set the Divisor Latch bit (7), so that the baud rate generator can be subsequently set:

```
XOR   AL,AL
OR    AL,PARITY+DATBIT+STOPBIT
OR    AL,BAUDDIV
MOV   DX,COM+3      ;Refer to Line Control Register.
OUT   DX,AL
```

Next write the desired baud rate to the 8250's 16-bit Divisor Latch:

```
MOV   AX,BAUD
SUB   DX,3          ;Refer to serial I/O base port.
OUT   DX,AL         ;Divisor Latch low byte.
MOV   AL,AH
INC   DX
OUT   DX,AL         ;Divisor Latch high byte.
```

Now retrieve the Line Control Register and re-write it, this time with the Divisor Latch bit zero, so that the port COM can transmit/receive data, and COM+1 addresses the Interrupt Enable Register:

```
MOV   DX,COM+3
IN    AL,DX         ;Read Line Control Register.
XOR   AL,BAUDDIV    ;Zero bit 7.
OUT   DX,AL         ;Re-write register.
```

Finally clear the Interrupt Enable Register so that interrupt conditions do not result in an interrupt actually being issued. It is then the concern of the transmitting/receiving routine to select types of event which are to result in an interrupt:

```
MOV   DX,COM+1
XOR   AL,AL
OUT   DX,AL         ;Write Interrupt Enable Register
                    ;with all bits zero.
```

Transmitting Data

The following routines assume a character ready for transmission in AH.
First assert the DTR/ and RTS/ signals by means of the Modem Control Register:

```
        MOV   DX,COM+4     ;Refer to Modem Control Register.
        MOV   AL,00000011B ;DTR and RTS bits.
        OUT   DX,AL
```

Now poll the DSR/ and CTS/ until both are asserted by the external device, or until timeout, whichever is the sooner:

```
        MOV   CX,TIMEOUT
        MOV   DX,COM+6     ;Point to Modem Status Register.
TIME1:  IN    AL,DX
        AND   AL,30H       ;Blank out all but bits 4 (CTS)
                           ;and 5 (DSR).
        CMP   AL,30H
        JZ    MODRDY       ;Jump if device ready.
        LOOP  TIME1        ;Otherwise check again, unless
                           ;count exhausted.
        JMP   MODXRDY      ;Jump if device timeout.
```

Assuming that the device was not disqualified by timeout, read the Line Status Register, looking for the Transmit Holding Register to become empty (bit 5). This condition is fulfilled if no character has been transmitted so far, or if the previous character has been passed to the Transmit Shift Register. Again a timeout condition can be applied:

```
        MOV   CX,TIMEOUT
        MOV   DX,COM+5      ;Refer to the Line Status Register.
TIME2:  IN    AL,DX
        AND   AL,20H        ;8250 can accept CPU char., as soon
                            ;as THRE (bit 5) set.
        JNZ   SEND          ;Jump if THRE empty.
        LOOP  TIME2         ;Otherwise check again, unless
                            ;count exhausted.
        JMP   MODXRDY       ;Jump if device timeout.
```

Assuming no timeout condition arose, a character can now be transmitted. The Transmit Hold Register is already known to be empty. No further status checks are required:

```
SEND:   MOV   DX,COM        ;Refer to Transmit Holding Register/
                            ;Receive Buffer.
        MOV   AL,AH         ;(Character assumed in AH.)
        OUT   DX,AL
```

Finally do not forget to deal with the possible timeout conditions (MODXRDY). This can take the form of retrying or terminating the operation.

Receiving Data

First activate the DTR/ output, so that the external device knows that it can start transmitting. This is achieved by setting bit 0 of the Modem Control Register:

```
        MOV   DX,COM+4      ;Refer to Modem Control Register.
        MOV   AL,1          ;Set bit 0.
        OUT   DX,AL
```

Now check that the device is ready by polling the DSR bit (5) in the Modem Status Register until it is set, or until device timeout:

```
          MOV   CX,TIMEOUT
          MOV   DX,COM+6     ;Refer to Modem Status Register.
TIME3:    IN    AL,DX
          AND   AL,20H       ;Check bit 5.
          JNZ   DATCHK       ;Jump if DSR active.
          LOOP  TIME3        ;Loop if DSR not active, unless
                             ;count exhausted.
          JMP   MODXRDY
```

Even when the device announces its readiness, it is still necessary to wait until a serial input character has been completely bit-shifted into the Receiver Buffer. Again a timeout condition or, alternatively, some kind of user (keyboard) break check can be applied:

```
DATCHK:   MOV   CX,TIMEOUT
          MOV   DX,COM+5     ;Data Ready bit is
                             ;in Line Status register.
TIME4:    IN    AL,DX
          TEST  AL,1         ;Bit 1 is Data Ready bit.
                             ;Do not AND AL, so that possible
                             ;error status return is retained.
          JNZ   RECEIVED     ;Jump if data ready.
          LOOP  TIME4        ;Otherwise, check bit again, unless
                             ;count exhausted.
          JMP   MODXRDY
```

Before actually reading the character from the Receiver Buffer, it is advisable to check that the character received was valid, that is, that there was no overrun, parity, or framing error, and that a break condition was not present on the serial input line:

```
RECEIVED:  AND   AL,1EH      ;Blank out all but overrun [1],
                             ;parity [2], framing [3] error
                             ;and break [4] bits.
           JNZ   ERROR       ;Jump if any error or break.
```

The valid character can now be read from the Receiver Buffer. (Do not forget to provide suitable handling routines for MODXRDY and ERROR.)

```
           MOV   DX,COM      ;Refer to Transmit Holding Register/
                             ;Receiver Buffer.
           IN    AL,DX       ;Valid character in AL.
```

CENTRONICS INTERFACE

The Centronics interface included on the main processor board provides a parallel connection to an external input/output device. The interface provides 8 TTL buffered data lines, 4 TTL buffered control lines, and 5 lines for status reading. Three basic operations can be performed:

* Write data to device.
* Read data from device.
* Read status of device.

Figure 4.13 shows the logical significance of the interface signals. Physical pin connections of the D-connector, situated second from top at the rear of the computer, are set out in Figure 4.14. The pins are capable of sourcing 2.6 mA and sinking 24 mA.

Control data comprises the following lines:

STROBE/ The pulse of at least 0.5 microseconds used to clock data read. Data is read when this signal is active (low).

AUTO/ This signal is commonly recognized by printers as an instruction to perform an automatic line feed when end of line is reached.

INIT/ This signal is commonly recognized by printers as a reset instruction, usually entailing erasure of remaining printer buffer contents.

SELECT INPUT/ Signal to select the external device (printer).

Status information comprises the following lines:

ERROR/ Indicates device (e.g. printer) error, for example, at paper-end, off-line, safety switch open, failure status.

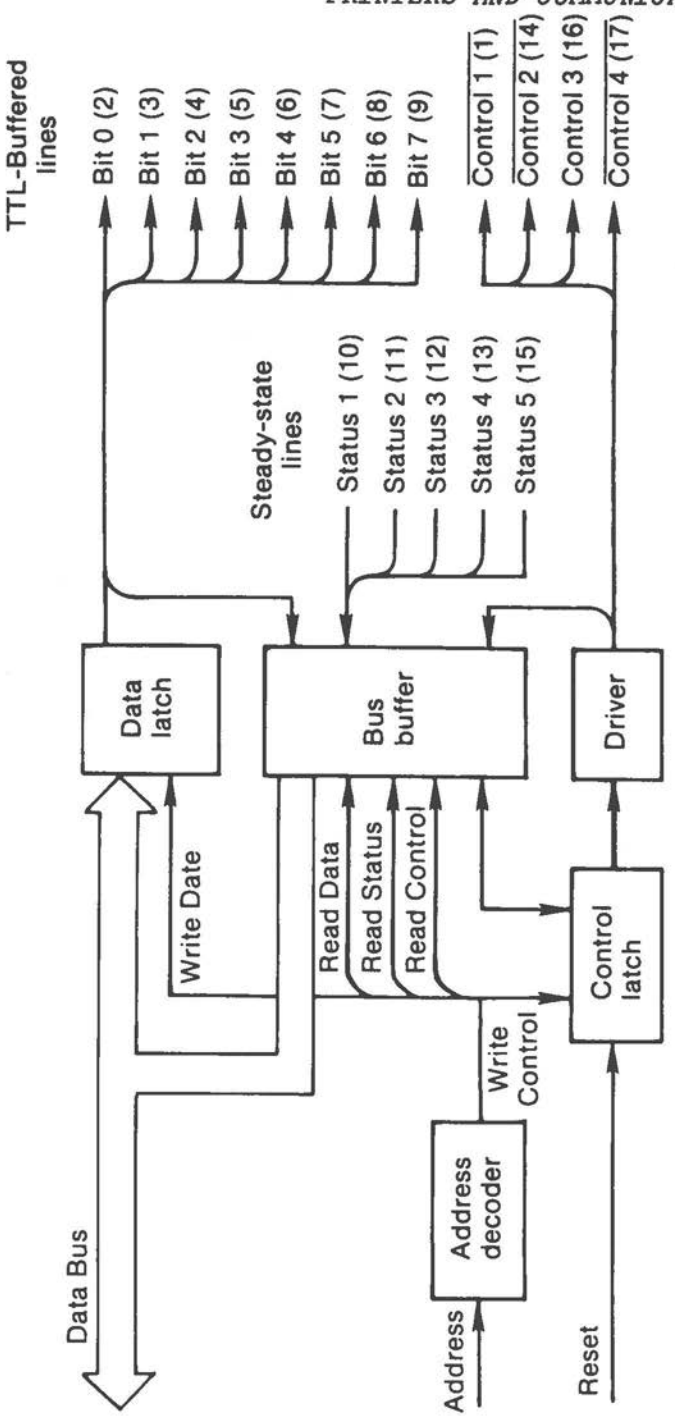


Figure 4.13 Centronics Interface Signals

Pin	Signal
1	STROBE/
2-9	Data lines 0-7
10	ACKNOWLEDGE/
11	BUSY
12	PE
13	SELECT
14	AUTO/
15	ERROR/
16	INIT/
17	SELECT IN/
18-25	Ground

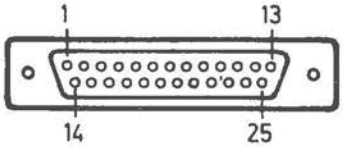


Figure 4.14 Centronics D-Connector

SELECT Confirms that the device is selected.

PE This signal is used by printers to denote an out-of-paper condition.

BUSY When this signal is active, the device cannot receive data. With printers, this signal is active under the following circumstances:

during data entry (depending on printer)
during printer operation
in off-line status
upon printer error status
during line feed

ACKNOWLEDGE/
Indicates that data has been received by the printer. New data cannot be received before this signal is reset.

SOFTWARE CONTROL

Data transfer between the microprocessor and the interface is via the system data bus. To write data to the external device the data is transmitted via port 378H. The interface captures the data from the bus and presents it on the data lines at the external connector. Appropriate control information must then be issued via port 37AH.

To read data from the Centronics interface, it is first necessary to transmit the appropriate control information via port 37AH. Data can then be read via port 378H.

Status reading is via port 379H.

Status and Control Registers

The status byte is made up as shown in Figure 4.15, the control byte as shown in Figure 4.16.

Device Status (IN 379H)							
Bit:							
7	6	5	4	3	2	1	0
BUSY	ACKNOW-LEDGE	PE	SELECT	ERROR	<—— not used ——>		

Figure 4.15 Centronics Status Register

Interface Control (OUT 37AH)							
Bit:							
7	6	5	4	3	2	1	0
0	0	0	Enable Int.	SELECT INPUT	INIT	AUTO	STROBE

Figure 4.16 Centronics Control Register

Centronics Printer Control

The short routines in this section illustrate fundamental programming steps involved in the initialization of the Centronics interface for parallel printers, and how to output data.

Initialization of the Centronics printer interface is essentially as follows:

```
        MOV   DX,37AH       ;Control port.
        MOV   AL,00001000B  ;Set bit to select external
                            ;device (SELECT INPUT).
        OUT   DX,AL
        MOV   CX,2000
WAIT1:  LOOP  WAIT1         ;Allow device time to select.
        MOV   AL,00001100B  ;Reset printer (INIT).
                            ;Here, auto line feed is
                            ;disabled.
        OUT   DX,AL
```

The following routine prints one character assumed to be available in AL. Supply a TIMEOUT value suitable for your printer and a routine (PRTXRDY) for the eventuality that a timeout situation arises.

```
        MOV   DX,378H       ;Data output port.
        OUT   DX,AL         ;Place character in output buffer.
        INC   DX            ;Refer to status port.
        MOV   CX,TIMEOUT
WAIT2:  IN    AL,DX         ;Read status.
        TEST  AL,10000000B  ;Test BUSY bit.
        JNZ   STROBE        ;Jump if not busy.
        LOOP  WAIT2         ;Otherwise loop ...
        JMP   PRTXRDY       ;... until count exhausted.
STROBE: INC   DX            ;Refer to control port.
        MOV   AL,00001101B  ;Strobe high.
        OUT   DX,AL
        DEC   AL            ;Thus resetting Strobe bit (0).
        OUT   DX,AL         ;Strobe low.
```

Status reading beyond that of the Busy bit is often required in order to establish the cause of a breakdown of printer activity. The following routine reads printer status and masks out insignificant bits:

```
MOV   DX,379H      ;refer to status port.
IN    AL,DX
AND   AL,11111000B ;Ensure unused bits 0,1,2 are zero.
```

Printer Status Analysis

This section consists of a short program which enables you to inspect the 5 interface status lines under test print conditions.

The printer continuously prints the characters in the code range 20H-7FH. Whenever one or more of the bits in the interface status register changes state, the new status is displayed on the screen. This happens, for example, at printer de-selection, end of paper, power on/off.

The timeout examination described in the previous section is also applied. The BUSY line status is constantly changing because the microprocessor is trying to output data faster than the printer can read it. The greater the TIMEOUT value specified, the less often a timeout condition is recognized. Each time a timeout condition is recognized in the character output routine PRNTOUT, where on the previous pass no such condition occurred, the information "Device Timeout" appears on the screen. Subsequent consecutive occurences of the timeout condition are each notified by means an asterisk. You can vary TIMEOUT so that this condition is recognized after every character or after every line feed. Observing this value can be useful when creating background printing software, in order to gain maximum printer efficiency while avoiding over-zealous printer status polling on the part of the microprocessor. The optimum value varies, of course, from one printer to another.

Recognition of timeout is here dependent on the status of the BUSY line. This means that timeout continues to be notified to the screen while the printer is de-selected or during paper out, but not while power is off. To include power off time in the timeout status check, you could include the SELECT bit in this check.

You will notice that a single operational change of state can produce more that one status change. This is due to transitional signal conditions.

```
;
PDAT        EQU   378H          ;Parallel I/O data    port.
PSTAT       EQU   379H          ;                status      .
PCONT       EQU   37AH          ;                control     .
;
FLAGSEL     EQU   11111000B     ;Mask out unused bits in status.
TIMERR      EQU   0FFH          ;Flags timeout error.
;
CR          EQU   0DH
LF          EQU   0AH
TIMEOUT     EQU   0FFFFH        ;Timeout while print character is
                                ;waiting in buffer (example value).
;
;
            CSEG  SEGMENT
            ASSUME CS:CSEG,DS:CSEG,ES:CSEG
;
            ORG   100H
            PUSH  CS
            POP   DS
;
                                ;<-- START.
            CALL  INITIF
            MOV   ERRFLAG,0FFH
;
;
```

```
PRTLOOP:  MOV   AH,20H       ;Start printer iteration with space.
NEXTCH:   MOV   AL,AH
          CALL  PRNTOUT      ;Attempt to print a character.
          CMP   AL,TIMERR
          JNZ   NEXTCH3      ;Jump if no device timeout.
          MOV   DX,OFFSET TIMEMES2   ;Assume timeout also on
                             ;previous pass through PRNTOUT
          CMP   TIMEFLAG,0
          JNZ   NEXTCH2      ;Jump if timeout also at previous
                             ;pass through PRNTOUT.
          MOV   TIMEFLAG,0FFH
          MOV   DX,OFFSET TIMEMES1
NEXTCH2:  CALL  SCRNMES      ;Notify timeout on screen.
NEXTCH3:  CALL  STATCHK      ;Full status analysis.
          INC   AH
          CMP   AH,80H
          JNZ   NEXTCH       ;Take next char. in ASCII sequence.
          JMP   PRTLOOP      ;Repeat printer iteration
                             ;indefinitely.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
STATCHK:                     ;Analyse status.
          CALL  RDSTAT
          CMP   AL,ERRFLAG
          JZ    STATCHK1     ;Return if status unchanged since
                             ;last poll.
          MOV   ERRFLAG,AL   ;Save new, changed error status.
                             ;Now build up status display.
          MOV   BX,OFFSET STATLN2 - 1
          MOV   CX,5         ;No. of status bits.
DISPLBIT: ADD   BX,6         ;Status print line tabulator.
          MOV   BYTE PTR [BX],'0'    ;Assume zero bit.
          SHL   AL,1
          JNC   NEXTBIT      ;Jump if bit is zero.
          MOV   BYTE PTR [BX],'1'
NEXTBIT:  LOOP  DISPLBIT
          MOV   DX,OFFSET STATLN1
          CALL  SCRNMES      ;Print status information.
STATCHK1: RET
```

```
;
;;;;;;;;;;;;;;;;
;
INITIF:                     ;Initialize interface
                            ;[see this Chapter].
          MOV   DX,PCONT
          MOV   AL,08H      ;SELECT INPUT
          OUT   DX,AL
          MOV   CX,2000
WAIT1:    LOOP  WAIT1
          MOV   AL,0EH      ;INIT, AUTO.
          OUT   DX,AL
          RET
;
;;;;;;;;;;;;;;;;
;
PRNTOUT:                    ;Accept character in AL
                            ;for printing [see this Chapter].
          MOV   DX,PDAT
          OUT   DX,AL       ;Character in output buffer.
          INC   DX          ;Point to status port.
          MOV   CX,TIMEOUT
WAIT2:    IN    AL,DX       ;Read status.
          TEST  AL,10000000B ;Examine BUSY.
          JNZ   STROBE
          LOOP  WAIT2       ;Otherwise loop until timeout.
          MOV   AL,TIMERR
          JMP   PRINTED
STROBE:   MOV   TIMEFLAG,0  ;Flag no [end of] timeout.
          INC   DX          ;Refer to control port.
          MOV   AL,0FH      ;Strobe data.
          OUT   DX,AL
          DEC   AL
          OUT   DX,AL
PRINTED:  RET               ;Character already sent to printer,
                            ;unless timeout.
;
;;;;;;;;;;;;;;;;
;
RDSTAT:                     ;Return printer status in AH.
          MOV   DX,PSTAT
          IN    AL,DX
          AND   AL,FLAGSEL  ;Zero insignificant bits.
          RET
```

```
;
;;;;;;;;;;;;;;;;
;
SCRNMES:                    ;Display error message on screen
                            ;using NCR-DOS fuction call 9,
                            ;display string up to $.
                            ;Entry: address of string in DX.
          PUSH  AX
          MOV   AH,9
          INT   21H
          POP   AX
          RET
;
; *** Status Message ***
;
ERRFLAG   DB  0             ;0FFH flags repetition of an error
                            ;on consecutive status polls.
TIMEFLAG  DB  0             ;0FFH flags repeated timeout.
TIMEMES1  DB  CR,LF,'Device Timeout ','$'
TIMEMES2  DB  '*','$'       ;Asterisk only for repeated timeout.
STATLN1   DB  CR,LF
          DB  '    7    6    5    4    3    2    1    0'
          DB  CR,LF
          DB  '  BUSY  ACK   PE  SELECT ERROR  -    -    -'
          DB  CR,LF
          DB  '-------------------------------------------------'
          DB  CR,LF
STATLN2   DB  '                                  ',CR,LF,'$'
;
          CSEG  ENDS
          END
```

Inhouse DLC Interface

The inhouse data link control (K201) is an interface enabling your NCR PERSONAL COMPUTER, as the primary station, to communicate over dedicated lines with up to 31 secondary stations, for example, cash terminals (NCR 2126) and dialog stations (NCR 7900).

The primary station continuously polls the secondary terminals, checking whether a data transfer is requested. The secondary terminal can then transmit the requested data, after which it returns to the receive mode. Transmission speed is 48000 bits per second.

The maximum advisable length of cable in the entire network is 3600 metres (12000 feet).

The interface can communicate data with the system microprocessor under DMA or under direct microprocessor I/O control.

INTERFACE CONNECTIONS

The interface uses only 4 wires, two for transmission (Ta, Tb), the other two for receiving (Ra, Rb). These wires are connected to a standard 9-pin D-connector as shown in Figure 5.1.

Wire	Color	Pin
Ta	white	1
Tb	green	2
Ra	yellow	3
Rb	brown	4

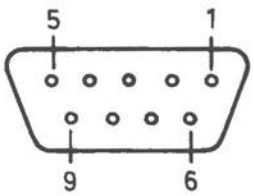


Figure 5.1 DLC Interface Connections

The interface hardware includes two jumpers which select the type of interrupt and the DMA channel used (Figure 5.2).

Jumper		Selection		
W1	W2	IRQ	Int. type	DMA channel
A – B	A – B	3	11 *	1 *
B – C	B – C	2	10	2

* Factory setting

Figure 5.2 DLC Interface Jumper Settings

DLC CONTROLLER

The DLC interface is controlled by an integrated circuit clocked by a 2.304 MHz oscillator. Figure 5.3 illustrates the integration of this controller in the NCR PERSONAL COMPUTER.

CONTROLLER PIN CONFIGURATION

Figure 5.4 illustrates the pin connections of the DLC integrated circuit. The significance of the individual connections is explained in this section.

CLOCK Input - Pin B20
System clock input frequency of 4.77 MHz.

Reset DRV Input - B2
Initialization (reset) signal synchronized to the falling edge of the clock.

Address lines Input - Pins A22 ... A31
The 10 least significant system address bus lines (A0 to A9). Pin A31 represents the least significant address line.

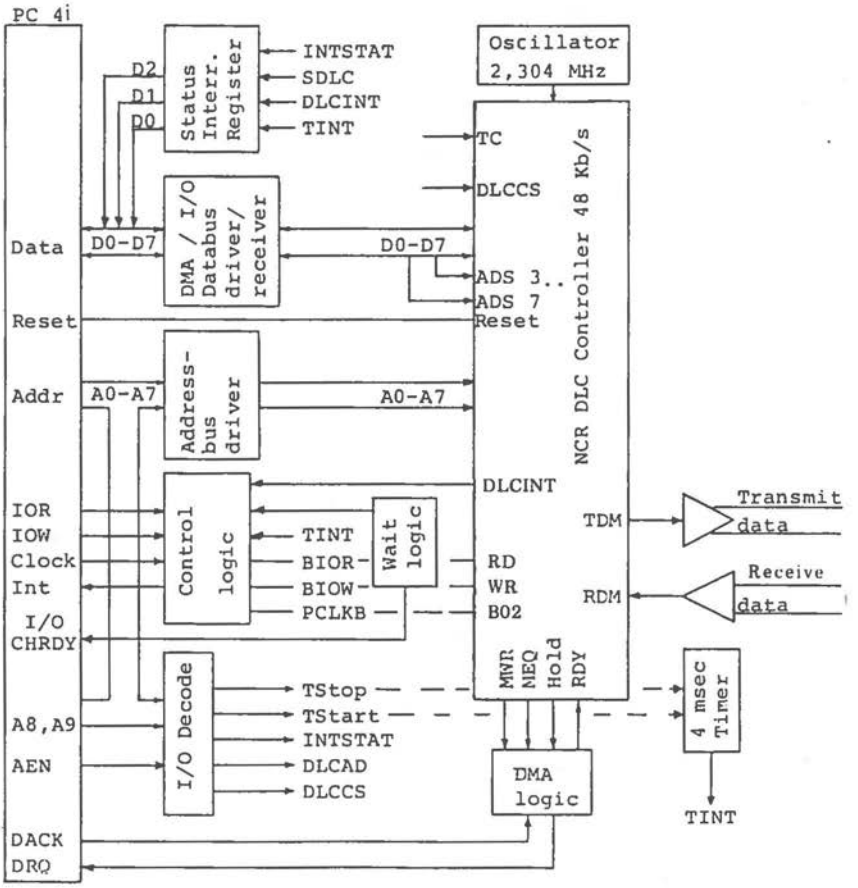


Figure 5.3 DLC Controller: Signal Flow

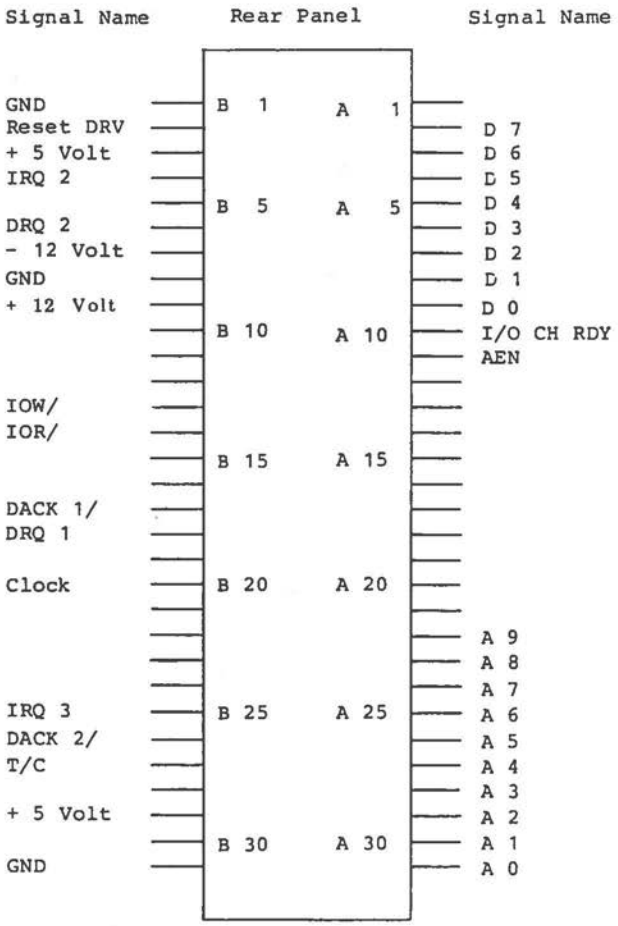


Figure 5.4 DLC Integrated Circuit

Data lines - Pins A2 ... A9
The system data bus (D0 to D7). The least significant data bit is represented at pin A9.

I/O CH RDY Output - Pin A10
I/O Channel Ready signal. This line is normally high (ready). It is pulled low by a memory or I/O device to lengthen I/O or memory cycles. Thus, slow devices can be attached to the interface with a minimum of difficulty. Any slow device should should drive this line low immediately upon detecting a valid address and a read/write command. This line is never held low for longer than 10 clock cycles.

IRQ2, IRQ3 Output - Pins B4, B25
These interrupt request lines are used to signal to the processor that the DLC controller requires attention. Jumpers are factory set to issue IRQ3 (Figure 5.2). The interrupt line is held high until it is acknowledged by the microprocessor.

IOR/ Input - Pin B14
I/O Read command. This signal instructs the DLC controller to place its data on the syatem data bus. It may be controlled by microprocessor or DMA.

IOW/ Input - Pin B13
I/O Write command. This signal instructs the DLC controller to read data from the system data bus. It may be controlled by microprocessor or DMA,

DRQ2, DRQ1 Output - Pins B6, B18
The DMA request lines. The selected DRQ line must remain high until acknowledged by the corresponding DACK/ line. The factory setting of the corresponding jumpers is that DRQ1 is selected (see Figure 5.2).

DACK1/, DACK2/ Input - Pins B17, B24
Acknowledge signal lines corresponding to the respective DMA request lines.

AEN Input - Pin A11
When this line is active, the microprocessor is degated so that a DMA transfer can take place.

T/C Input - Pin B27
Terminal Count. This line is pulsed active when the terminal count for the DMA channel has been attained.

The following voltages are present at the DLC controller:

+ 5 V 5%	- Pins B3, B29
+ 12 V 5%	- Pin B9
- 12 V 10%	- Pin B7
Ground	- Pins B1, B10, B31

SOFTWARE CONTROL
The interface address lines are decoded as follows:

IN 380H	Data Receive Hold Register
OUT 380H	Data Transmit Hold Register
IN 381H	Status Register
OUT 381H	Command Register
OUT 382H	Group Address Register
OUT 383H	Transmit DMA Address (low) Register
OUT 384H	DMA Counter Transmit Register
OUT 385H	Receive DMA Address (low) Register
OUT 386H	DMA Counter Receive Register
OUT 388H	Unique Address Register
IN 389H	Interrupt Status
OUT 38AH	Start Timer (4 ms)
OUT 38BH	Stop and Reset Timer

Command Register

This register controls the functions of the DLC integrated circuit. Each of the eight bits is dedicated to a different task. The structure of this Command Register byte is shown in Figure 5.5.

Bit:							
7	6	5	4	3	2	1	0
TXFR	TXFLG	RXA	RXB	IO/DMA	DIAG	SWRST	0

Figure 5.5 Command Register

TXFR

Transmit Frame. Setting this bit instructs the DLC controller to begin transmission of a frame. Subsequently data for transmission is expected from the system bus. In DMA mode the controller internally resets this bit when the transmission is complete. Under direct microprocessor control of data transfer this bit is reset when a Transmitter Not Serviced condition arises.

TXFLG

Setting this bit causes the controller to transmit contiguous flags. If TXFR is set simultaneously the transmitted frame is completed before the contiguous flags are transmitted.

RXA, RXB

The binary value contained in these two bits (RXA is MSB) determines which receive mode is active:

0 Receive mode not active
1 Accept all addresses (Primary)
2 Accept group, global, or unique address (Secondary)
3 Accept unique addresses only

These two bits are reset internally by the controller at the conclusion of frame reception. Global address is F9.
The transmit bits (TXFR, TXFLG) take priority over the receive bits. In the event of a conflict, all transmission is completed before receive mode is entered.

IO/DMA
As long as this bit is zero, data transfer between DLC controller and system (both transmit and receive) takes place under DMA control. Otherwise data is transferred under microprocessor I/O control.

DIAG
Setting this bit causes the integrated circuit to carry out internal diagnostics. In transmit diagnostics the output lines Ta and Tb are disabled. Transmitter output is then routed back into the Receive Hold Register. Receive diagnostics can also be performed.

SWRST
Setting this bit resets the DLC controller. This reset must be performed after every introduction of power to the integrated circuit.

Status Register
Figure 5.6 summarizes the bits of the Status Register.

Bit:							
7	6	5	4	3	2	1	0
EOF	RNS	TNS	FCS ERROR	RX ABORT	not used	DMATRM	SVCRQ

Figure 5.6 Status Register

EOF
End of Frame. This bit is set after a frame has been received whereupon the DLC controller generates an interrupt.

RNS
Receiver Not Serviced. This bit is received after a frame has been received if the Receive Hold Register was not read in time. This condition can occur in both DMA and processor I/O modes of operation.

TNS
Transmitter Not Serviced. The Transmit Hold Register was not written in the required time. This error causes the issue of an interrupt. The TNS condition can occur in both DMA and processor I/O modes of operation, but not under the following circumstances:

* The transmit command was aborted by controller command.
* The condition was recognized during the last byte transfer in a microprocessor I/O sequence.

FCS ERROR
Frame Check Sequence Error. After a complete frame has been received, the received frame's FCS bytes are compared with those generated by the DLC controller during data byte reception. Any discrepancy results in this status bit being set.

RX ABORT
If seven contiguous "ones" are detected on a received frame, this bit and the EOF bit are set and an interrupt is issued.

DMATRM
This bit is used for DMA transfers between DLC controller and the system. In the receive mode of operation DMATRM is set when the count written to the DMA Counter Receive Register is exhausted. In transmit mode it is set when the count in the DMA Counter Transmit Register is exhausted, or when a T/C (Terminal Count) signal is received from the DMA controller.

SVCRQ
Service Request. This bit is used for transfer under direct microprocessor control, that is, when the Transmit Hold Register is waiting for a byte of data (transmit mode) or when a byte of data is waiting to be read from the Receive Hold Register (receive mode). An interrupt is generated when this bit is set.

With the exception of SVCRQ reading the Status Register resets all its bits. SVCRQ is reset only after the data transfer via the bus has occurred.

Interrupt Status Register
Three bits of the Interrupt Status Register read via port 389H are significant:

Bit 0 A DLC controller interrupt sets this bit.
Bit 1 4 ms timer interrupt.
Bit 2 Reflects DRQ/IRQ jumper selection:
0 - DRQ2 and IRQ2
1 - DRQ1 and IRQ3

Unique Address Register
This register identifies the NCR PERSONAL COMPUTER in the DLC configuration. Normally, the computer will be functioning as the primary unit. The Unique Address Register is automatically set to 01 (primary) at DLC reset (which should always be performed after power introduction and before a new command), so that this register need not normally be written. If you have cause to use your NCR PERSONAL COMPUTER as a secondary device in a DLC configuration, you can write this register with a binary value as shown in Figure 5.7.

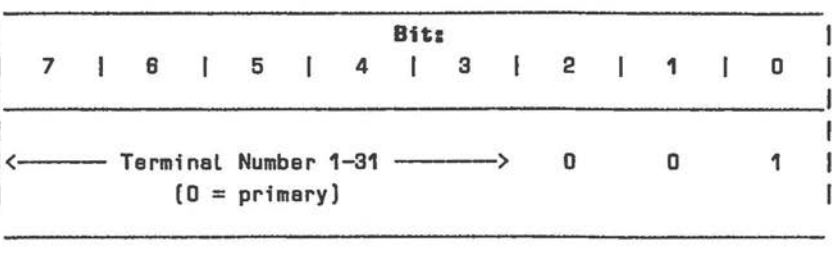


Figure 5.7 Unique Address Register

In addition the transmit and receive lines from the adapter exchange significance.

PROGRAMMING CONSIDERATIONS

With the aid of flow charts the remainder of this Chapter discusses some fundamental points to observe when designing software for the DLC interface.

Transmitting via CPU

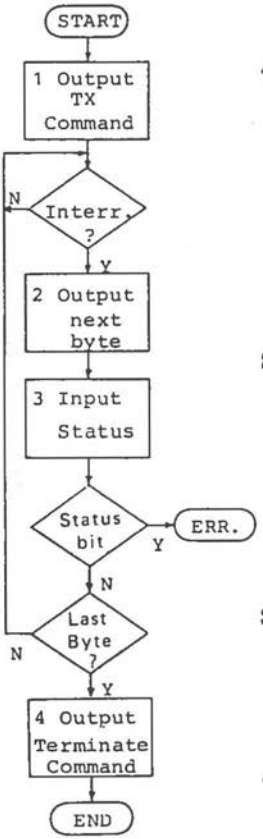


Figure 5.8 (1 of 2) CPU Transmit

1. The Command Register is written with bit 3 set, indicating that transfer will take place under direct CPU control. The values for bits 6 and 7 conform to the type of transmission required. When the Command Register has been written, the transmission sequence is started and an interrupt is generated.

2. Transmission circuits are enabled following servicing of the first interrupt. Whenever an interrupt occurs, it must be serviced (that is, data must be output) within six bit times, otherwise a Transmitter Not Serviced condition arises and the frame is terminated.

3. After the interrupt has been serviced, check the Status Register for possible errors. Normally the register should have all bits zero.

4. To terminate the frame in progress issue a transmission terminate command (Command Register bits: 0xxx1000), again within six bit times, otherwise a TNS error will

Figure 5.8 (1 of 2) CPU Transmit occur.

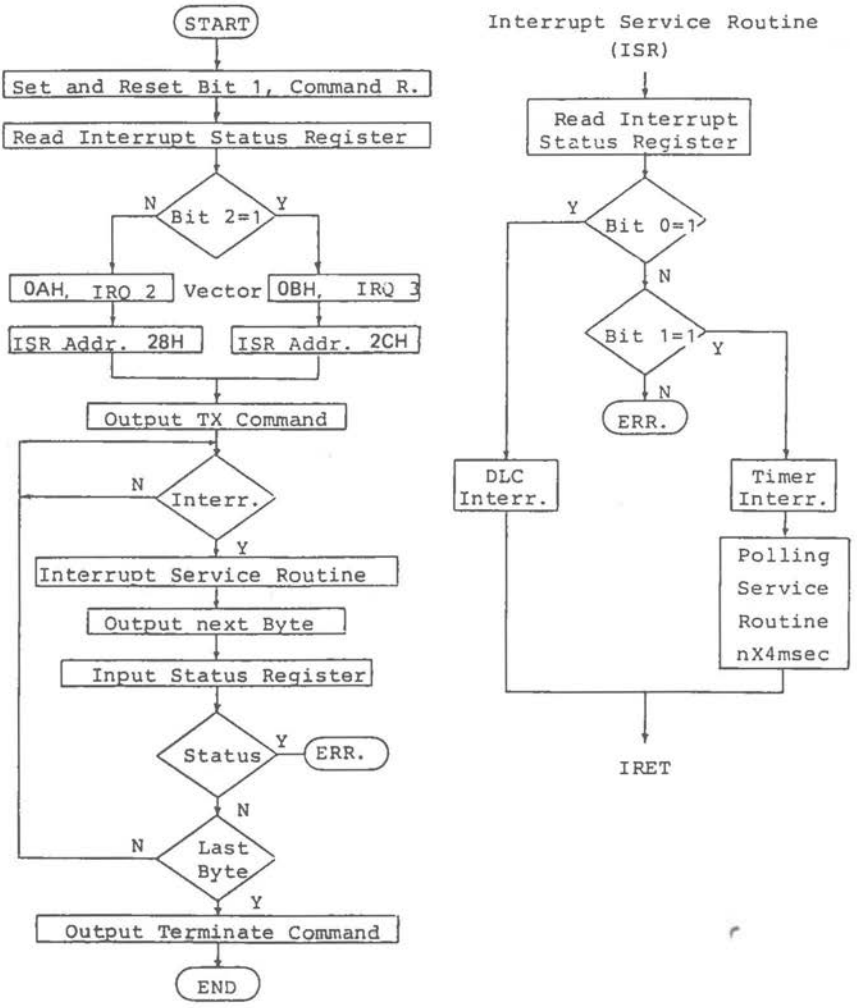


Figure 5.8 CPU Transmit (2 of 2)

Receiving via CPU

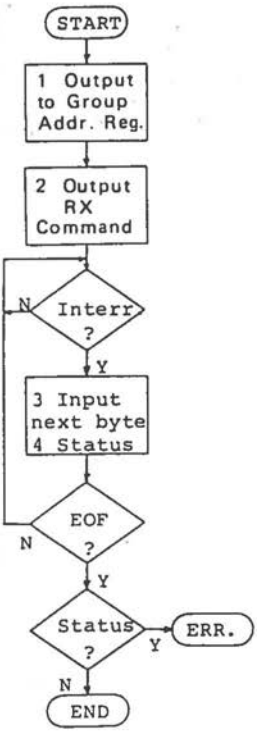


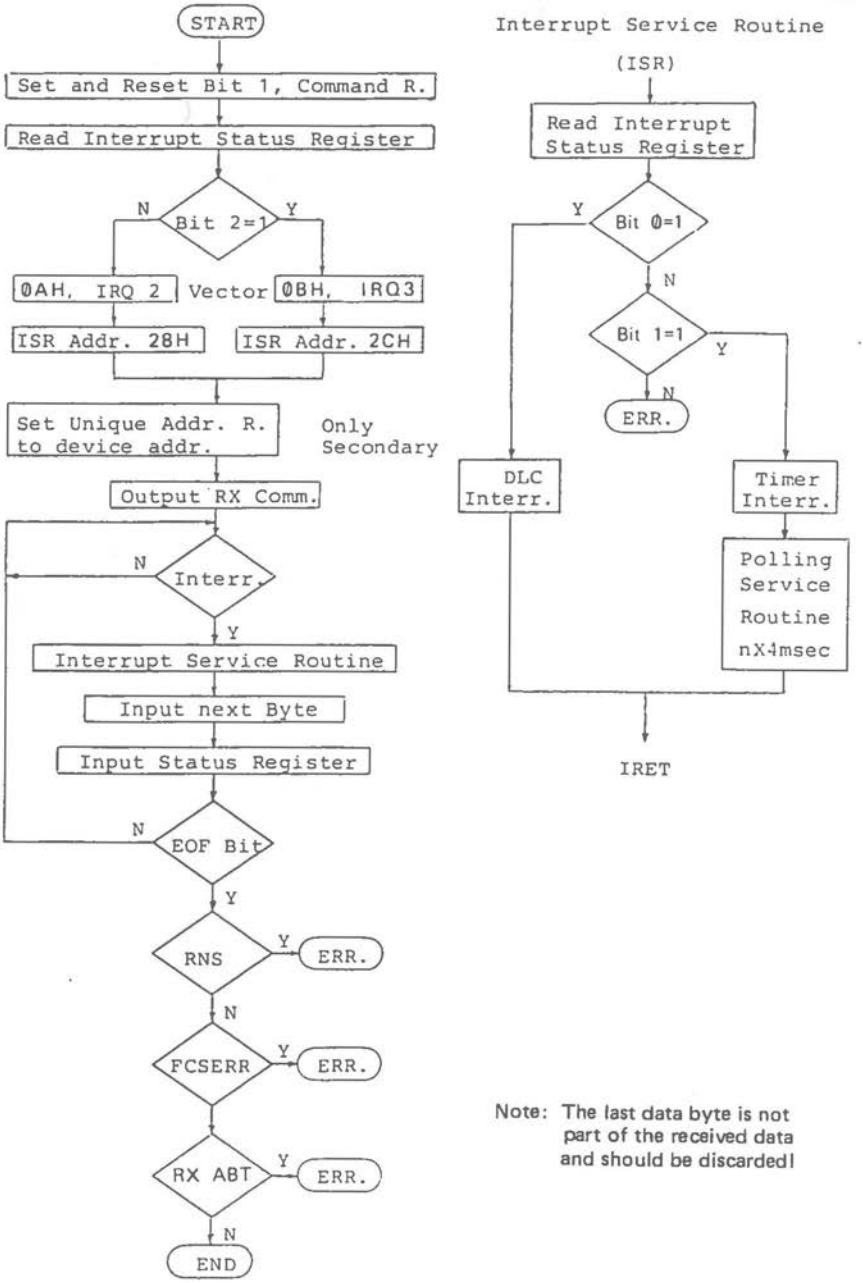
Figure 5.9 (1 of 2)
CPU Receive

1. Assuming that the DLC interface is the primary device, the Group Address Register need not be written. If a unique, group, or global command is to be issued (Command Register bits 4 and 5), this register must be written accordingly.

2. Issue a receive command detailed in bits 4 and 5 (RXA, RXB), with the IO/DMA bit set. A receive command may be given at the same time as a transmit command. In this case the latter is performed first.

3. The receive command starts by activating the receiver circuits. First a valid addressed frame is searched for. Upon detection an interrupt is issued, indicating that a data transfer to the system is required. This data byte must now be read by the CPU within 6 bit times after the interrupt, otherwise a Receiver Not Serviced condition will arise.

4. After the transfer of each data byte examine the Status Register. If the End of Frame (EOF) bit is set the data transfer is complete. If any other bit is set an error condition exists. If no bits are set there is still data to be received from the terminal.
   Note: the last byte of data received (the byte which caused EOF to be set) is extraneous and should be discarded.



Note: The last data byte is not part of the received data and should be discarded!

Figure 5.9 CPU Receive (2 of 2)

Transmitting via DMA

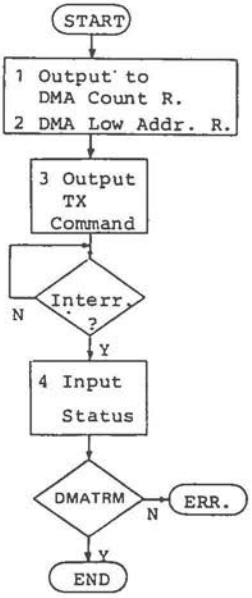


Figure 5.10 (1 of 2)
DMA Transmit

1. Initialize the DMA Counter Transmit register to OFFH.

2. Write the 8 LSBs of the first memory location to the Transmit DMA Address Register. (The higher bits are to be supplied by the DMA controller).

3. Issue a transmit command in accordance with the type of interframe transmission required. If both TXFR and TXFLG are set (Command Register: 11xx0000) interframe time will be filled with contiguous flags. If TXFR is set and TXFLG is zero (Command Register: 10xx0000) the frame is followed by an idling sequence of 25 ones.

4. The DMA controller issues an interrupt after the last transfer. The Status Register should now be read. Two conditions are significant. The DMATRM bit set denotes successful completion of the DMA transfer. Any error results in a premature termination of the transfer and the setting of the Transmitter Not Serviced bit.

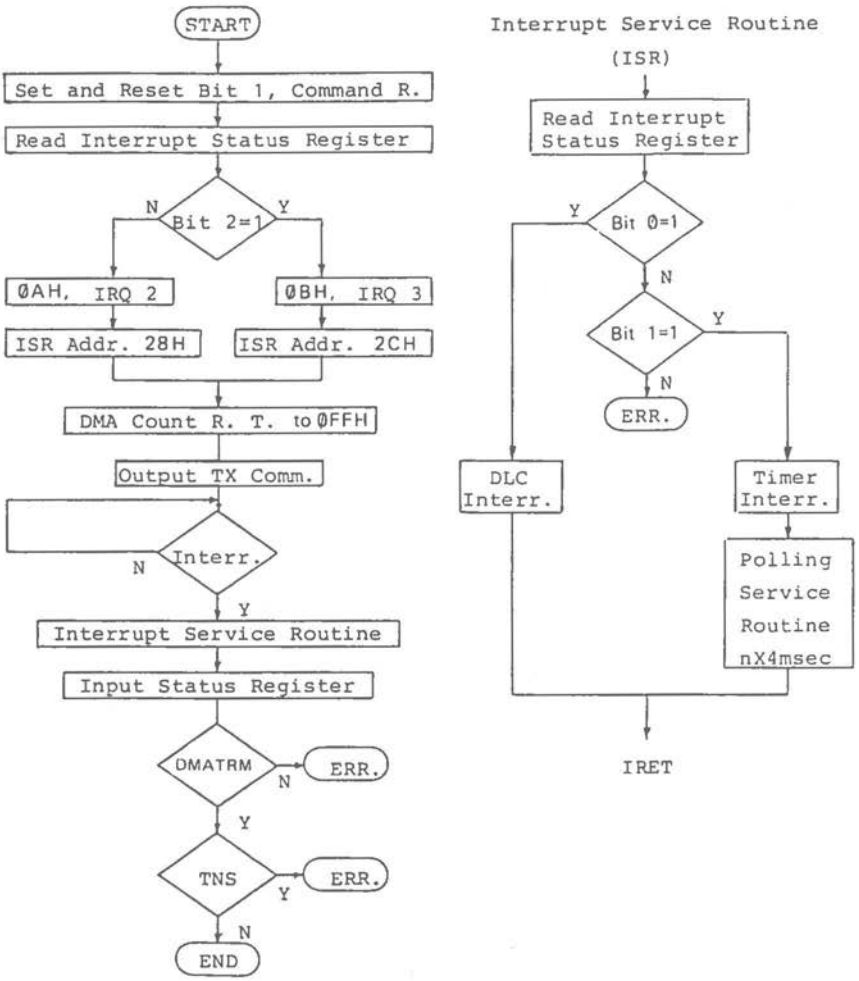


Figure 5.10 DMA Transmit (2 of 2)

Receiving via DMA

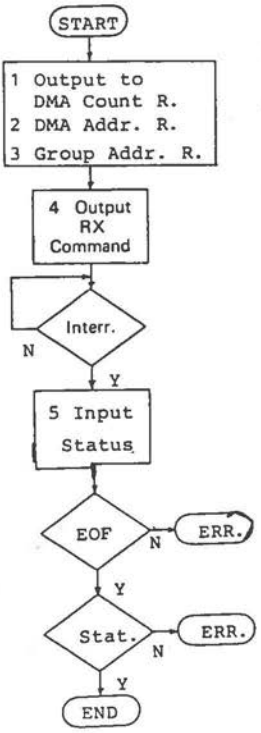


Figure 5.11 (1 of 2)
DMA Receive

1. Initialize the Receive DMA Count Register to OFFH.

2. Write the 8 LSBs of the first memory location to the Receive DMA Address Register (the higher bits are to be supplied by the DMA controller).

3. If the receive command is to be unique, group, or global, write the Group Address Register (4 MSBs only; LSBs must be all set).

4. Write the receive command to the Command Register, specifying RXA and RXB in accordance with the type of receive required and with the IO/DMA bit zero.

5. After a valid addressed frame has been detected data is transferred until all data has been received. Following the last byte an interrupt is generated indicating that a frame has been received. Reading the Status register should show that only the EOF bit and possibly the DMATRM bit are set. (DMATRM is set if the DMA count was exhausted before the end of frame was Figure 5.11 (1 of 2) detected). Any other bits set denote an DMA Receive error condition.

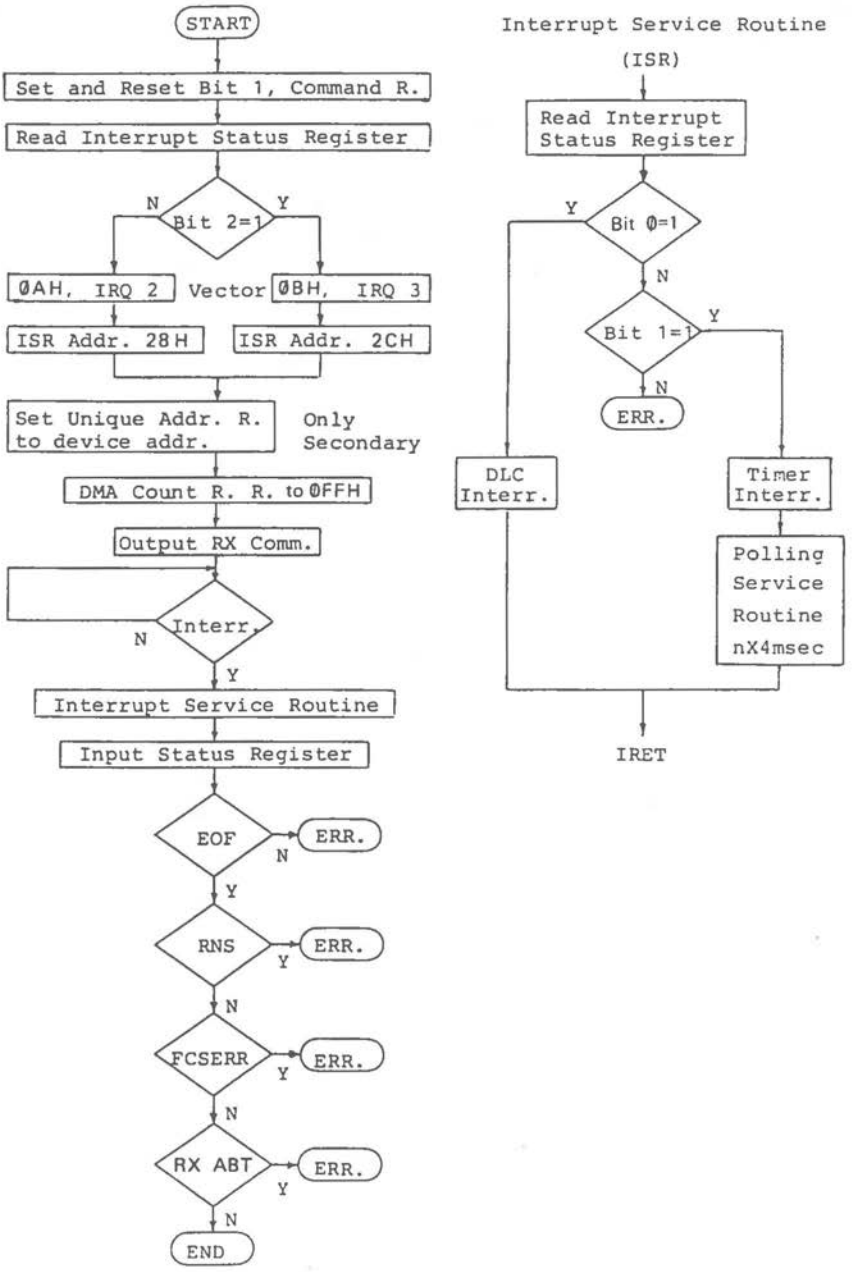


Figure 5.11 DMA Receive (2 of 2)

Disk Storage

INTRODUCTION

Both flexible disks and fixed (Winchester or hard) disks can serve as disk storage for the NCR PERSONAL COMPUTER. Capacity of an internal fixed disk is 10 MB, standard diskette drives store data in 48 TPI format.

A typical system includes either two flexible disk drives or one flexible disk drive in conjunction with a hard disk drive. You may alter your system by means of the following kits:

3299-K702 - 48 TPI flexible disk drive
3299-K701 - 1.2 MB (96 TPI) flexible disk drive
3299-K705 - Controller and cables for 3299-K701
3299-K750 - Internal fixed disk drive
3282-1212 - External fixed disk drive
3299-K725 - Controller for fixed disk drives

K702 makes use of the flexible disk drive controller incorporated in the system main processor board.

48 TPI FLEXIBLE DISK DRIVE

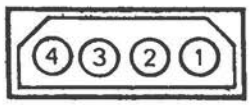
The computer has one or two 5 1/4-inch flexible disk drives integrated to provide mass storage of programs and data. The drives contain read/write controller electronics, driver mechanics, read/write heads, and head positioning mechanisms. Systems with an integrated fixed drive have only one integrated flexible disk drive; this is mounted on the leftmost side of the computer's front panel.

Flexible disk drives are connected in a daisy-chain configuration. A DIP resistor module provides the terminator in the last unit in the chain.

Tracks per inch (TPI)			48
Tracks per disk side			40
Unformatted capacity			500 KB
Motor rotation speed			300 r.p.m
Motor start time			<400msec
Drive timeout			After 15 s
Head movement (track to track)			< 6 ms
Maximum seek time			495 ms
Latency time (at 300 r.p.m)			100 ms
Data transfer rate			250K bits/sec
Data recording format			MFM
Power requirements (at controller on main board)	12Vdc 0.6V		250 mA (average)
			900 mA (peak)
	5Vdc 0.25V		500 mA (average)
			800 mA (peak)

Figure 6.1 Flexible Disk Drive Technical Data

Power connections to the flexible disk drive are illustrated in Figure 6.2.



Pin Number	
1	+12VDC
2	GND (+12V Return)
3	GND (+5V Return)
4	+5VDC

Figure 6.2 Flexible Disk Drive Power Connections

The flexible disk drive contains sensors to detect the diskette index hole, the head at track 0, and whether the write protect notch is covered.

Figure 6.3 illustrates the main components of the flexible disk drive.

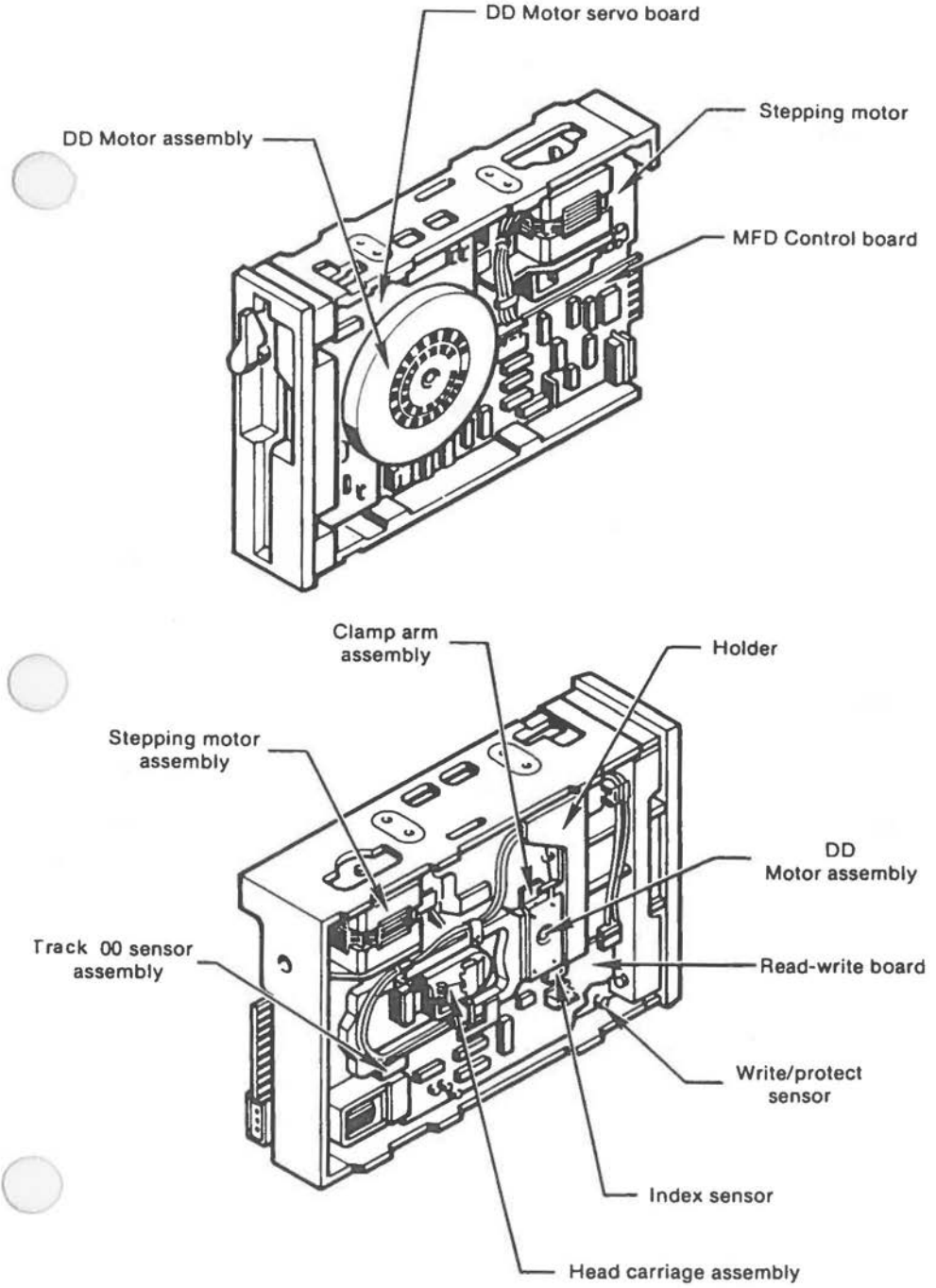


Figure 6.3 Flexible Disk Drive Components

The flexible disk drive is strapped internally. Of particular interest is the strap which determines to which Drive Select signal the drive is to respond. Internal strapping is illustrated in Figure 6.4.

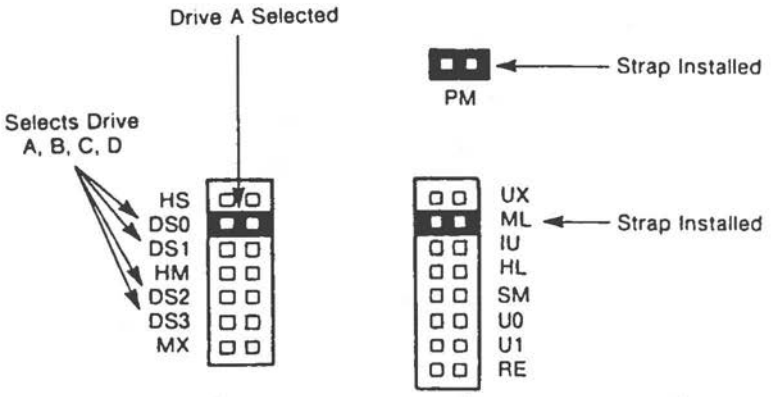


Figure 6.4 Flexible Disk Drive Straps

CONTROL AND DATA SIGNALS

The flexible disk drives use the standard pin assignments as shown in Figure 6.5. Figure 6.6 shows the corresponding edge connector on the flexible disk drive.

The input and output signals to the disk drive are standard industry-compatible signals. An overview of flexible disk control signals is provided in Figures 6.7 and 6.8. These signals are:

Drive Select (0,1,2,3)
  Active low input signal to select one of two flexible disk drives (only 0 and 1 are used).

Head Select
  An active low input selects head 1, otherwise head 0 is selected.

Signal	Signal Pin	Signal Direction
Ready/	34	OUT
Head Select/	32	IN
Read Data/	30	OUT
Write Prot./	28	OUT
Track 0/	26	OUT
Write Gate/	24	IN
Write Data/	22	IN
Step/	20	IN
Direction/	18	IN
Motor On (Motor On 1)	16	IN
Drive S. 2/ (Drive S. 0/)	14	IN
Drive S. 1/ (Drive S. 1/)	12	IN
Drive S. 0/ (Motor On 0)	10	IN
Index/	8	OUT
Drive S. 3/	6	IN
Head Load/	4	IN

NOTE: All odd numbered pins are signal ground. Designations in parentheses are for system with internal fixed disk drive.

Figure 6.5 FDD Pin Assignments

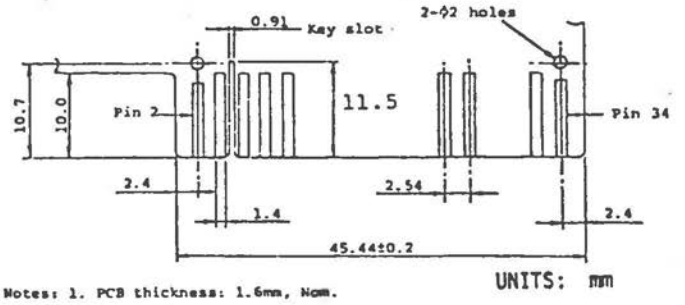


Figure 6.6 FDD Edge Connector

Index
Active low output signal which is at an active level each time the index hole is sensed. This signal is active for one pulse each disk revolution to indicate the beginning of a track. Index is held active when no flexible disk is inserted in the system.

Motor On
Active low input signal to turn the motor on. Time has to be allowed by the system before reading or writing to allow the motor to start.

Direction Select
Input signal to define the direction the heads move when the step line is pulsed. Active low causes the head to move toward the center of the disk. Active high causes the head to move toward the outside of the disk.

Step
Active low input signal to move the head in the direction specified by Direction Select. Each step pulse is delayed by 6ms from the preceding step.

Write Data
Input signal to provide the data to be written on the flexible disk. Each transition from high to low causes the current through the read/write heads to reverse, causing a data bit to be written. This line is enabled by Write Gate active. Write Data is not active during a read operation. The write pulse width can range from 150 nanoseconds to 2.5 microseconds.

To ensure data integrity, MFM write data can be pre-compensated on tracks 26-39. This bit shifting can be early or late with respect to the nominal bit cell position (see Figure 6.13)

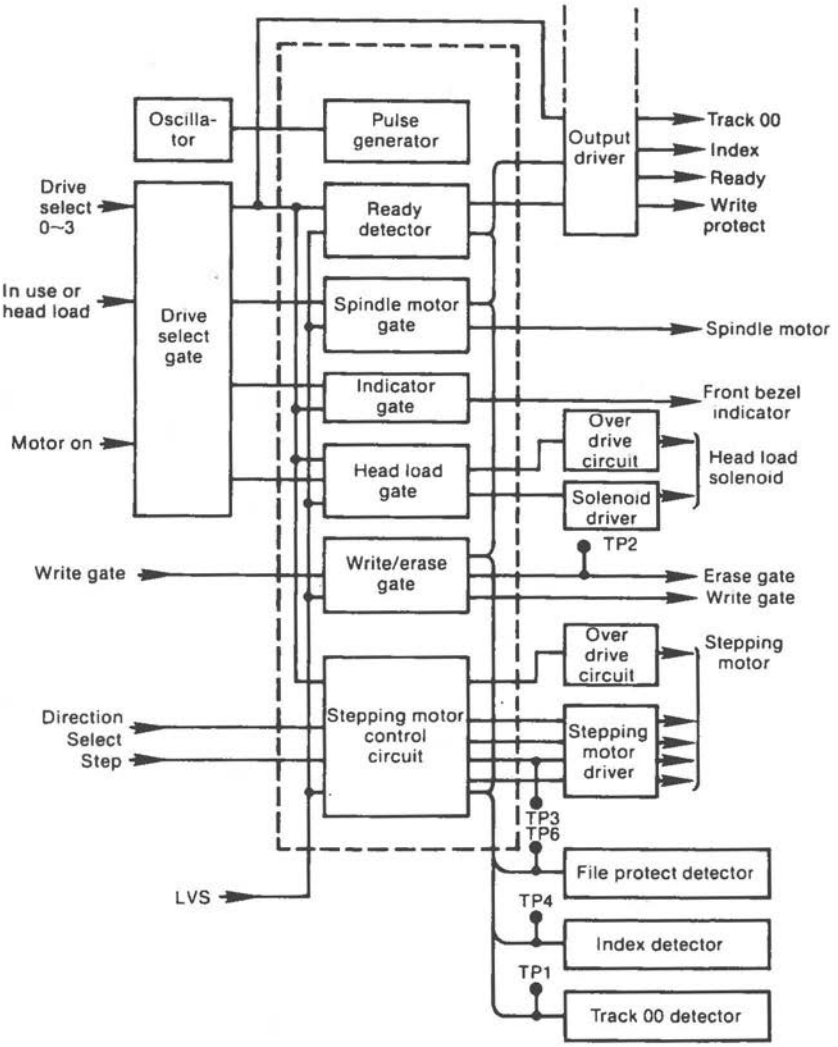


Figure 6.7 Flexible Disk Drive Control

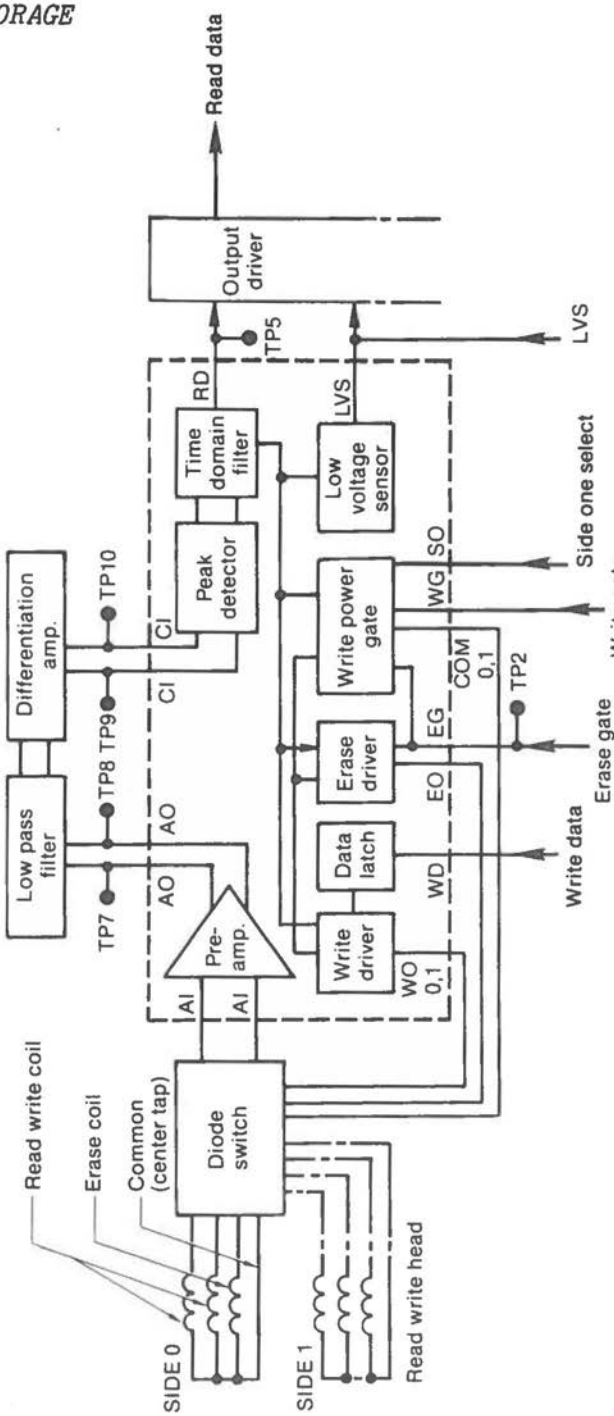


Figure 6.8 Flexible Disk Drive Read/Write Control

Write Gate
Active low input signal to enable data to be written on the flexible disk. Write Gate not active enables the stepper logic and read data logic.

Track 00
Active low output signal to indicate when the read/write heads are positioned at track zero. Track 00 is not active when the read/write heads are not at track zero.

Write Protect
Active low output signal to indicate when a write protected disk is installed in the drive. The drive inhibits writing and provides the write protect signal, irrespective of the state of the Write Data and Write Gate signals.

Read Data
Output signal containing composite clock and data pulses.

Head Load
Active low input signal to load the head. The indicator on the front panel is turned on.

Ready
This output signal is issued when the following conditions are satisfied:

* There is power at the unit.
* The diskette is installed.
* Disk rotation is more than half nominal speed.
* Two Index pulses have been counted since disk rotation exceeded half nominal speed.

Figures 6.9 - 6.15 illustrate signal timing.

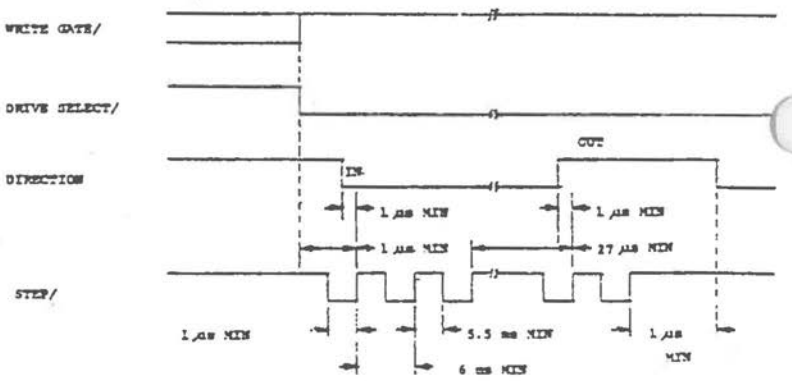


Figure 6.9 Track Access Timing

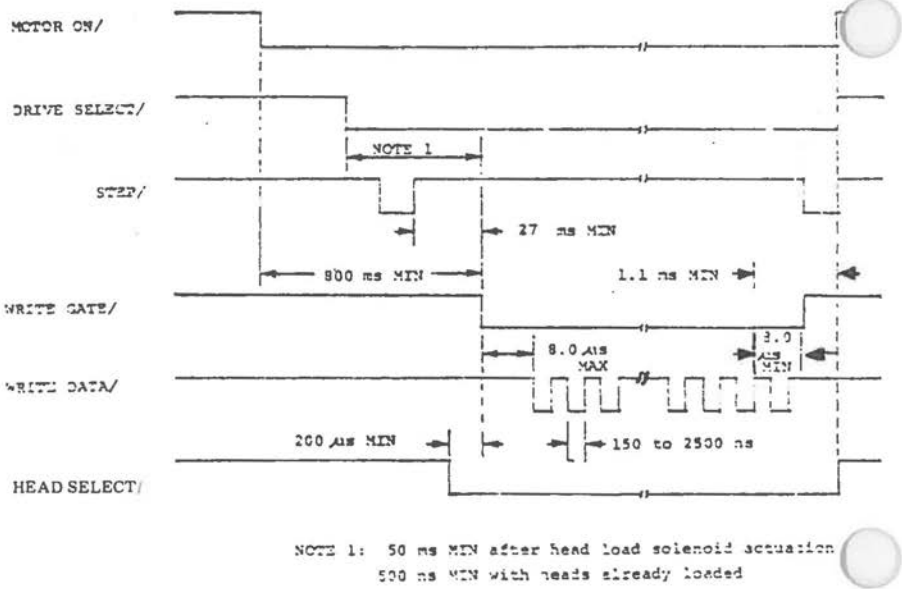


Figure 6.10 Write Initiate Timing

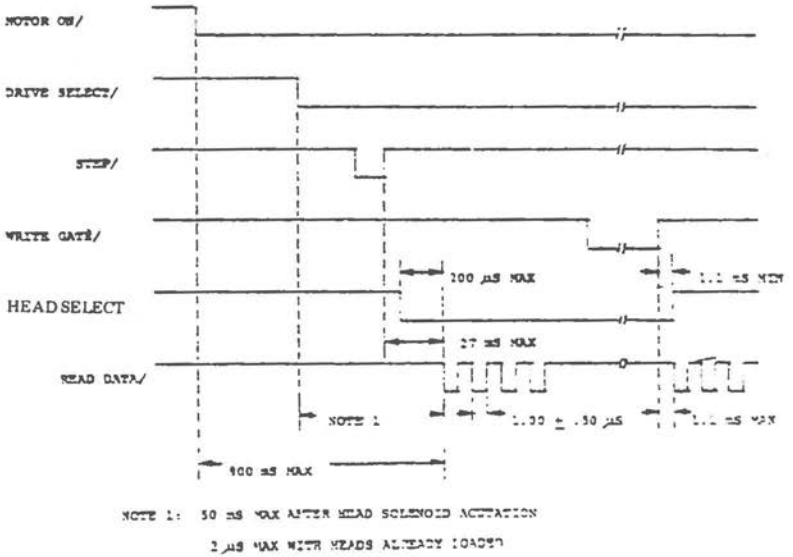


Figure 6.11 Read Initiate Timing

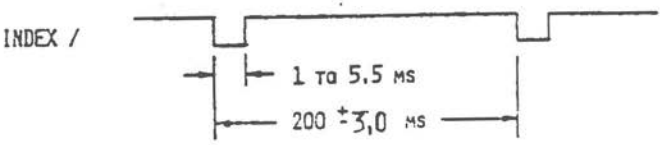


Figure 6.12 Index Timing

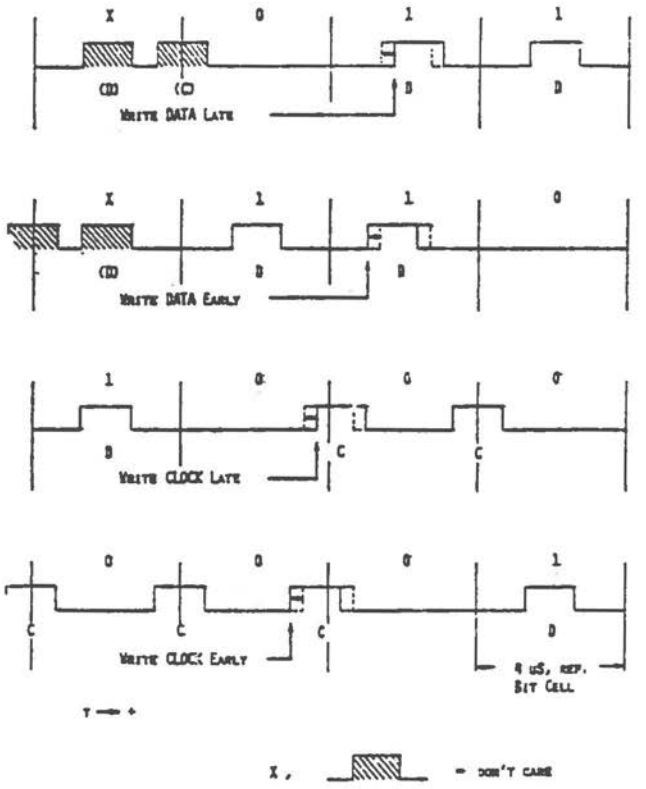


Figure 6.13 MFM Write Pre-compensation Patterns

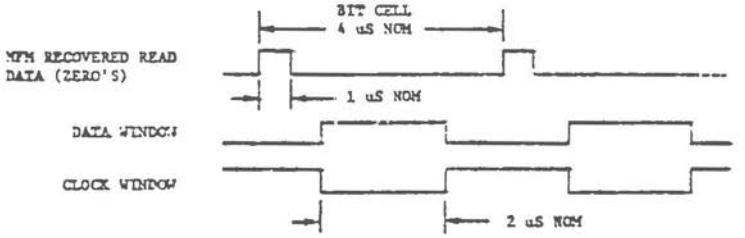


Figure 6.14 Nominal Clock And Data Window Timing

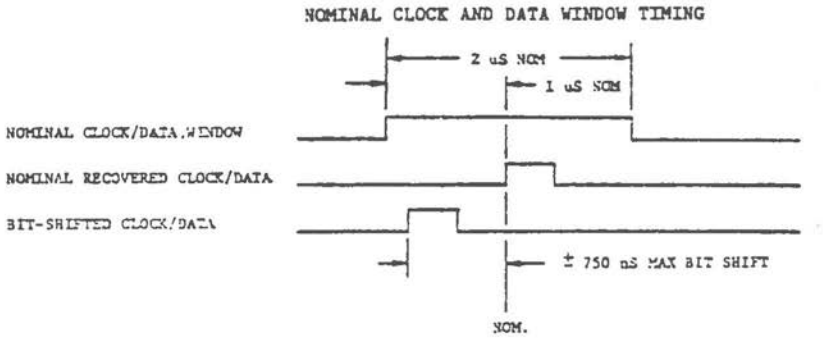


Figure 6.15 Bit Shift Timing

MINIMUM FORMAT REQUIREMENTS

Figure 6.14 illustrates index requirements and tolerances for flexible disk formatting.

Post-Index

Before initial recording of data upon a selected track, a minimum gap time of one millisecond is required to allow for drive-to-drive adjustment tolerances. This is the gap (Gap 1) from the edge of the index pulse to the beginning Sync Field for the ID field address mark. This gap allows for variations in Index pulse width, speed variations, and interchange tolerances between drives.

Pre-Index

The Pre-Index gap (Gap 4) is required to compensate for maximum speed variations between drives. Physically the gap is the space between the last sector and the beginning of the index pulse. Its minimum length is 7.2 milliseconds.

Pre-Data

The Pre-Data gap (Gap 2) has a minimum timing requirement of 489 microseconds. Gap timing is determined by the erase turn-on circuit tolerance and the tunnel-erase structure of the read/write head.

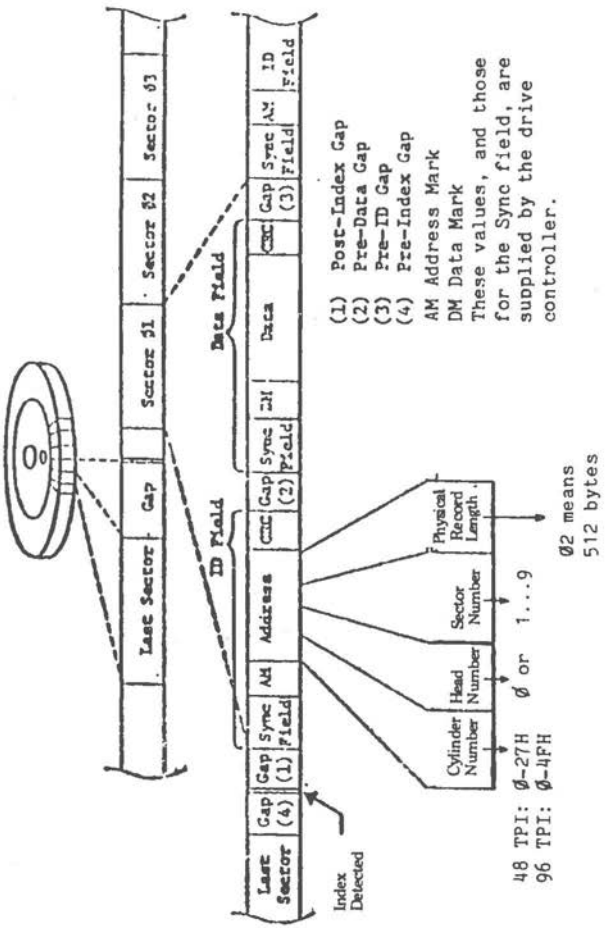


Figure 6.16 Flexible Disk Format

Pre-ID
The Pre-ID gap length is a minimum of 1.64 milliseconds for 128-byte FM or 256-byte MFM sector lengths, but varies with sector size. The Pre-ID gap (Gap 3) is based on the tunnel-erase structure of the read/write head and maximum erase-circuit turn-off delay.

FLEXIBLE DISK CONTROLLER
The PD765 Flexible Disk Controller (FDC) makes use of three interface ports to the microprocessor:

OUT 3F2H Drive select command output.

IN 3F4H Read the FDC's Main Status Register.

OUT 3F5H Output up to nine commands to the command stack.

IN 3F5H Input one of up to seven results from the result stack, including up to four 8-bit status registers,

The FDC can perform 15 different operational commands, each initiated by the transfer of a byte of command identification and any parameters belonging to that command via port 3F5H. Each operation in the process of flexible disk control can be regarded as consisting of three phases:

* Command Phase
  During the Command Phase, the FDC receives commands and inherent parameters from the microprocessor.

* Execution Phase
  During this phase, the actual execution of the command takes place.

* Result Phase
  During this phase, useful operational status information can be read by the microprocessor via the Main Status Register and/or the other four status registers. If the operation involved microprocessor reading of flexible disk data, that data is made available.

Main Status Register
Figure 6.17 indicates the status information which can be read via port 3F4H.

Bits 6 (DIO) and 7 (RQM) of the Main Status Register are of particular importance for the transfer of information between FDC and microprocessor. Before writing a byte to the FDC, these bits must be 0 and 1, respectively. Reading a byte from the FDC requires both these bits to be 1.

The FDC in your NCR PERSONAL COMPUTER operates in DMA mode, with the result that interrupt detection before reading individual bytes is not required. In this mode, the FDC communicates with the DMA controller by means of an exchange of DRQ and DACK/ signals.

The significance of the FDC Main Status Register when identifying the three operational phases is as follows:

* Command Phase
  Commands/data for operation are received from the microprocessor. During this phase, DIO and RQM must be 0 and 1, respectively.

* Execution Phase
  Performance of the requested operation. At the beginning of this phase, DIO and RQM go to 1 and 0, respectively.

* Result Phase
  Information available to the microprocessor, denoted by bits 6 and 7 of the Main Status Register being set.

Bit	Name	Symbol	Description
D0	FDD 0 Busy	D0B	FDD 0 is in the Seek mode
D1	FDD 1 Busy	D1B	FDD 1 is in the Seek mode
D2	FDD 2 Busy	D2B	FDD 2 is in the Seek mode
D3	FDD 3 Busy	D3B	FDD 3 is in the Seek mode
D4	FDC Busy	CB	A read or write command is in process
D5	Non-DMA mode	NDM	The FDC is in non-DMA mode. This bit is set only during execution phase in non-DMA mode. Transition to zero indicates execution phase has ended
D6	Data Input /Output	DIO	Indicates direction of data transfer between FDC and Data Register. If DIO is set, then transfer is from Data Register to the Processor. If DIO is zero, then transfer is from Processor to Data Register
D7	Request for Master	RQM	Indicates Data Register is ready to send or receive data to or from the Processor. Both bits DIO and RQM can be used to perform handshaking functions of "ready" and "direction" to the processor

Figure 6.17 FDC Main Status Register

Disk Select Register
The Disk Select Register is a write only register (Port 3F2H) used to:

* Select disk drive units.
* Control the motors of these units.
* Enable controller interrupts and DMA requests.

Figure 6.18 explains the significance of the individial bits of this register.

Bit	Significance
0 1	} Binary value (bit 1 is MSB) selecting one of the } units 0-3, corresponding to drives A-D, respectively
2	Bit set enables FDC
3	Bit set enables DMA request and FDC interrupts to be to I/O interface. Bit zero disables I/O interface drivers.
4	Turns motor of drive A on (bit set) or off (bit zero)
5	B
6	C
7	D

Figure 6.18 Disk Select Register

FDC Commands and their Parameters
Figure 6.19 presents a summary of commands and parameters transmitted via port 3F5H. A command phase consists of up to 9 bytes. The tables comprising Figure 6.20 explain the abbreviations used in Figure 6.19.

A system interrupt (type 0EH) denotes that an execution phase is completed or prematurely terminated. Status information and/or data is then

available for input at port 3F5H. The sigificance of this result phase (consisting of up to seven bytes) is also set out in Figure 6.19. The significance of the status information which can be read during the result phase is set out in detail in Figure 6.26. (This status information is not to be confused with the Main Status which can be read at any time via port 3F4H.)

It is important that commands and their data are input in the correct order and that return information is read completely, even if elements of this information are not required.

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
		READ DATA								
	W	MT	MFM	SK	0	0	1	1	0	Command Codes
C	W	0	0	0	0	0	HDS	DS1	DS0	
o	W	C								Sector ID info.
m	W	H								prior to Com-
m	W	R								mand execution
a	W	N								
n	W	EOT								
d	W	GPL								
	W	DTL								
E										Data transfer
x										between the FDD
e										and main system
	R	ST0								Status info
R	R	ST1								after Command
e	R	ST2								execution
s	R	C								
u	R	H								Sector ID info.
l	R	R								after Command
t	R	N								execution

Figure 6.19 FDC Command Summary [1 of 10]

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
READ DELETED DATA										
	W	MT	MFM	SK	0	1	1	0	0	Command Codes
Command	W	0	0	0	0	0	HDS	DS1	DS0	
	W	———	———	———	———	C	———	———	———	Sector ID info.
	W	———	———	———	———	H	———	———	———	prior to Com-
	W	———	———	———	———	R	———	———	———	mand execution
	W	———	———	———	———	N	———	———	———	
	W	———	———	———	———	EOT	———	———	———	
	W	———	———	———	———	GPL	———	———	———	
	W	———	———	———	———	DTL	———	———	———	
Execution										Data transfer between the FDD and main system
	R	———	———	———	———	ST0	———	———	———	Status info
Result	R	———	———	———	———	ST1	———	———	———	after Command
	R	———	———	———	———	ST2	———	———	———	execution
	R	———	———	———	———	C	———	———	———	
	R	———	———	———	———	H	———	———	———	Sector ID info.
	R	———	———	———	———	R	———	———	———	after Command
	R	———	———	———	———	N	———	———	———	execution

Figure 6.19 FDC Command Summary (2 of 10)

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
		WRITE DATA								
	W	MT	MFM	0	0	0	1	0	1	Command Codes
C	W	0	0	0	0	0	HDS	DS1	DS0	
o	W	——— C ———								Sector ID info.
m	W	——— H ———								prior to Com-
m	W	——— R ———								mand execution
a	W	——— N ———								
n	W	——— EOT ———								
d	W	——— GPL ———								
	W	——— DTL ———								
E										Data transfer
x										between the FDD
e										and main system
	R	——— ST0 ———								Status info
R	R	——— ST1 ———								after Command
e	R	——— ST2 ———								execution
s	R	——— C ———								
u	R	——— H ———								Sector ID info.
l	R	——— R ———								after Command
t	R	——— N ———								execution

Figure 6.19 FDC Command Summary (3 of 10)

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
WRITE DELETED DATA										
	W	MT	MFM	0	0	1	0	0	1	Command Codes
C	W	0	0	0	0	0	HDS	DS1	DS0	
o	W	C								Sector ID info.
m	W	H								prior to Com-
m	W	R								mand execution
a	W	N								
n	W	EOT								
d	W	GPL								
	W	DTL								
E										Data transfer
x										between the FDD
e										and main system
	R	ST0								Status info
R	R	ST1								after Command
e	R	ST2								execution
s	R	C								
u	R	H								Sector ID info.
l	R	R								after Command
t	R	N								execution

Figure 6.19 FDC Command Summary (4 of 10)

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
		READ A TRACK								
	W	MT	MFM	SK	0	0	0	1	0	Command Codes
C	W	0	0	0	0	0	HDS	DS1	DS0	
Command	W	C								Sector ID info. prior to Command execution
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the FDD and main system FDC reads all of cylinder's contents from index hole to EOT
	R	ST0								Status info after Command execution
Result	R	ST1								
	R	ST2								
	R	C								Sector ID info. after Command execution
	R	H								
	R	R								
	R	N								

Figure 6.19 FDC Command Summary (5 of 10)

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
READ ID										
Command	W	MT	MFM	0	0	1	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution										The first correct ID info on the Cylinder is stored in Data Register
Result	R	ST0								Status info after Command execution
	R	ST1								
	R	ST2								
	R	C								
	R	H								Sector ID info. during Execution Phase
	R	R								
	R	N								

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
FORMAT A TRACK										
	W	MT	MFM	0	0	1	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Command	W	N								Bytes/Sector
	W	SC								Sectors/Cylinder
	W	GPL								Gap 3
	W	D								Filler Byte
Execution										FDC formats an entire cylinder
Result	R	ST0								Status info after Command execution
	R	ST1								
	R	ST2								
	R	C								
	R	H								In this case the ID info has no meaning
	R	R								
	R	N								

Figure 6.19 FDC Command Summary (6 of 10)

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
SCAN EQUAL										
	W	MT	MFM	SK	1	0	0	0	1	Command Codes
C	W	0	0	0	0	0	HDS	DS1	DS0	
o	W	.....			C					Sector ID info.
m	W				H					prior to Com-
m	W				R					mand execution
a	W				N					
n	W				EOT					
d	W				GPL					
	W				STP					
E										Data compared
x										between the FDD
e										and main system
	R				ST0					Status info
R	R				ST1					after Command
e	R				ST2					execution
s	R				C					
u	R	.....			H					Sector ID info.
l	R				R					after Command
t	R	.....			N					execution
SEEK										
C	W	0	0	0	0	1	1	1	1	Command Codes
m	W	0	0	0	0	0	HDS	DS1	DS0	
d	W				NCN					
										Head positioned
E										over proper
x										Cylinder on
e										Diskette

Figure 6.19 FDC Command Summary (7 of 10)

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
SCAN LOW OR EQUAL										
	W	MT	MFM	SK	1	1	0	0	1	Command Codes
C	W	0	0	0	0	0	HDS	DS1	DS0	
o	W	---------- C ----------								Sector ID info.
m	W	---------- H ----------								prior to Com-
m	W	---------- R ----------								mand execution
a	W	---------- N ----------								
n	W	---------- EOT ----------								
d	W	---------- GPL ----------								
	W	---------- STP ----------								
E										Data compared
x										between the FDD
e										and main system
	R	---------- ST0 ----------								Status info
R	R	---------- ST1 ----------								after Command
e	R	---------- ST2 ----------								execution
s	R	---------- C ----------								
u	R	---------- H ----------								Sector ID info.
l	R	---------- R ----------								after Command
t	R	---------- N ----------								execution

Figure 6.19 FDC Command Summary (8 of 10)

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
		SCAN HIGH OR EQUAL								
	W	MT	MFM	SK	1	1	1	0	1	Command Codes
C	W	0	0	0	0	0	HDS	DS1	DS0	
o	W	··	··	··	··	C	··	··	··	Sector ID info.
m	W	——	——	——	——	H	——	——	——	prior to Com-
m	W	——	——	——	——	R	——	——	——	mand execution
a	W	——	——	——	——	N	——	——	——	
n	W	——	——	——	——	EOT	——	——	——	
d	W	——	——	——	——	GPL	——	——	——	
	W	——	——	——	——	STP	——	——	——	
E										Data compared
x										between the FDD
e										and main system
	R	--	--	--	--	ST0	——	——	——	Status info
R	R	——	——	——	——	ST1	--	--	--	after Command
e	R	——	——	——	——	ST2	··	··	··	execution
s	R	——	——	——	——	C	——	——	——	
u	R	——	——	——	——	H	——	——	——	Sector ID info.
l	R	——	——	——	——	R	--	--	--	after Command
t	R	··	··	··	··	N	··	··	··	execution

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
		RECALIBRATE								
C	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS1	DS0	
E										Head retracted
										to Track 0

Figure 6.19 FDC Command Summary (9 of 10)

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
SENSE INTERRUPT STATUS										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R	ST0								Status info. at the the end of each seek operation about the FDC
Result	R	PCN								
SPECIFY TIME										
Command	W	0	0	0	0	0	0	1	1	Command Codes
Command	W	SRT						HUT		
Command	W	HLT							ND	
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R	ST3								Status info about FDD
INVALID										
Command	W	Invalid Codes								Invalid Command Codes (NoOp-FDC goes into Stand-by Status)
Result	R	ST0								ST0 = 80

Figure 6.19 FDC Command Summary (10 of 10)

SYMBOL	NAME	DESCRIPTION
A0	Address Line 0	A0 controls selection of Main Status Register (A0 = 0) or Data Register (A0 = 1)
C	Cylinder No.	C stands for the current selected Cylinder number
D	Data	D stands for the data pattern for writing to a Sector
D7-D0	Data Bus	8-bit Data Bus where D7 is the most significant bit, and D0 is the least significant bit
DS0, DS1	Drive Select	DS stands for a selected drive number 0 or 1
DTL	Data Length	When N is defined as 00, DTL stands for the data length which users are going to read out of, or write into the Sector. Otherwise, DTL should be 0FFH
EOT	End of Track	EOT stands for the final Sector number of a Cylinder
GPL	Gap Length	GPL stands for the length of Gap 3 (specing between Sectors excluding VCO Sync Field)
H	Head Address	H stands for head number 0 or 1, as specified in ID field
HDS	Head Select	HDS stands for a selected head number 0 or 1 (H = HDS in all command words)

Figure 6.20 Abbreviations Used In FDC Command Summary (1 of 3)

SYMBOL	NAME	DESCRIPTION
HLT	Head Load Time	HLT stands for the head load time in the FDD (4 to 512ms in 4ms increments)
HUT	Head Unload Time	HUT stands for the head unload time after a read or write operation has occurred (0 to 480ms in 32ms increments)
MFM	FM or MFM Mode	If MF is low, FM mode is selected; if it is high, MFM mode is selected
MT	Multi-Track	If MT is high, a multi-track operation is to be performed (a cylinder under both HD0 and HD1 will be read or written)
N	Number	N stands for the number of data bytes written in a Sector
NCN	New Cylinder Number	NCN stands for new Cylinder number, which is going to be reached as a result of the Seek operation. Desired position of Head
ND	Non-DMA Mode	ND stands for operation in the non-DMA Mode
PCN	Present Cylinder Number	PCN stands for the Cylinder number at the completion of SENSE INTERRUPT STATUS Command. Head position at present time
R	Record	R stands for the Sector number which will be read or written

Figure 6.20 Abbreviations Used In FDC Command Summary (2 of 3)

SYMBOL	NAME	DESCRIPTION
R/W	Read/Write	R/W stands for either Read (R) or Write (W) signal
SC	Sector	SC indicates the number of Sectors per Cylinder
SK	Skip	SK stands for Skip Deleted Data Address Mark
SRT	Step Rate Time	SRT stands for the Stepping Rate for the FDD (2 to 32ms in 2ms increments). Same Stepping Rate applies to all drives
ST0 ST1 ST2 ST3	Status 0 Status 1 Status 2 Status 3	ST0-3 stands for one of four registers storing the status information after a command has been executed. This info is available during the result phase after command execution. These registers are not to be confused with the main status register (selected by A0 = 0). ST0-3 may be read only after a command has been executed. They contain information relevant to that particular command.
STP	Scan Test	During a Scan operation, if STP = 1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA), and if STP = 2 , then alternate sectors are read and compared

Figure 6.20 Abbreviations Used In FDC Command Summary (3 of 3)

Notes concerning individual FDC commands (the status registers referred to here are described in the subsequent section):

READ DATA

After head location and settling time, the requested sector is located by means of the ID marks and ID fields on the located track. The FDC then outputs the data from the flexible disk data field one byte at a time. This operation is a multi-sector read, that is, sectors are output by the FDC until the DMA issues a TC (terminal count signal). If there are sectors remaining on the track, these are read but only a Cyclic Reduncancy Check is performed; they are not output to the data bus. Figure 6.21 states factors affecting transfer capacity of a single read command.

MT	MFM	Byte/ Sector N	Max.Transfer Capacity (Bytes/Sector) (Number of Sectors)	Final Sector Read from Diskette
0	0	00	(128)(26) = 3,328	26 at Side 0
0	1	01	(256)(26) = 6,656	or 26 at Side 1
1	0	00	(128)(52) = 6,656	26 at Side 1
1	1	01	(256)(52) = 13,312	26 at Side 1
0	0	01	(256)(15) = 3,840	15 at Side 0
0	1	02	(512)(15) = 7,680	or 15 at Side 1
1	0	01	(256)(30) = 7,680	15 at Side 1
1	1	02	(512)(30) = 15,360	15 at Side 1
0	0	02	(512) (8) = 4,096	8 at Side 0
0	1	03	(1024) (8) = 8,192	or 8 at Side 1
1	0	02	(512)(16) = 8,192	8 at Side 1
1	1	03	(1024)(16) = 16,384	8 at Side 1

Figure 6.21 Transfer Capacity

If the MT bit is set data is read from both sides of the diskette (the entire cylinder), starting with side 0, sector 1 and finishing with the last sector on side 1.
If N is 0 the data length of each sector is defined by DTL. If there are more bytes actually in the sector the excess bytes are subjected to CRC but do not appear on the data bus.
After accessing the sector(s), there is normally a head unload interval. However, this interval does not apply if the FDC has already detected a subsequent command entailing an access operation. This can save time when accessing large amounts of data.
When transferring data from FDC to the microprocessor, the FDC requires microprocessor attention every 27 (FM mode) or 13 (MFM mode) microseconds. Failing this, OR in Status Register 1 (see below) is set.

Possible error conditions:
Failure to find the specified sector sets the ND flag in Status Register 1. In Status Register 0, bit 6 is set and bit 7 is zero, and the command is terminated. CRC failure sets the DE flag in Status Register 1, and terminates the command. In addition, bit 6 in Status Register 0 is set, bit 7 is zero. If the CRC failure lies in the Data rather than the ID field, the DD flag in Status Register 2 is also set.
External termination (TC signal) influences the C, H, R, and N information. This information further depends on the MT and EOT values given in the command (see Figure 6.22) and can be read in the Result Phase.

READ ONE TRACK
All Data Fields on a track are read, even if CRC failure is encountered. In this case the NDR flag in Status Register 1 is set. Failure to find an ID Address Mark sets the MA flag in Status Rgister 1 and terminates the command.

MT	EOT	Final Sector Transferred to Processor	ID Info at Result Phase C	H	R	N
0	1A	Sector 1 to 25 at Side 0	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	C+1	NC	R=01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C+1	NC	R=01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				
1	1A	Sector 1 to 25 at Side 0	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	NC	LSB	R=01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R+1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C+1	LSB	R=01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				

Notes: 1. NC [No Change]: Same value as at beginning of command execution.

2. LSB [Least Significant Bit]: The least significant bit of H is complemented.

Figure 6.22 Status At External Termination

READ DELETED DATA
As READ DATA exept that upon encountering a Deleted Data Address Mark at the beginning of a Data field and assuming that SK was 0 when the command was issued, the data in the sector is transmitted, the CM flag in Status Register 2 is set and the command is terminated.

WRITE DATA
When the sector R has been found the FDC accepts data from the data bus for that sector. Upon occurrence of a TC (Terminal Count) signal, the remainder of the data field currently being written is filled with zeros. CRC failure in an ID field sets the DE flag in Status Register 1 and terminates the command. Other details as in READ DATA.

WRITE DELETED DATA
As WRITE DATA, except that a Deleted Data Address Mark is written at the beginning of the Data field.

READ ID
This command returns the current head position by means of the first ID field the FDC can read. Failure to find an ID Address Mark sets the MA flag in Status Register 1. If there is no data ND is set in the same register.

FORMAT ONE TRACK
The format is determined by the values specified for N, SC, GPL, and D in the command. The sector is located by means of the C, H, R, and N values. The R value is incremented automatically until index hole detection indicates the end of the track. If an error situation arises, the EC flag in Status Register 0 is set and the command is terminated. The N, SC and GPL values for MFM recording are shown in Figure 6.23.

Sector Size	N	SC	GPL 1	GPL 2
256	01	12	0A	0C
256	01	10	20	32
512	02	08	2A	50
1024	03	04	80	F0
2048	04	02	C8	FF
4096	05	01	C8	FF

Figure 6.23 Sector Size Variables

GPL 1 avoid splicing between data field and ID field of contiguous sections. GPL 2 figures are for formatting purposes.

SCAN ONE TRACK

Data read by the FDC from disk is compared byte by byte with data on the data bus (supplied by DMA controller or microprocessor). The comparative condition can be =, <=, or >=, using one's complement arithmetic. The operation is carried out for the specified track with automatic incrementing of R until the condition is fulfilled, end of track is reached or a TC signal occurs.

If the scan condition is fulfilled, the SH bit in Status Register 2 is set; else SN is set. Figure 6.24 shows the possible conditions of comparison and the status of SH and SN (FDD = data read from disk, Bus = data read from data bus).

If a Deleted Data Address Mark is encountered and SK was issued in the scan command as a zero bit, this is regarded as the last sector and CM in Status Register 2 is set. If SK was issued set, CM is likewise set to indicate that the deletion mark was detected but the scan process skips that sector and continues to scan the remainder of the track.

Command	Status Register 2		Comments
	Bit 2 = SN	Bit 3 = SH	
Scan Equal	0	1	D FDD= D Bus
	1	0	D FDD<>D Bus
Scan Low or Equal	0	1	D FDD= D Bus
	0	0	D FDD< D Bus
	1	0	D FDD> D Bus
Scan High or Equal	0	1	D FDD= D Bus
	0	0	D FDD> D Bus
	1	0	D FDD< D Bus

Figure 6.24 Scan Status Codes

SEEK

The present cylinder number (PCN) is compared with the new cylinder number to be located (NCN). If PCN < NCN the Direction line to the flexible disk drive is set high; if PCN > NCN this line is low. When the cylinder has been located SE in Status Register 0 is set and the command terminated.

If the disk drive is not ready the NR flag in Status Register 0 is set and the command terminated. This command does not include a result phase. However, the termination of the command is effected by means of the Sense Interrupt Status command.

NOTE: FDC read and write commands affect the track/cylinder at which the head is currently positioned. To locate the track/cylinder itself an explicit SEEK command must be issued.

RECALIBRATE

With the Direction line high the read/write head retracts 77 step pulses or to track 0, whichever occurs first. The SE flag in Status Register 0 is set high. If the Track 0 signal is still low the EC flag in Status Register 0 is also set high. This command

does not include a result phase. However, the termination of the command is effected by means of the Sense Interrupt Status command.

SENSE INTERRUPT STATUS

An interrupt signal occurs upon entering the result phase of one of the above commands (or upon termination of SEEK or RECALIBRATE), or the Ready line changes state, or during execution in non-DMA mode. Figure 6.25 shows the significance of the three interrupt status bits affected in Status Regiser 0. It is important to check this interrupt status before attempting to read the information resulting from an FDC command.

Seek End	Interrupt Code		Cause
Bit 5	Bit 6	Bit 7	
0	1	1	Ready Line changed state, either polarity
1	0	0	Normal Termination of Seek or Recalibrate Command
1	1	0	Abnormal Termination of Seek or Recalibrate Command

Figure 6.25 Sense Interrupt Status

SPECIFY TIME

Sets Head Load Time (HLT) in the range 4 ms to 512 ms in increments of 4 ms, Head Unload Time (HUT) in the range 0 ms to 480 ms in 32 ms increments, Step Rate Time (SRT) in the range 2 ms to 32 ms in 2 ms increments. In addition, non-DMA mode is determined by bit ND being zero.

SENSE DRIVE STATUS
Returns information in Status Register 3.
If the FDC does not recognize a valid command, no interrupt is generated but DIO and RQM in the Main Status Register are set (as during a normal result phase). Status Register 0, containing 80H, must be read before a command can be issued to the FDC.

Motor On/Off
To turn a flexible disk drive motor on or off a byte must be issued via the Drive Select Command port 3F2H (Drive Code is a two-bit binary value: 0=A, 1=B):

	Bit:							
	7	6	5	4	3	2	1	0
Motor On	0	0	1=B	1=A	1	1	Drive Code	
Motor Off	0	0	0	0	1	1	0	0

NOTE: Motor Off is provided by the ROM BIOS after decrementation to zero of a timeout counter serviced by hardware timer ISR.

FDC Status
The Main Status Register is accessed via port 3F4H. There are four status registers which can be read via port 3F5H. It is imperative that all status registers affected by a particular command (see Figure 6.19) are actually read during the result phase in the specified order, even if the information they yield is not required. Figure 6.26 summarizes the four Status Registers.

Bit No.	Bit Name	Symbol	Description
	Status Register 0		
D7	Interrupt Code	IC	D7 = 0 and D6 = 0. Normal Termination of Command, (NT). Command was properly executed
D6			D7 = 0 and D6 = 1. Abnormal Termination of Command, (AT). Execution of Command was started, but was not successfully completed
			D7 = 1 and D6 = 0. Invalid Command issue, (IC). Command issued was never started
			D7 = 1 and D6 = 1. Abnormal Termination because during Command execution the Ready Signal from FDD changed state
D5	Seek End	SE	When the FDC completes the Seek Command, this flag is set to 1 (high)
D4	Equipment Check	EC	If a fault Signal is received from the FDD, or the Track 0 Signal fails to occur after 77 Step Pulses (Recalibrate Command) then this flag is set

Figure 6.26 Status Registers (1 of 6)

Bit			Description
No.	Name	Symbol	
Status Register 0			
D3	Not Ready	NR	When the FDD is in Not-Ready state and a Read or Write command is issued, this flag is set. If a Read or Write command is issued to Side 1 of a single sided drive, then this flag is set
D2	Head Address	HD	This flag is used to indicate the state of the head at interrupt
D1	Unit Select 1	US1	These flags are used to indicate a Drive Unit number at interrupt
D0	Unit Select 0	US0	

Figure 6.26 Status Register (2 of 6)

Bit No.	Bit Name	Bit Symbol	Description
	Status Register 1		
D7	End of Cylinder	EN	When the FDC tries to access a Sector beyond the final Sector of a Cylinder, this flag is set
D6			Not used. This bit is always 0
D5	Data Error	DE	When the FDC detects a CRC error in either the ID field or the data field, this flag is set
D4	Over Run	OR	If the FDC is not serviced by the main system during data transfers, within a certain time interval, this flag is set
D3			Not used. This bit is always 0 (low)
D2	No Data	ND	During execution of READ DATA, WRITE DELETED DATA or Scan Command this flag is set if the FDC cannot find the Sector specified in the ID Register This flag is set when FDC cannot read the ID field without error when executing READ ID command Flag is set if starting sector is not found during execution of READ A Cylinder command

Figure 6.26 Status Registers (3 of 6)

Bit No.	Bit Name	Bit Symbol	Description
	Status Register 1		
D1	Not Writable	NW	During execution of WRITE DATA, WRITE DELETED DATA or FORMAT a Cylinder command, if the FDC detects a write protect signal from the FDD, then this flag is set
D0	Missing Address Mark	MA	If the FDC cannot detect the ID Address Mark after encountering the index hole twice, then this flag is set
			If the FDC cannot detect the Data Address Mark or Deleted Data Address Mark, this flag is set. Also, at the same time the MD (Missing Address Mark in Data Field) of Status Register 2 is set

Figure 6.26 Status Registers (4 of 6)

Bit No.	Bit Name	Symbol	Description
Status Register 2			
D7			Bit not used, is always 0 (low)
D6	Control Mark	CM	Flag is set when FDC encounters a sector containing a Deleted Data Address Mark when executing READ DATA or SCAN Command
D5	Data Error in Data Field	DD	Flag is set when FDC detects a CRC error in the data field
D4	Wrong Cylinder	WC	This bit is related to ND bit; flag is set when contents of C differs from that of ID reg.
D3	Scan Equal Hit	SH	Flag is set when condition "equal" is satisfied upon Scan
D2	Scan Not Satisfied	SN	Flag is set when FDC cannot find a sector during Scan meeting the condition
D1	Bad Cylinder	BC	This bit is related to ND bit; flag is set when contents of C is OFFH and differs from the contents stored in ID register
D0	Missing Address Mark in Data Field	MD	Flag is set when during Read the FDC cannot find a Data Address Mark or a Deleted Data Address Mark

Figure 6.26 Status Registers (5 of 6)

Bit			Description
No.	Name	Symbol	
Status Register 3			
D7	Fault	FT	Bit indicates status of Fault signal from the FDD
D6	Write Protected	WP	Bit indicates status of Write Protected signal from the FDD
D5	Ready	RDY	Bit indicates status of Ready signal from the FDD
D4	Track 0	T0	Bit indicates status of Track-0 signal from the FDD
D3	Two Side	TS	Bit indicates status of Two-Side signal from the FDD
D2	Head	HD	Bit indicates status of Side Select signal to the FDD
D1	Unit Select 1	US1	Bit indicates status of Unit Select 1 signal to the FDD
D0	Unit Select 1	US0	Bit indicates status of Unit Select 0 signal to the FDD

Figure 6.26 Status Registers (6 of 6)

FIXED (WINCHESTER) DISK DRIVE

The Winchester disk drive uses a non-removable 5 1/4-inch disk as storage media. The disk has two recording surfaces, each of which is served by two heads. The total formatted capacity is 10M bytes over 306 cylinders (internal) or 20 MB over 615 cylinders (external). The Western Digital Winchester disk controller interfaces the disk drive to the host processor. All necessary buffers and receivers/drivers are included on the Winchester disk controller board to allow direct connection to the drive. Power requirements for the fixed disk drive are given in Figure 6.27, pin assignments for the DC power connector are illustrated in Figure 6.28.

Voltage	Current
+12V 5%	1.0A typical, 2.5A max.
+5V 5%	1.0A typical

Figure 6.27 Power Requirements

Pin	Connection
1	+ 12V
2	5V return
3	12V return
4	+ 5V

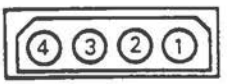


Figure 6.28 DC Power Connector

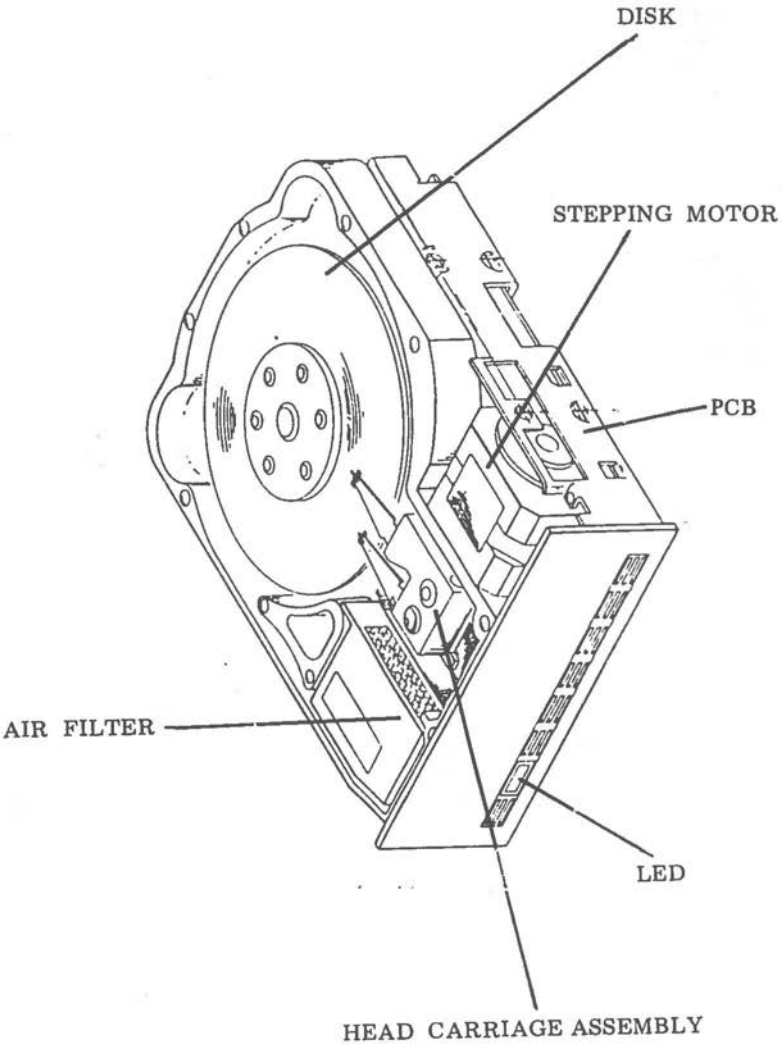


Figure 6.29 Fixed Disk Drive Components

	10 MB	20 MB
MBytes per unit [formatted]	12.75 (10)	25.62 (20)
Tracks per inch	350	700
Tracks per surface	310	615
Bytes per trk. [formatted]	10416 (8192)	10416 (8192)
Bits per inch [MFM]	9000	9000
Motor rotation speed [rpm]	3546	3546
Motor start time	15 s	15 s
Head movement:		
Average Seek	85 ms	85 ms
Maximum Seek [buffered]	215 ms	215 ms
Data transfer time	625 KB/s	625 KB/s

Figure 6.30 Technical Data

CONTROL AND DATA SIGNALS

Figure 6.31 specifies the pin assignments of the control signal interface between controller and the drive. The corresponding edge connector is shown in Figure 6.32. Data in MFM format is transferred by means of a separate cable: pin assignments and edge connector are illustrated in Figures 6.33 and 6.44.

The input and output signals for the drive are:

Write Gate

The active state of this signal enables data to be written on the disk. The inactive state of this signal enables data to be transferred from the drive. During power fail or power up all data recorded on the drive is retained, regardless of the frequency of occurrence of dc power transitions at the drive itself and regardless of the sequence and rate of dc voltage decay or rise. The Write Gate input is provided with an open circuit (inactive, high impedance driver) during these conditions.

Signal	Signal Pin	Signal Direction
Reserved	2	
HEAD SELECT2	4	
WRITE GATE/	6	OUT
SEEK COMPLETE/	8	IN
TRACK0/	10	IN
WRITE FAULT/	12	IN
HEAD SELECT0/	14	OUT
Reserved	16	
HEAD SELECT1/	18	OUT
INDEX/	20	IN
READY/	22	IN
STEP/	24	OUT
DRIVE SELECT1/	26	OUT
DRIVE SELECT2/	28	OUT
DRIVE SELECT3/	30	OUT
DRIVE SELECT4/	32	OUT
DIRECTION IN/	34	OUT
Odd numbered connections are Ground		

Figure 6.31 Fixed Disk Controller/Drive Control Signals

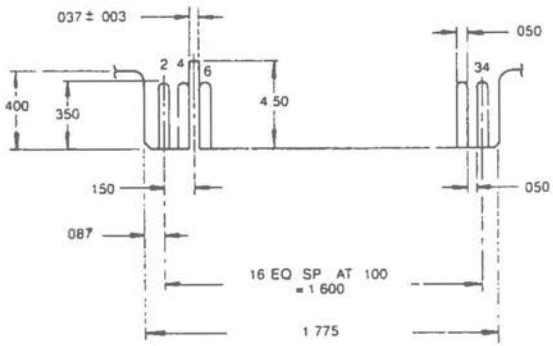


Figure 6.32 Fixed Disk: Edge Connector For Control Signa

Signal	Signal Pin
SELECT/	1
Reserved	3
Reserved	5
Reserved	7
Reserved	9
Reserved	10
GND	11
MFM WRITE DATA	13
MFM WRITE DATA/	14
GND	15
MFM READ DATA	17
MFM READ DATA/	18
GND	19
Other connections are Ground	

Figure 6.33 Fixed Disk Dontroller/Drive Data Signals

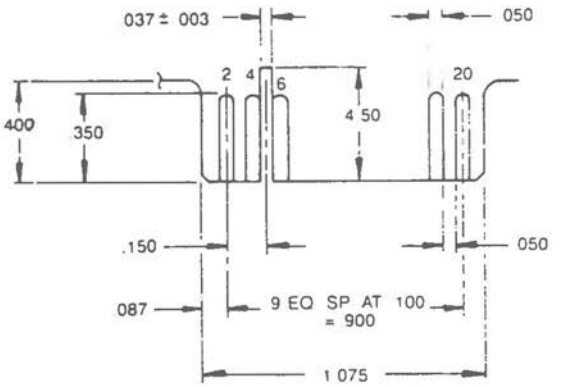


Figure 6.34 Fixed Disk: Edge Connector For Data Signals

Head select

The Head Select lines 0 and 1 provide for the selection of each individual read/write head (0, 1, 2, or 3) in a binary coded sequence. Head Select 0 is the least significant line. Example: Head Select 0 low with Head Select 1 high selects head 1 (signals are active low).

Direction In

This signal defines direction of motion of the R/W head when the Step line is pulsed. Signal high defines the direction as "out" and if a pulse is applied to the Step line, the R/W head moves away from the center of the disk. If this line is signal low, the direction of motion is defined as "in" and the R/W head moves toward the center of the disk. Direction must not change during step time.

Step

This interface line is a control signal which causes the R/W head to move with the direction of motion defined by the Direction In line. Any change in the Direction In line must be made at least 100 ns before the leading edge of the step pulse.

The drive accepts step pulses from the controller in one of two possible modes, track-to-track or buffered. Track-to-track mode is possible when the interval between step pulses is 3 ms or greater. In buffered mode steps are accumulated and then issued at a rate between 3 and 200 microseconds. The drive's own microprocessor monitors the step pulses and decides on the step mode to use as well as the optimum seek algorithm.

Drive Select 1 - 3

Drive Select, when signal active connects the drive interface to the control lines and activates the LED on the front panel of the drive. Switches are provided on the drive which are set in a specified pattern so as to determine which unique select line on the interface will activate that particular drive (see Figure 6.35).

Drive	Switch			
	1	2	3	4
DS1	ON	OFF	OFF	OFF
DS2	OFF	ON	OFF	OFF
DS3	OFF	OFF	ON	OFF
DS4	OFF	OFF	OFF	ON

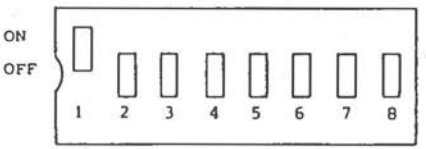


Figure 6.35 Drive Selection Switches

Seek Complete

This line will go from inactive to active when the R/W heads have settled on the final track at the end of a seek. Reading or writing should not be attempted when Seek Complete is inactive. Seek Complete must go inactive for any of the following cases:

1. A recalibration sequence is initiated (by drive logic).
2. 12 microseconds (typical) after the leading edge of a step pulse or series of step pulses.
3. At power on or after a power interruption.

Track 0

This interface signal is active only when the drive's R/W heads are positioned at track 0 (the outermost data track).

Write Fault

This signal indicates that a condition exists at the drive that would cause improper writing on the disk. When this line is active, further writing is inhibited at the drive until the condition is corrected. Once corrected, this line can be reset by de-activating the Write Gate or deselecting the drive.

There are a number of conditions which can cause Write Fault to be activated:

* Write current malfunction:
  Write current in a head without Write Gate active
  or
  Write Gate and Drive Select active with no write current in a head.
* A drive malfunction causing more than, or less than, one head to be selected, or a malfunction causing a head to be selected for read during write or for write during read.
* DC voltages are grossly out of tolerance.
* Spindle speed out of tolerance
* Abnormality at Track 0 sensor during recalibrate at power on.

Index

This interface signal is provided for 1.5 ms by the drive once each revolution (16.67ms nom.) to indicate the beginning of the track. This signal is normally inactive and makes the transition to active to indicate Index.

Ready

This interface signal, when active together with Seek Complete, indicates that the drive is ready to read, write, or seek and that the I/O signals are valid. When this line is inactive all writing, reading, and seeking is inhibited. Ready should be inactive only as a result of and during recovery from a power off condition: Ready is normally asserted about 20 seconds after power on.

MFM Read Data

The data recovered by reading a pre-recorded track is transmitted to the host system via a differential pair of MFM Read Data lines. The transition of the +MFM Read Data line going more positive than the -MFM Read Data line represents a flux reversal on the track of the selected head.

MFM Write Data

This is a differential pair that defines the transitions to be written on the track. The transition of +MFM Write Data line going more positive than the -MFM Write Data will cause a flux reversal on the track, provided Write Gate is active. This signal must be driven to an inactive state (+MFM Write Data more negative than -MFM Write Data) by the host system when in a read mode. Integrity of data writing, especially on inner tracks, is ensured by pre-compensation being active. Data patterns which cause a large amount of bit shift have appropriate data bits shifted early or late with respect to the nominal bit cell position. Bit shift compensation, whether early or late with respect to the nominal bit cell position, is 12ns.

The fixed disk drive provides auto recalibration at dc power up, or when a command to the controller would otherwise cause the head to move beyond the innermost cylinder, or when the Write Fault circuit is activated.

Figures 6.36 to 6.41 illustrate sample timing considerations.

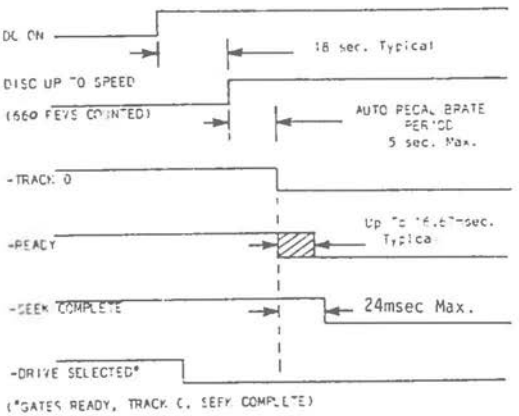


Figure 6.36 Power-On Timing

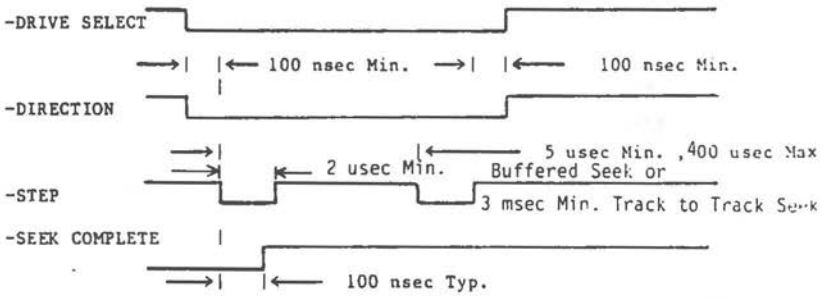


Figure 6.37 Seek Timing

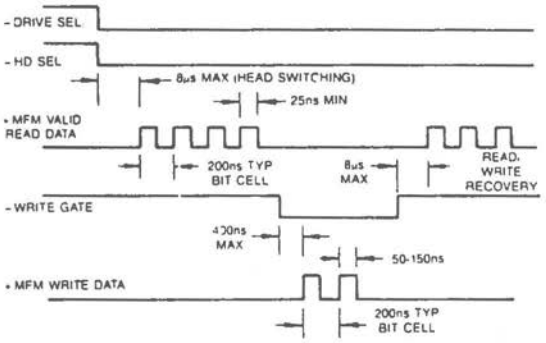


Figure 6.38 Read/Write Data Timing

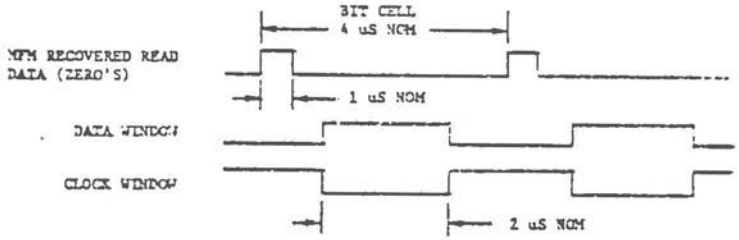


Figure 6.39 Nominal Clock And Data Window Timing

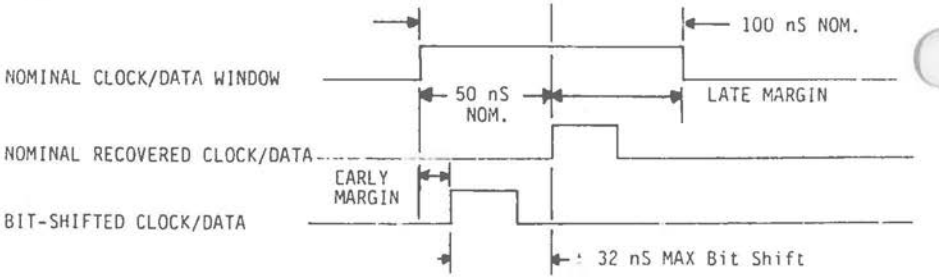


Figure 6.40 Bit Shift

	PREVIOUS	SENDING	NEXT
WRITE DATA LATE	x	0	1	1
WRITE DATA EARLY	x	1	1	0
WRITE CLOCK LATE	1	0	0	0
WRITE CLOCK EARLY	0	0	0	1

x denotes "don't care"

Figure 6.41 Write Pre-Compensation Patterns

?IXED DISK CONTROLLER

The fixed disk controller includes features which considerably offload the microprocessor:

* Retries are automatically performed, even on ECC (Polynomial Error Check) and head positioning errors.

* Basic read/write functions include an implied seek operation, thus saving head positioning control software.

* Head positions are noted internally by the controller. This avoids the need to maintain corresponding tables in your software.

* Data reading and writing is buffered internally, thus facilitating interruption.

* Multiple sector reading and writing.

The interface between controller and the system is via 4 ports with I/O addresses 320H to 323H:

IN 320H Read status/data from the controller.
OUT 320H Write commands/parameters to the controller.

IN 321H Read controller status.
OUT 321H Reset controller.

IN 322H Read drive configuration byte.
OUT 322H Select controller.

OUT 323H DMA and Interrupt Request.

Figure 6.42 sets out the significance of the status bits read via port 321H. Writing to this port effects a hardware of the hard disk drive system. The value actually written is of no significance.
Reading port 322H returns, in bit positions 0-3, a code representing the drive configuration. The two LSBs correspond to the first drive , the two MSBs of the four correspond to the second drive. The reset default settings can be set by jumpers. Writing to port 322H while the BSY bit read via port 321H is not set prepares the controller to receive a command. The controller acknowledges this request by setting this bit. The actual value written via port 322H is immaterial.
Only bits 0 and 1 of the write only port 323H are significant: bit 0 is the DMA Request Enable bit, enabling DMA requests to the microprocessor, when set. Bit 1 (IRQEN) set enables interrupts to the microprocessor.

Bit		Significance
0	REQ	System/controller handshake bit. 1 indicates that the controller is ready for transfer via port 320H
1	I/O	Direction of transfer: 1 = input from controller 0 = output to controller
2	C/D	Specifies whether the controller is expecting a control/status (1) or data (0) transfer
3	BSY	Bit set indicates that the controller is busy executing a command
4	DRQ/DMA	Bit set indicates that the controller is ready for a DMA transfer in the direction indicated by the bit I/O
5	IRQ	Bit set indicates that an interrupt is pending
6,7		Not used

Figure 6.42 Controller Status

Programming the Fixed Disk Controller

Writing to port 322H indicates to the controller that you wish to issue a command. In response, the controller sets the BSY and REQ bits (bits 0 and 3 read via port 321H). The controller now expects to receive a 6-byte command block, of which details are given in Figure 6.43.

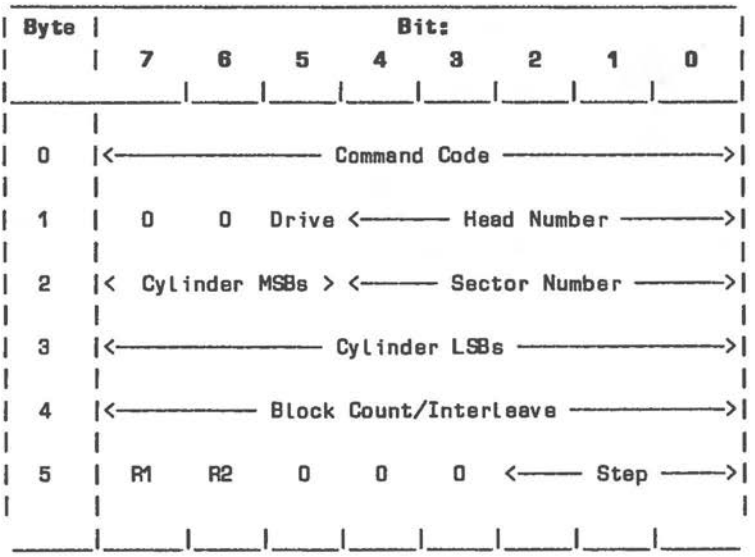


Figure 6.43 Fixed Disk Controller Command Block

R1 and R2 are data retry bits: setting R2 disables the one automatic retry which otherwise occurs upon a data ECC error being encountered. Setting R1 prevents the controller from executing up to 10 retries upon any other type of error (an ID Not Found error allows a second run of 10 retries after an intermediate return to Track 0). A non-recoverable error aborts the command and reports the fact in the status return described later in this section.

The 3-bit value Step determines the stepping rate: a binary value below four or above five specifies 3 ms, value four specifies 200 microseconds, value five specifies 70 microseconds.

If executing the command requires DMA service, the DRQ line (read via port 321H) is asserted. Processor I/O is requested by means of the REQ, I/O, and C/D lines (DMA mode of transfer is specified by means of port 323H). Completion of the command is notified by the BSY bit being reset and the issuing of an interrupt. A command completion byte is then available for reading at port 320H: bit 5 represents

a binary value according to which of the two drives was involved in execution of the command, bit 1 is set if an unrecovered error occurred, other bits are zero.

There are 19 commands which can be executed by the fixed disk controller. This number includes three diagnostic commands which are described in a later section of this Chapter. The 16 main commands are:

Test Drive Ready
Recalibrate
Read Status of Last Operation
Format Drive
Verify Sectors
Format Track
Format Bad Track
Read Sectors
Write Sectors
Seek
Initialize Drive Parameters
Read ECC Burst Error Length
Read Sector Buffer
Write Sector Buffer
Read
Write long

The following descriptions deal with these commands in detail. Where values are required for the 6-byte command block, this is indicated. Parameters of the command block which are not explicitly mentioned should be issued as all bits zero for that command.

TEST DRIVE READY - Command Code 00H
Specific parameters: Drive.
An error condition (Drive Not Ready or Drive Still Seeking) occurs if the drive is active or its status lines show an error.

RECALIBRATE - Command Code 01H
Specific parameters: Drive, R1.
The read/write heads are moved, one cylinder at a time, until Track 0 is encountered. This is somewhat slower than a direct seek Track 0 command, as the

latter moves according to a calculated number of step pulses, and does not check the identity of the cylinders along the way.

READ STATUS OF LAST OPERATION - Command Code 03H
Specific parameters: Drive.
Following execution of this command, 4 status bytes are available for reading via port 320H. These status bytes are described in the next section.

FORMAT DRIVE STARTING AT SPECIFIED TRACK -
Command Code 04H
Specific parameters: Drive, Head, Cylinder, Sector (must be valid, but is ignored), Interleave, R1, Step.
After drive recalibration formatting starts at the first sector of the specified track. Address marks, header fields, and data fields are written for all sectors. The data pattern defaults to that currently in the sector buffer. Interleave tells the controller where the next logical sector is in relation to the current one: the value 1 specifies that logical sectors are contiguous, 2 specifies that the next logical sector is two sectors ahead of the current physical sector, and so on. Therefore, Interleave may be a value between 1 and one less than the number of sectors per track. Out of range values result in an Interleave value of 1.
In the event of an error, formatting can be continued at the beginning of the next track.

VERIFY SECTORS - Command Code 05H
Specific parameters: Drive, Head, Cylinder, Sector, Block Count, R1, R2, Step.
This command reads from 1 to 256 sectors, as specified by Block Count and beginning at the Sector specified. In the event of an error, the number of the sector concerned is returned as part of the status return (Command 03H).

FORMAT TRACK - Command Code 06H
Specific parameters: Drive, Head, Cylinder, Sector (must be valid, but is ignored), Interleave, R1, Step.

This command is similar to Command 04H. The difference is that formatting is confined to the specified track. FORMAT TRACK can be used to clear the Bad Track flag.

FORMAT BAD TRACK - Command Code 07H
Specific parameters: as Command 06H.
The difference between this command and Command 06H is that the Bad Track flag is set in the ID field.

READ SECTORS - Command Code 08H
Specific parameters: Drive, Head, Cylinder, Sector, Block Count, R1, R2, Step.
This command reads from 1 to 256 sectors, as specified by Block Count and beginning at the Sector specified. In the event of an error, the number of the sector concerned is returned as part of the status return (Command 03H).

WRITE SECTORS - Command Code 0AH
Specific parameters: Drive, Head, Cylinder, Sector, Block Count, R1, Step.
This command functions as Command 08H, except that sectors are written and not read.

SEEK - Command Code 0BH
Specific parameters: Drive, Head, Cylinder, Sector (must be valid, but is ignored), R1, Step.
This command seeks a specified track on a drive which has already been formatted.
As the drive itself supports buffered step seeks, step pulses can be issued at high speed (70 or 200 microseconds), thus freeing the controller during the seek process. Completion status is returned even before the drive has concluded the seek operation. The status return can only confirm that the SEEK command was properly issued. Once the status has been read, a new command can be issued to the controller. As long as the drive is still performing the seek (BSY line is asserted), the new command (except Test Drive Ready) must wait to be executed. The controller does not recognize a timeout condition.

INITIALIZE DRIVE PARAMETERS - Command Code 0CH
Specific parameters: Drive.
Following the command block, the controller expects an 8-byte initialization block:

2 bytes	Max. no. of cylinders	- max. value = 1024
1 byte	Number of heads	- max. value = 8
2 bytes	Outermost cylinder with reduced write current	- max. value = 1024
2 bytes	Outermost cylinder with write pre-compensation	- max. value = 1024
1 byte	Max. ECC data burst length	- max. value = 11

If this command is not issued, the following default values apply: Cylinders = 306, Heads = 4, Reduced write current and write pre-compensation from cylinder 153, 11-bit burst error length.

READ ECC BURST ERROR LENGTH - Command Code 0DH
Specific parameters: Drive.
This command is valid only after a correctable data error. It transfers one byte indicating the length of the error corrected. The error length is the number of bits between the first and the last bit in the error, inclusive.

READ SECTOR BUFFER - Command Code 0EH
Specific parameters: none.
Reads 512 bytes of data from the controller's sector buffer.

WRITE SECTOR BUFFER - Command Code 0FH
Specific parameters: none.
Writes 512 bytes of data to the controller's sector buffer.

READ LONG - Command Code 0E5H
Specific parameters: Drive, Head, Cylinder, Sector, Block Count, R1, Step.
In addition to reading the data bytes from a sector, the four ECC bytes are also read into the sector buffer.

WRITE LONG - Command Code 0E6H
Specific parameters: Drive, Head, Cylinder, Sector, Block Count, R1, Step.
Following the data bytes, four ECC bytes are written to the specified sector. This command can be used to test the controller's ECC circuitry.

Fixed Disk Controller Status

In addition to the status read via port 321H, detailed status information is returned in 4 bytes via Command 03H - Read Status of Last Disk Operation. After this command has been issued, the four status bytes can be read via port 320H.

Byte	Bit: 7	6	5	4	3	2	1	0
0	< DA >	0	<---------- Error Code ---------->					
1	0	0	Drive <------- Head Number ------->					
2	< Cylinder MSBs >		<------ Sector Number ------>					
3	<---------- Cylinder LSBs ---------->							

NOTE: DA bit set signifies that the most recent command required a disk location.

Figure 6.44 Fixed Disk Controller Status

6-Bit Value (hex)	Significance
00	No error detected
03	Write Fault signal from drive
04	Drive Not Ready.
06	Track 0 Not Found. This error is returned if, on a recalibrate command, Track 0 could not be found within 200 steps
08	Drive Still Seeking. Returned in response to a Test Drive Ready command if a buffered step seek is still in progress
11	Uncorrectable Data Error in Data Field
12	Sector Address Mark Not Found. The controller did not detect a data address mark within its timing window
15	Seek Error. Disk location could not be found due to an unmatched ID field or bad CRC
18	Correctable Data Error. A media error encountered while reading was corrected by ECC. This error notification is, therefore, essentially for information purposes
19	Bad Track. The most recent data transfer command encountered a track already flagged as bad by an earlier Format Bad Track command. (No retries are performed)
20	Invalid Command was issued to the controller
21	Illegal Disk Location issued to the controller
30	Sector Buffer Error. A data error was detected during Sector Buffer Diagnostics
31	Controller ROM Checksum Error returned by diagostic command
32	ECC Polynomial Error detected during diagnostic command

Figure 6.45 Fixed Disk Controller Error Codes

The significance of the fixed disk location returned in the status bytes differs in accordance with the command to which it refers:

* Error on multiple sector data transfer (read or write) command
  - location of the defective sector.

* Error on format or verify sector command
  - location of the defective track.

* No error on format or verify sector command
  - location of the sector following the last sector/track checked or formatted.

Figure 6.45 gives the significance of Error Code values returned in the 6 least significant bits of the first status byte.

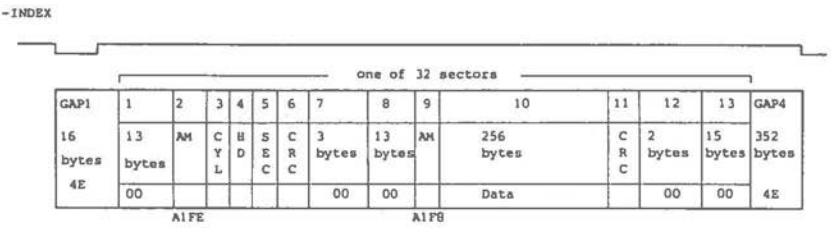
Fixed Disk Controller Diagnostics
As the result of an explicit command from the microprocessor, one of three internal diagnostic commands can be performed by the fixed disk controller:

SECTOR BUFFER DIAGNOSTIC - Command Code 0E0H
Specific parameters: none.
The sector buffer is subjected to a 9-pass memory integrity test, using a 9-byte pattern (no bits set, bit 0 set, bit 1 set ... bit 7 set). After each pattern write and verification reading, the whole pattern is shifted by one byte and the test is repeated.

DRIVE DIAGNOSTIC - Command Code 0E3H
Specific parameters: Drive, R1, Step.
The controller sends Recalibrate and Seek commands to the specified drive, and reads sector 0 of each track to check that ID and data fields are correct. The disk must already be formatted (no writing to disk takes place).

CONTROLLER DIAGNOSTIC - Command Code 0E4H
Specific parameters: none.
This command performs a controller RAM integrity check (similar to Command 0E0H), a ROM checksum test, an ECC integrated circuit diagnostic check, and a number of other circuit verifications.

Fixed Disk Format
The sector format is illustrated in Figure 6.46.



1.	13 bytes of 00	VFO SYNC area
2.	A1,FE	ID address mark (with missing clock)
3.	1 byte	Cylinder address
4.	1 byte	Head address, except bit 5 = cylinder MSB bit 7 set = bad sector
5.	1 byte	Sector address
6.	2 bytes	CRC
7.	3 bytes of 00	Write switching gap
8.	13 bytes of 00	VFO SYNC area
9.	A1,F8	Data address mark (with missing clock)
10.	256 bytes	Data
11.	2 bytes	CRC
12.	3 bytes of 00	Write switching gap
13.	15 bytes of 4E	Sector separator

Figure 6.46 Fixed Disk Format

EXTERNAL FIXED DISK CONNECTION

Signal connection to the external fixed disk unit is by means of a 37-pole D-connector attached to the end of the cable coming from the unit. Figure 6.47 shows pin significance of this connector.

Pin	Signal	Pin	Signal
1	DIRECTION IN	19	MFM WRITE DATA +
2	DRIVE SELECT2	20	GND
3	DRIVE SELECT1	21	GND
4	STEP	22	GND
5	READY	23	GND
6	INDEX	24	GND
7	HEAD SELECT1	25	GND
8	HEAD SELECT0	26	GND
9	WRITE FAULT	27	GND
10	TRACK0	28	GND
11	SEEK COMPLETE	29	GND
12	WRITE GATE	30	GND
13	HEAD SELECT2	31	GND
14	Reserved	32	GND
15	GND	33	GND
16	MFM READ DATA -	34	GND
17	MFM READ DATA +	35	GND
18	MFM WRITE DATA -	36	GND
		37	GND

Figure 6.47 Connector To External Fixed Disk Unit

1.2 MB FLEXIBLE DISK DRIVE

The 1.2 MB flexible disk drive, available as an option, provides double-sided storage over 80 tracks per side. Each track is formatted to 15 sectors of 512 bytes.
The drive is used in its high density, high speed mode of operation and requires a separate controller board.

Tracks per inch (TPI)	96
Tracks per disk side	80
Unformatted capacity	1.604 MB
Motor rotation speed	360 r.p.m
Motor start time	< 500 ms
Head movement (track to track)	3 ms
Average seek time	94 ms
Latency time (at 300 r.p.m)	83.3 ms
Data transfer rate	300 K bits/s
Data recording format	MFM
Power consumption (typical with disk rotating in drive)	12 Vdc 0.22 A 5 Vdc 0.35 A

Figure 6.48 1.2 MB Flexible Disk Drive Technical Data

Jumpers on the 1.2 MB flexible disk drive are illustrated in Figure 6.49.

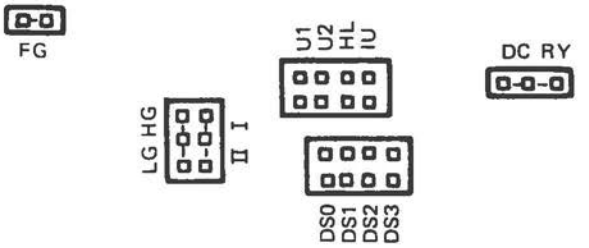


Figure 6.49 1.2 MB Flexible Disk Drive Jumpers

DS1 - DS4
Address selection of the drive. For the NCR PERSONAL COMPUTER DS1 is to be selected, even if the drive being installed is the second in the system. (The cable band from the first to the second connector has two DS lines transposed.)

IU
When closed, enables the IN USE/ signal at pin 4 of the control/data connector. This signal indicates that all the daisy-chained flexible disk drives are in use under system control.

U1, U2
In conjunction with IU, these straps select one of five turn-on conditions for the drive indicator LED:

Strap installed			LED turn-on condition(s)
IU	U1	U2	
no	no	no	Drive Select
yes	no	no	Drive Select + In Use
yes	yes	no	In Use
no	yes	yes	Drive Select or Ready
yes	yes	yes	In Use and (Drive Select or Ready)

HL
Not used (open)

FG
Connects frame to 0 V.

LG, HG
One of these straps must be installed:

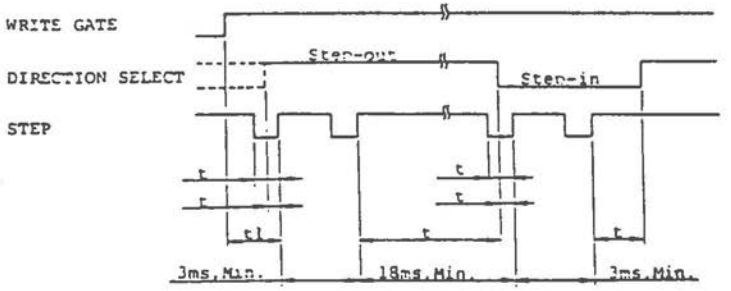
Installed	Density Signal "low" selects
LG	High density
HG	Low density

I, II
One of these straps must be installed:

Installed	Density uses	Rot. Speed (r.p.m)
I	High	360
	Low	300
II	High	360
	Low	360

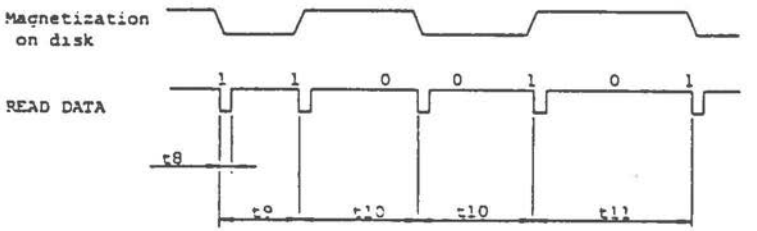
RY, DC
One of these straps must be installed: pin 34 of the control/data connector provides Ready (RY installed) or Disk Change (DC installed) signal.

Figures 6.50 - 6.53 illustrate signal timing.



t= 0.8us,Min. t1= 590us, Min.(360rpm)
1ms,Min.(300rpm)

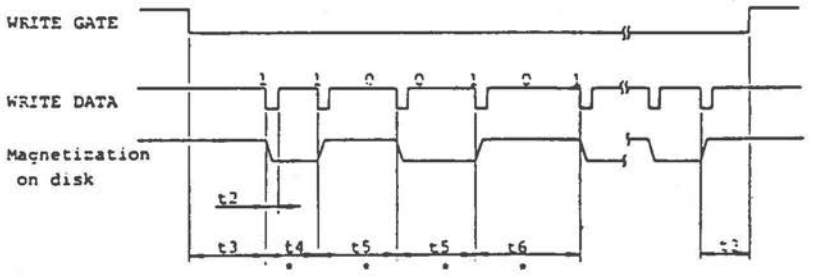
Figure 6.50 Step Timing



Note: The displacement of any bit position does not exceed "t12" from its nominal position. (When PLO separator is used with zero write pre-compensation.)

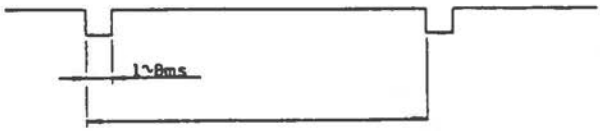
Density & Disk speed	t8	t9	t10	t11	t12
High, 360rpm	0.5±0.25us	2us,Nom.	3us,Nom.	4us,Nom.	±350ns
Normal, 360rpm	0.5±0.25us	3.3us,Nom.	5us,Nom.	6.7us,Nom.	±580ns
Normal, 300rpm	0.5±0.25us	4us,Nom.	6us,Nom.	8us,Nom.	±700ns

Figure 6.51 Read Data Timing (MFM)



Density & Disk Speed	t2	t3	t4	t5	t6
High,360rpm	0.15~1.1us	4us,Max.	2us,Nom.	3us,Nom.	4us,Nom
Normal,360rpm	0.15~2.1us	6.7us,Max.	3.3us,Nom.	5us,Nom.	6.7us,Nom.
Normal,300rpm	0.15~2.1us	8us,Max.	4us,Nom.	6us,Nom.	8us,Nom.

Figure 6.52 Write Data Timing (MFM)



t7= 166.7±2.5ms (360rpm)
200±3ms (300rpm)

Figure 6.53 Index Timing

Installation of the 1.2 MB flexible disk controller requires removal of the jumper JP6 from the main processor board (not present on all boards). This disables the flexible disk controller IC on the main processor board. The controller IC on the newly installed board is then responsible for both the 1.2 MB drive and the original (48 TPI) drive.

Jumpers are provided on the 1.2 MB flexible disk controller board for selection of an alternative I/O area for the controller. These address areas follow (standard addresses are given in parentheses):

OUT 372H (3F2H) Drive select command

IN 374H (3F4H) Main Status Register

OUT 375H (3F5H) Command Register

IN 375H (3F5H) Result input

Otherwise I/O addressing of the controller IC is as for the equivalent IC on the main processor board.

ROM BIOS ensures correct recognition of the type of flexible disk being used in the 1.2 MB drive. ROM BIOS flexible disk drive control (see Chapter 3) applies to the 1.2 MB drive in the same way as to the standard flexible disk drive, with the obvious difference that track and sector parameters for the higher flexible disk capacity are also recognized.

NOTE: ROM BIOS support of the 1.2 MB flexible disk drive is included in ROM BIOS Version 3.5 and later versions.

The Screen Display

The NCR PERSONAL COMPUTER can drive both a monochrome and a color display by means of adapter cards. Two types of adapter card are available: the character display card enables the monochrome display of all 256 characters in the character generator and includes 4 KB of memory; the graphic display card additionally supports "all points addressable" graphics and contains a 32 KB memory.

The 32 KB memory of the graphic display card can be upgraded to 64 KB, thus enabling you to make full use of the high resolution (640 x 400 pixels) capability of the NCR PERSONAL COMPUTER color CRT.

Both the monochrome and color CRT are driven by a 6845 controller. A dual port configuration is used so that CPU access and CRT controller can access video RAM independently of one another.

AVAILABLE ADAPTERS

Only one version of the NCR PERSONAL COMPUTER contains the character display card as standard: the monochrome, single diskette drive, 256 KB memory system. All other versions contain a graphic display card. The video memory address areas of the two cards do not clash (all video RAM lies within the CPU address area). It is possible to connect a monochrome monitor to a graphic display card and it is possible to add an external monitor to the NCR PERSONAL COMPUTER by means of an additional adapter card. A number of industry standard monitors can be connected: if the internal monitor is driven by a graphic display card, the external monitor must be connected by means of a character display card, and vice versa. Different cards are used for a type of display (character/graphic), according to whether the

monitor is internal or external. This is because the frequency characteristics of internal and external monitors are not identical.

The display faciliites of your NCR PERSOAL COMPUTER can be changed/upgraded using the following Kits:

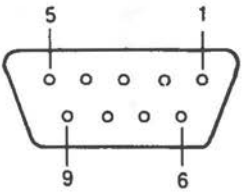
K510 - Character display card with 4 KB video memory for NCR internal monitor.
K511 - Graphic display with 32 KB video memory for NCR internal monitor.
K512 - Upgrade 32 KB graphic display to 64 KB.
K140 - Graphic display card with 16 KB video memory for an external monitor.
K141 - Character display card with 4 KB video memory for an exteral monitor.

CAUTION: Before connecting an external monitor, you must ensure that it is TTL compatible. A number of monitors available claim to conform to this norm, but in fact drain excessive current from the adapter card.

CONNECTION AND CONFIGURATION

A 9-pin D-shell connector interfaces to the internal CRT assembly or external monitor. Figure 7.1 shows the pin connections for the monochrome and color adapters for both internal and external monitors. Figure 7.2 illustrates the signals for the light pen connector.

	Pin	Internal	External
C	1	VIDEO	GROUND
H	2	HSYNC	GROUND
A	3		
R	4		
A	5		
C	6		INTENSITY
T	7	VSYNC/	VIDEO
E	8		HSYNC
R	9	GROUND	VSYNC/



	Pin	Internal	External
	1	VIDEO-LO	GROUND
G	2	HSYNCP	GROUND
R	3	RED/	RED
A	4	GREEN/	GREEN
P	5	BLUE/	BLUE
H	6	INTENSITY/	INTENSITY
I	7	VSYNC/	
C	8	HSYNC/	HSYNC
	9	GROUND	VSYNC

Figure 7.1 D-Connector Pin Assignments

Pin	Signal
1	LPENINPUT/
3	LPENSW/
4	LGRD
5	+5V
6	+12V

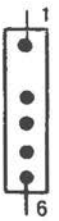
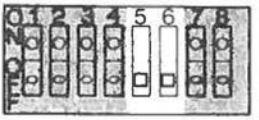
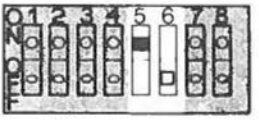


Figure 7.2 Light Pen Connector

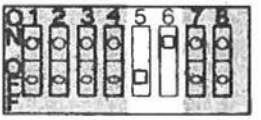
Main Board display switches 5 and 6 can be used to set an initial display configuration (see Figure 7.3). These determine at system initialization which display card is to be driven. These settings can be overridden by software control. This is explained later in this Chapter. The adapter cards for the NCR PERSONAL COMPUTER use separate TTL load lines for horizontal sync, vertical sync, and video signal; there is no provision for a composite TV-signal. You should set these switches to the 80 x 25 setting, if you are using the graphic display card.



Monochrome



Color/graphics 80 x 25



Color/graphics 40 x 25

Figure 7.3 Display Switch Settings

CHARACTER SET

The NCR PERSONAL COMPUTER draws upon two character sets. In the character mode of display (the only possible mode in the case of the character display card), a set of 256 characters can be displayed on the screen. This character set, for the most part consisting of 6 (horizontal) x 9 pixel patterns in 8 x 16 matrices, is produced by a character generator which is not accessible to software. If your application requires the pixel patterns of this character set (for example, for character magnification), you can load a corresponding graphic

table into user memory by means of the GRAFTABL utility under your NCR-DOS operating system. This table is required for the 640 by 400 pixel graphic resolution of GW-BASIC.

The lower part of this graphic table (for character codes 0 to 7FH) is held in read only memory, starting at the machine address 0D800H offset to the beginning of the ROM at paragraph 0F000H in the normal machine memory map (the ROM is not switched), and extending over 2048 bytes (128 characters each of 16 bytes). This lower part is always available for program reading.

The higher part (character codes 80H to 0FFH) is loaded with the NCR-DOS executable file GRAFTABL into user memory. Assuming that you specify high resolution in response to the GRAFTABL menu, this higher part of the table is retained in user memory even after the return to the parent process (using the NCR-DOS function call "Terminate but Keep Process"). The exact location of the first 16 x 8 pattern byte (that is, for the character with code 80H) is the byte with a positive offset of 400H to the Code Segment + Instruction Pointer value held in the 8088 interrupt vector entry for interrupt type 1FH. This entry is set by the GRAFTABL program. Example: if the four bytes starting at machine address 7CH (= vector entry for interrupt type 1FH) are 03H-01H (IP) 91H-06H (CS) in that order (the CS value can vary according to the program segment set up by NCR-DOS), the beginning of this higher part graphics table is located at the hexadecimal paragraph:offset address 0691:0503, and the top 8-pixel line for the graphics character with code 81H is stored in the byte 0691:0513.

The pixel patterns for this 16 x 8 character set are illustrated in Figure 7.4.

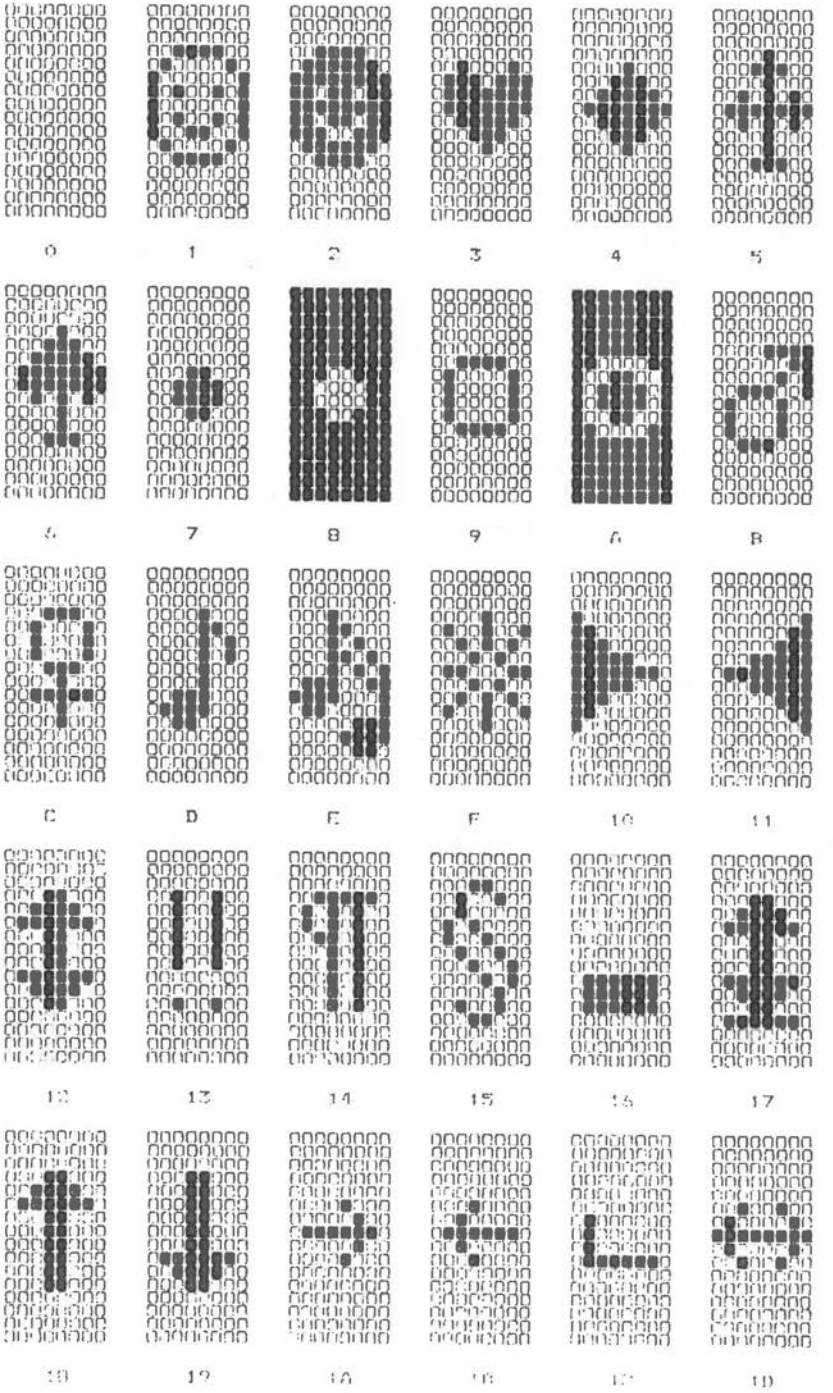


Figure 7.4 The Main Character Set (1 of 9)

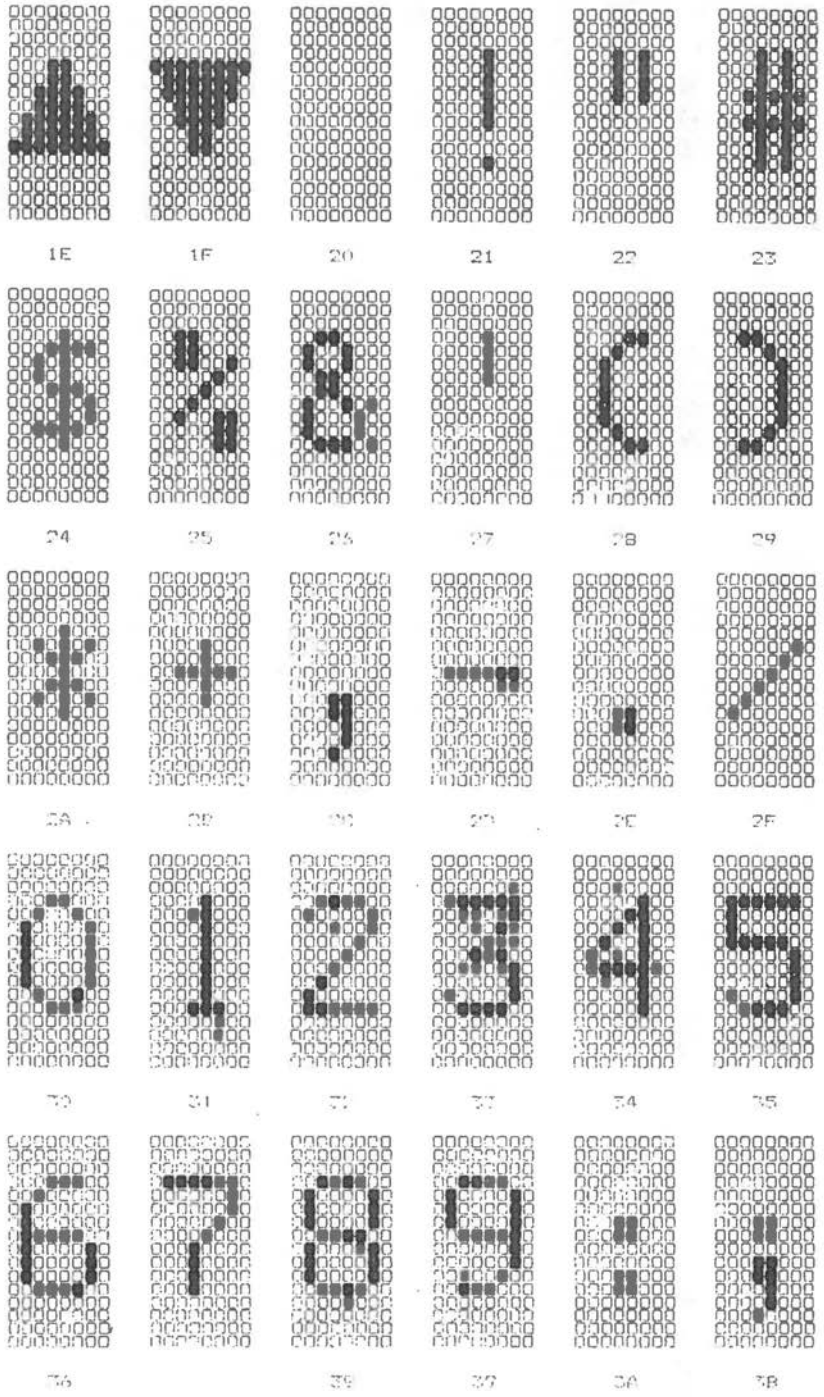


Figure 7.4 The Main Character Set (2 of 9)

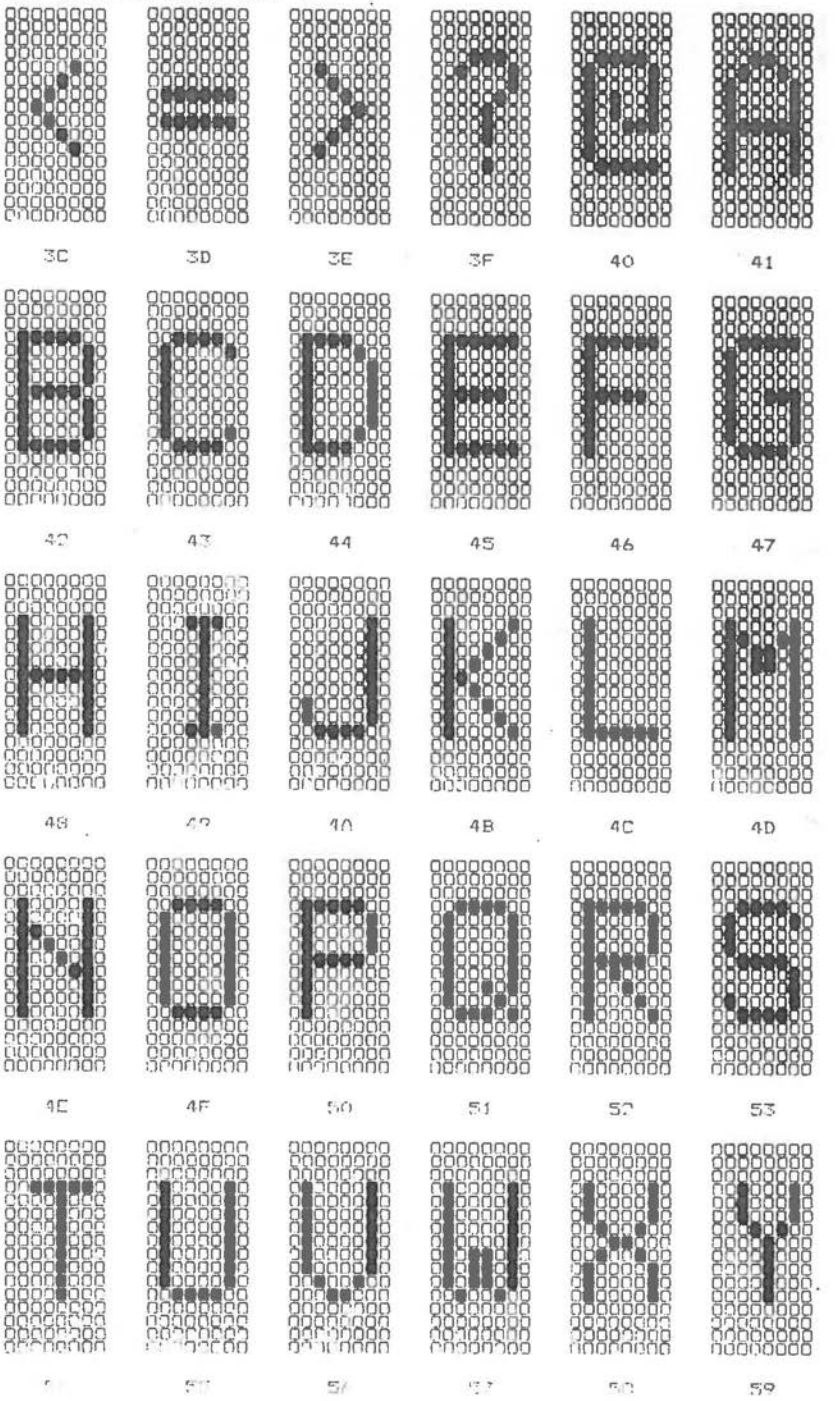


Figure 7.4 The Main Character Set (3 of 9)

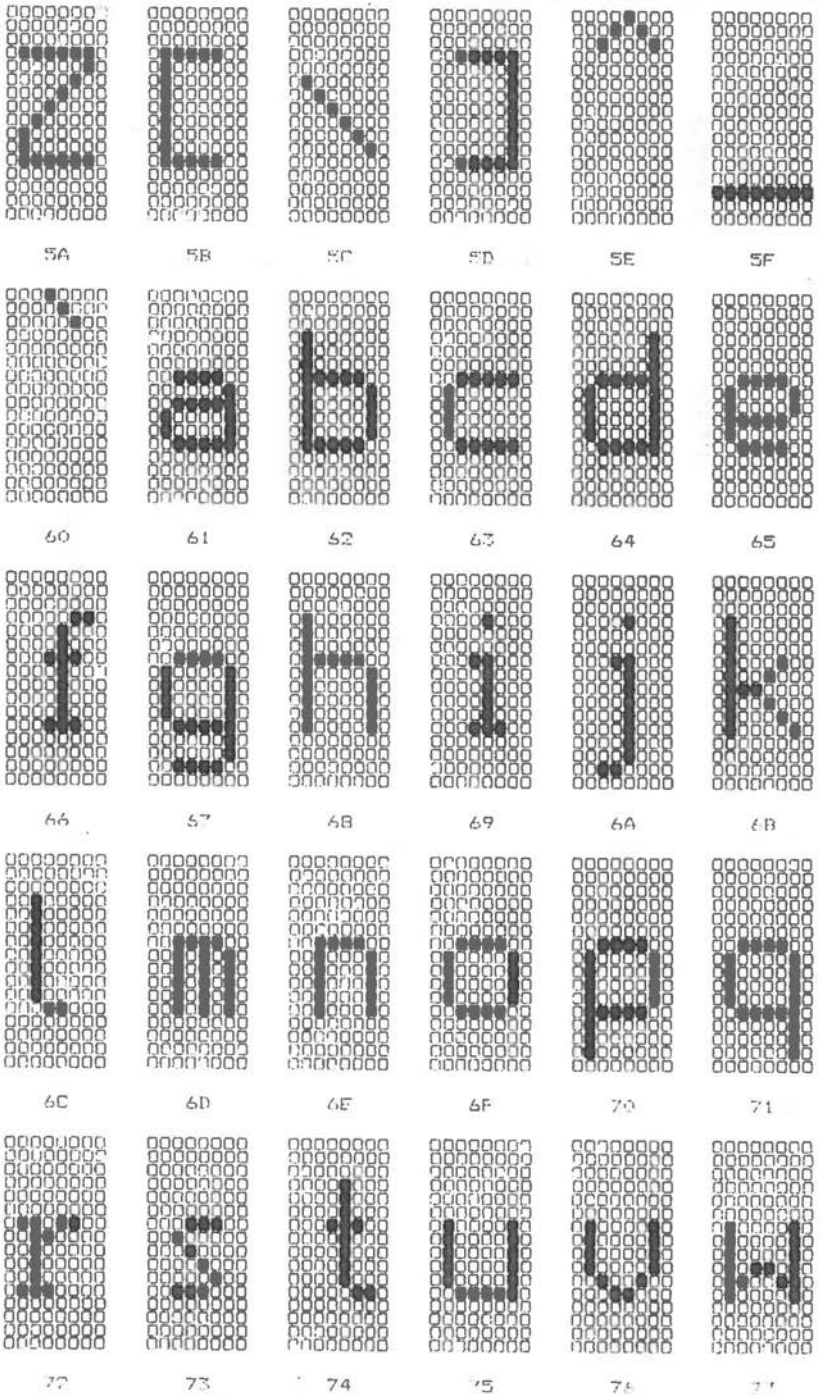


Figure 7.4 The Main Character Set (4 of 9)

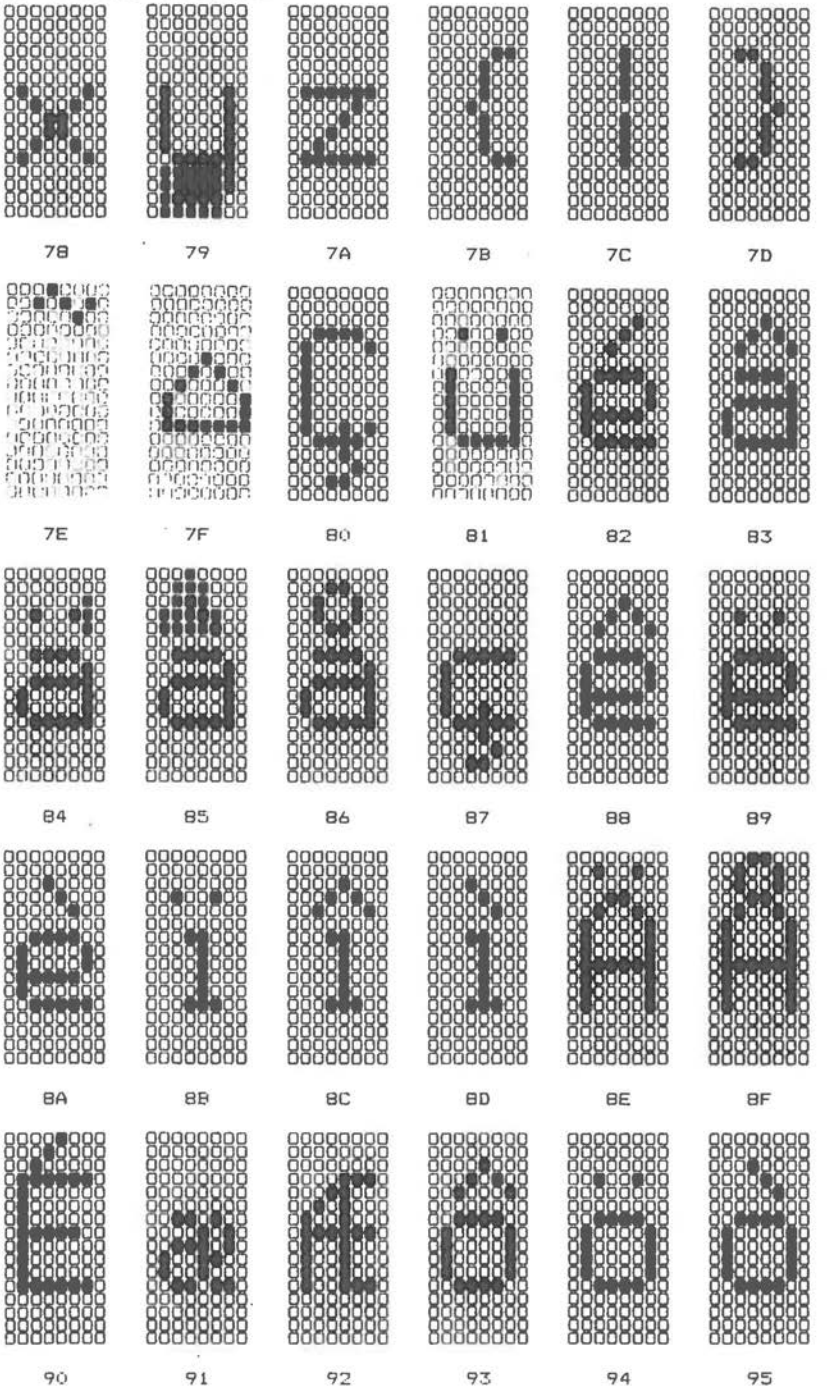


Figure 7.4 The Main Character Set (5 of 9)

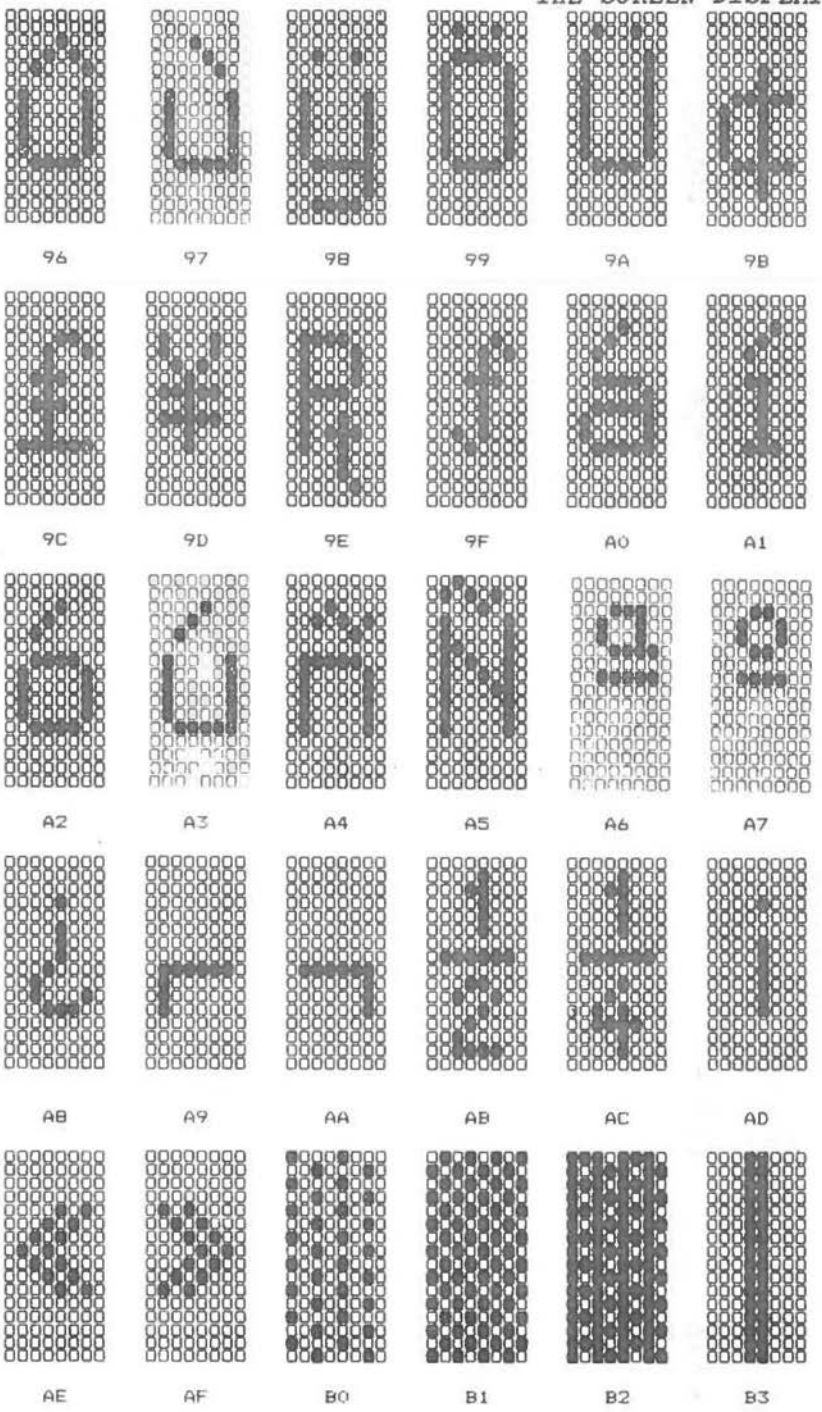


Figure 7.4 The Main Character Set (6 of 9)

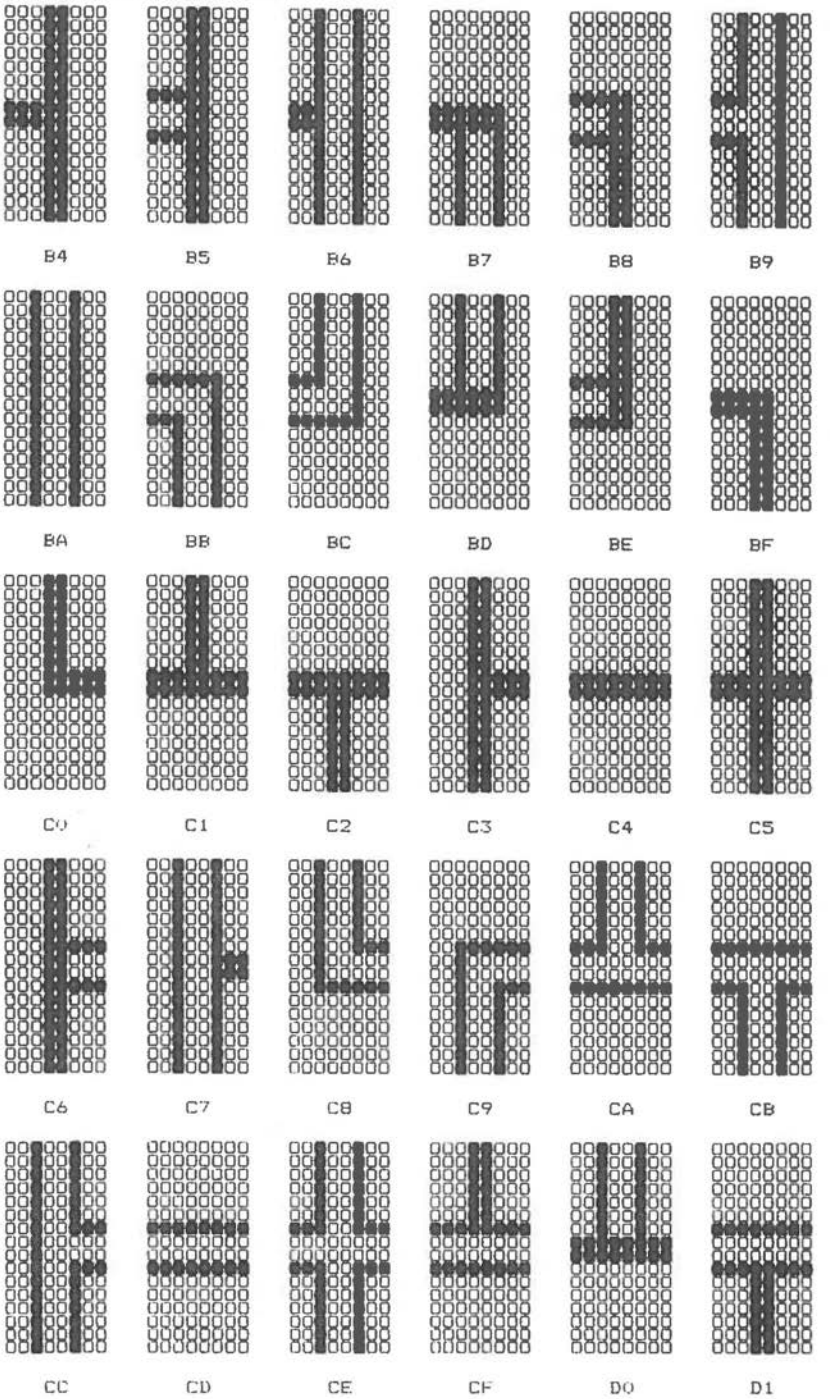


Figure 7.4 The Main Character Set (7 of 8)

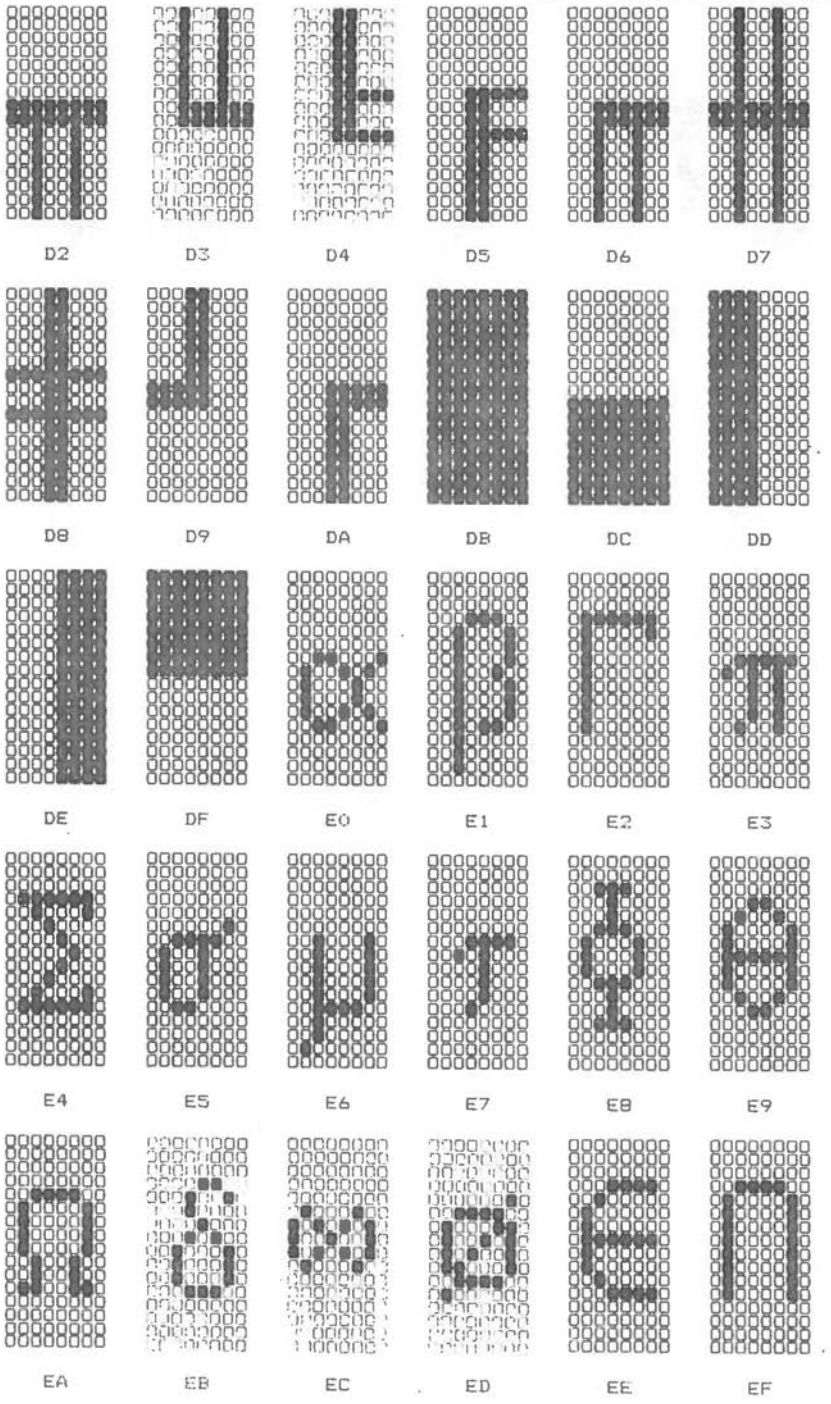


Figure 7.4 The Main Character Set (8 of 9)

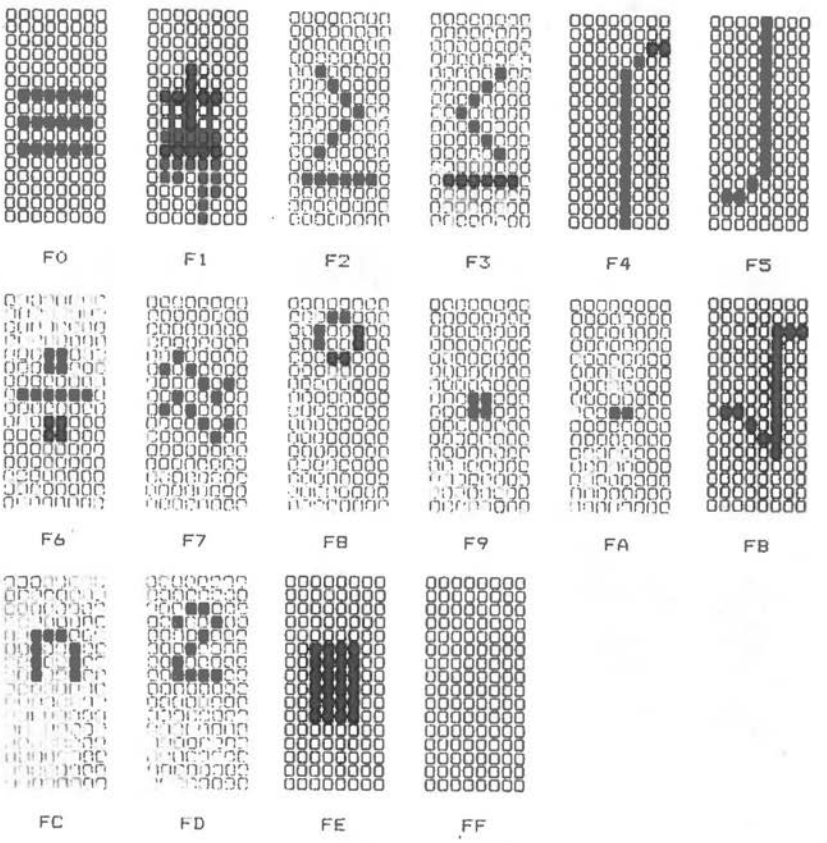


Figure 7.4 The Main Character Set (9 of 9)

An additional character set is available for reading under software control, as used by the low resolution graphics mode of display in GW-BASIC. Essentially this consists of the same characters as available in the character set already described. The main difference is that pixel patterns for the characters in this set are in a 7 x 7 format (except for downward tails) contained in an 8 x 8 area.

The pixel patterns for the lower part of this character set (that is, characters with codes 0 - 7FH) can be read starting at address 0FA6EH, offset to the beginning of the ROM at paragraph address 0F000H: the pixel pattern for the character with code 0 is stored in the first 8 bytes, the pattern for code 1 in the next 8 bytes, and so on. The last character pattern (for code 07FH) is stored between offset 0FE66H and 0FE6DH.

The procedure for accessing the higher part of this Alternative Character Set is the sames as for the 16 x 8 character set, except that

* You may specify either low/medium or high resolution in response to the GRAFTABL menu.

* The pixel pattern for any character occupies only 8 bytes of memory.

* The beginning of the character table is at a positive offset of one byte to the location pointed to by the interrupt vector.

This character set is not implemented in the hardware character generator.

Figure 7.3 illustrates the pixel patterns of the 8 x 8 character set.

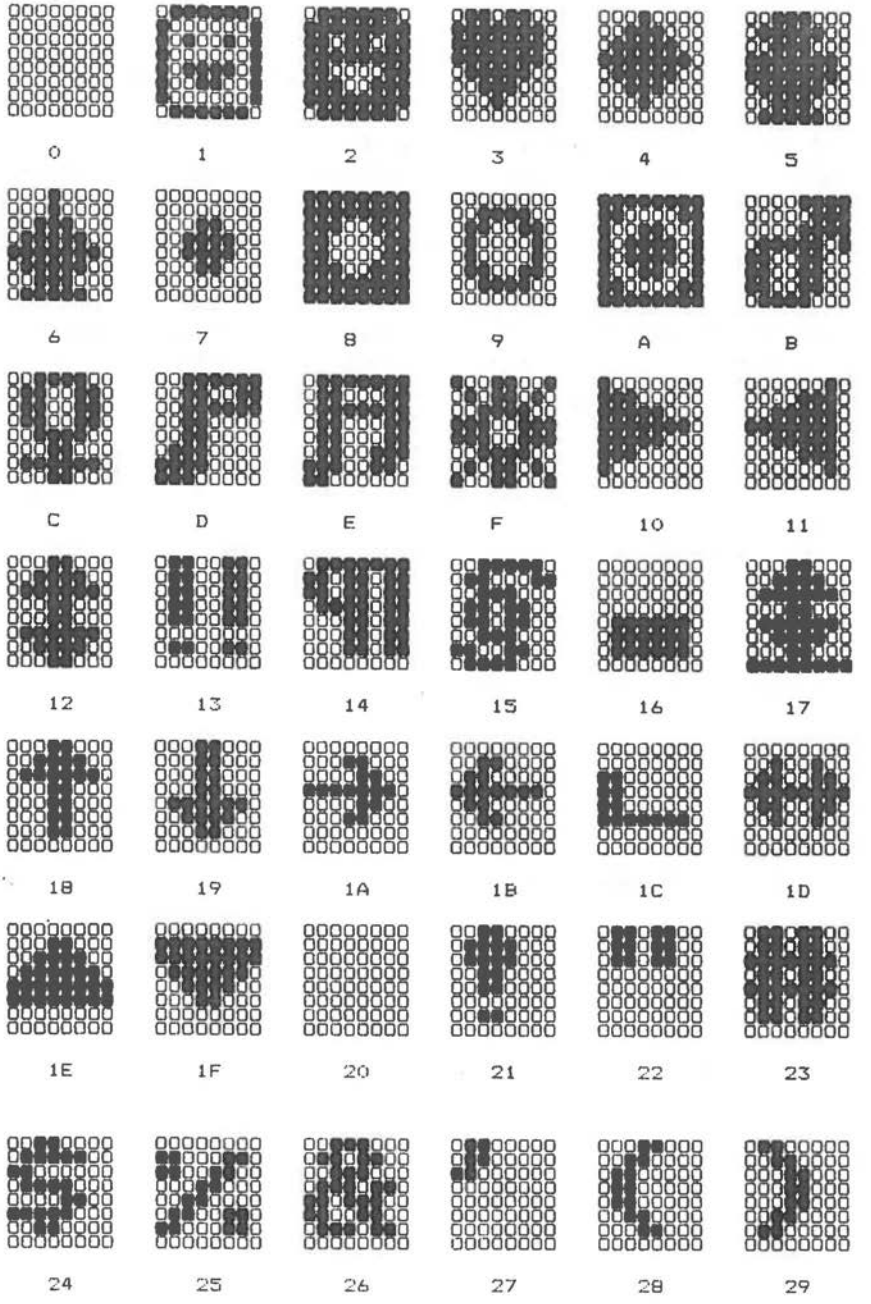


Figure 7.5 The Alternative Character Set (1 of 7)

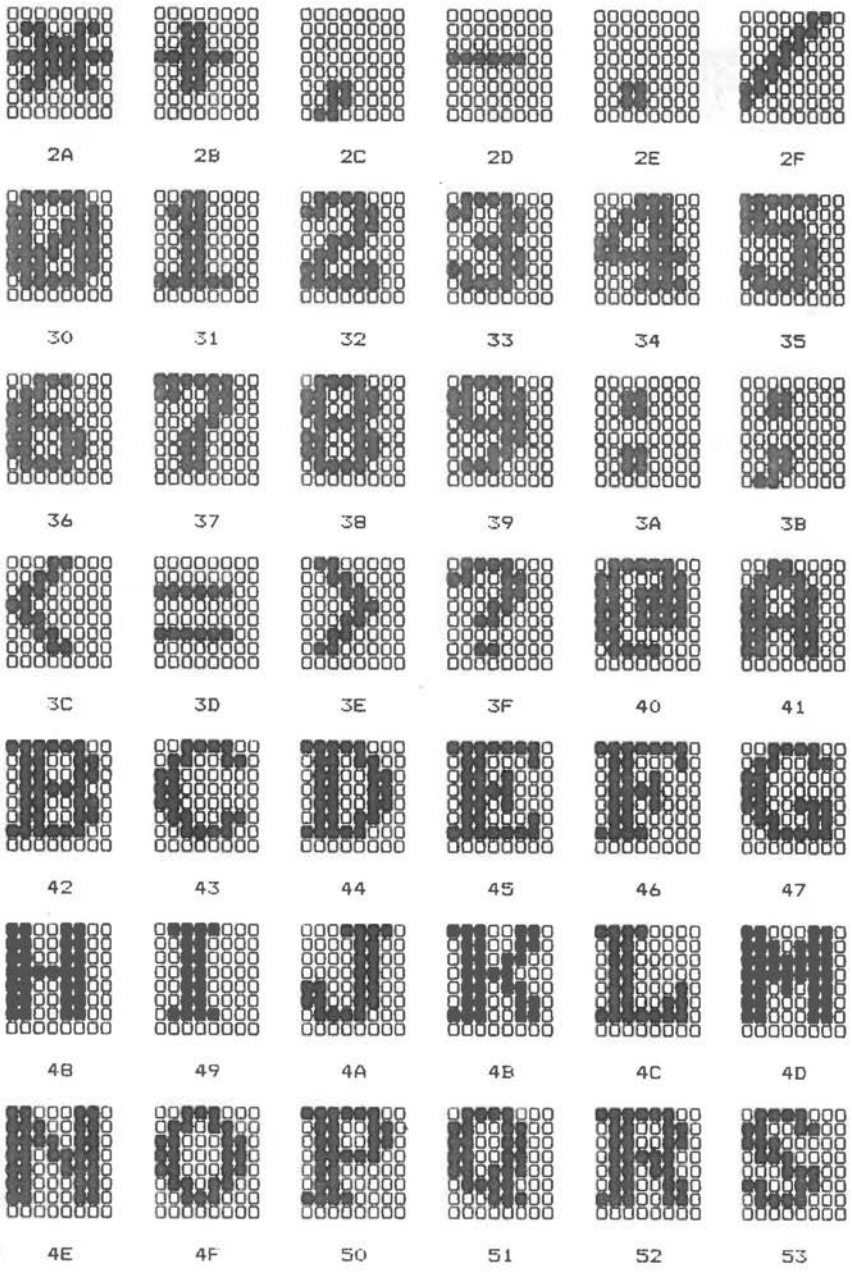


Figure 7.5 The Alternative Character Set (2 of 7)

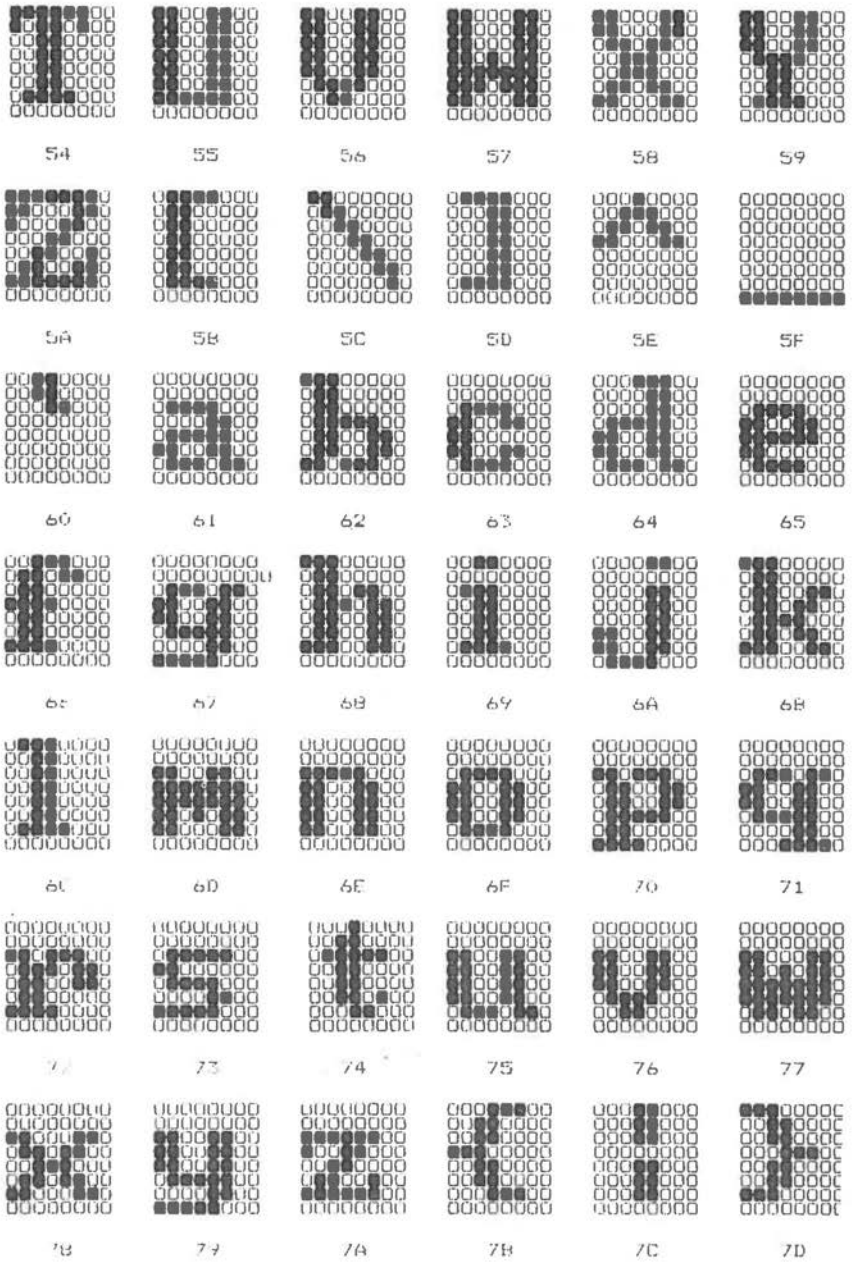


Figure 7.5 The Alternative Character Set (3 of 7)

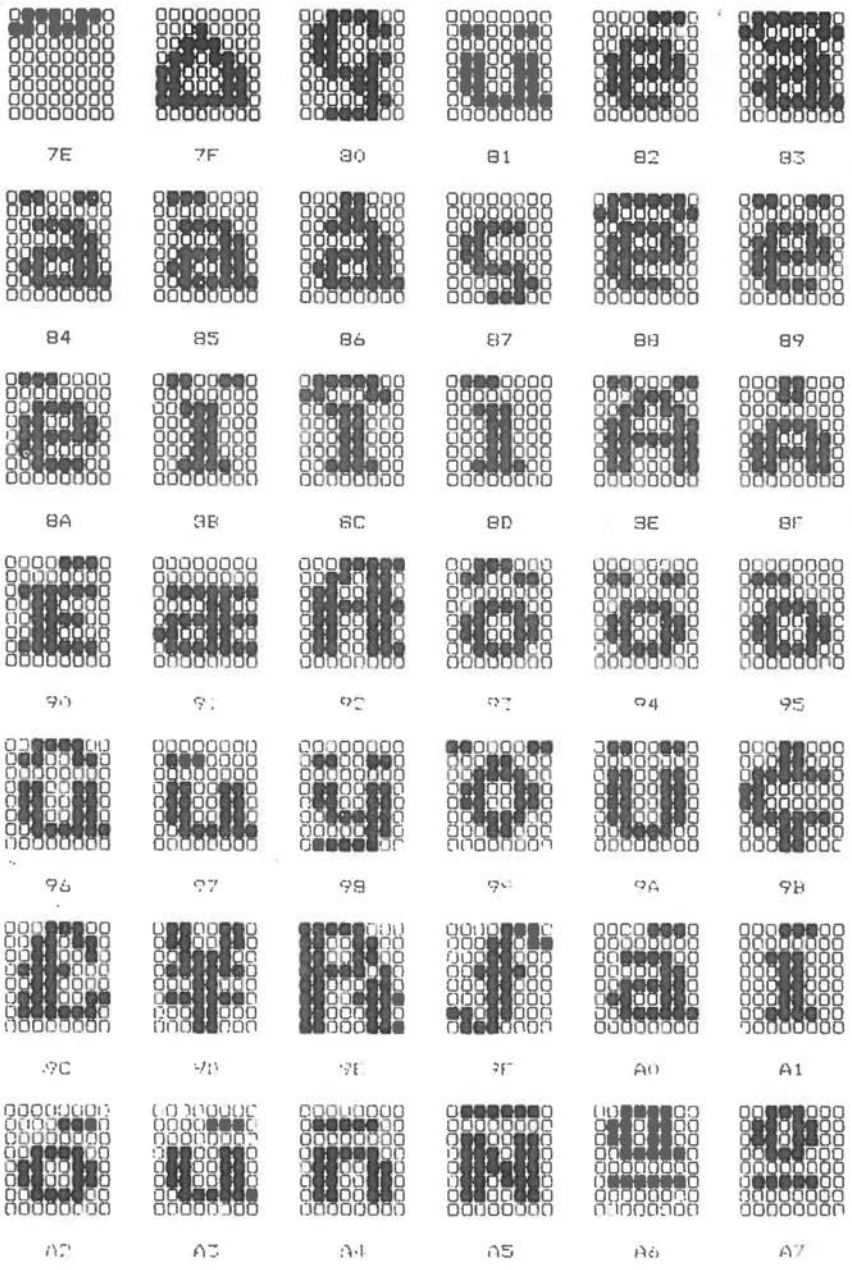


Figure 7.5 The Alternative Character Set (4 of 7)

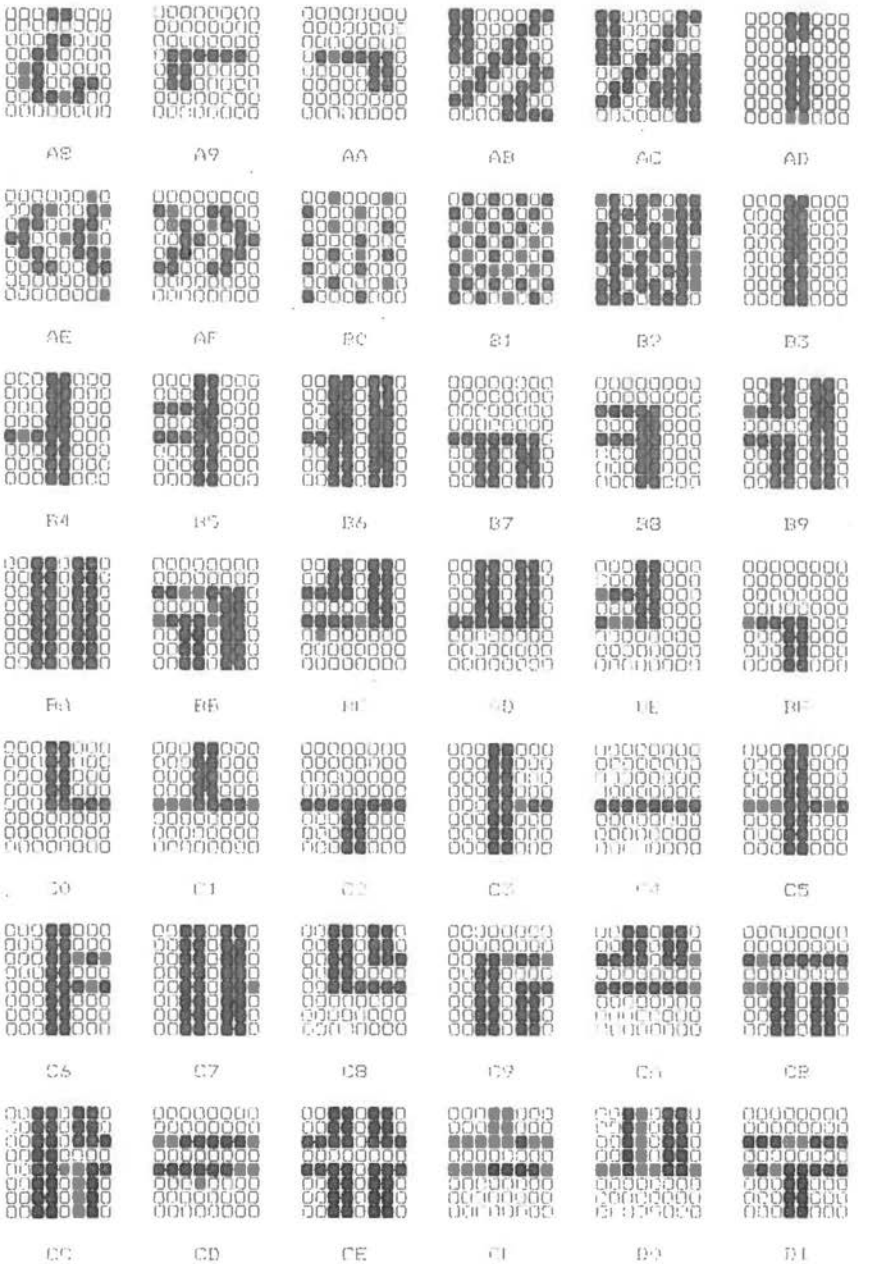


Figure 7.5 The Alternative Character Set (5 of 7)

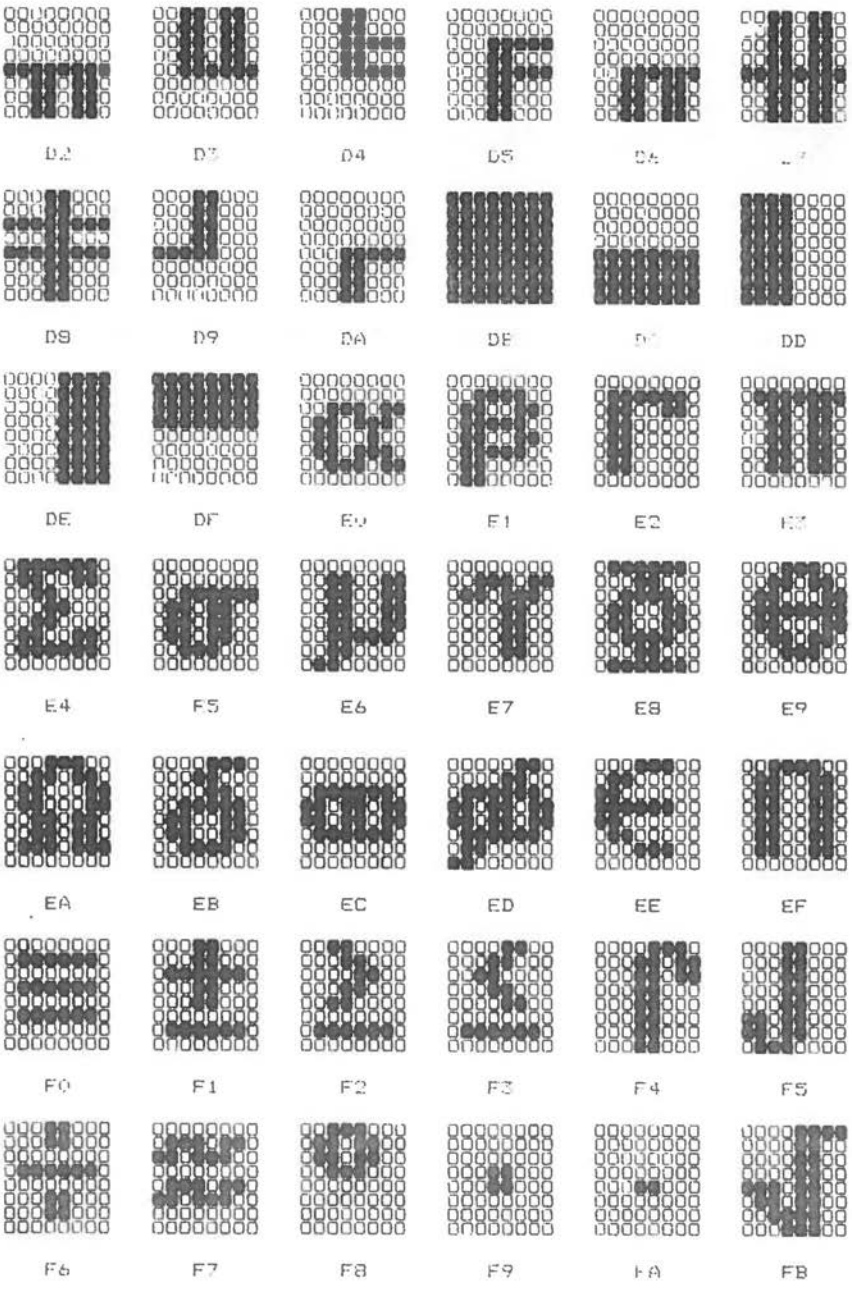


Figure 7.5 The Alternative Character Set (6 of 7)

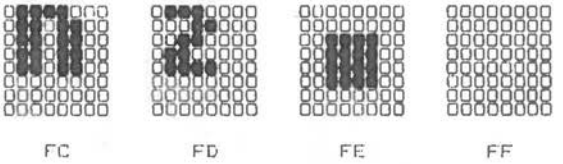


Figure 7.5 The Alternative Character Set (7 of 7)

MONOCHROME CHARACTER DISPLAY

The character display of the NCR PERSONAL COMPUTER can accomodate 80 characters in the horizontal and 25 characters in the vertical direction. The hardware characteristics of the monochrome character display:

* Screen dimensions	12 inches (295 mm)
* Deflection angle	90 degrees
* Phosphor	P31
* Display	Scale of green
* Definition	640 (horizontal) x 400 (vertical)
* Bandwidth	23 MHz
* Refresh rate	56.2 Hz
* Horizontal frequency	23.8 KHz
* Anti-glare surface	
*	

Figures 7.6 and 7.7 illustrate the signal flow which results in the generation of the character display. Figure 7.8 gives details about connections to the monochrome deflection board.

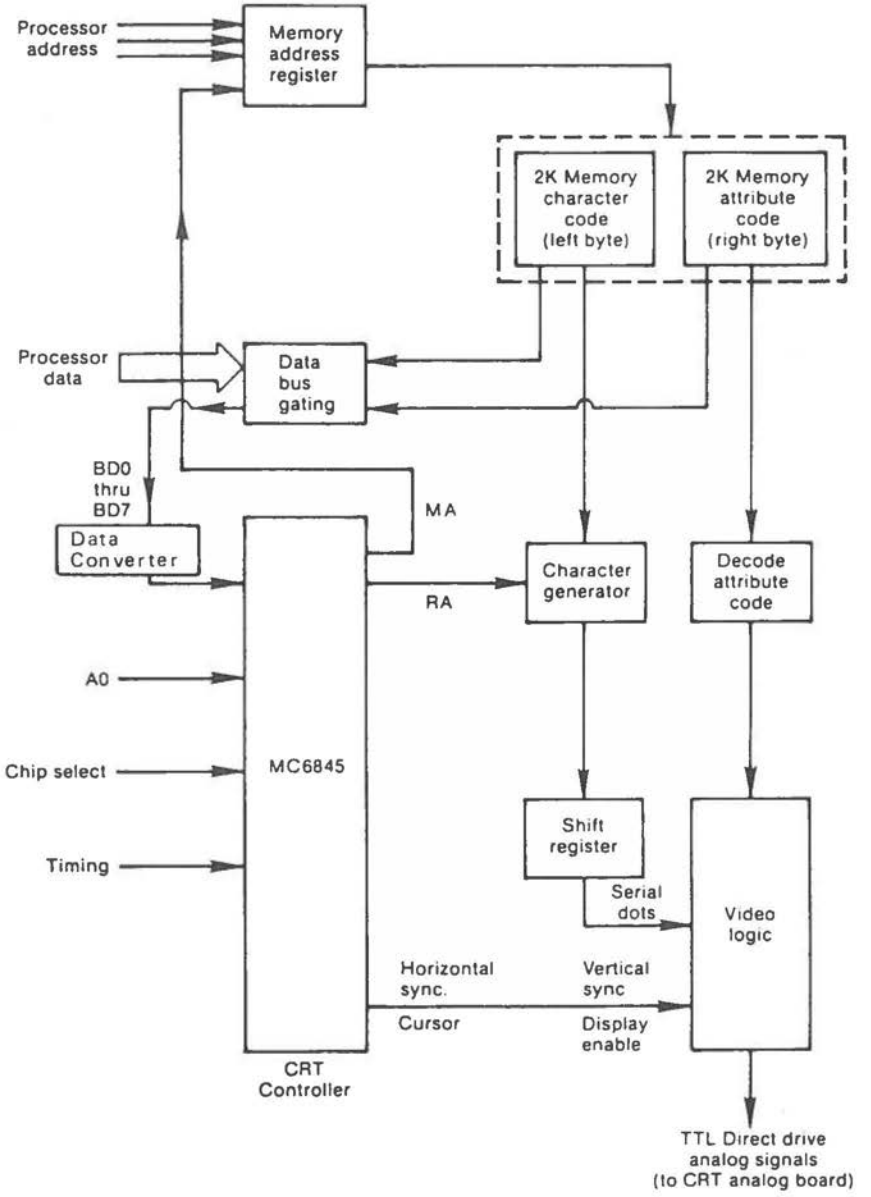


Figure 7.6 Character Display Logic

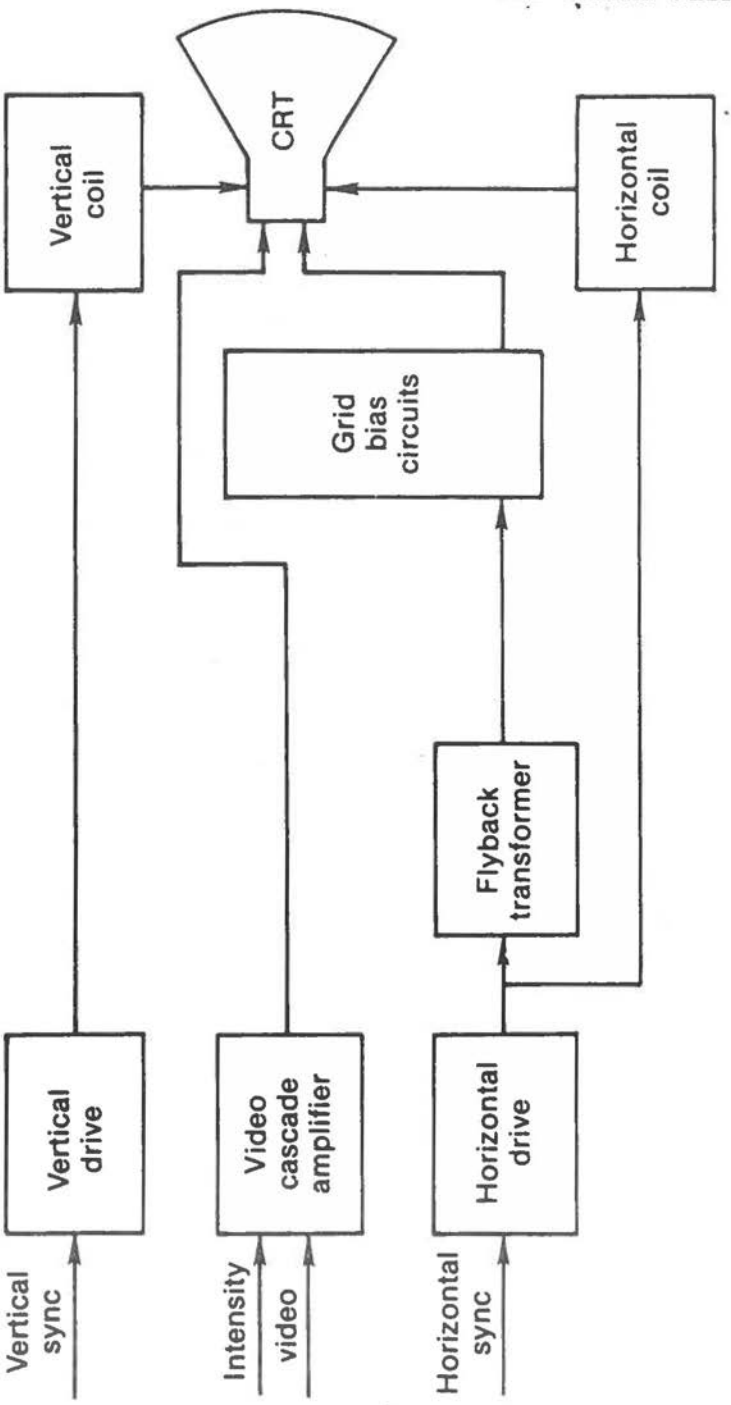
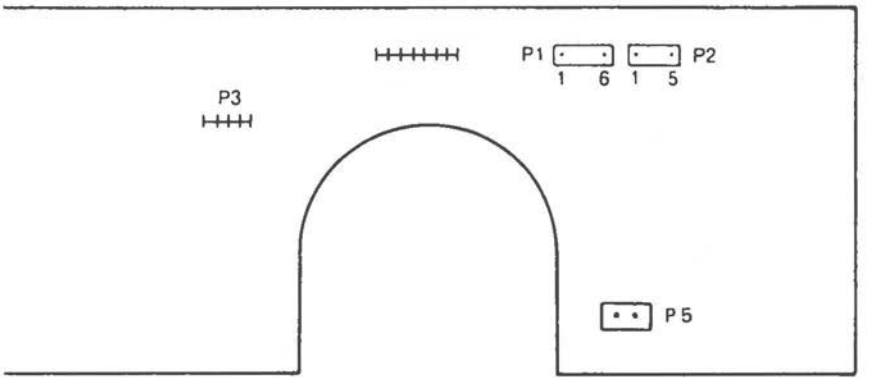


Figure 7.7 Monochrome CRT Signals



Monochrome Deflection
Board

P1-1 GND
2 Vertical Sync.
3 Video
4 Horizontal Sync.

} from character adapter card

P2-1 Contrast W
2 Contrast LO
3 Contrast HI

P5-1 +12V
2 GND

Figure 7.8 Monochrome CRT Deflection Board Pin Assignments

CHARACTER DISPLAY MEMORY

The character display adapter stores the character screen image in byte pairs (character code - attribute) in 4 KB of read/write memory, starting at CPU memory address 0B0000H. The even address of each pair stores the character code in the range 0 to 0FFH, the odd address immediately above it stores the attribute byte. The CPU/monitor access is synchronized with the character clock. Therefore, each CPU access to video memory is accompanied by at least one processor WAIT state. Alternatively, video memory can be accessed by DMA. The significance of the attribute byte is set out in Figure 7.9.

Function	Background bits			Writing bits		
	6	5	4	2	1	0
Invisible writing (black on black)	0	0	0	0	0	0
Invisible writing (green on green)	1	1	1	1	1	1
Green writing on black background	0	0	0	1	1	1
Black writing on green background (inverse video)	1	1	1	0	0	0
Green writing with underlining	0	0	0	0	0	1

Figure 7.9 The Character Display Attribute Byte

Notes:

Bits 7 and 3 set result in blinking and increased intensity, respectively.

Underline attribute results in a blue character on a color monitor.

The character display memory comprises 4 KB. The screen memory address lines are decoded in a way that this display memory is addressed a further 7 times, starting at 4 KB boundaries. This means that the single-page character display memory occupies a total address range of 32 KB.

CHARACTER DISPLAY CONTROLLER

Because video data can be written to and read from read/write memory with CPU MOVE instructions, programming the 6845 is required only in order to initialize the character display and for certain changes of the display mode. These programming steps are described in this section.

The following hexadecimal port addresses are dedicated to the 6845 for the character display card:

- 3B4 - Controller Index Register (value output via this port points to one of eighteen 6845 internal registers)
- 3B5 - Data Register (value output to register selected via port 3B4)
- 3B8 - Control functions (write only)
  - bit 0 - set to denote 80 x 25 display
  - bit 3 - set to enable video signal
  - bit 5 - set enables display blinking
- 3BA - Control information (read only)
  - bit 0 - set denotes horizontal retrace
  - bit 1 - set denotes light pen trigger set
  - bit 2 - set denotes light pen switch made
  - bit 3 - set denotes monochrome video
- 3BB - Clear light pen latch (write only)
- 3BC - Preset light pen latch

The controller register selected via port 03B4H can be one of those given in Figure 7.10. These controller registers are write only, except:

0EH, 0FH - read/write 10H, 11H - read only

Registers 0CH and 0DH are of special interest: when set to zero, they address the first character of the 4 KB screen page starting at CPU address 0B0000H as the top left corner of the screen. Changing this value, you can displace characters on the physical screen without having to transfer them within screen memory.

Reg.	Function/Value via Port 3B5
0	Horizontal synchronization period: the number of displayed characters plus the number of non-display characters (retrace time) minus 1.
1	Number of displayed characters per line (must be less than number specified in register 0).
2	Horizontal Sync position: defines horizontal sync and scan delays. Increasing this value shifts the display to the left, decreasing this value shifts the display to the right.
3	Sync width: the value of the vertical sync pulse and the horizontal sync pulse width.
4	Elapsed time of the vertical scan: the integer number of physically displayable character lines, minus 1.
5	Elapsed time of vertical scan: lines not involved with display of video memory image (for vertical adjustment).
6	Number of displayed character row times (must be less than value in register 4).
7	Vertical Sync Position: increasing this value shifts the display up, decreasing this value shifts the display down.

Figure 7.10 The Controller Registers (1 of 2)

Reg.	Function/Value via Port 385
8	Interlace Mode and Skew: combined value in bits 0,1: 0 or 2 = non interlace 1 = interlace synch 3 = interlace synch and video 4,5: 0 = no skew 1 = 1 character skew 2 = 2 character skew 6,7: 0 = no cursor skew 1 = 1 character cursor skew 2 = 2 character cursor skew
9	Number of scan lines per character row, minus 1.
0AH	Cursor definition: combined value in bits 0-3 = start scan line of cursor in character block; combined value in bits 5 and 6: 0 = slow blinking 1 = no cursor display 3 = fast blinking
0BH	Cursor definition: combined value in bits 0-3 = last scan line of cursor in character block.
0CH 0DH	First address output by CRT controller. The initial value is 0, thus addressing the beginning of video memory. In the case of the character display card, this represents the CPU address 0B0000H. The most significant bits of this address are in 0CH.
0EH 0FH	Cursor position in read/write memory (most significant bits in 0FH).
10H 11H	Light Pen address (most significant bits in 10H).

Figure 7.10 The Control Registers (2 of 2)

Control registers 0AH-11H can be manipulated in accordance with the requirements of the individual application. The other registers are set for a 80 x 25 character display as shown in Figure 7.11.

Register	Value - hex	Register	Value - hex
0	67 (61)	5	0 (06)
1	50	6	19
2	54 (52)	7	19
3	14 (0F)	8	0 (02)
4	1A (19)	9	0F (0D)

Figure 7.11 Control Registers For Character Display

Note:

Values given in parentheses apply to certain industry standard monitors which can be used in conjunction with suitable adapters (see section "Available Adapters" in this Chapter). The BIOS sets values for an internal monitor, but you can override these values in your own application.

CHARACTER CURSOR

The character display mode cursor of the NCR PERSONAL COMPUTER consists of up to 16 scan lines. In order to maintain compatibility with applications which assume a cursor of lower definition, a small PROM intercepts data to control registers 0AH and 0BH (via port 3B5H) before they reach the 6845 controller. The following conversions are performed:

* The value 2 is added to cursor start and end values between 8 and 13 inclusive. As a result, scan lines 8 and 9 are not accessible.

* Start and end values 14 and 15 are both interpreted as value 15.

Scan lines 0 to 7 are passed to the controller without intervening conversion. The scan line number specified for the cursor start must not be greater than that specified for the end scan line.

COLOR GRAPHICS DISPLAY

The graphics display card supports display of the 256-characters - set in an 80 (horizontal) x 25 format. All points addressable graphics can be implemented in three degrees of resolution. This card is capable of driving a color CRT. The color CRT of the NCR PERSONAL COMPUTER MODEL 4E has the following hardware characteristics:

* Screen dimensions	12 inches (295 mm)
* Digital input signals	Red, Green, Blue, Intensity
* Display	8 colors, each in normal and high intensity: black, red, green, blue, brown, magenta, cyan, white
* Definition	640 (horizontal) x 400
* Bandwidth	20 MHz
* Refresh rate	55.6 Hz
* Horizontal frequency	23.6 KHz
* Anti-glare surface	

Figures 7.12 and 7.13 illustrate the signal flow which results in the generation of the color display. Figure 7.14 illustrates connections to the color CRT deflection board.

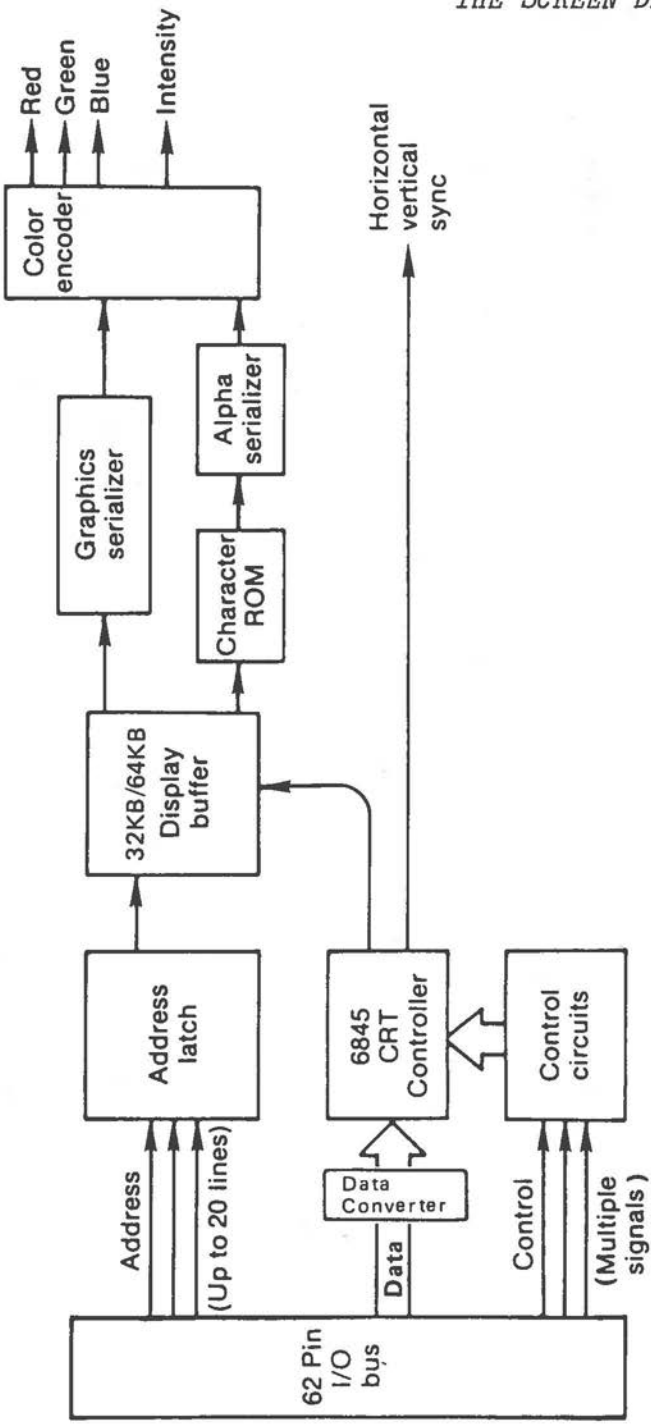


Figure 7.12 Color Graphic Display Logic

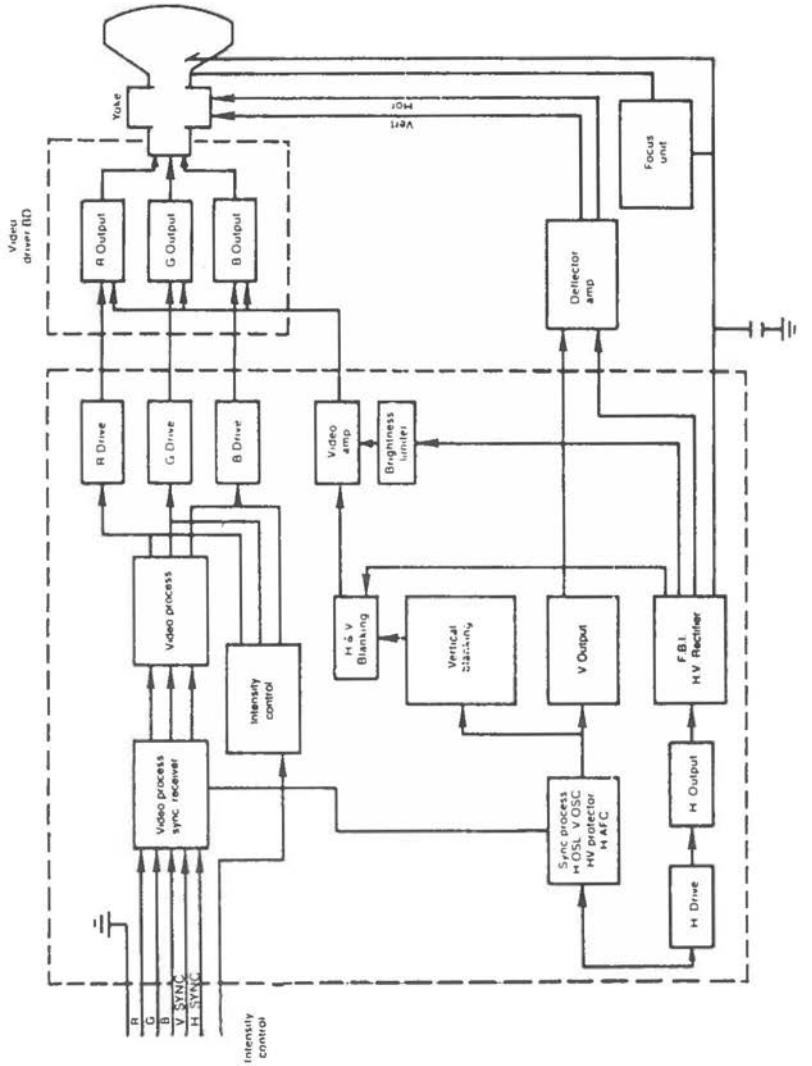
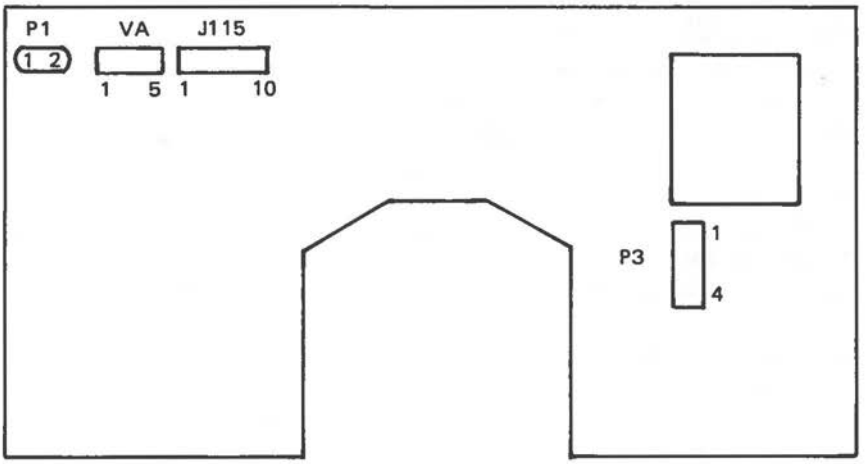


Figure 7.13 Color CRT Signals



P1 1 Power
2 Ground

VA RGB drive to CRT
Pin 2 is Ground

J115 To controller D-connector pins

Pin		D-connector pin
1	-	3
3	-	4
4	-	5
5	-	9
6	-	8
7	-	7
8	-	6

Figure 7.14 Color CRT Deflection Board Pin Assignments

GRAPHICS DISPLAY MEMORY

The graphic display card has as much as 64 KB of read/write memory at its disposal, according to the type of graphic display card being used. This memory resides in the CPU address areas 0B8000H - 0BFFFFH and 0A8000H - 0AFFFFH (upper 32 KB if video memory extension installed). The use of these memory areas varies according to the display mode selected, and is therefore given for each mode separately in this section. The video memory is dual ported, so that the CPU can access it at any time. To avoid display disturbances applications should confine access to the horizontal and vertical retrace intervals.

GRAPHICS DISPLAY CONTROLLER

The graphic display controller is the same as that used in the character display adapter - the 6845. The general concept of video display initialization discussed in "The Character Display Controller" also applies to initializing the graphic display character, both in its character and in its graphic display modes. The significant differences lie in the port addresses used and the more extensive mode control and status information needed to account for the variety of modes in which the adapter can be used.

The following hexadecimal port addresses are used by the graphic display card:

3D4 - Controller Index Register (value output via this port points to one of eighteen 6845 internal registers)

3D5 - Data Register (value output to register selected via port 3D4)

3D8 - Mode Control (write only). The effects of the individual bits, when set, are as follows:

0 - select fast CRT-controller clock (necessary for 80 x 25 character mode and 640 x 400 graphics mode)

1 - select graphics mode; zero selects character mode

2 - activate video signals for monochrome display; otherwise, a color display is assumed
3 - enable video signal
4 - select monochrome graphics (640 x 200 or 640 x 400)
5 - enable blinking function
6 - select 400 lines display
7 - select page: zero = page 1 or 2, set = page 3 or 4

3D9 - Color Selection (write only). The effects of the individual bits, when set, are as follows:

0, 1, 2 - activate blue, green, and red guns, respectively for border color (character mode); in graphics modes, this color setting applies to the background color. The border color in character mode is a software convention; it is not actually displayed on the CRT of the NCR PERSONAL COMPUTER.
3 - set high intensity for color selected by bits 0,1,2, provided these bits apply to character mode border or 320 x 200 graphics.
4 - set high intensity for background color in character display
5- palette selection in color graphics through activation and de-activation of the CRT blue gun
6,7 - not used

3DA - Status (read only). The significance of the individual bits, when set:
0- display enabled (controller not reading display memory at the moment)
1 - light pen trigger set
2 - light pen switch closed
3 - vertical retrace in progress
3DB Clear light pen latch
3DC - Preset light pen latch

CHARACTER MODE

The memory image of the character screen display on the graphics display card is built up in much the same way as that of the character display card described in the previous section: two bytes are devoted to each character area, of which the lower (even) byte contains a value representing one of the 256 characters in the character generator, the upper (odd) byte containing the display attribute.

The important difference between the two cards is that the graphics card interprets the attribute byte as color information for the color CRT.

The significance of the attribute byte is explained in Figure 7.15.

Color	Background bits			Writing bits		
	6	5	4	2	1	0
Black	0	0	0	0	0	0
Red	1	0	0	1	0	0
Green	0	1	0	0	1	0
Blue	0	0	1	0	0	1
Cyan	0	1	1	0	1	1
Magenta	1	0	1	1	0	1
Brown	1	1	0	1	1	0
White	1	1	1	1	1	1

Figure 7.15 The Character Display Attribute Byte [Color]

Notes:

Bits 7 and 3 set blinking and increased intensity, respectively. When blinking is disabled [see mode select register - port 3D8H], bit 7 controls background intensity.

The initial settings of 6845 registers (set via ports 3D4H and 3D5H) for the character modes of display of

the graphics display card for a color CRT are given in Figure 7.16. Registers 0AH-11H can be manipulated in accordance with the requirements of the individual application.

Register	Value - hex		Register	Value - hex	
	40 x 25	80 x 25		40 x 25	80 x 25
0	34 (38)	69 (71)	5	0 (06)	0 (06)
1	28	50	6	19	19
2	2C (2D)	58 (5A)	7	19 (1C)	19 (1C)
3	5 (0A)	0A	8	0 (02)	0 (02)
4	1A (1F)	1A (1F)	9	0F (07)	0F (07)

Figure 7.16 Control Registers For Color Character Display

Note:

Values given in parentheses apply to certain industry standard monitors which can be used in conjunction with suitable adapters (see section "Available Adapters" in this Chapter). The BIOS sets values for an internal monitor, but you can override these values in your own application.

Critical settings for the Mode Control Register (port 3D8H) are given in Figure 7.17.

Bit								Character Display Mode
7	6	5	4	3	2	1	0	
0	0	1	0	1	1	0	0	40 x 25 monochrome
0	0	1	0	1	0	0	0	40 x 25 color
0	0	1	0	1	1	0	1	80 x 25 monochrome
0	0	1	0	1	0	0	1	80 x 25 color

Figure 7.17 Character Mode Control On Color/Graphics Adapter

GRAPHICS MODES

The graphics display controller supports a number of degrees of graphic resolution. The graphics display adapter and initialization firmware of your NCR PERSONAL COMPUTER allow three graphic display modes: low, medium, and high resolution. Low and medium resolution graphics make use of a 16 KB video memory for one screen design, high resolution requires 64 KB.

Note:

A number of non-NCR personal computers apply the term "low resolution" to a 160 x 100 pixel graphic display, as, for example, transmitted to home televisions by means of a composite VHF or UHF signal. The low resolution graphic mode of your NCR PERSONAL COMPUTER supports 320 x 200 pixel color graphics, medium resolution supports a 640 x 200 pixel monochrome graphic display. The NCR high resolution mode of display, not usually supported by other personal computers, gives a high quality 640 x 400 pixel color graphic display. This enhanced feature of the NCR PERSONAL COMPUTER does not adversely affect the compatibility stated in the foreword to this Manual.

Low Resolution

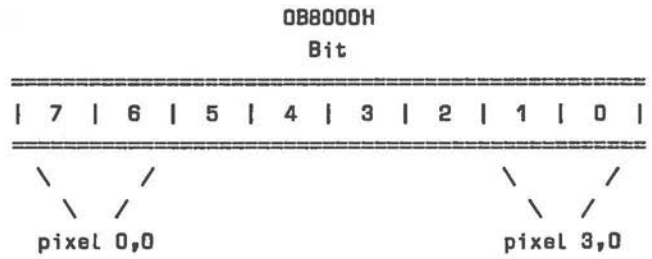
The low resolution graphic display consists of 320 horizontal by 200 vertical pixels. Each pixel is represented by two bits, so that a choice of four colors is available for each pixel. Each line is output twice (duplicated in the following odd scan), so that the physical screen is actually filled by 400 graphic lines.

Low resolution uses 16 KB of video memory as follows:

0B8000H - 0B9F3FH: even line scans (0,2,...198)
0B9F40H - 0B9FFFH: not used

0BA000H - 0BBF3FH: odd scan lines (1,3,...199)
0BBF40H - 0BBFFFH: not used

The first byte (0B8000H) contains color information for the four leftmost pixels of the top display line, and so on:



Of each bit pair the more significant bit controls the red gun, the less significant bit controls the green gun. If neither of these bits is set, the pixel takes on the current background color of the Color Select Register (port 3D9H, bits 0-3). The blue gun is governed by the Color Select Register (port 3D9H, bit 5). The CRT display uses one of two color palettes, according to whether the blue gun is on or not. By activating the monochrome bit in the Mode Control Register Port 3D8H, it is possible to create a third color palette. Color selection is set out in Figure 7.18.

The initial settings of 6845 registers (set via ports 3D4H and 3D5H) for the low resolution mode of graphic display on the internal color CRT are given in Figure 7.19. Registers 0AH-11H can be manipulated in accordance with specific programming requirements. When both set to zero, registers 0CH and 0DH set the graphics display controller to regard video memory address 0B8000H as the first byte of display information. You can change these two registers to address the upper 16 KB of the 32 KB graphic memory, enabling your programs to acccess a second graphic page.

Video Memory MSB	Video Memory LSB	Port 3D8H, bit 2	Port 3D9H, bit 5	Pixel Color
0	0	0	0 or 1	Current Background
0	1	0	0	Green
1	0	0	0	Red
1	1	0	0	Brown
0	1	0	1	Cyan
1	0	0	1	Magenta
1	1	0	1	White
0	0	1	0 or 1	Current Background
0	1	1	0 or 1	Cyan
1	0	1	0 or 1	Red
1	1	1	0 or 1	White

Figure 7.18 Low Resolution Graphics - Color Selection

Notes:
Background can be any one of 16 colors (8 basic colors, with intensity bit on or off).
Foreground color can be given high intensity (bit 4 of color select register, port 3D9H).

Register	Value - hex	Register	Value - hex
0	34 (38)	5	0 (06)
1	28	6	64
2	2C (2D)	7	64 (70)
3	5 (0A)	8	0 (02)
4	6B (7F)	9	3 (01)

Figure 7.19 Control Registers For Low Resolution Graphics

Note:

Values given in parentheses apply to certain industry standard monitors which can be used in conjunction with suitable adapters [see section "Available Adapters" in this Chapter]. The BIOS sets values for an internal monitor, but you can override these values in your own application.

If your graphics display card contains a 64 KB video memory, you can use these two registers to switch between two graphic pages. With additional manipulation of the page switch bit in the Mode Control register (port 3D8H, bit 7) up to 4 graphic pages can be addressed.

The memory map for the full 64 KB video memory is as follows:

```
0B8000H - 0B9F3FH: even line scans  }
0B9F40H - 0B9FFFH: not used         }  page 1
0BA000H - 0BBF3FH: odd scan lines   }
0BBF40H - 0BBFFFH: not used         }

0BC000H - 0BDF3FH: even scan lines  }
0BDF40H - 0BDFFFH: not used         }  page 2
0BE000H - 0BFF3FH: odd scan lines   }
0BFF40H - 0BFFFFH: not used         }


0A8000H - 0A9F3FH: even line scans  }
0A9F40H - 0A9FFFH: not used         }  page 3
0AA000H - 0ABF3FH: odd scan lines   }
0ABF40H - 0ABFFFH: not used         }

0AC000H - 0ADF3FH: even scan lines  }
0ADF40H - 0ADFFFH: not used         }  page 4
0AE000H - 0AFF3FH: odd scan lines   }
0AFF40H - 0AFFFFH: not used         }
```

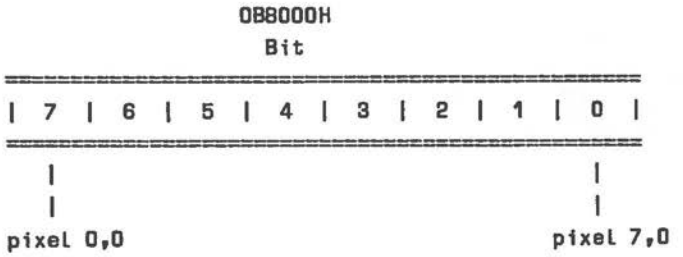
Critical settings for the Mode Control Register (port 3D8H) are to be found in Figure 7.20 (bit 5 is "don't care").

Bit								Low Resolution Graphics
7	6	5	4	3	2	1	0	
0	0	-	0	1	0	1	0	Color
0	0	-	0	1	1	1	0	Monochrome

Figure 7.20 Graphics Mode Control - Low Resolution

Medium Resolution

The medium resolution graphic display consists of 640 horizontal by 200 vertical pixels. Each pixel is represented by a single bit, so that each pixel can be displayed as on or off in 16 KB of video memory. The video memory addresses used are the same as those for low resolution graphics. The only difference is that each byte contains display information (monochrome on or off) for 8 pixels:



The 6845 registers (port 3D4H/3D5H) are set as for medium resolution graphics.

The Mode Control Register (port 3D8H) is set as shown in Figure 7.21 (bit 5 is "don't care".

Bit								Medium Resolution Graphics
7	6	5	4	3	2	1	0	
0	0	-	1	1	1	1	0	Monochrome (640 x 200) pages 1 and 2
1	0	-	1	1	1	1	0	Monochrome (640 x 200) pages 3 and 4

Figure 7.21 Graphics Mode Control - Medium Resolution

High Resolution

The high resolution mode of the graphic display enables you to use the internal color CRT to its full advantage. This mode makes use of the maximum of 64 KB video memory which can be installed on the graphics display card. In the case of color graphics, this memory is used as follows:

```
0B8000H - 0BBF3FH: even scans (0,4,8,...396)
0BBF40H - 0BBFFFH: not used
0BC000H - 0BFF3FH: odd scans  (1,5,9,...397)
0BFF40H - 0BFFFFH: not used
0A8000H - 0ABF3FH: even scans (2,6,10,...398)
0ABF40H - 0ABFFFH: not used
0AC000H - 0AFFFFH: odd scans  (3,7,11,...399)
0AFF40H - 0AFFFFH: not used
```

As in low resolution graphics, each pixel is represented by two bits. Bits 7 and 6 of byte 0B8000H determine the color of the pixel in the top left corner of the screen: either current background color as specified by the Color Select Register, or one of three colors in the current palette selected by the Color Select Register.

The same resolution (640 x 400) can be confined to monochrome (pixel on or off) graphics. As in medium resolution graphics, each pixel is represented by one bit starting with bit 7 at 0B8000H (top left pixel). An alternate screen image can be stored at 0A8000H. The video memory map for the screen image starting at 0B8000H (the equivalent is also available at 0A8000H):

```
0B8000H - 0B9F3FH: even scans (0,4,8,...396) }
0B9F40H - 0B9FFFH: not used                  }
0BA000H - 0BBF3FH: odd scans  (2,6,10,...398)}
0BBF40H - 0BBFFFH: not used                  }image 1
0BC000H - 0BDF3FH: even scans (1,5,9,...397) }
0BDF40H - 0BDFFFH: not used                  }
0BE000H - 0BFF3FH: odd scans  (3,7,11,...399)}
0BFF40H - 0BFFFFH: not used                  }
```

The 6845 registers (port 3D4H/3D5H) are set for high resolution graphics as shown in Figure 7.22.

Register	Value – hex	Register	Value – hex
0	69	5	0
1	50	6	64
2	58	7	64
3	0A	8	0
4	6B	9	3

Figure 7.22 Control Registers For High Resolution Graphics

Figure 7.23 illustrates Mode Control Register settings (port 3D8H) for high resolution graphics.

Bit								High Resolution Graphics
7	6	5	4	3	2	1	0	
–	1	–	1	1	1	1	0	Monochrome
–	1	–	0	1	0	1	1	Color

Figure 7.23 Graphics Mode Control – High Resolution

The remainder of this section consists of a program (NCR-DOS .EXE file) enabling you to plot pixels on a high resolution color display.

The graphics "cursor" is a single pixel, initially at the center of the screen, which can be moved in the compass directions N, NE, E, SE, S, SW, W, and NW by depressing the numeric pad keys 8, 9, 6, 3, 2, 1, 4, and 7, respectively.

A status line displays the x/y co-ordinates of the current cursor position, the top left corner of the screen being the origin. Range checking (maximum x is 639, maximum y is 399) is included.

The L key is a toggle between screen drawing and simply moving the cursor. The C key is used for stepping through the colors which can be displayed in high resolution color graphics. The range of colors is made up of two "palettes": the palette is selected by the blue gun being on or off (port 3D9H). Both the L and C keys are reflected in the status line.

The status line is initially situated at the foot of the display, but when pixel plotting or cursor movement approaches the status line, it is transposed to the top of the screen and the former contents of the vacated area are restored.

Pressing the Z key gives the cursor fast movement in the direction last specified by a direction key.

You may wish to add routines of your own to store the graphic screen on disk. If so, execute the SHOWBAND routine before saving the screen. This discards the status line in favour of the full-height graphic design. You can use the STOREBAND routine already provided for saving the status line area, to read the screen 16 scan lines at a time.

If you are adding figure drawing capabilities by way of trigonometric calculation, you may wish to take into consideration that the internal monitor has a horizontal/vertical pixel ratio of 6/5: the vertical pixels are a little further apart than the horizontal pixels.

```
            PAGE ,132

/IDEOINT      EQU  10H           ;ROM BIOS initialize video mode
                                 ;interrupt.
:BCHINT       EQU  16H           ;ROM BIOS read keyboard interrupt.
;
/IDEORAM      EQU  0B800H        ;Paragraph address of lowest location
                                 ;occupied by first scan line
                                 ;of hi-res color display.
;
(MAX          EQU  639           ;Maximum x and y screen co-ordinate
/MAX          EQU  399           ;values for high-resolution color.
;
:IPFACT       EQU  20            ;Repetition factor for fast screen
                                 ;movement.
;
:ODEC         EQU  2EH           ;Scan code for C.
:ODEL         EQU  26H           ;              L.
:ODEF         EQU  21H           ;              F.
:ODEZ         EQU  2CH           ;              Z.
;
            CSEG SEGMENT
            ASSUME CS:CSEG,DS:CSEG,ES:NOTHING
;
            ORG  100H
            PUSH CS
            POP  DS
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                               ;
; Setting up:                                                   ;
;    Display is high resolution (640 x 400) color               ;
;    Status display area is lower screen                        ;
;    Display starting co-ordinates                              ;
;                                                               ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
            CALL HIRESINIT
            CALL LOGISWITCH  ;Set logic to move.
            CALL COLSWITCH   ;Set initial color to 0.
;
;
CYCLE:
```

```
CYCLE1:     CMP   ZIPCOUNT,0
            JZ    CYCLE2        ;Jump if direction repetition non-
                                ;existent or exhausted.
            DEC   ZIPCOUNT
            MOV   AL,ZIPKEY     ;Get direction for repetition.
            JMP   SHORT CYCLE4  ;No need to analyze key further.
CYCLE2:     CALL  KBREAD
                                ;Scan code is in AH.
            CMP   AH,CODEZ      ;Key Z for zip.
            JNZ   CYCLE3        ;Jump if no zip to be activated.
            MOV   ZIPCOUNT,ZIPFACT
            JMP   SHORT CYCLE1
CYCLE3:     CMP   AH,47H        ;Direction arrow scan codes are in
            JB    CYCLE5        ;range 47H to 51H.
            CMP   AH,51H
            JA    CYCLE5
            MOV   AL,AH
            MOV   ZIPKEY,AL     ;Store direction in case needed for
                                ;subsequent zip.
CYCLE4:     CALL  ARROW         ;Analyze direction key.
            CALL  DISPCOORDS    ;Display new co-ordinates.
            CALL  PLOT          ;Plot or simply move.

CYCLE5:     CMP   AH,CODEC      ;Check key C for color.
            JNZ   CYCLE51
            CALL  COLSWITCH
            JMP   CYCLE1
CYCLE51:    CMP   AH,CODEL      ;Check L for logic.
            JNZ   CYCLE52
            CALL  LOGISWITCH
CYCLE52:    JMP   CYCLE1
;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                     ;
; Calculate new x/y coordinates, relative to screen top-left;         ;
;   restrain out of range values.                                     ;
; Entry:  scan code in range 47H to 51H in AL                         ;
; Return: new x in CX and at HORIPIX + 2                              ;
;             y    DX        VERTIPIX + 2                             ;
;                                                                     ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
ARROW:      MOV  CX,HORIPIX
            MOV  DX,VERTIPIX
            SUB  AL,47H
            MOV  BX,OFFSET DIRTAB           ;Movement direction table.
            XLAT                            ;4 MSBs are zero,
                                            ;bit 0 set = right
                                            ;    1     = left
                                            ;    2     = down
                                            ;    3     = up
                                  ;Shift 4 LSBs through Carry Flag,
                                  ;adjusting horiz. (CX) and vertical
                                  ;(DX) co-ords according to new pos.
            SHR  AL,1
            ADC  CX,0
            SHR  AL,1
            SBB  CX,0
            CMP  CX,XMAX+1        ;Range check - horizontal.
            JNAE HORISET          ;Jump if not 640 or 0FFFFH.
            DEC  CX               ;Horizontal max. is 639.
            CMP  CX,XMAX
            JZ   HORISET          ;Jump if drawing would have run off
                                  ;right edge of screen.
            XOR  CX,CX            ;Horizontal co-ord. had decremented
                                  ;from 0 to 0FFFFH, so zero it.
HORISET:                          ;Legal horizontal value now in CX.
            SHR  AL,1             ;Equivalent for vertical co-ordinate.
            ADC  DX,0
            SHR  AL,1
            SBB  DX,0
            CMP  DX,YMAX+1        ;Range check - vertical.
            JNAE VERTISET         ;Jump if not 400 or 0FFFFH.
            DEC  DX               ;Vertical max. is 399.
            CMP  DX,YMAX
```

```
            JZ    VERTISET      ;Jump if drawing would have run off
                                ;bottom edge of screen.
            XOR   DX,DX
VERTISET:                       ;New legal x/y co-ords established.
            MOV   HORIPIX + 2,CX
            MOV   VERTIPIX + 2,DX
            PUSH  CX
            PUSH  DX
                                ;Now check whether the status line
                                ;ought to be moved.
            CMP   DX,360
            JZ    STATMOV1
            CMP   DX,40
            JZ    STATMOV2
            JMP   SHORT POSSET      ;No need to move status line.
STATMOV1:   CMP   CURROW,24
            JZ    STATMOV3      ;Must move bottom to top.
            JMP   SHORT POSSET
STATMOV2:   CMP   CURROW,24
            JNZ   STATMOV4      ;Must move top to bottom.
;
POSSET:     POP   DX
            POP   CX
            RET
;
STATMOV3:
STATMOV4:   CALL  SHOWBAND          ;Change position of status line.
            MOV   AX,BANDPARAG
            MOV   BX,BANDPARAGT
            MOV   BANDPARAG,BX
            MOV   BANDPARAGT,AX
            CALL  STOREBAND
            CALL  CLEARBAND
            MOV   AX,WORD PTR CURROW
            XCHG  AL,AH
            MOV   WORD PTR CURROW,AX
            MOV   CURCOL,0
            CALL  STATLINE
            JMP   SHORT POSSET
;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                        ;
; Plot one pixel [2 bits in hi-res color] on screen                      ;
;                                                                        ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
PLOT:       CALL XOLDPIX
            MOV  AX,HORIPIX + 2         ;New co-ords now become
            MOV  HORIPIX,AX             ;current co-ords.
            MOV  AX,VERTIPIX + 2
            MOV  VERTIPIX,AX
            CALL ILLUMINE
            RET
;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
              ;                                     ;
              ; Leave old pixel position as it      ;
              ; was before illuminating next one    ;
              ;                                     ;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
XOLDPIX:    MOV  AL,PIXLOGIC
            AND  AL,1         ;Ignore all but LSB.
            JNZ  DOTPLOT
DOTUNPLOT:  JMP  ILLUMINE     ;RET is from this routine.
DOTPLOT:    RET
;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
              ;                                     ;
              ; Illuminate pixel calculated by      ;
              ; PIX2MEM, in accordance with         ;
              ; PIXLOGIC                            ;
              ;                                     ;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
ILLUMINE:   MOV  BX,OFFSET VERTIPIX
            CALL PIX2MEM      ;Isolate current pixel.
            MOV  AL,BYTE PTR ES:[BX]
            MOV  AH,AL
            MOV  DH,PIXLOGIC
            AND  DH,1
            JNZ  REPLACE
```

```
MOVEPOS:    NOT   AL
            JMP   SHORT SEL2BITS
REPLACE:    MOV   AL,PIXCOLOR
SEL2BITS:   MOV   DH,DL         ;Copy pixel mask.
            NOT   DH
            AND   AL,DL         ;Blank out bits not concerned
                                ;with pixel.
            AND   AH,DH         ;Blank out bits concerned with pixel.
            OR    AL,AH         ;Reconstruct video byte.
            MOV   BYTE PTR ES:[BX],AL
            RET
;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
              ;                                       ;
              ; Return at VIDEOPG:VIDEOOFFS+VMASK;
              ; video RAM address and 2-bit mask ;
              ; of the color pixel whose y value ;
              ; is addressed by BX, with the x    ;
              ; value at BX - 4.                  ;
              ; Return info also available in     ;
              ; ES:BX+DL.                         ;
              ;                                   ;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
PIX2MEM:    MOV   SI,VIDEORAM - 400H
                                ;Base paragraph of graphics video
                                ;memory (will be raised by 400H on
                                ;first pass through loop).
            MOV   CX,[BX]
            MOV   AX,CX         ;For offset calculations below.
            AND   CX,0003H      ;Blank out all but 2 LSBs of
                                ;vertical position.
            INC   CX
PIX2MEM1:   ADD   SI,400H       ;Add amount (paragraph) of one
                                ;scan line block.
            LOOP  PIX2MEM1
            TEST  SI,4000H      ;Now test whether upper hex digit
            JZ    PIX2MEM2      ;went from B to C. If it did, convert
            AND   SI,0BFFFH     ;it to A and add 800H to SI, thus
            OR    SI,2000H      ;addressing correctly NCR extra video
            ADD   SI,800H       ;RAM (paragraph).
PIX2MEM2:   MOV   ES,SI
            MOV   VIDEOPG,SI    ;1 of 4 scan line blocks selected;
```

```
            SHR   AX,1          ;now calculate offset of beginning of
            SHR   AX,1          ;actual line into that paragraph.
                                ;AH is zero, as max. value in AX
                                ;was 18FH (399).
            MOV   CL,0A0H       ;160 bytes per line.
            MUL   CL            ;Byte offset of beginning of line
            MOV   DX,AX         ;now in DX.
            SUB   BX,4
            MOV   AX,[BX]       ;Get horizontal position.
            PUSH  AX            ;Save for 2-bit/pixel calculation.
            SHR   AX,1
            SHR   AX,1
            ADD   AX,DX         ;AX = offset due to horizontal coord,
                                ;DX =                vertical        .
            MOV   BX,AX         ;As return parameter.
            MOV   VIDEOOFFS,AX  ;Offset into scan block stored.
            POP   CX            ;Now establish which two bits of the
            AND   CX,0003H      ;byte calculated refer to the color
                                ;pixel.
            INC   CX
            MOV   AX,0300H      ;Initially assume that pixel
                                ;addressed is the leftmost of the
                                ;4 pixels in the byte.
PIX2MEM3:   SHR   AX,1
            SHR   AX,1
            LOOP  PIX2MEM3
            MOV   DL,AL
            MOV   VMASK,AL      ;1 of 4 pixel selection mask stored.
            RET
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                              ;
; Initialize to high-resolution color graphics                                 ;
;                                                                              ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
HIRESINIT:  MOV   AH,0          ;ROM BIOS INT 10H function to
                                ;set video mode.
            MOV   AL,9          ;Selects 640 x 400 color.
            INT   VIDEOINT
            RET
;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                       ;
; Routines to write binary pos.[status line] to graphics screen         ;
;                                                                       ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
DISPCOORDS: PUSH   CX           ;Save pixel position in regs
            PUSH   DX           ;for plotting.
            MOV    AX,CX        ;X co-ordinate.
            MOV    CURCOL,4     ;Position of first digit in line.
            CALL   SEEDIGITS
            MOV    AX,DX        ;Same for Y co-ordinate.
            MOV    CURCOL,14
            CALL   SEEDIGITS
            POP    DX
            POP    CX
            RET
;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
              ;                                    ;
              ; Entry: binary number in AX,        ;
              ; cursor row/col in CURROW/CURCOL    ;
              ;                                    ;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
SEEDIGITS:  PUSH   DX           ;So as not to corrupt Y co-ordinate.
            MOV    DX,WORD PTR CURCOL
            CALL   CURSET
            MOV    BL,100       ;Decimal divisor.
            DIV    BL           ;Max. dividend is 639.
                                ;Quotient in AL, remainder in AH.
            CALL   DISPNUMB     ;Display hundreds.
            XCHG   AL,AH
            CBW                 ;AX = new quotient = old remainder.
            MOV    BL,10
            DIV    BL
            CALL   DISPNUMB     ;Display tens.
            XCHG   AL,AH
            CALL   DISPNUMB     ;Display units.
            POP    DX
            RET
;
```

```
DISPNUMB:   PUSH  AX
            PUSH  BX
            OR    AL,30H      ;Convert binary dec. digit to ASCII.
            MOV   AH,0EH      ;ROM BIOS INT 10H: teletype.
            MOV   BL,3        ;Character color.
            INT   VIDEOINT
            POP   BX
            POP   AX
            RET
;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
              ;                                    ;
              ; Display item titles/parameters    ;
              ;   on status line                   ;
              ;                                    ;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
STATLINE:
            PUSH DS           ;Build up status line (DI) with
            POP  ES           ;Color+Logic+Coords (SI).
            MOV  SI,OFFSET LOGITEXT
            MOV  DI,OFFSET STATTEXT1 + 69
            MOV  CX,7
            CLD
        REP MOVSB             ;SI correct following previous MOVSB.
            MOV  CURCOL,0
            MOV  DX,WORD PTR CURCOL
            CALL CURSET
            MOV  BX,OFFSET STATTEXT1
            MOV  CX,78        ;Length of status line.
STAT1:      MOV  AH,0EH       ;ROM BIOS INT 10H: write teletype.
            MOV  AL,[BX]
            PUSH BX
            MOV  BL,2         ;Palette color for status items.
            INT  VIDEOINT
            POP  BX
            INC  BX
            LOOP STAT1        ;Until items/parameters
                              ;[without color] displayed.

            MOV  BX,OFFSET STATTEXT1 + 51
            MOV  CURCOL,51
            MOV  DX,WORD PTR CURCOL
```

```
                CALL CURSET       ;Address "COLOR" in status line.
                MOV  CX,5
STAT2:          MOV  AH,0EH       ;ROM BIOS INT 10H: teletype.
                MOV  AL,[BX]
                PUSH BX
                MOV  BL,PIXCOLOR
                AND  BL,3         ;Teletype display color in 3 bits.
                INT  VIDEOINT     ;Display one character of "COLOR".
                POP  BX
                INC  BX
                LOOP STAT2

                MOV  CX,HORIPIX   ;Re-display current co-ordinates, as
                MOV  DX,VERTIPIX  ;they were erased by new status line
                CALL DISPCOORDS   ;build-up.
                RET
;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;                                           ;
                ; ROM BIOS read from keyboard               ;
                ;   buffer: scan code in AH                 ;
                ;                                           ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
KBREAD:         MOV  AH,0         ;Keyboard read function.
                INT  KBCHINT      ;ROM BIOS read keyboard buffer.
                RET               ;Returns scan code in AH, char.in AL,
                                  ;provided Zero Flag = 0.
;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;                                           ;
                ; Set cursor for character writing          ;
                ; in hi-res color display.                  ;
                ; Entry: row in DH, column in DL            ;
                ;                                           ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
CURSET:         PUSH AX
                PUSH BX
                MOV  AH,2         ;ROM BIOS INT 10H: set cursor.
                MOV  BH,0         ;Only display page possible
                INT  VIDEOINT     ;in hi-res color.
                POP  BX
                POP  AX
```

```
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                          ;
; Routines to maintain status line at top or bottom of screen              ;
;                                                                          ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
              ;                                  ;
              ; Clear band 16 pixels high,       ;
              ; starting at BANDPARAG:0          ;
              ;                                  ;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
CLEARBAND:  MOV   AX,BANDPARAG
            MOV   DL,2          ;1 pass thru conventional PC video
                                ;memory, 1 pass thru NCR extra mem.
CLB1:       MOV   ES,AX
            MOV   SI,AX         ;For paragraph addition to address
                                ;subsequent  scan blocks.
            XOR   AX,AX         ;Fill band with zeros.
            MOV   DI,AX         ;Offset pointer to video memory.
            MOV   BL,2          ;2 scan blocks each in normal
                                ;PC video and NCR extra memory.
CLB2:       MOV   CX,320        ;80 words per line.
            CLD                 ;Block fill increment DI/SI.
        REP STOSW
            ADD   SI,400H       ;Paragraph size of one scan block.
            MOV   ES,SI
            XOR   DI,DI
            DEC   BL
            JNZ   CLB2
            MOV   AX,BANDPARAG
            AND   AX,0EFFFH     ;Point to NCR extra memory.
            DEC   DL
            JNZ   CLB1          ;Jump if scan blocks in NCR extra
                                ;memory still to be written.
            RET
;
```

```
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;                                           ;
                ; Copy 16 scan lines BANDSTORE to           ;
                ; video memory                              ;
                ;                                           ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
SHOWBAND:   MOV  DL,2                   ;As CLEARBAND, except that
            MOV  SI,OFFSET BANDSTORE    ;AX not required for zero
            MOV  AX,BANDPARAG           ;character.
SHOWB1:     MOV  ES,AX
            MOV  BL,2
SHOWB2:     MOV  CX,320
            XOR  DI,DI
            CLD
        REP MOVSW
            ADD  AX,400H                ;Address 2nd scan block in
            MOV  ES,AX                  ;B800 or A800 video memory.
            DEC  BL
            JNZ  SHOWB2
            MOV  AX,BANDPARAG           ;Get paragraph addr. as for
            AND  AX,0EFFFH              ;B800 block, change B to A.
            DEC  DL
            JNZ  SHOWB1
            RET
;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;                                           ;
                ; Copy 16 scan lines video memory           ;
                ; to BANDSTORE                              ;
                ;                                           ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                            ;This is the corollary procedure to
                            ;SHOWBAND: DS must point to video
                            ;memory, ES to normal data memory.
STOREBAND:  MOV  DL,2
            MOV  DI,OFFSET BANDSTORE    ;Destination is not video.
            MOV  AX,BANDPARAG           ;Source is video.
            PUSH DS
            PUSH DS
            POP  ES             ;Destination in effect refers to DS.
STOREB1:    MOV  DS,AX
            MOV  BL,2
STOREB2:    XOR  SI,SI
```

```
            MOV  CX,320
            CLD
        REP MOVSW
            ADD  AX,400H
            MOV  DS,AX
            DEC  BL
            JNZ  STOREB2
            MOV  AX,CS:BANDPARAG   ;Segment override prefix needed
                                   ;as DS is still corrupted.
            AND  AX,0EFFFH
            DEC  DL
            JNZ  STOREB1
            POP  DS
            RET
;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
              ;                                       ;
              ; Color and display mode switching       ;
              ;                                       ;
              ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
COLSWITCH:  CALL ILLUMINE      ;Neutralize current cursor point.
            PUSH DX
            MOV  DX,3D9H       ;Graphic controller color selection.
            MOV  AL,COLNUMBER  ;Value 0-7.
            INC  AL            ;Step to next color.
            MOV  COLNUMBER,AL  ;Store next color in sequence.
            PUSH AX
            TEST AL,4          ;This bit controls blue gun.
            JZ   BLUEOFF
BLUEON:     MOV  AL,20H
            JMP  SHORT COL2BITS
BLUEOFF:    MOV  AL,0
COL2BITS:   OUT  DX,AL
            POP  AX
            AND  AL,3          ;Duplicate 2 LSBs through byte.
            MOV  AH,AL
            SHL  AL,1
            SHL  AL,1
            OR   AL,AH
            MOV  AH,AL
            MOV  CL,4
            SHL  AL,CL
            OR   AL,AH
```

```
            MOV  PIXCOLOR,AL  ;Mask for video memory bytes.
            CALL STATLINE
            CALL ILLUMINE     ;Illuminated, but eliminated by next
                              ;call to XOLDPIX.
            POP  DX
            RET
;
LOGISWITCH: INC  PIXLOGIC     ;Toggle Move/Replace.
            MOV  AL,PIXLOGIC
            AND  AL,1         ;Discard all but LSB.
            MOV  SI,OFFSET LOGIC1
            MOV  CL,7
            MUL  CL           ;Address status parameter CL bytes
                              ;higher if Replace. AH need  not be
                              ;zeroed.
            ADD  SI,AX
            MOV  DI,OFFSET LOGITEXT
            PUSH DS
            POP  ES
            MOV  CX,7
            CLD
        REP MOVSB
            CALL STATLINE
            CALL ILLUMINE
            RET
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                              ;
;                          Data                                                ;
;                                                                              ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
ZIPKEY      DB   0            ;Occupied by scan code of fast move
                              ;direction.
ZIPCOUNT    DB   0            ;Decrements from ZIPFACTOR during
                              ;fast movement.
;
HORIPIX     DW   320          ;Current y co-ordinate.
            DW   320          ;New     "      "      (initial).
VERTIPIX    DW   200          ;Current x co-ordinate.
            DW   200          ;New     "      "      (initial).
VIDEOPG     DW   1 DUP (0)    ;Paragraph of video memory byte.
VIDEOOFFS   DW   1 DUP (0)    ;Offset of video memory byte.
```

```
                                ;to be affected by plotting action.
VMASK       DB   1 DUP (0)      ;2 adjacent bits set to select one
                                ;of four color pixels in video byte.
;
PIXLOGIC    DB   1              ;0 = move only, 1 = replace.
COLNUMBER   DB   0              ;Color in 3 LSBs, 5 MSBs irrelevant.
PIXCOLOR    DB   55H            ;To contain 4 2-bit copies of color.
;
                                ; 0000^_<>
DIRTAB      DB                    00001010B
            DB                    00001000B
            DB                    00001001B
            DB                    0
            DB                    00000010B
            DB                    0             ;key 5
            DB                    00000001B
            DB                    0
            DB                    00000110B
            DB                    00001100B
            DB                    00001101B
            DB                    0
;
;
BANDPARAG   DW   0BBC0H         ;Paragraph address of first byte of
                                ;status line. Here initial value, but
                                ;will change when drawing active gets
                                ;dangerously close, to ...
BANDPARAGT  DW   VIDEORAM
;
STATTEXT1   DB   'X =       Y =
            DB   'COLOR    Logic is              '
LOGITEXT    DB   'MOVE   '
;
LOGIC1      DB   'MOVE   '
LOGIC2      DB   'REPLACE'
BANDSTORE   DW   1600 DUP (0) ;Store graphics area currently
                                ;occupied by status line.
CURCOL      DB   0              ;Cursor row and column storage
CURROW      DB   24             ;for ROM BIOS character output.
CURROWT     DB   0
;
;
            CSEG ENDS
            END
```

GRAPHICS CURSOR

The graphic cursor of the NCR PERSONAL COMPUTER, like the character display mode cursor, consists of up to 16 scan lines. In order to maintain compatibility with applications which assume a cursor of lower definition, a small PROM intercepts data to control registers 0AH and 0BH (via port 3D5H) before they reach the 6845 controller. The following conversion is performed:

* A scan line specified in the range 0 to 7 affects **two** scan lines: 0 is interpreted as 0 and 1, 1 is interpreted as 2 and 3, ... 7 is interpreted as 14 and 15.

Scan lines 8 to 15 are passed to the controller without intervening conversion. The scan line number specified for the cursor start must not be greater than that specified for the end scan line.

CONTROLLER DATA CONVERSION

In addition to the conversion applied to the character and graphics cursor, display controller conversion is applied to most of the other registers of the 6845 written via ports 3B4H/3B5H. This conversion ensures that display control parameters written by some applications are received in a way that will produce the display intended, even though the particular application may have been designed for a non-NCR monitor. Non-conforming parameters are converted to the respective values given for the NCR **internal** CRTs in this Chapter (that is, the leftmost value, not the value given in parentheses).
The controller registers affected (in addition to the cursor definition registers already discussed), are the registers 0 to 9, selected via port 3B4H and written via port 3B5H. These registers convey parameters setting up display synchronization and skew/interlace modes. These parameters are supplied by the ROM BIOS, and need not normally be disturbed.

The conversion does not apply to the display cards intended for external monitors (K140, K141).

The Keyboard

The keyboard is supplied with those keys already fitted, which have the same markings and the same position for all the country versions implemented. The remaining keys are fitted by the user from a set supplied for a specific version. Figure 8.3 shows the keys already fitted, Figure 8.4 shows the complete keyboards for the various country versions.

Special features of the keyboard are:

* Cursor movement keypad in addition to dual function cursor movement/ numeric keypad
* Tactile point on key 5 of the numeric keypad; deeper finger moulds on F and J keys
* LED indication of status of NumLock and CapsLock keys - LEDs are extinguished at power-up initialization, so that unshifted (not shifted) and cursor movement (not digit) keys are active
* Tilting positions the keyboard in one of two possible planes: 5 and 9 degrees

The keyboard is also available separately as Kit 021. The only significant difference between the standard keyboard and K021 is that the latter uses a slightly modified clock signal and firmware in order to extract optimum performance from rarely used non-NCR hardware enhancements.

Connection to the NCR PERSONAL COMPUTER is via a screened 5-core cable. The cable is approximately 300mm (11.8") in its coiled state, giving a fully extended operating length of 990mm (38.9"). The computer connection end of the cable is a DIN connector. The pin configuration is shown in Figure 8.1.

Pin	Color of wire	to PCB
5	green	5V
4	black	Ground
1	brown	Clock
3	red	Reset
2	white	Data
screen		Boundary line

connector viewed from solder side

Figure 8.1 The Keyboard Connector

The working voltage for the keyboard is 4.75 V - 5.25 V, the maximum power drain is 150 mA. Required signal levels for Clock and Data are minimum 2.4 V (for high) and maximum 0.7 V (for low). Data transmission is clocked as illustrated in Figure 8.2.

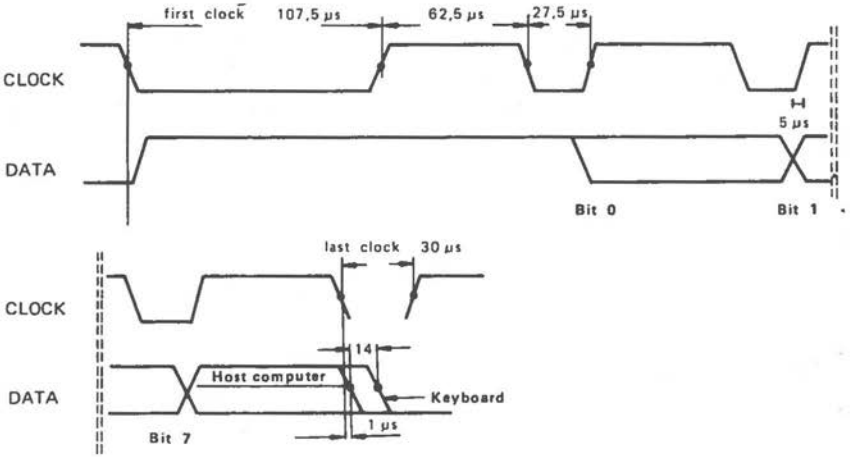


Figure 8.2 Data/Clock Signals

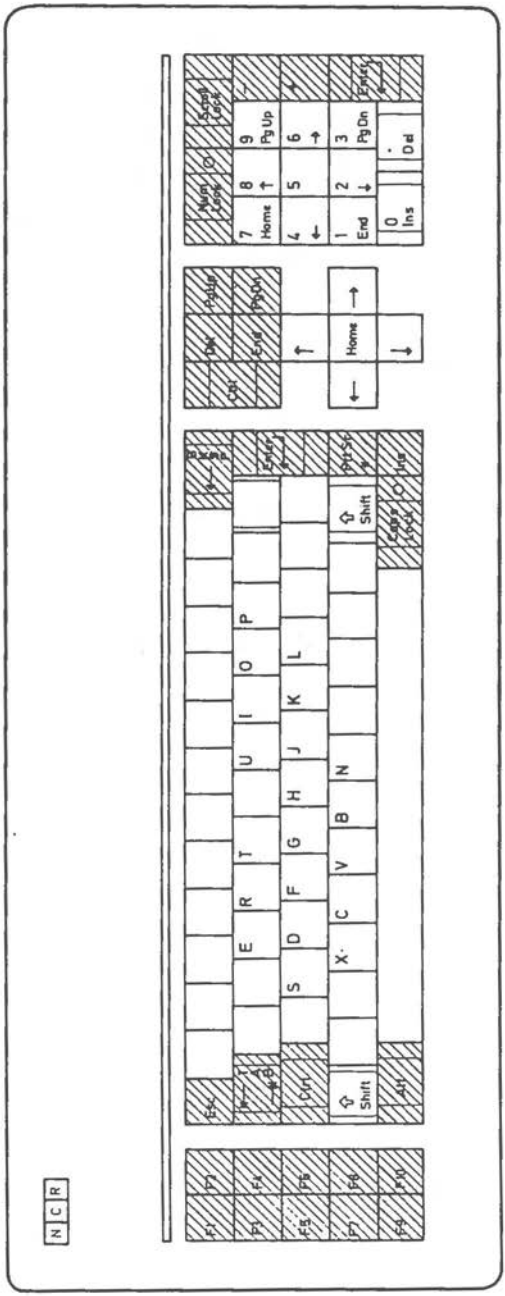


Figure 8.3 Keyboard - Common Keys

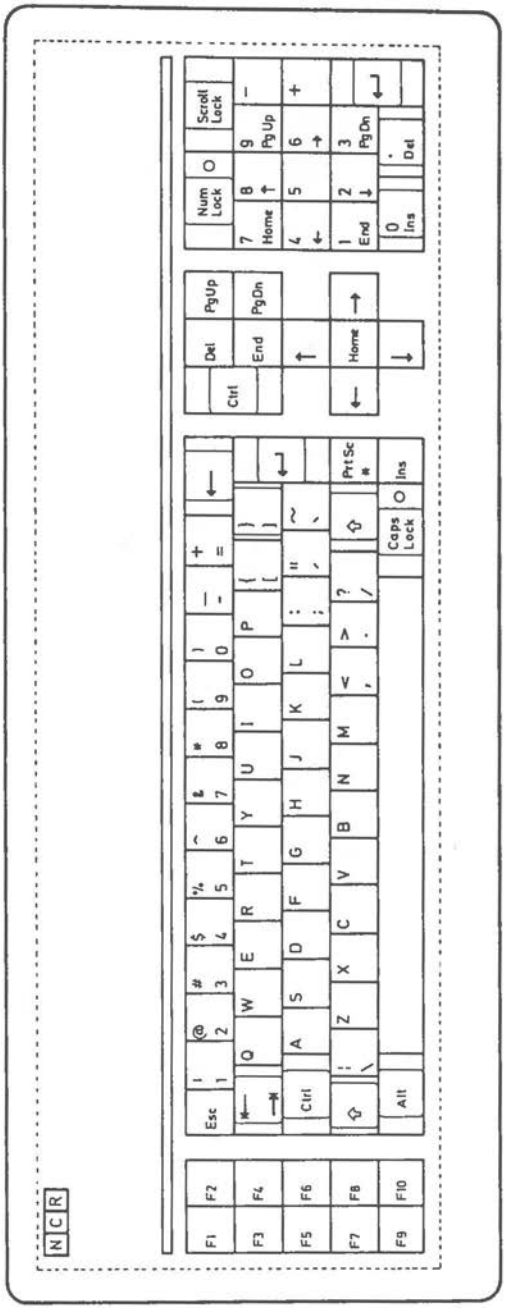


Figure 8.4 (1 of 6) Keyboard – US English

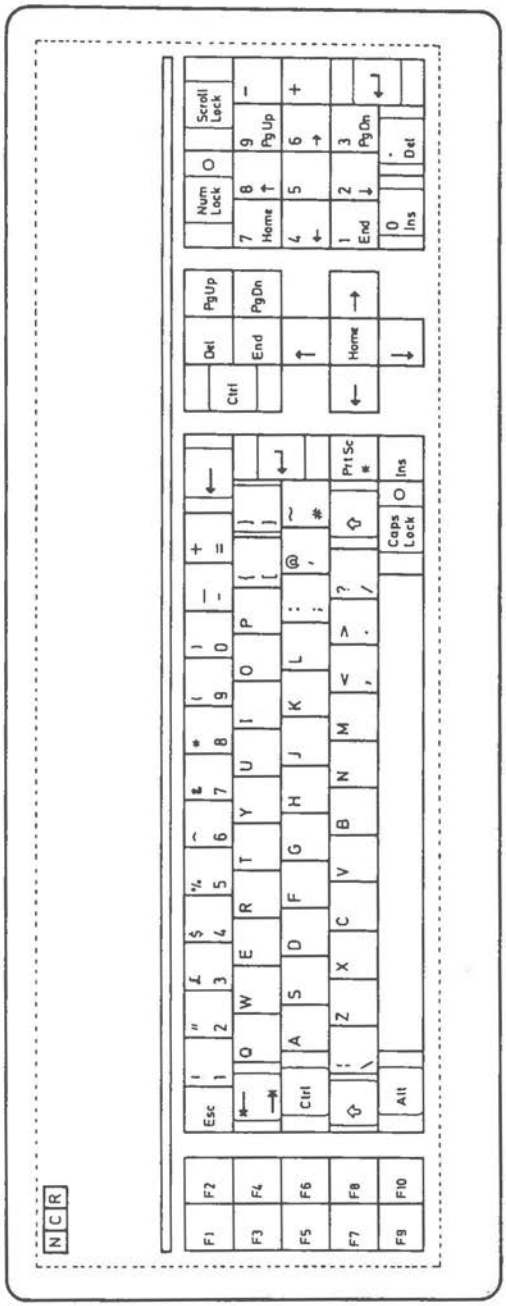


Figure 8.4 (2 of 6) Keyboard – UK English

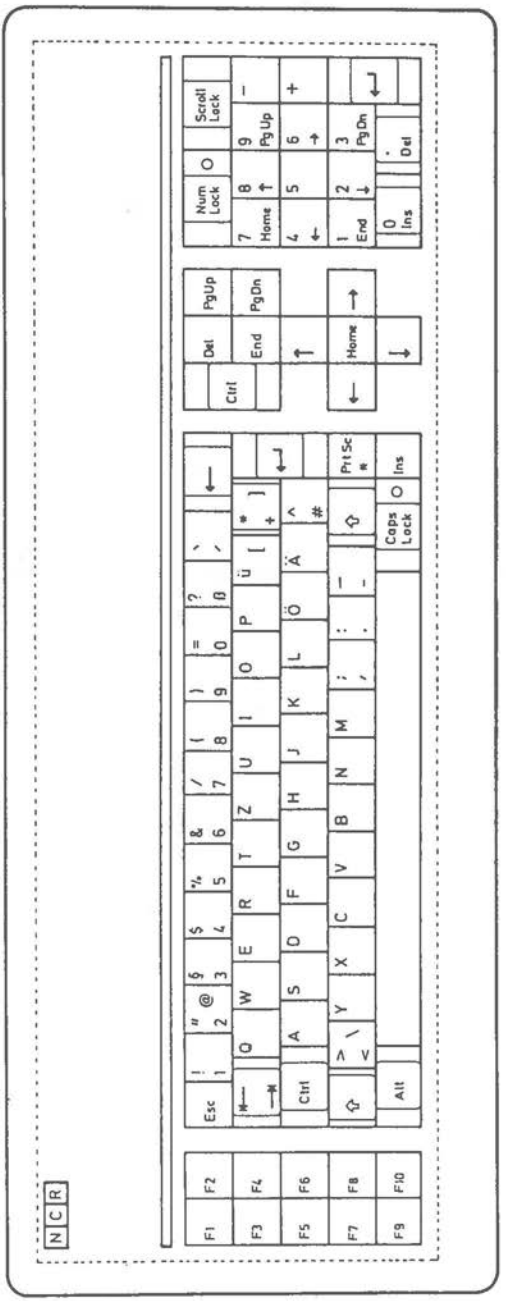


Figure 8.4 (3 of 6) Keyboard - German

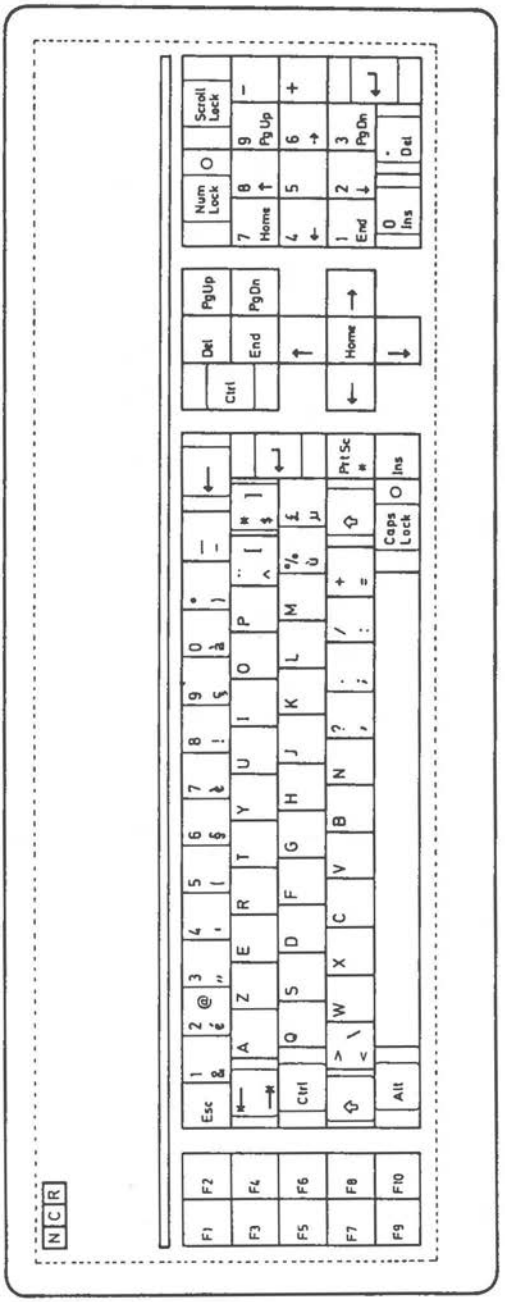


Figure 8.4 [4 of 6] Keyboard - French

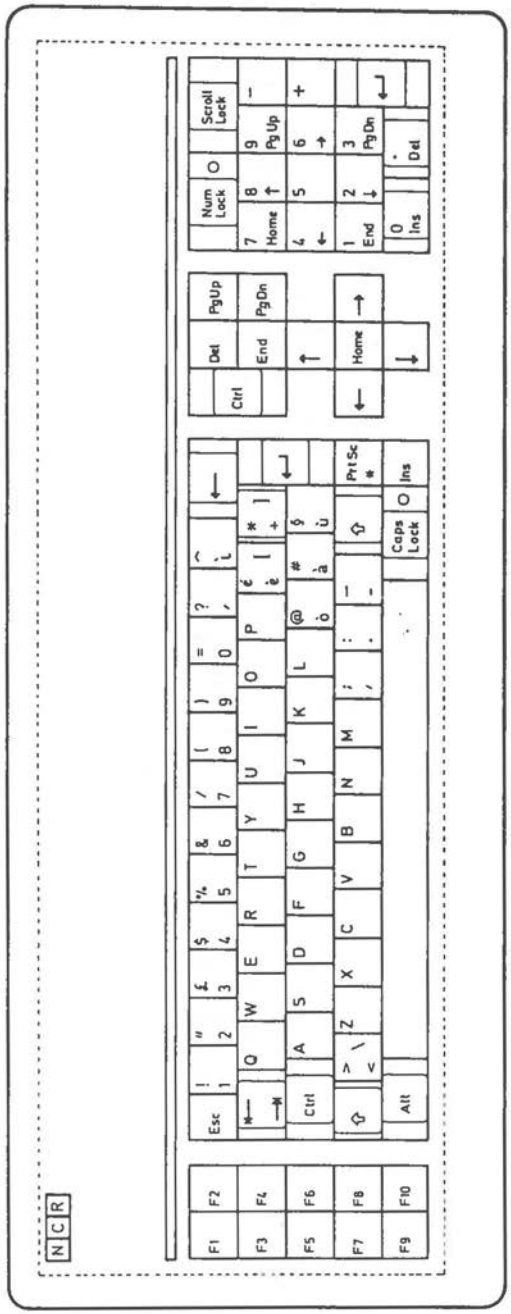


Figure 8.4 (5 of 6) Keyboard - Italian

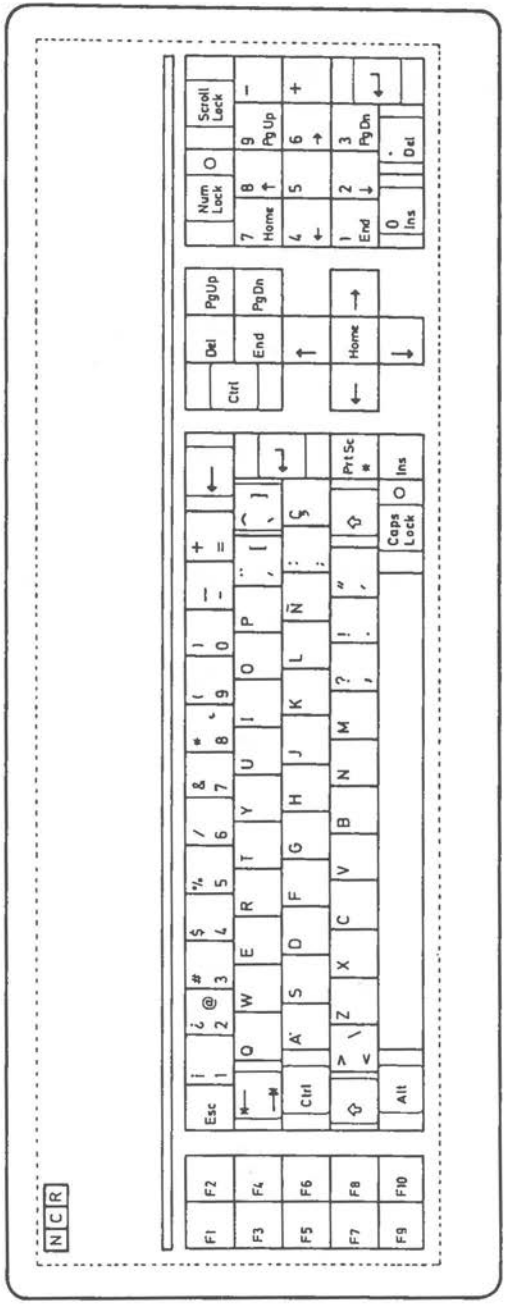


Figure 8.4 (6 of 6) Keyboard - Spanish

KEYBOARD PERFORMANCE

A First In - First Out buffer accomodating up to 32 key codes provides a type-ahead facility. All keys, except NumLock, repeat automatically if depressed for longer than 96 keyboard scan cycles (one scan cycle = 8 ms). The key is then repeated once every eight scan cycles.

Debounce time for the initial depression and release of a key is one scan cycle. Following detection of the initial key depression, that key is checked again after an interval of eight scan cycles.

CODE GENERATION

An 8039 or 8049 microprocessor controls generation of key codes for serial output to the main unit (see Figure 8.5). Note that the code generated represents a specific position on the keyboard; it does not conform to the ASCII code for the sign marked on the specific key. This means that the code returned for a key position is always the same, irrespective of country-dependent keys fitted.

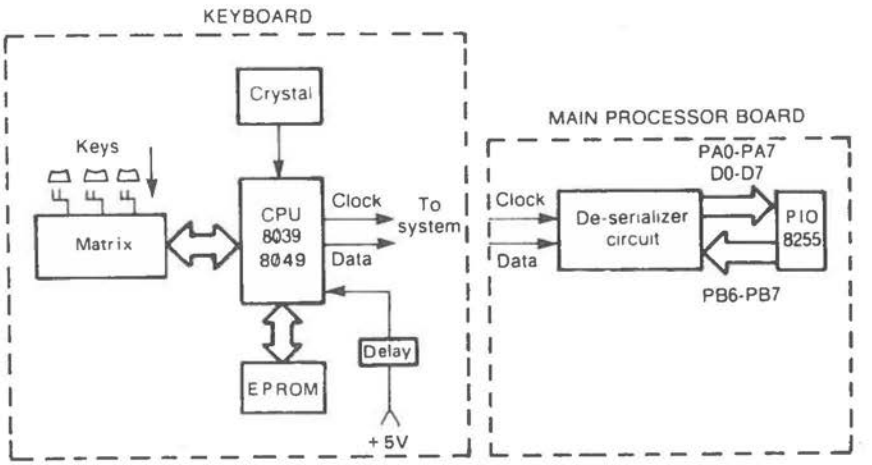


Figure 8.5 Keyboard Logic

Figure 8.6 shows the code generated for each key, and for your orientation, the US English key markings. When the key is depressed, its position code is transmitted (bit 7 is zero). Upon releasing the key this code is transmitted again, with the difference that bit 7 is now set. The only exception is the NumLock key: at key actuation (not release) both key down (45H) and key up (0C5H) codes are generated. It is the concern of the Operating System or application to give significance to this special key action. The same applies to the position codes generated by the keyboard. Figure 8.7 sets out the stages of logic between keyboard scan and code output.

The keyboard read routine contained in the system ROM is activated by hardware interrupt request 1 (CPU interrupt type 9). The priority of this interrupt, as set by the initialization firmware (fully nested mode), is second only to that of the 8253 timer.

The keyboard position code itself is read via the 8255A parallel interface at port 60H. To ensure that the keyboard data input and not the switch status input is selected, bit 7 of the 8255 output port (PB at I/O address 61H) should be transmitted to the 8255 in a reset (0) condition:

```
MOV   DX,61H
IN    AL,DX
AND   AL,7FH
OUT   DX,AL
```

The keyboard position code can then be read:

```
MOV   DX,60H
IN    AL,DX
```

After the position code has been read, this must be notified to the keyboard by transmitting bit 7 set to port 61H (this is also the instruction which selects the main processor board switch information):

```
MOV    DX,61H
IN     AL,DX
OR     AL,80H
```

This clearing command does not affect the keyboard input buffer set up and maintained by the BIOS.

The keyboard hardware interrupt writes the character value of the keyboard position code into the keyboard buffer. There is a software interrupt (1BH) accessible from the interrupt 9 service routine. However, INT 1BH is issued only in the event of the Ctrl-Break key combination. Therefore, if you wish to interrogate the keyboard position code by means of your own software (for example, if you only want to check for a limited number of codes and ignore the keyboard buffer), you will have to write your own keyboard handling routine and set the four interrupt vector bytes starting at 6CH to address your own interrupt service routine. This routine must be concluded by the 8259A Progammable Interrupt Controller non-specific End-of-Interrupt command.

Chapter 2 includes an example of using the Ctrl-Break keyboard interrupt.

DIAGNOSTICS

At power-up the keyboard performs an initialization check of internal RAM and ROM. When no fault is detected after the clock line has been low for longer than 32 ms, 0AAH is transmitted.

Figure 8.6 Keyboard Position Codes

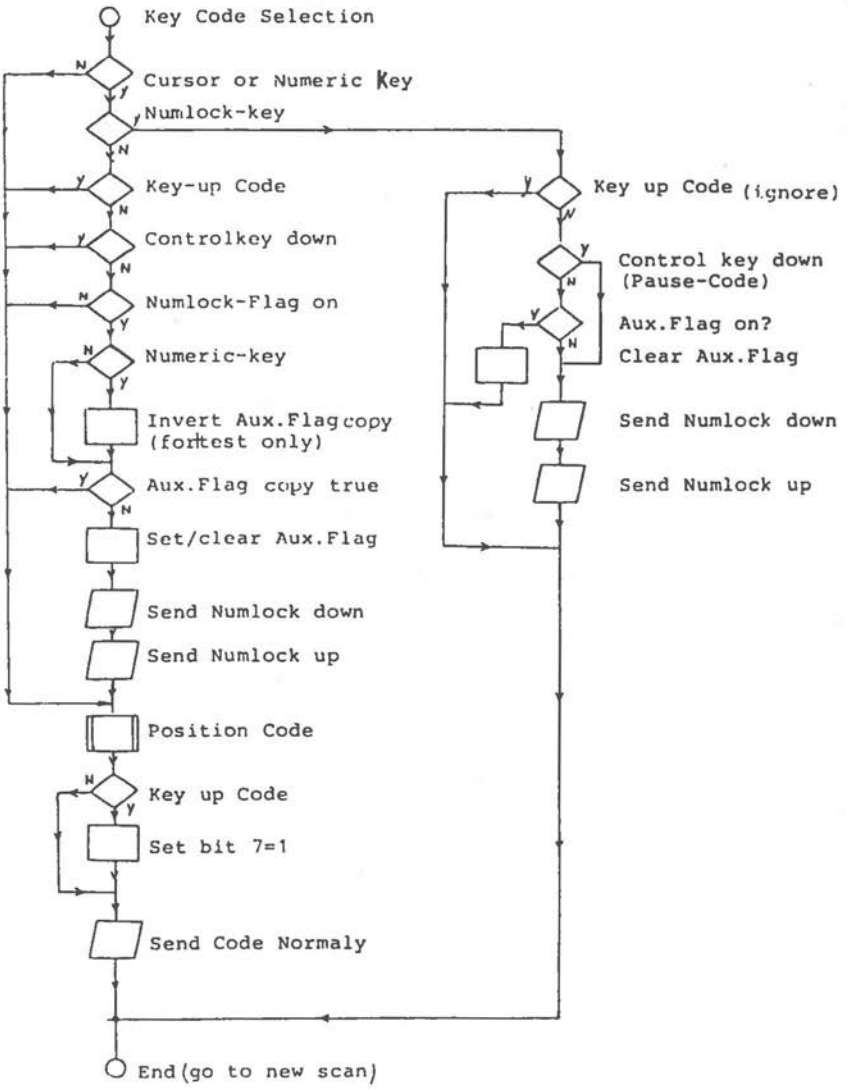


Figure 8.7 Signal Output Logic

ADVANCED KEYBOARD

An advanced keyboard is available as 3299-K420 (US) and 3299-K421 (International). This keyboard is designed not only for the NCR PERSONAL COMPUTER PC4i but also for other equipment in the NCR Personal Computer range. For operation with the NCR PERSONAL COMPUTER PC4i (and compatibles), a small slider switch on the keyboard must be set as shown in Figure 8.8.

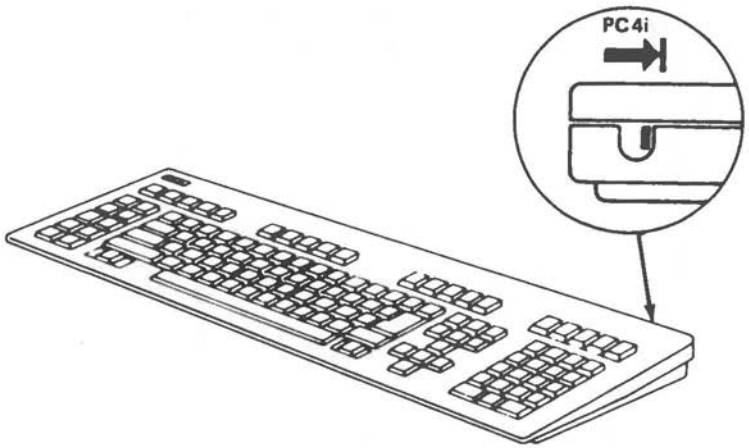


Figure 8.8 Advanced Keyboard Equipment Selection

Possible tilting planes are 5 and 12 degrees.

An LED is present in the ScrollLock key, just like the NumLock and CapsLock LEDs on the standard keyboard.

As on the standard keyboard, all keys repeat automatically with the exception of NumLock: this key is set to a non-repeat power-on default, but this can be changed using a special function. Debounce time for the initial depression and release of a key is one scan cycle (8 ms).

Auto-repeat can be varied locally at the keyboard.

The slowest repeat rate is 2/s, the fastest 30/s, with 32 incremental steps. Initialization default is 10/s.

The initial repeat delay is also variable - between 1/4 s and 1 s in four incremental steps. Initialization default is 1/2 s.

The advanced keyboard includes twenty additional Function Keys F11 - F30.

Special keyboard functions can be accessed while system activity is suspended: following depression of the Control-NumLock combination (= suspension of system activity until another key is pressed), Function Keys F11 - F30 do not issue codes to the system but take on the following functions internal to the keyboard:

F11 toggle to enable/disable autonomous codes for F11 to F30 in normal (non-suspension) mode. As long as autonomous codes are disabled, F11 - F30 provide Shift and Control duplicate functions of F1 - F10:

F11 ... F20 = Shift-F1 ... Shift-F10
F21 ... F30 = Control-F1 ... Control-F10

With autonomous codes enabled, F11 - F30 generate codes in the range 55H to 68H.

This function requires a small modification on some advanced keyboards (see below)

F12 increment auto-repeat rate

F13 decrement " "

F14 increment repeat delay

F15 decrement " "

F17 toggle LED on CapsLock

F18 " " " NumLock

F19 " " " ScrollLock

F20 toggle to exchange 0E and 50 codes transmitted to system (required if exchanging between US/International keyboard use). Initialization default is standard use in the country for which keyboard was provided.

F21 toggle NumLock repeat/non-repeat. Default is non-repeat.

Autonomous codes are active when the LED in the Alt key is illuminated. This Function Key extension can be installed in advanced keyboards which do not already include it: remove diode CR 121 and replace the Alt keytip by one containing an LED window (the LED is already installed in the key mechanism).

Figure 8.9 gives the scan codes for the advanced keyboard. Keyboard layouts are provided in Figures 8.10 and 8.11.

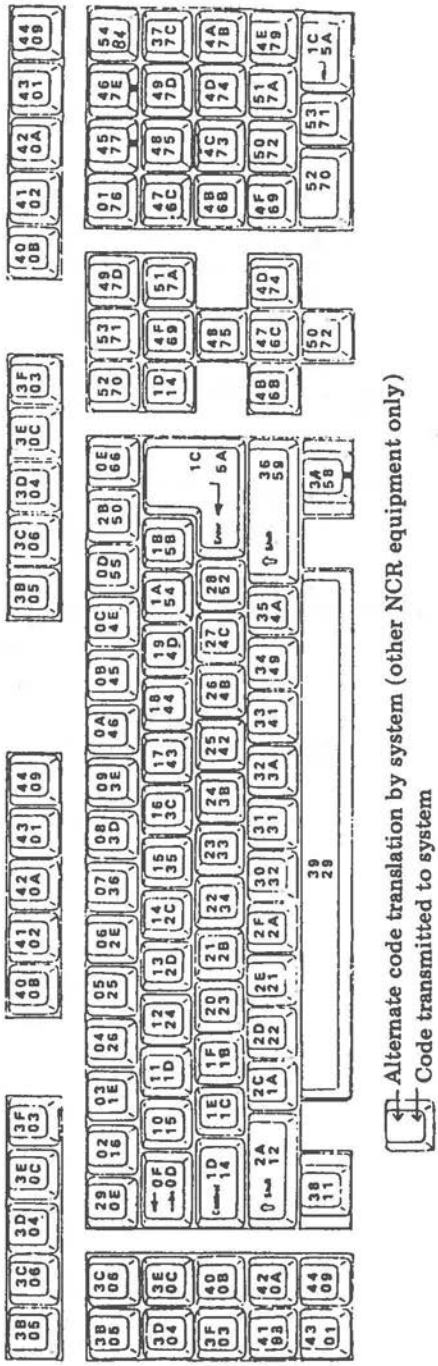


Figure 8.9 Advanced Keyboard Scan Codes

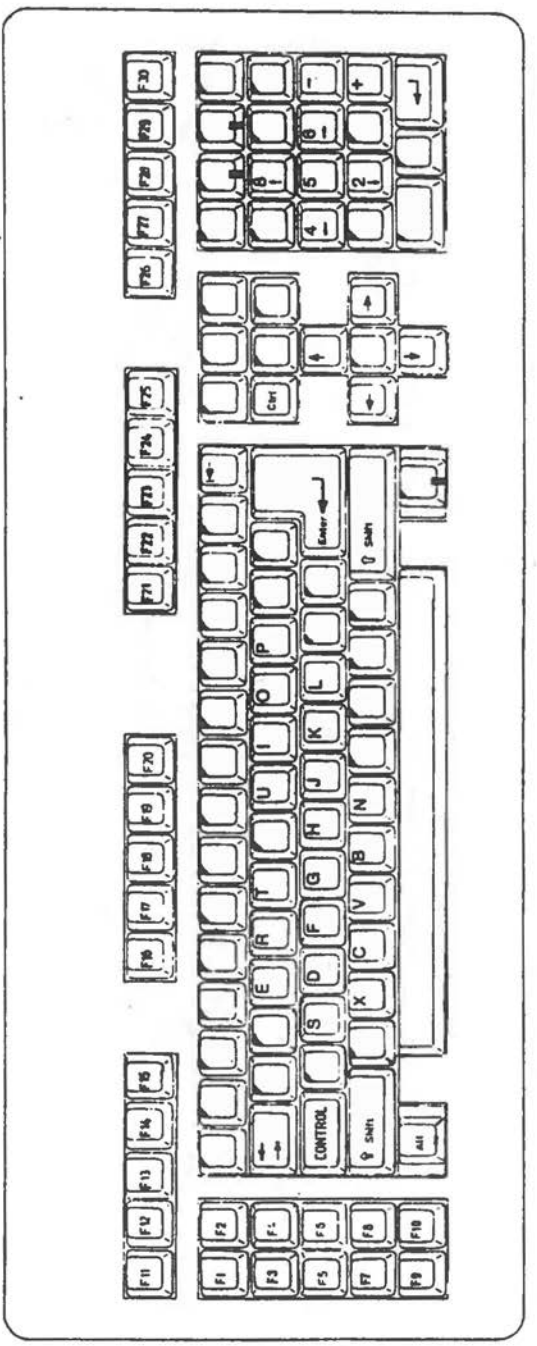


Figure 8.10 Advanced Keyboard – Common Keys

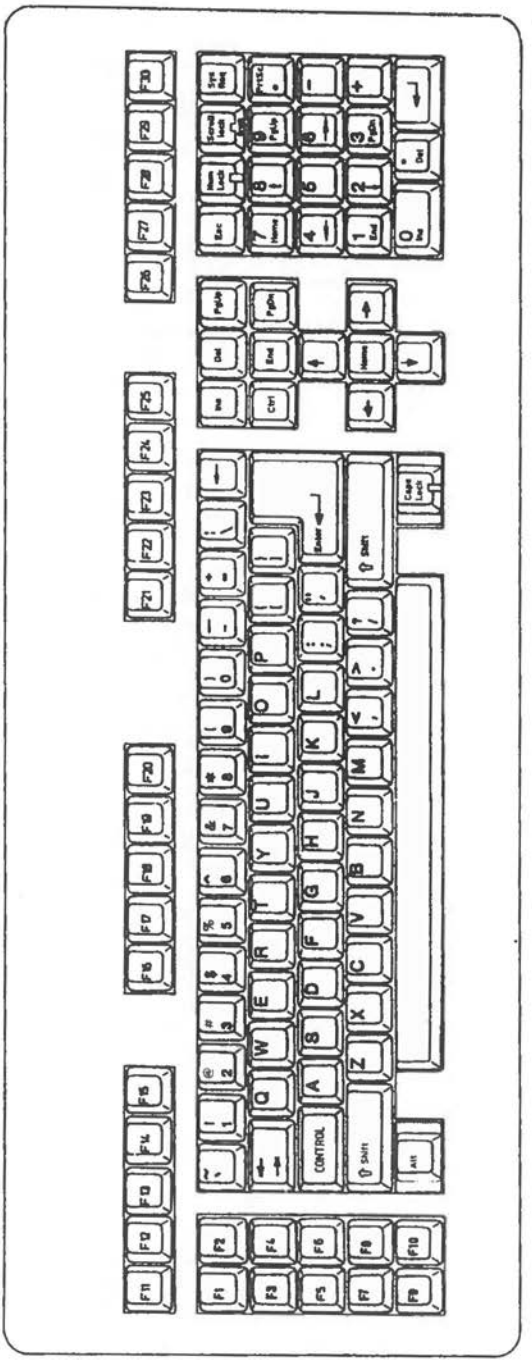


Figure 8.11 (1 of 9) Advanced Keyboard - US English

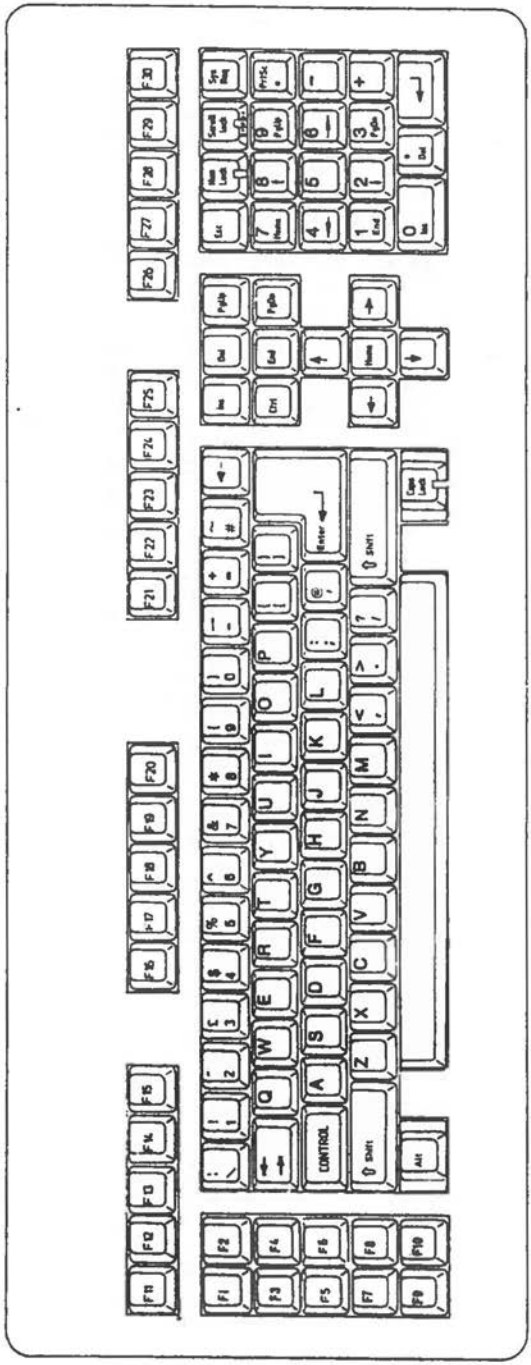


Figure 8.11 (2 of 9) Advanced Keyboard - UK English

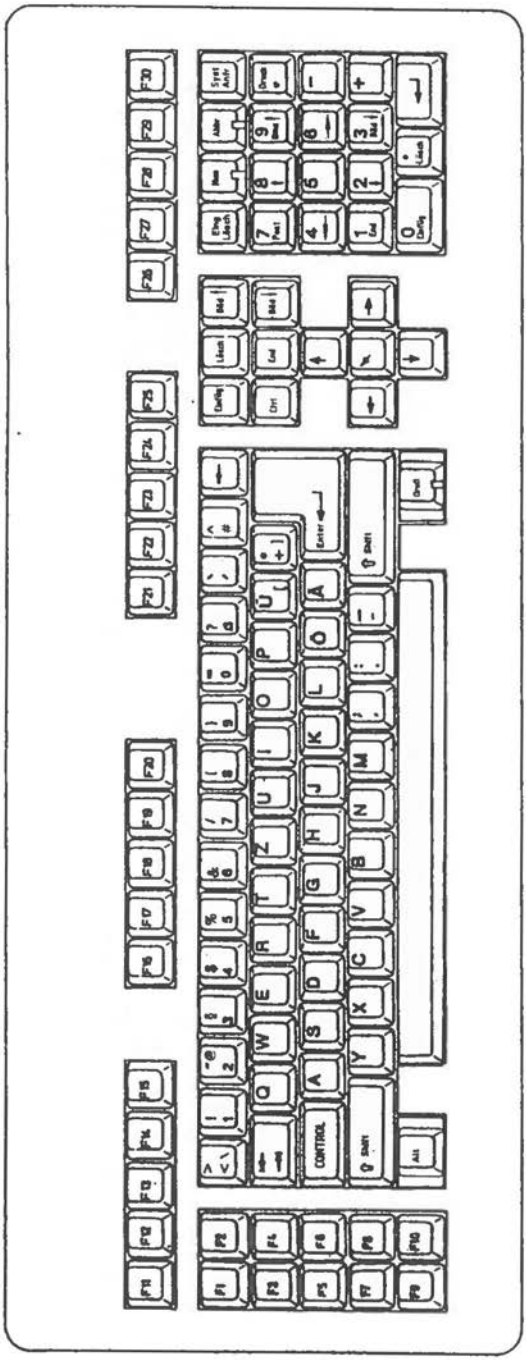


Figure 8.11 (3 of 9) Advanced Keyboard - German

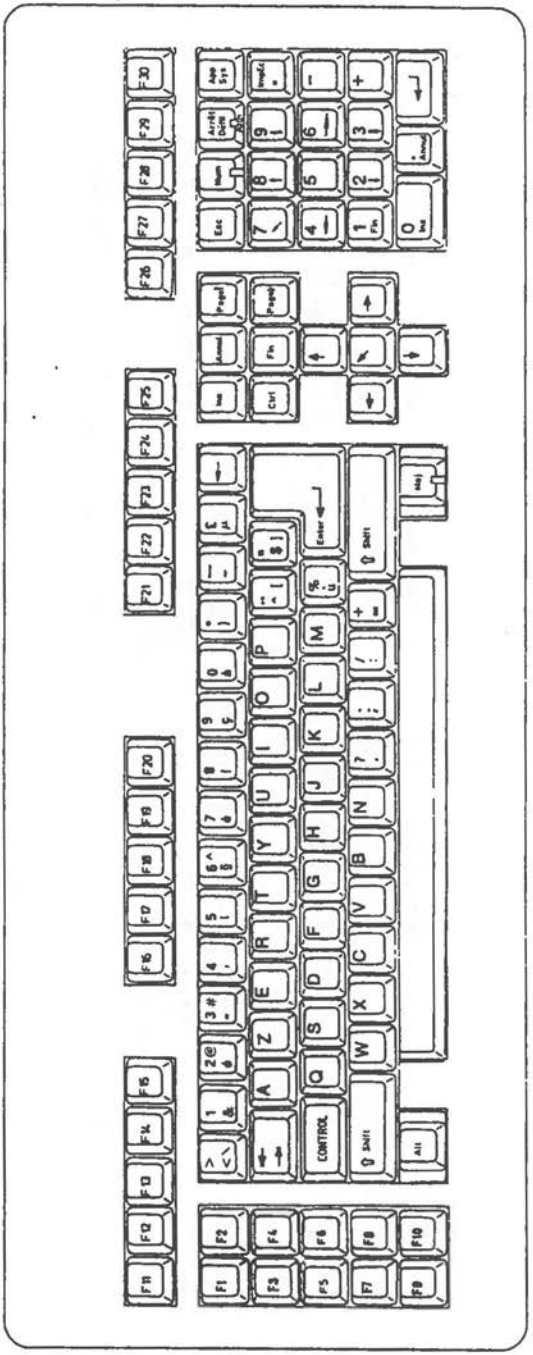


Figure 8.11 (4 of 9) Advanced Keyboard – French

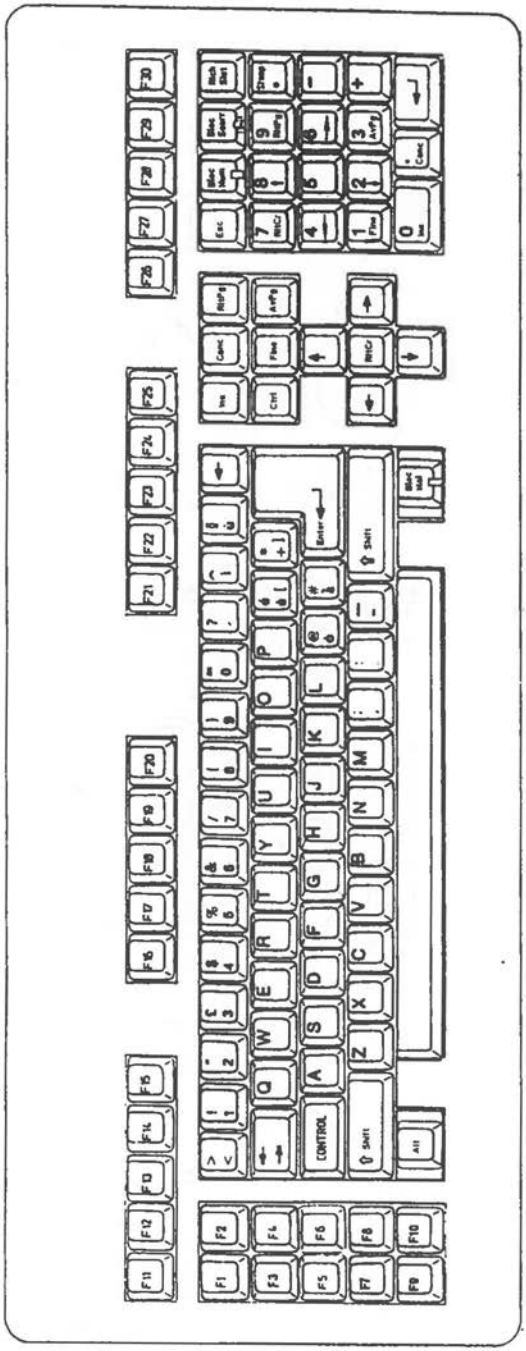


Figure 8.11 (5 of 9) Advanced Keyboard - Italian

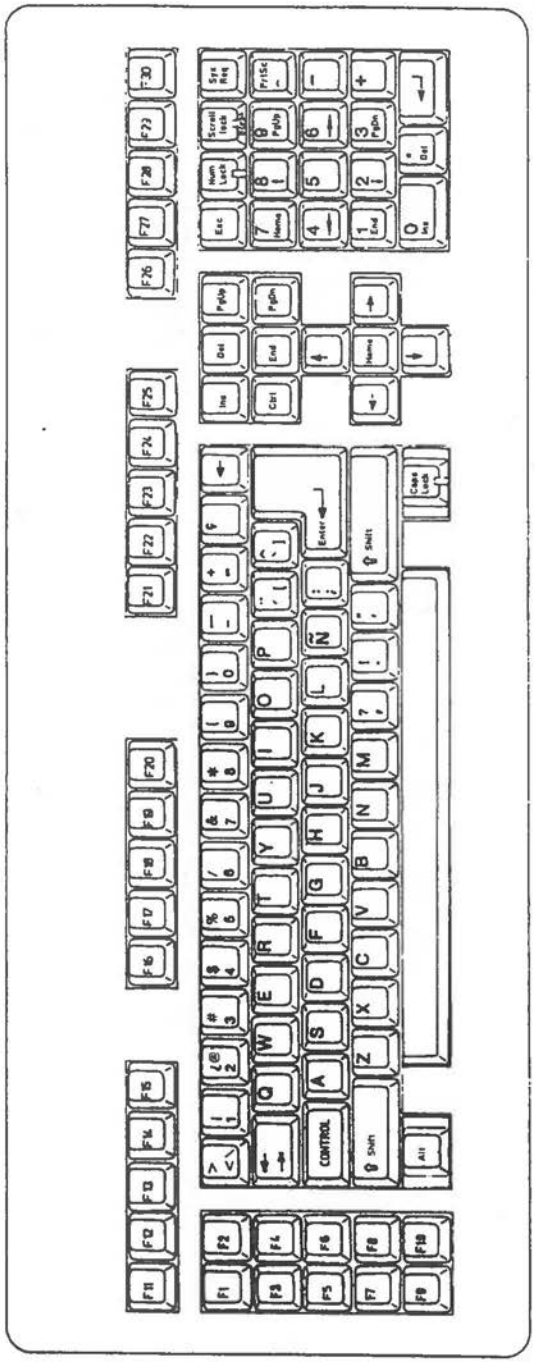


Figure 8.11 (6 of 9) Advanced Keyboard - Spanish

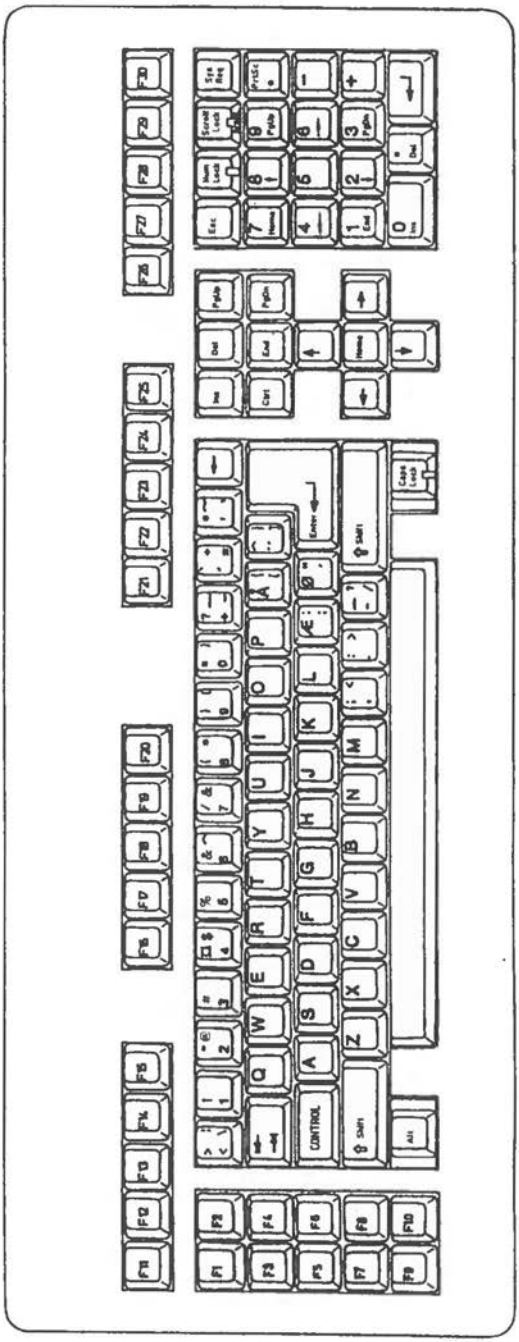


Figure 8.11 (7 of 9) Advanced Keyboard - Danish

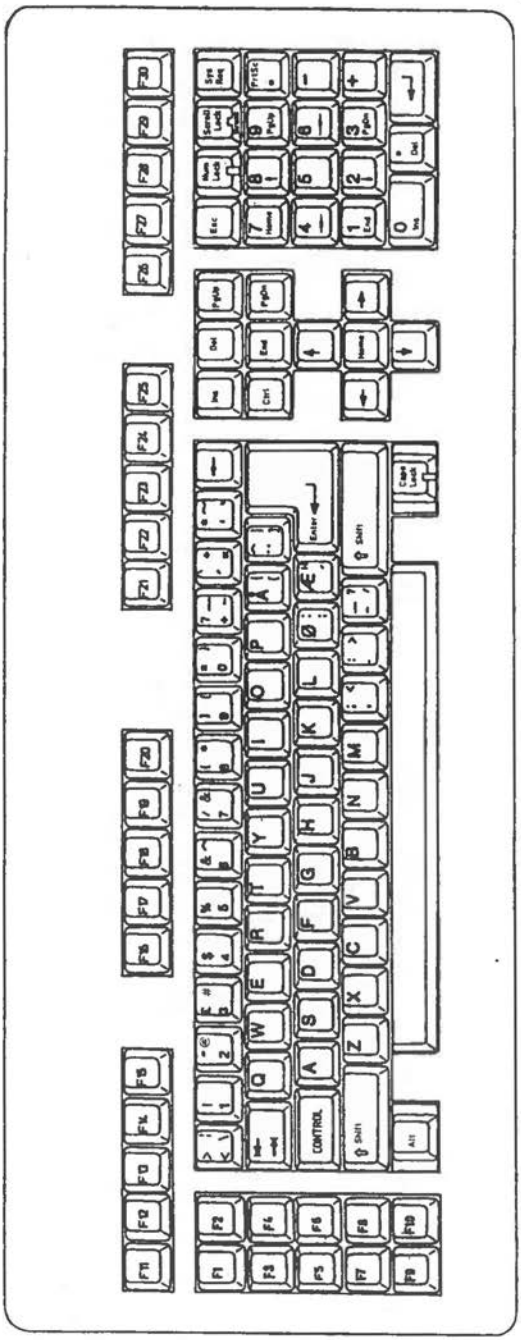


Figure 8.11 (8 of 9) Advanced Keyboard - Norwegian

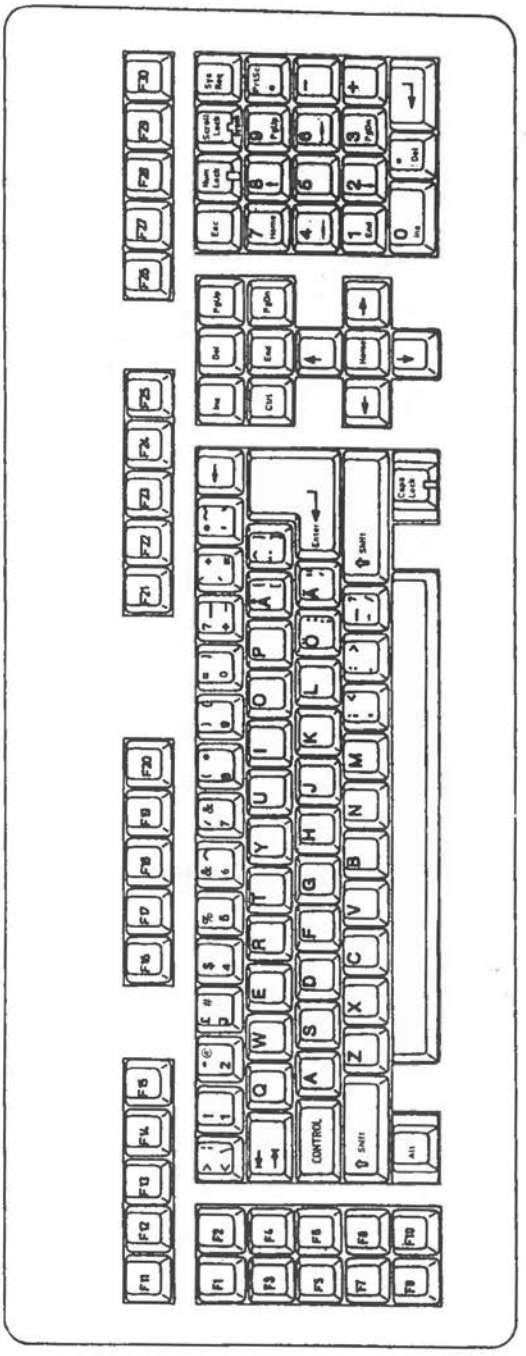


Figure 8.11 (9 of 9) Advanced Keyboard - Swedish/Finnish

The Loudspeaker

HARDWARE CHARACTERISTICS

The NCR PERSONAL COMPUTER MODEL 4i contains a small loudspeaker, mainly used to attract operator attention in application programs. The loudspeaker is driven by a TTL signal amplified by a transistor. The driving signal is derived from output line PB1 of the 8255A-5 integrated circuit used for parallel data I/O (described in Chapter 2). In principle, it is possible to drive this line under direct instruction from the CPU. This has the disadvantage that each signal requires CPU time. It is preferable to use the programmable timer integrated circuit to provide signals at programmed intervals. This method, outlined below, offloads the CPU to the extent that sound effects can take place while other instructions are being processed. (The timer integrated circuit is described in Chapter 2.)

The signal from the parallel input/output IC can be controlled by OUT 2 from the 8253-5 timer. CLK 2 is timed by the same signal as clocks 1 and 2, but is gated separately by means of PB0 from the 8255A. Therefore, the loudspeaker is capable of "driving itself" in a synchronous fashion, using the square wave generator mode of operation (Mode 3) of the timer IC. This mode starts the counter after the rising edge of the trigger input. The OUT signal is high for the first half of the count and low for the second half. Loudspeaker activity can be terminated by software blanking of PB1 of the PIO integrated circuit.

PROGRAMMING SOUND

The examples given in this section show how a variety of sound effects can be achieved by control of the following factors affecting loudspeaker output:

* The frequency of the tone (determined by timer counter 2).
* The duration of the tone.
* The period of silence between tones.
* The number of times a tone, with intervening periods of silence, is repeated.

The following routine sets the timer to provide a square wave frequency and programs the PIO integrated circuit to send signals to the loudspeaker in accordance with this strobing. The duration of the tone and the length of periods of silence between tones are achieved by disabling and re-enabling PB1.

Parameters are accepted by this routine in CPU registers as follows:

DX - number of CLK 2 triggers to be counted before OUT 2 is activated. Because this process is self-repeating (see above) without software intervention, this yields the frequency of the tone independent of CPU workload
DI - the length of time this tone signal may actually be passed to the loudspeaker (CPU LOOP count)
ES - the length of ensuing silence
BX - the number of times this alternation of tone and silence is to be produced

```
SPEAK:    MOV   AL,0B6H     ;initialize timer 2 as follows -
          OUT   43H,AL      ;  access both high and low counter
                            ;  bytes,
                            ;  counter byte values are binary
                            ;  (not BCD),
                            ;  operation mode is 3 (square wave
                            ;  generator).
;
          MOV   AX,DX       ;accept high and low bytes for counter
                            ;from DX.
;
          OUT   42H,AL      ;write byte to counter 2, low ...
```

```
        MOV   AL,AH
        OUT   42H,AL      ; ... then high.
;
AGAIN:  IN    AL,61H      ;read PB0-PB7 from PIO and store in AH.
        MOV   AH,AL
;
        OR    AL,3        ;enable PB1, thus enabling loudspeaker.
        OUT   61H,AL      ;enable PB0 to strobe GATE2 on Timer.
;
        MOV   CX,DI
TONE:   LOOP  TONE        ;wait until count exhausted.
;
        MOV   AL,AH       ;restore PB0-PB7, so that
        OUT   61H,AL      ;tone now turned off.
;
        MOV   CX,ES       ;silence counter  ==> rougher tone.
QUIET:  LOOP  QUIET
;
        DEC   BX
        JNZ   AGAIN       ;repeat tone/silence pattern
                          ;until count exhausted.
;
;
```

The following code creates the sound of a warning siren when repeated continuously. The value in CX sets the count for the duration of silence between units of the same tone, SI sets the incremental value for the next value to be written to the timer. The interplay between silence (ES), tone duration (DI), repetition of this pattern (BX and CX), and frequency change can be used to create interesting sound effects.

```
        MOV   AX,80H
        MOV   ES,AX       ;Length of silence.
        MOV   DI,200H     ;Length of tone.
        MOV   DX,300H     ;Highest tone of glissando.
        MOV   SI,20H      ;Incremental unit in glissando.
DROP:   ADD   DX,SI       ;New, increased value for timer count
                          ;to lower note.
        MOV   BX,6        ;Repetition of tone.
        CALL  SPEAK
```

```
        MOV   CX,1          ;Insignificant delay,
                            ;go on to next tone.
WAIT:   LOOP  WAIT
        CMP   DX,900H       ;Lowest tone (greatest timer
                            ;counter value).
        JB    DROP
```

If you wish to program sounds of a more musical variety, you will find the table of frequencies and their respective notes in musical notation useful (Figure 9.1).

Note	Frequency (Hz)	Note	Frequency (Hz)
A	220	F#	740
A#	233	G	784
B	247	G#	830
C	262	A	880
C#	277.2	A#	930
D	293.6	B	987.8
D#	311.6	C	1046.4
E	329.6	C#	1106
F	349.2	D	1174.6
F#	370	D#	1244
G	392	E	1318.6
G#	416	F	1397
A	440	F#	1480
A#	466	G	1568
B	493.2	G#	1660
Middle C	523.2	A	1760
C#	554.8	A#	1864
D	587.4	B	1975.6
D#	622	C	2093
E	659.2	C#	2217.4
F	698.4	D	2349.4

Figure 9.1 Music Frequencies

To obtain timer counter 2 values to produce the above frequencies, it is necessary to perform jumbo arithmetic: place the frequency in Hz in CX. The value 1234DCH in DX-AX is then divided by CX to yield a 16-bit result in AX. This value is then output to the counter, low byte then high.

The following example plays a note of frequency (Hz) contained in CX for a period of approximately 65536 times BX times 9 processor clocks (9 clocks for each jump in LOOP instruction):

```
        MOV   CX,20BH     ;middle C.
        MOV   BX,0FFH     ;longest period.
        MOV   DX,12H
        MOV   AX,34DCH    ;reference dividend.
        DIV   CX          ;quotient in AX, ignore remainder.
        PUSH  AX
        MOV   AL,0B6H ;initialize timer 2 as follows -
        OUT   43H,AL  ;  access both high and low counter bytes,
                      ;  counter byte values binary (not BCD),
                      ;  mode 3 (square wave generator).
        POP   AX
        OUT   42H,AL      ;write byte to counter 2, low ...
        MOV   AL,AH
        OUT   42H,AL      ; ... then high.
        IN    AL,61H      ;read PB0-PB7 from PIO and store in AH.
        MOV   AH,AL
        OR    AL,3        ;enable PB1, thus enabling loudspeaker;
        OUT   61H,AL      ;enable PB0 to strobe GATE2 on Timer.
        MOV   CX,BX
PLAY1:  PUSH  CX
        MOV   CX,0FFFFH
PLAY2:  LOOP  PLAY2
        POP   CX
        LOOP  PLAY1       ;wait until count exhausted.
        MOV   AL,AH       ;restore PB0-PB7, so that
        OUT   61H,AL      ;tone now turned off.
```

The program which now follows enables you to use an easy to understand music programming language, thus avoiding the need to perform individual calculations for note and length. The program inspects the string

contained at TUNE and ending with a $, interpreting the following elements as music commands. Each element is preceded by a comma, except the terminating $. Optional information is given in [].

,[n][+ or -]N[# or b][.]

Play the note N natural, N sharp or N flat. N stands for a note in the range A to G.
+ or - specifies that the note is to be played one octave higher (+) or lower (-) than the currently selected octave.
The single digit n specifies a duration of note n times the currently selected metronome beat.
One or more full stops may follow note specification. A full stop results in the note not being played in its full length. Instead, the end of the note is actually transmitted to the loudspeaker output routine as silence (OFFFFH). This prevents notes from running on directly, thus creating a staccato effect. The more full stops given, the more pronounced the staccato effect.

,+ or - Select a new octave one octave above (+) or below (-) the current octave. Notes are now played in the new octave until the next octave command.

,M+ Increase metronome beat by one minimum beat unit. The actual length of this unit can be varied by adjusting the value of the inner CX loop in the MUSIC routine.

,M- Decrease metronome beat by one minimum beat unit.

,S+ Increase base staccato (audible note truncation) factor by one minimum beat unit.

,S-	Decrease base staccato (audible note truncation) factor by one minimum beat unit.
,K+n ,K-n	Subsequent notes are played n (single digit) semitones higher (+) or lower (-) than their their nominal pitch. This new 'key' remains in force until the next key command.
,[n]Q	Determines a period of silence of length n (default 1) times the current metronome beat.
$	End of music string.

Illegal syntax and out of range values result in end of range values (note = silence, length = maximum) or are ignored.

The EQUate BEAT sets the metronome beat rate as a multiple of the minimum beat unit. A digit at the beginning of a note specifying command results in the note being played for a corresponding length of time. For example, if BEAT specifies 16, the music command

```
,4C
```

plays a C for 64 minimum beat units.

The EQUate BREAK determines the base element used for note truncation (staccato) in minimum beat units. For example, if BEAT is 16 and BREAK is set to 2, the command

```
,C..
```

means that C will be audible for 12 minimum beat units followed by 4 units of silence.

The loudspeaker driver part of the program is contained in MUSIC. The music interpreter occupies the rest of the program.

```
BEAT      EQU   16      ;Base metronome beat * minimum beat
                        ;used by MUSIC.
BREAK     EQU   2       ;Base unit for note truncation.
                        ;EQUs are example values.
;
TIMER     EQU   40H     ;Timer base port.
PORTB     EQU   61H     ;PB0-PB7 on PIO interface.
;
          CSEG  SEGMENT
          ASSUME CS:CSEG,DS:CSEG
;
          ORG   100H
          PUSH  CS
          POP   DS
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                      ;;
;;        Initialize octave, key, and beat frequencies                  ;;
;;                                                                      ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
READTUNE: MOV   SI,OFFSET TUNE      ;<— START at this instruction.
          MOV   BX,OFFSET A2
          MOV   OCTAVE,BX     ;Therefore, notes A-G are in this
                              ;octave until further octave command.
          MOV   KEYDISP,0     ;No change of key.
          MOV   AL,BEAT
          MOV   METRO,AL      ;One metronome beat = AL * min. beat.
          MOV   AL,BREAK
          MOV   TRUNCATE,AL   ;Base unit of note truncation =
                              ;AL * minimum beat: staccato effect.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                      ;;
;;        Check for end of music ($)                                    ;;
;;                                                                      ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
ENDCHK:   MOV   AL,[SI]       ;One character of tune string.
          CMP   AL,'$'        ;Check for end of music completion.
          JNE   IDMUSIC
          JMP   ENDTUNE       ;Intra-segment (too far for SHORT)
;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                             ;;
;;        Identify beginning of music item (,)                 ;;
;;                                                             ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
IDMUSIC:  MOV   AL,[SI]
          INC   SI              ;SI now points to the char. following
                                ;the one currently in AL.
          CMP   AL,','          ;Comma separates music items.
          JNE   IDMUSIC         ;Continue search for separator.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                             ;;
;;        Set default values for note and length               ;;
;;                                                             ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
          MOV   CL,0H           ;Default for note is silence.
          MOV   CH,METRO        ;Length default = metro * min. beat.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                             ;;
;;        Parse string for 'key' command: Kn(+ or -)           ;;
;;        and, if found, return to IDMUSIC to look for next item;;
;;                                                             ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
KEY:      MOV   AL,[SI]
          INC   SI
          CMP   AL,'K'          ;Check for change of key.
          JNE   TEMPO           ;Jump if no key command.
          MOV   AL,[SI]         ;This char. must be digit 1-9.
          CMP   AL,'1'
          JB    ENDCHK          ;Syntax error,
                                ;look for $ or next comma.
          CMP   AL,'9'
          JA    ENDCHK          ;Syntax error, ...
          SUB   AL,30H          ;Convert ASCII to binary number.
          MOV   AH,AL           ;No. of semitones for key change.
          INC   SI              ;Valid key command so far,
                                ;so point to next char.
          MOV   AL,[SI]         ;This char must be + or -.
          CMP   AL,'+'          ;Specifies key AH semitones
```

```
                        ;above current key.
        JNE   NOKEYUP
        ADD   KEYDISP,AH
        INC   SI            ;Key command was valid,
                            ;so point to next char.
        JMP   IDMUSIC       ;[SI] should be comma.
NOKEYUP: CMP  AL,'-'        ;Specify key AH semitones
                            ;below current key.
        JNE   ENDCHK    ;Syntax error, look for $ or next comma.
        NEG   AH        ;2's complement for negative number.
        ADD   KEYDISP,AH
        INC   SI            ;Key command was valid, ...
        JMP   IDMUSIC       ;[SI] should be comma.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                          ;;
;;    Parse string for one of                                               ;;
;;      increment/decrement metronome beat: M[+ or -]                       ;;
;;      increment/decrement staccato factor: S[+ or -]                      ;;
;;        [one incremental/decremental unit = minimum beat]                 ;;
;;    If found, return to search for next music item                        ;;
;;                                                                          ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
TEMPO:    CMP   AL,'M'
          JNE   TEMPO2      ;Jump if no change to metronome beat.
          MOV   AL,[SI]     ;This char. must be + or -.
          CMP   AL,'+'
          JNE   TEMPO11     ;No metronome increment.
          INC   METRO       ;Store increased metronome speed.
          INC   SI          ;Metronome command was valid,
                            ;so point to next char.
          JMP   IDMUSIC     ;[SI] should be comma.
TEMPO11:  CMP   AL,'-'
          JNE   ENDCHK  ;Syntax error, look for $ or next comma.
          DEC   METRO   ;Store decreased metronome speed.
          INC   SI      ;Metronome command was valid, ...
          JMP   IDMUSIC ;[SI] should be comma.
;
;Command S increments[+] or decrements[-] the base unit for
;the period of silence subtracted from the nominal length of a
;note. This period = TRUNCATE * minimum beat * no. of full stops
;immediately following note [see SPACE].
;
```

```
TEMPO2:   CMP   AL,'S'
          JNE   LENGTH0  ;Jump if no change to truncation factor.
          MOV   AL,[SI]  ;This char. must be + or -.
          CMP   AL,'+'
          JNE   TEMPO21  ;No increase in length of truncation.
          INC   TRUNCATE ;Store new increased truncation factor.
          INC   SI       ;Truncate command was valid, so point to
                         ;next character.
          JMP   IDMUSIC  ;[SI] should be comma.
TEMPO21:  CMP   AL,'-'
          JNE   ENDCHK   ;Syntax error, ...
          DEC   TRUNCATE ;Store new reduced truncation factor.
          INC   SI       ;Truncate command was valid, ...
          JMP   IDMUSIC  ;[SI] should be comma.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                          ;;
;;    Parse string, looking for a note/length/octave command:               ;;
;;          [n][+ or -]N[# or b][one or more .]                             ;;
;;                                                                          ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;Check for digit specifying length of note.
;Length = digit[n] * METRO.
;
LENGTH0:  CMP   AL,'1'
          JB    OCTVCHK      ;Jump if length not specified.
          CMP   AL,'9'
          JA    OCTVCHK      ;Jump if length not specified.
          SUB   AL,30H       ;Convert ASCII to binary number.
          MOV   BL,METRO     ;Metronome unit.
          XOR   AH,AH    ;Zero AH to enable check for AL overflow.
          MUL   BL       ;Length is AL * minimum beat.
          MOV   CH,AL    ;Length for speaker routine now in CH,
                         ;subject to following legality check.
          AND   AH,0FFH  ;Result should be zero, otherwise AL
                         ;overflowed during calculation of length
                         ;(max. length accounted for is 0FFH).
          JZ    LENGTH1
          MOV   CH,0FFH  ;If length illegal, set it to maximum.
LENGTH1:  MOV   AL,[SI]  ;Length set, so fetch next char.
          INC   SI
;
;
```

```
;A single + or - may precede the actual note, thus raising or
;lowering the note by one octave. If the + or - is a separate
;music item (i.e. followed immediately by a comma), the change
;of octave applies until a further such command; otherwise,
;the raised/lowered octave applies to the subsequent note only.
;
OCTVCHK:  MOV   BX,OCTAVE  ;Get current base note (nominally A).
          CMP   AL,'+'
          JNE   OCTVCHK1
          ADD   BX,7       ;Number of notes natural
                           ;to attain next octave.
                           ;NOTE: There is no range checking.
                           ;Illegal notes are detected in RDNOTE.
          JMP   OCTVCHK2
OCTVCHK1: CMP   AL,'-'
          JNE   RDNOTE     ;Jump if no octave command.
          SUB   BX,7
OCTVCHK2: MOV   AL,[SI]    ;Octave command detected,
                           ;so fetch next char.
          INC   SI
          CMP   AL,','
          JNE   RDNOTE     ;Jump if octave change only temporary.
          MOV   OCTAVE,BX  ;Otherwise, store new base note.
          DEC   SI         ;So that MUSICID detects this comma.
          JMP   IDMUSIC
;
;Look for a note A-G, optionally followed by a # (sharp)
;or b (flat). If none is found, the default 'silence' will be
;sent to the speaker routine.
;Q, instead of A-G, specifies silence.
;
RDNOTE:   CMP   AL,'A'
          JB    SPACE        ;Jump if no note (illegal syntax).
          CMP   AL,'Q'       ;Q specifies silence instead of note.
          JE    QUIET
          CMP   AL,'G'
          JA    SPACE
          SUB   AL,41H       ;Note A-G represents +ve displacement
                             ;0 to 6 to base note in BX.
          XOR   AH,AH
          ADD   BX,AX        ;[BX] is natural note in FREQTAB.
          MOV   CL,[BX]
          MOV   AL,[SI]      ;Valid note detected,
                             ;so fetch next character.
```

```
          INC   SI
          CMP   AL,'#'        ;Look for sharp suffix.
          JNE   NOTSHARP
          INC   CL            ;One semitone for sharp.
          MOV   AL,[SI]       ;Valid note suffix #,
                              ;so fetch next character.
          INC   SI
          JMP   NEWKEY
NOTSHARP: CMP   AL,'b'        ;For keys requiring flat notation.
          JNE   NOTFLAT
          DEC   CL            ;One semitone for flat.
          MOV   AL,[SI]       ;Valid note suffix b, ...
          INC   SI
          JMP   NEWKEY
QUIET:    MOV   AL,[SI]       ;CL already contains default value.
          INC   SI
          JMP   SPACE
;
;
;An experimental semitone offset specified by the 'key' command
;affects the note in CL.
;Check that note is within legal range 1 - 2AH
;
NOTFLAT:
NEWKEY:   MOV   AH,KEYDISP    ;KEYDISP zero,
                              ;unless set by key command.
          ADD   AH,CL         ;AL is 2's complement for -ve value.
NOTECHK:  CMP   AH,1
          JB    OUTRANGE
          CMP   AH,2AH
          JNA   NOTECHKD
OUTRANGE: MOV   AH,0          ;Specify silence if note out of range.
NOTECHKD: MOV   CL,AH         ;Note finally in CL ready for
                              ;XLATion to loudsp. routine value.
;
;Count the number of full stops. For each one counted, the last
;TRUNCATE * minimum beat of the note will be silenced. This
;prevents note from running on the next one.
;
SPACE:    XOR   DH,DH
SPACE1:   CMP   AL,'.'
          JNE   LASTCHAR      ;Jump if music command complete.
          INC   DH
          SUB   CH,TRUNCATE   ;Reduce length of note accordingly.
```

```
          CMP   CH,1
          JB    SPACE2          ;Jump if length of truncation exceeds
                                ;length of note.
          MOV   AL,[SI]
          INC   SI
          JMP   SPACE1
SPACE2:   MOV   CH,0FFH         ;Sets maximum length if note
                                ;truncation was illegal.
          XOR   DH,DH
;
;Prepare command word in DX, specifying length of
;silence at the end of the audible part of the note.
;
LASTCHAR: MOV   AL,DH           ;No. of full stops (0 if none).
          MUL   TRUNCATE        ;Yields length of silence/min. beat
                                ;in AL.
          MOV   DH,AL
          MOV   DL,0            ;'Silence' length and note now in DX.
;
;One or two word pairs are now written to COMPTUNE. The second
;word pair (expanded from DX) is not written if the note is to
;be audible in its entire nominal length.
;
TOSPK:    PUSH  BX              ;BX now needed as XLAT pointer.
          PUSH  CX
          MOV   BX,OFFSET HERZTAB
          MOV   AL,CL           ;Intermediate note code to AL.
          SHL   AL,1            ;Double AL.
          PUSH  AX
          XLAT  BYTE PTR HERZTAB ;Look up low byte of note in Hz.
          MOV   CL,AL
          POP   AX
          INC   AX
          XLAT  BYTE PTR HERZTAB ;Look up high byte of note in Hz.
          MOV   CH,AL           ;(Length of note is still on stack.)
          POP   BX              ;Value was PUSHed as CX.
          MOV   BL,BH
          XOR   BH,BH           ;Length, but not note, now in BX.
          PUSH  DX              ;Contains possible staccato info.
          CALL  MUSIC
          POP   DX
          CMP   DH,0            ;Check for staccato.
          JE    PLAYED          ;Jump if note to be played in its
                                ;entire nominal length.
```

```
            MOV   CX,0FFFFH     ;Inaudible frequency for non-audible
                                ;end of note, if any.
            MOV   BL,DH
            XOR   BH,BH         ;Length of truncating silence in BX.
            CALL  MUSIC
PLAYED:     DEC   SI            ;[SI] should be comma or $.
            POP   BX
            JMP   ENDCHK
;
ENDTUNE:    NOP                 ;<-- BREAK after this instruction.
;
;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;                                                       ;
                ; Play note of specified frequency for                 ;
                ; specified length.                                    ;
                ; Entry:                                               ;
                ;   Frequency [Hz] in CX                               ;
                ;   Length in BX                                       ;
                ;                                                       ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
MUSIC:      MOV   DX,12H
            MOV   AX,34DCH      ;Reference dividend in DX-AX.
            DIV   CX            ;Quotient in AX, ignore remainder.
;
            PUSH  AX
            MOV   AL,0B6H       ;Initialize Timer 2 as follows -
            OUT   TIMER+3,AL    ;  access both high and low counter
                                ;  bytes,
                                ;  counter byte values are binary
                                ;  [not BCD],
                                ;  operation mode is 3 [square
                                ;  wave generator].
            POP   AX
            OUT   TIMER+2,AL    ;Write byte to Timer 2, low ...
            MOV   AL,AH
            OUT   TIMER+2,AL    ; ... then high.
;
            IN    AL,PORTB      ;Read PB0-PB7 from PIO and
                                ;store in AH.
            MOV   AH,AL
;
            OR    AL,3          ;Enable PB1, thus enabling
```

```
            OUT   PORTB,AL      ;loudspeaker; enable PB0 to strobe
                                ;Gate 2 on Timer.

;
            MOV   CX,BX         ;Now wait for BX*CX loops
                                ;before turning off speaker.
PLAY1:      PUSH  CX
            MOV   CX,1800H      ;Minimum beat (sample value).
PLAY2:      LOOP  PLAY2
            POP   CX
            LOOP  PLAY1         ;Wait until outer count exhausted.
;
            MOV   AL,AH         ;Restore PB0-PB7, so that
            OUT   PORTB,AL      ;speaker now turned off.
;
            RET
;
;
;FREQTAB:
A1          DB    1,3,4,6,8,9,0BH
A2          DB    0DH,0FH,10H,12H,14H,15H,17H
A3          DB    19H,1BH,1CH,1EH,20H,21H,23H
            DB    25H,27H,28H,2AH       ;Intermediate frequency
                                        ;(natural notes).
;
HERZTAB     DW    0FFFFH                ;Effectively silence.
HA1         DW    220,233,247,262,277,294,312,330,349,370,392,416
HA2         DW    440,466,493,523,555,587,622,659,698,740,784,830
HA3         DW    880,930,988,1046,1108,1175,1244,1317,1397,1480
            DW    1568,1660
HA4         DW    1760,1864,1976,2093,2217,2349
;
OCTAVE      DW    0           ;Will be set to address A2.
METRO       DB    0           ;Will be set to base metronome beat.
TRUNCATE    DB    0           ;Will be set to base truncation factor.
KEYDISP     DB    0           ;Used for key experimenting.
;
TUNE        DB    ',C.,D.,E.,F.,G.,G.,G.,G.,+A.,F.,+C.,+A.,3G...,Q,'
            DB    '+,+A.,F.,+C.,+A.,3G...,-,Q,'
            DB    'G.,F.,F.,F.,F.,E.,E.,E.,E.,D.,E.,D.,C.,E.,G.,Q,'
            DB    'G.,F.,F.,F.,F.,E.,E.,E.,E.,D.,E.,D.,2C,Q,Q,Q,Q,'
;
            DB    'K4-,'        ;Same tune, but 4 semitones lower,
            DB    'S+,S+,S+,S+,S+,S+,S+,S+,' ;more staccato,
```

```
        DB      'M+,M+,M+,M+'                   ;and slower.
;
        DB      ',C.,D.,E.,F.,G.,G.,G.,G.,+A.,F.,+C.,+A.,3G...,Q,'
        DB      '+,+A.,F.,+C.,+A.,3G...,-,Q,'
        DB      'G.,F.,F.,F.,F.,E.,E.,E.,E.,D.,E.,D.,C.,E.,G.,Q,'
        DB      'G.,F.,F.,F.,F.,E.,E.,E.,E.,D.,E.,D.,2C$'
;
;
        CSEG    ENDS
        END
```

Videotex

INTRODUCTION

The Videotex model of the NCR PERSONAL COMPUTER is designed to function as

* A PRESTEL or CEPT-standard Videotex terminal, with the following additional features:
  - full use of computer BIOS (except simultaneous CRT control), disk and peripheral device operations
  - connection of video long player (VLP) or video recorder, with screen intermix of Videotex and video information
  - Extensive software for page-editing, modem selection, and VLP support.
* An NCR PERSONAL COMPUTER with a color CRT. Possible display modes are character, low and medium resolution (as defined in Chapter 7). The display controller is configured as for the external color monitor (see Chapter 7).

Software switching between the two functions is controlled by a switch board (see Appendix A). It does not affect the contents of the video memories.

Figure 10.1 provides a block diagram of the NCR PERSONAL COMPUTER Videotex model, emphasizing the additional (Videotex) video controller and Videotex peripheral devices like VLP and modems.

Details of the CEPT standard and any regulations governing the generation of Videotex pages are available in documents issued by the post and telecommunications authorities of a number of countries. Regulations concerning the generation of Videotex pages and modem connection may vary from one country to another.

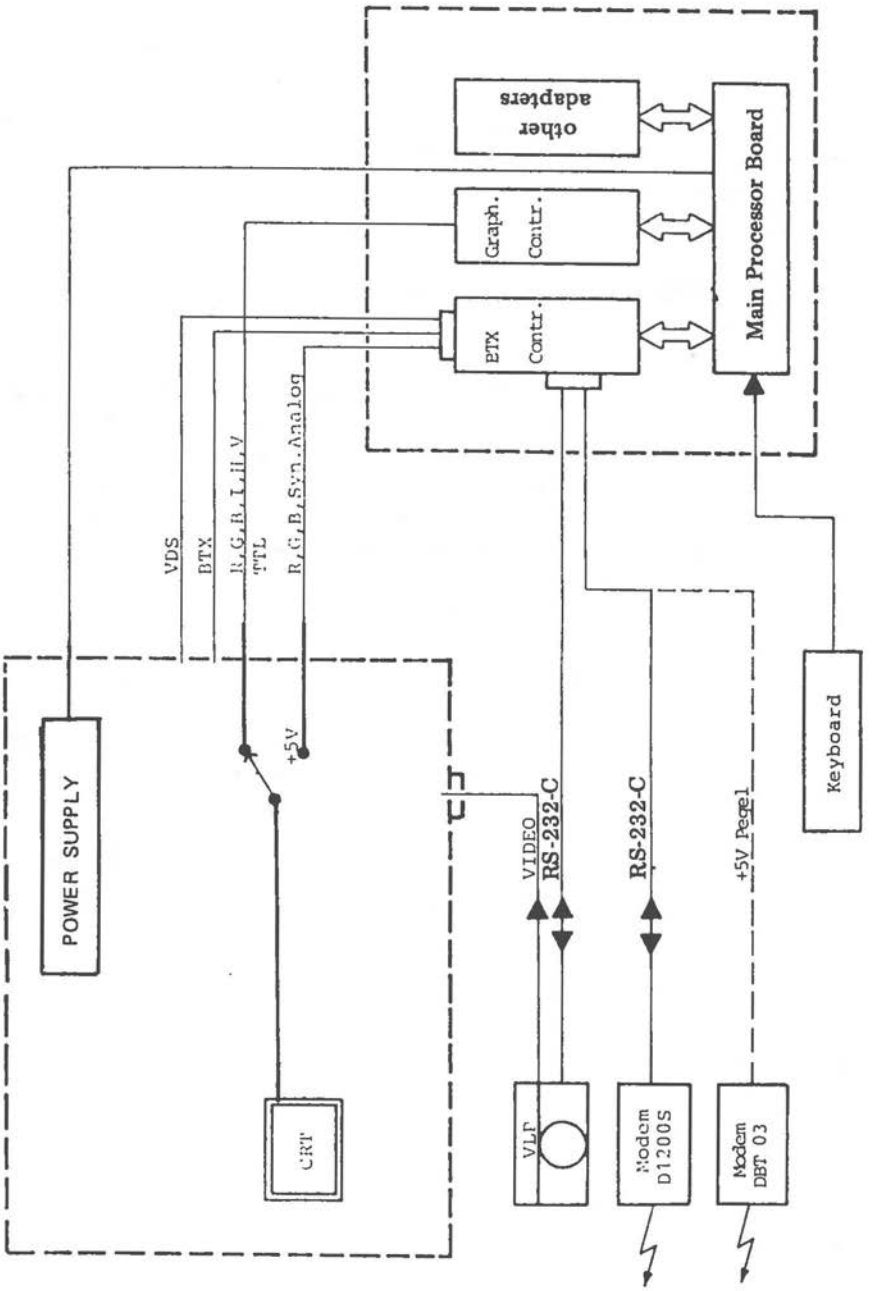


Figure 10.1 Videotex System – Block Diagram

SWITCHES AND CONNECTIONS

Videotex modem, video and VLP controllers make use of the interface signals shown in Figure 10.2.

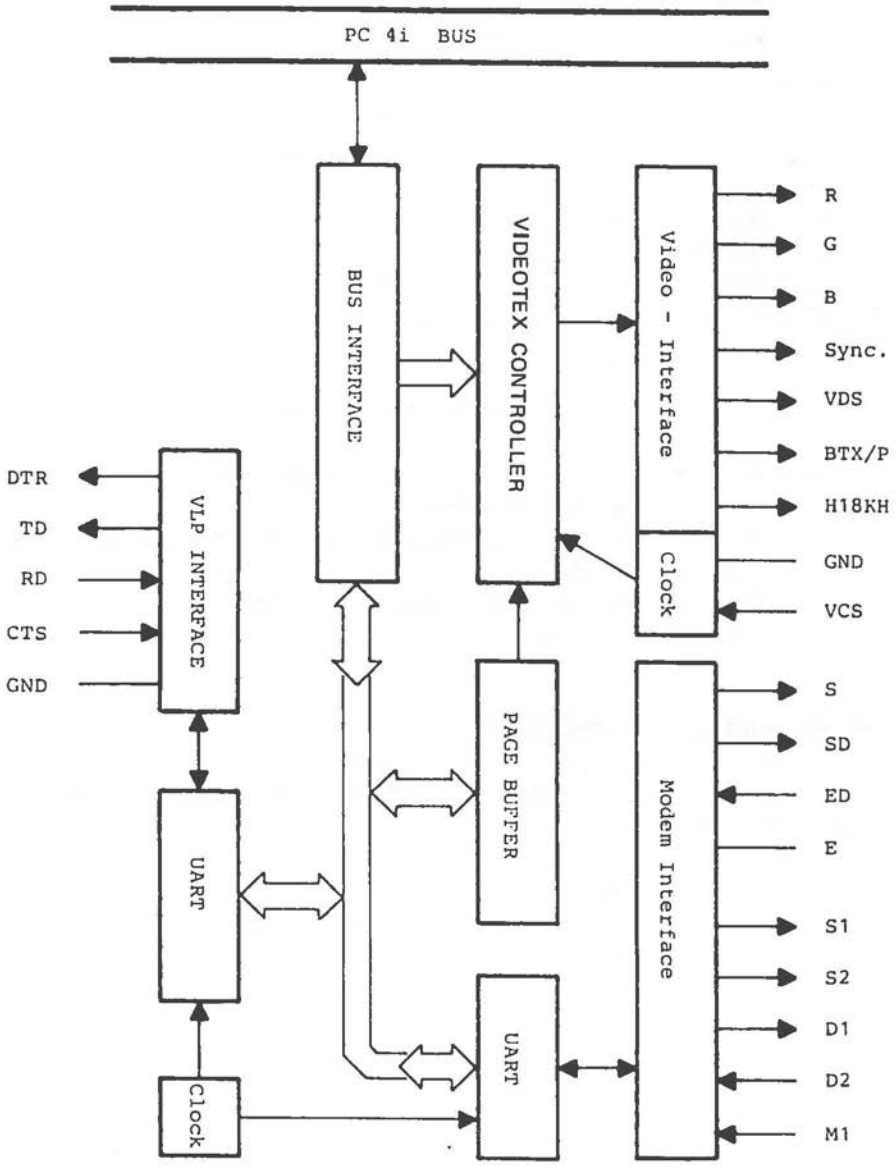


Figure 10.2 Videotex Interface Signals

MONITOR CONNECTION

Monitor connection is illustrated in Figure 10.3.

Pin	Signal
1	RED
2	GREEN
3	BLUE
4	TCS/
5	GND
6	SWITCH VIDEOTEX-PC
7	SWITCH VLP-VIDEOTEX
8	FBAS
9	HORIZ. FREQ SWITCH 18-15 KHz

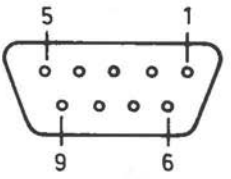


Figure 10.3 Monitor Connector

Note: TCS/ provides synchronization for RGB signals.
Pin 6 is high for Videotex
Pin 7 is high for Videotex RGB input, low for composite signal from VLP
FBAS is composite signal from VLP (75 Ohm)
Pin 9 is high for 18 KHz

MODEM AND VLP CONNECTION

The Videotex adapter is designed to work with three types of modem (suitable cables are identified in parentheses):

1 Full duplex with auxiliary data line, manual dialling (K301, or K303 with connection for video player).
Baud rate: Receive = 1200, Transmit = 75

2 Full duplex, manual dialling (K301, or K303 with connection for video player)
Baud rate: Receive = 1200, Transmit = 1200

3 Full duplex, automatic dialling (K300, or K302 with connection for video player)
Baud rate: Receive = 1200, Transmit = 75

Full duplex modem communications are supported on modern non-dedicated (standard telephone) lines, where incoming and outgoing "data" is clocked at different frequencies.

For VLP use, the NCR software assumes a receive baud rate of 1200, but this could be programmed at low level to another baud rate (see later section).

Figure 10.4 illustrates pin designations for the modem/video connector. Figures 10.5 and 10.6 show the use of lines 1 - 10 by the three types of modem. The actual designations of the modems vary according to the country in which the Videotex system is installed.

Pin	Signal	Pin	Signal
1	D2	9	ED
2	HD1	10	S
3	M1	11	GND_V
4	No connection	12	TD
5	S1	13	RD
6	S2	14	CTS
7	GND_M	15	DTR
8	SD		

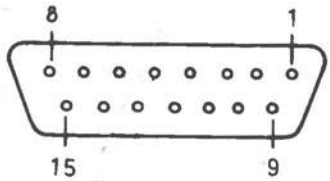


Figure 10.4 Modem/VLP Connector

Note: Pins 12-15 communicate with VLP,
other pins are for modem control (see below).

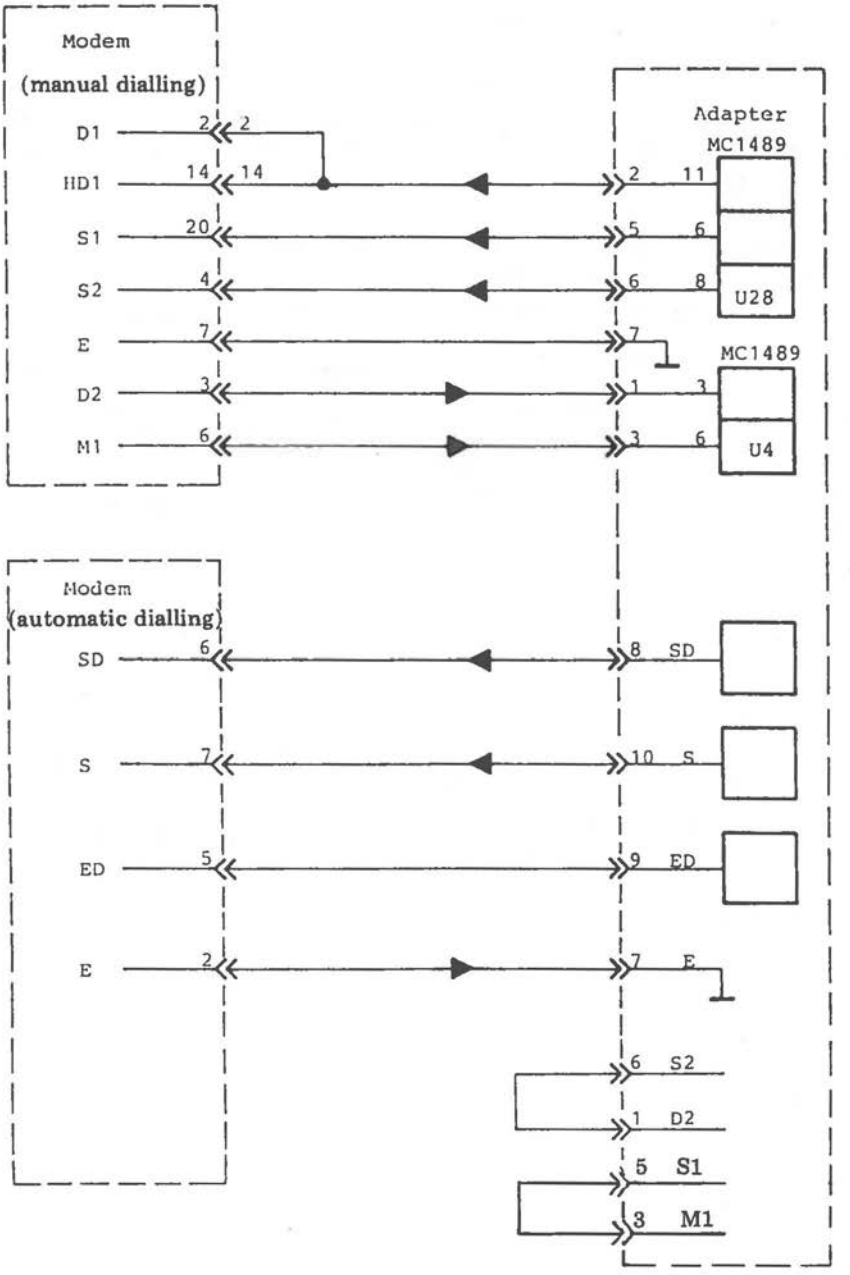


Figure 10.5 Videotex/Modem Signals

Signal	Modem: 1	2	3	Significance
D2	x	x		Receive Data
HD1	x	x		Transmit Data
M1	x	x		Data Set Ready - denotes modem activity
S1	x	x		Connect Data Set to Line - must be held high for modem activity
S2		x		Request to Send
GND	x	x	x	Ground
SD			x	Transmit Data
ED			x	Receive Data
S			x	Control

Figure 10.6 Modem Signal/Control Lines

SWITCHES

The functioning of the Videotex adapter is dependent on the switch settings. The locations of these switches is shown in Figure 10.7. Actual switch settings are given in Figure 10.8.

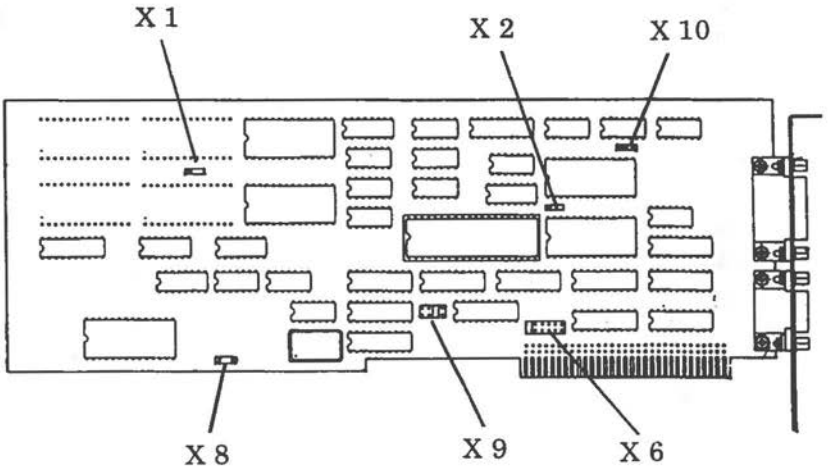


Figure 10.7 Videotex Adapter Switches

Switch	Switch Positions: Closed	Open	Significance
X1	1 - 2	2 - 3	2 KB RAM (= 1 page) display
	2 - 3	1 - 2	8 KB RAM *
X2	1 - 2	2 - 3	2 KB RAM (= 1 page)
	2 - 3	1 - 2	8 KB RAM *
X6	1 - 12	all others	Interrupt Request is IRQ2 *
	2 - 11		IRQ3
	3 - 10		IRQ4
	4 - 9		IRQ5
	5 - 8		IRQ6
	6 - 7		IRQ7
X8	yes	no	Phase lock loop filter set for video recorder
	no	yes	Phase lock loop filter set for VLP
X9	1 - 5	all others	Video memory starts at 70000H
	2 - 6		0B0000H
	3 - 7		0D0000H *
	4 - 8		0E0000H
X10	yes	no	Modem interface interrupt on transmit and receive
	no	yes	Modem interface interrupt on receive only *

* denotes factory settings

Figure 10.8 Videotex Adapter Switch settings

VIDEOTEX MEMORY ORGANIZATION

Video memory is organized from the machine address 0D0000H upwards, assuming the factory setting for switch X9 (see Figure 10.8).

Address (hex)	Bytes	Display use
D0000 - D03E8	1000	Character codes of 1 character screen 40 x 25 characters
D03F5	11	Line colors (lines 25-15, see "Character Attributes")
D0400 - D07E8	1000	Attribute bytes (1 KB)
D07F0	16	Line colors (lines 14-0 and -1)
D0800	6144	Re-definable character set (DRCS)
DC000 - DC03A	59	Videotex Controller registers 0-58

This memory map applies to the "stack" mode of display, this being the mode most conducive to implementation of the CEPT norm (other modes: 80 Character, Explicit Fill). The term "stack" is derived from the way in which character attributes are arranged (see later section).

Other memory areas in the address range 0DC000H to 0DC3FFH are used to control video display, modem and VLP interfaces. These interfaces are described in subsequent sections. Note that these interfaces are memory, not I/O, mapped.

Address (hex)	Interface
DC000 - DC0FF	Write Videotex display controller registers

'he memory addresses between 0DC100H and 0DC107H are ledicated to a number of switching functions:

Address (hex)	Switch Function (write only)
DC100	Enable (bit 7 = 1) or disable (bit 7 = 0) screen
DC101	Activate Videotex (bit 7 = 1) or NCR PERSONAL COMPUTER (bit 7 = 0) display
DC102	Select manual (bit 7 = 1) or automatic (bit 7 = 0) dialling modem
DC103	Select display horizontal sync: bit 7 = 0: 15 KHz (Videotex vertical sync 50 Hz) bit 7 = 1: 18 KHz (Videotex vertical sync 60 Hz)
DC104	Scroll during retrace only (bit 7 = 1) or at any time (bit 7 = 0)
DC105	Enable (bit 7 = 1) or disable (bit 7 = 0) VLP
DC106 - DC107	Not used

Modem and VLP control and data registers are mapped into memory as follows:

Address (hex)	Interface
DC200 - DC207	2651 USART registers (modem)
DC300 - DC307	2651 USART registers (VLP)

DISPLAY GENERATION

In the Videotex mode of operation, the video display may contain "alpha mosaic" characters from a ROM character set as well as from a user-defined character set in random access memory. The term "alpha mosaic" means that the character pixel patterns are written by the display controller in a fixed 40 (horizontal) x 25 (vertical) raster. Individual pixel plotting in video memory is not included but such an effect can be achieved indirectly by writing suitable user-defined graphic characters.

Characters can be written in one of up to 4096 colors. In addition, a number of display attributes can be set. A "lock bit" can be set when defining attributes for a character. Setting this bit has the effect that the character table currently selected will apply to further occurrences of that particular character code, until a new table (1-4) is specified for it. The default character table is Table 1 (see Figure 10.10).

Display line scrolling, or even a complete re-organization of the 25 display character lines, can be achieved without the need for CPU time-consuming block move software.

VIDEOTEX CONTROLLER

The SAA 5350 Videotex processor, in conjunction with a 8 KB video memory, provides a character generator and character display controller in one. Figure 10.9 shows the internal architecture of the Videotex Controller.

An external clock signal is used for horizontal synchronization provided by a 5240 integrated circuit on the Videotex adapter.

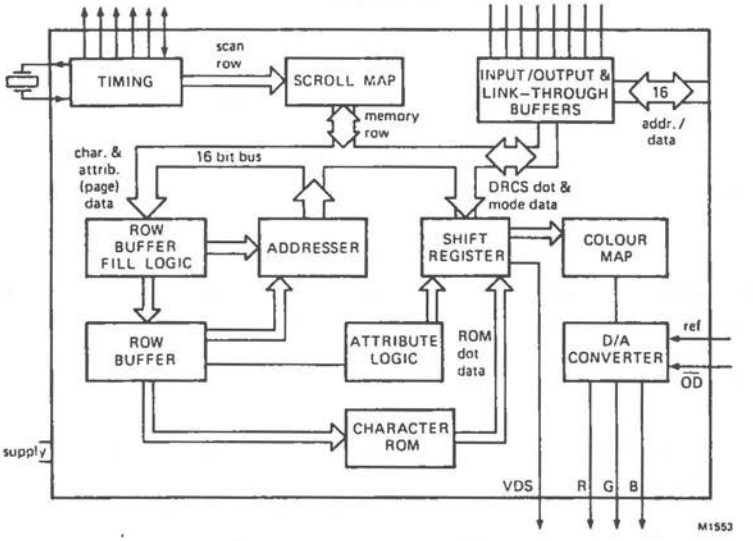


Figure 10.9 Videotex Controller - Internal Architecture

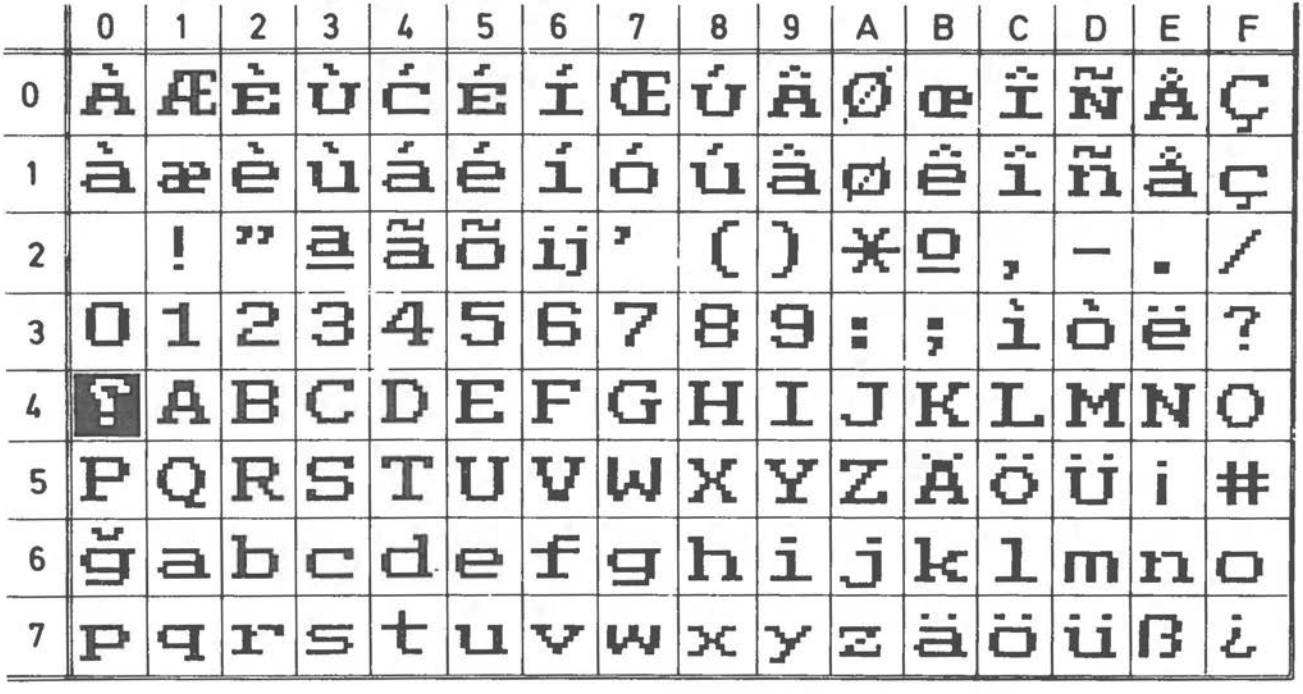
Prior to the Videotex controller taking over the system bus, it issues a bus request signal (BR/). The period of this advanced warning is a programmable feature (see below). A corresponding signal (DTACK/) is issued to denote completion of a data transfer.

CHARACTER SETS

Standard Characters

The Videotex Controller contains in its own read only memory the patterns of 512 alphanumeric and graphic characters in four separate tables of 128 characters each (see Figures 10.10 - 10.13).

The ROM character set is divided into four tables because only seven bits are available for the code of each character in video memory. The most significant bit of the character code byte is used to instruct the Videotex Controller to refer to a set of attributes for that character (see "Character Attributes"). One of these attributes is selection of the table to which the character code applies.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	À	Æ	È	Ù	Ć	É	Í	Œ	Ú	Â	Ø	œ	Î	Ñ	Å	Ç
1	à	æ	è	ù	á	é	í	ó	ú	â	ø	ê	î	ñ	å	ç
2		!	”	a	ã	õ	ij	’	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	ì	ò	ë	?
4	§	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ü	¡	#
6	ğ	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ü	ß	¿

Figure 10.10 ROM Character Set Table 1

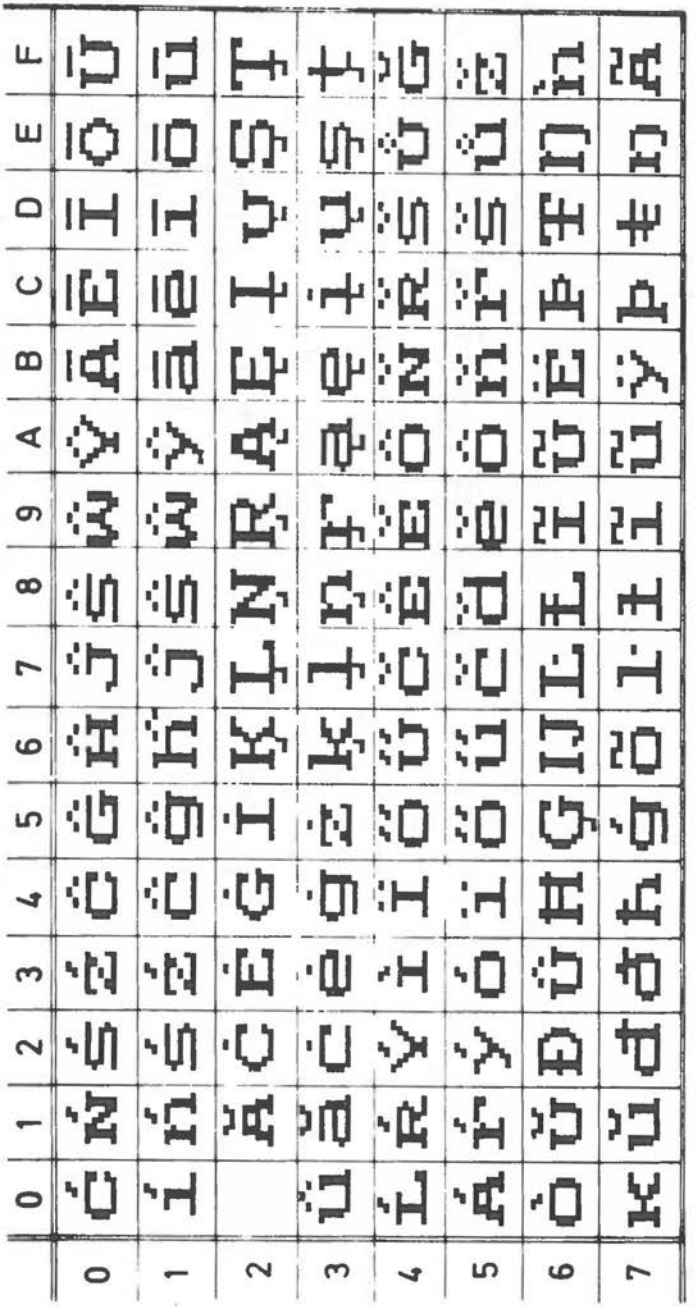


Figure 10.11 ROM Character Set Table 2

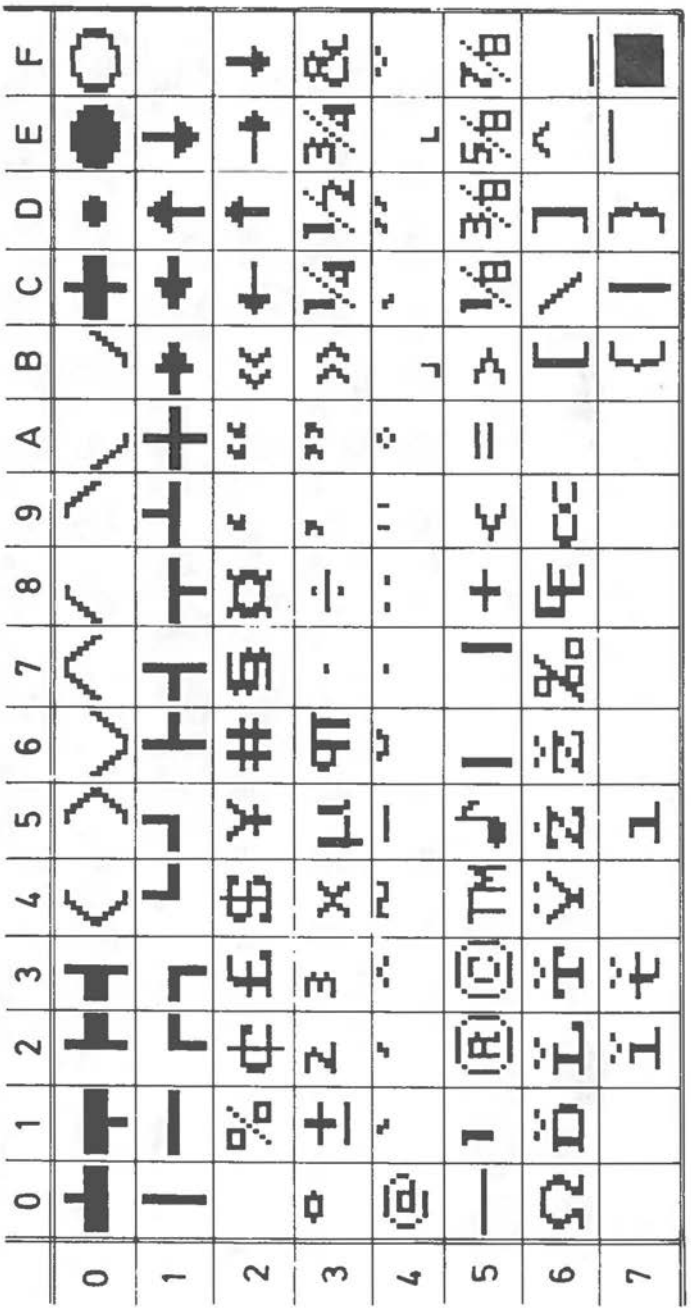


Figure 10.12 ROM Character Set Table 3

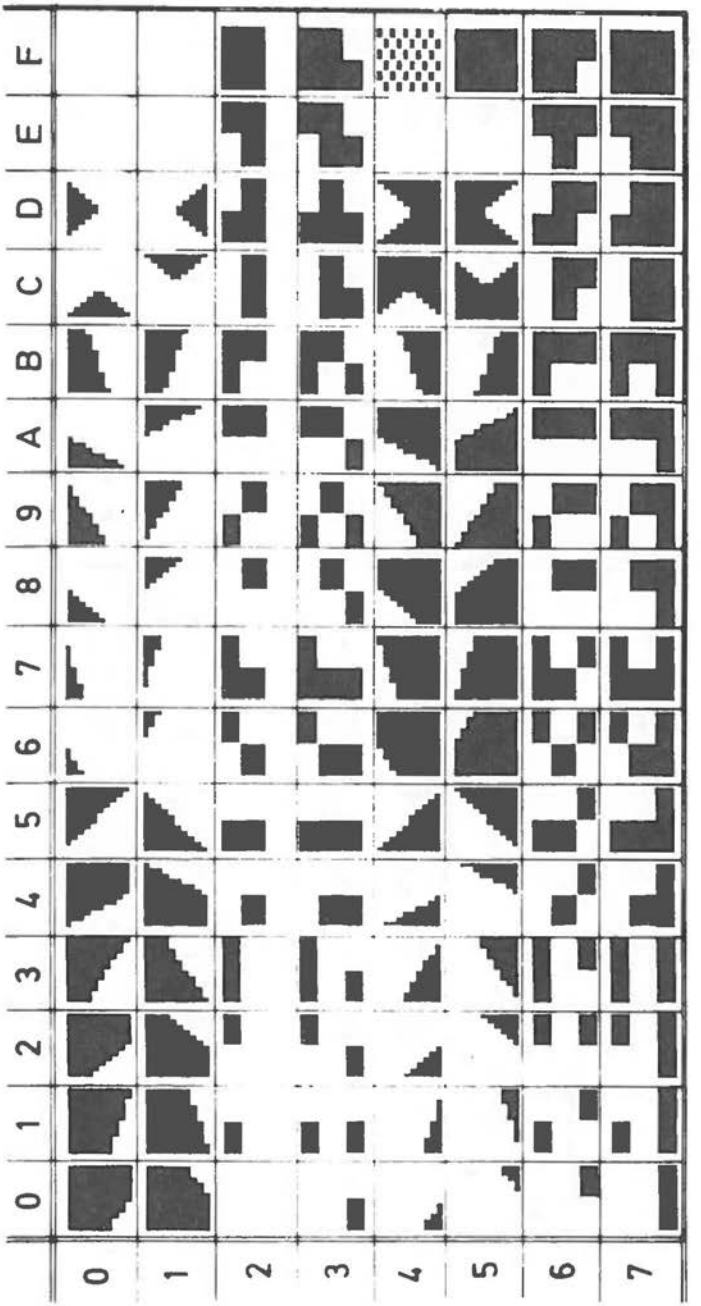


Figure 10.13 ROM Character Set Table 4

Definable Characters (DRCS)

The Videotex adapter is provided with 6 KB of RAM starting at machine address 0D0800H, from which the Videotex controller can read the pixel patterns for additional character tables.

The bit patterns for these "Dynamically Redefinable Character Sets" (DRCS) can be written under CPU control. Characters can be defined in rasters of 10 or 12 scan lines/character line. More than one such character format is possible, even within a single display line. The actual format used is encoded in the display information for each defined character. Figure 10.14 summarizes the DRCS dimensions available.

Format	Pixel Dimensions	Bits/Pixel	—> Max. characters in 1024 16-bit words (= 1 Chapter)
	Horiz. Vert.		
1	12 x 10	1	102
2	12 x 10	2	51
3	6 x 10	1	2 * 102
4	6 x 10	2	102
5	6 x 10	4	51
6	6 x 5	2	2 * 102
7	6 x 5	4	102
8	12 x 12	1	85
9	12 x 12	2	42
10	6 x 12	1	2 * 85
11	6 x 12	2	85
12	6 x 12	4	42
13	6 x 6	2	2 * 85
14	6 x 6	4	86

Figure 10.14 DRCS Alpha Mosaic Formats

Each character occupies 20, 24, 40, or 48 bytes of DRCS RAM, according to the Format selected. As the number of DRCS characters in 6 KB cannot exceed 306, all such characters can be addressed by selecting character table 5, 6, or 7. The Videotex Controller automatically calculates the offset in DRCS RAM for any DRCS character whose code is specified in the normal way in display memory (1 KB from 0D0000H). This is true even if you have occasion to mix character formats in your display.

Each display line is represented by a 16-bit word, the upper bits being in the even byte. The format 1 - 14 (Figure 10.14) selected is encoded in four bits in this 16-bit word - bits 15-14 and 6-7 (Figure 10.15). For most formats, the 4-bit format code is actually read from the first word specified for the character. For reasons of formal consistency, the remaining words should repeat this information.

Format	Bit:			
	15	14	7	6
1/8	0	0	0	0
2/9	0	0	0	1
3/10	1	0	1	0
4/11	1	0	0	1
5/12	0	0	1	1
6/13	1	1	1	1
7/14	1	1	0	1

Figure 10.15 DRCS Format Codes

The significance of the 12 display bits varies from format to format. In all formats, the display MSB (bit 13) of the least significant (even) byte is written left- and uppermost.

FORMAT 1, FORMAT 8
Each display word of 12 bits represents one pixel line. A pixel can be either on or off, the actual color being determined by the attributes set for that character (see "Character Attributes"). The only difference between these two formats is that Format 8 requires 12 such words, Format 1 requires 10. An example for Format 1 is given in Figure 10.16.

FORMAT 2, FORMAT 9
Each pixel is represented by **two** bits: the display MSB of byte 0 and the display MSB of byte 20 (Format 2) or byte 24 (Format 9) are interpreted as a 2-bit binary value (upper byte supplies the MSB of the two) which refers to one of four "DRCS color look-up tables" (CLUT, see "Character Attributes"); this determines the color of the first pixel of the character. Bit 12 of the same two bytes determines a color for the next pixel and so on. An example for Format 9 is given in Figure 10.17, where the color look-up tables for each pixel is denoted by a number 0 - 3.

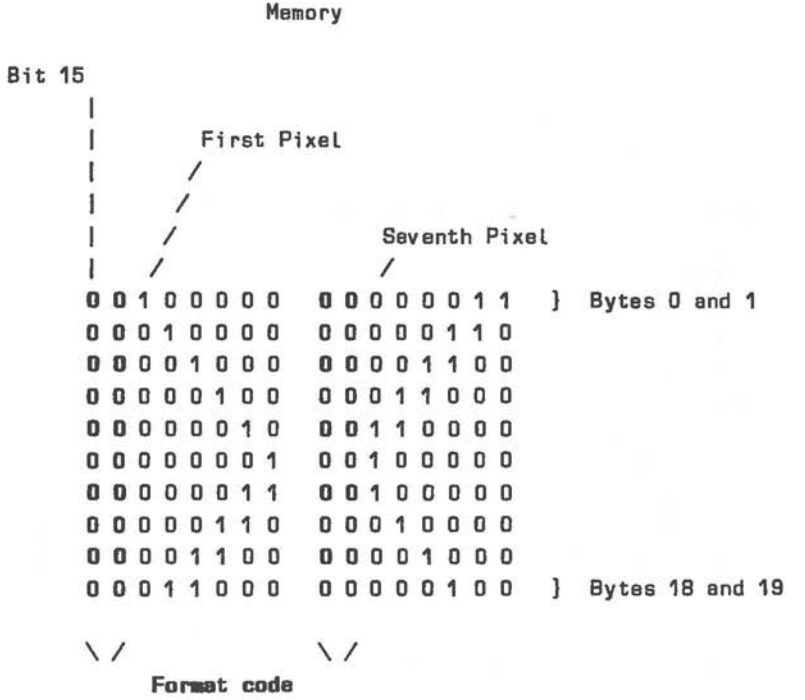
FORMAT 3, FORMAT 10
The 6-pixel line pattern is given twice (bits 13-8 and 5-0) within each of the 10 or 12 display words. Figure 10.18 gives an example for Format 3.

FORMAT 4, FORMAT 11
This is similar to Format 3 and 10, with the difference that the two 6-bit halves within a display word combine, as in Format 2 and 9, to refer to one of four color look-up tables. This is illustrated for Format 4 by the example in Figure 10.19.

FORMAT 5, FORMAT 12
These formats make possible the specification of any one of 32 colors for each pixel. Each pixel is referred to by four bits, which would normally restrict color selection to one of sixteen. A Videotex Controller register selects which of two color groups applies (see section below). Format 5 requires 40 bytes of DRCS memory (see example in Figure 10.20), Format 12 requires 48 bytes.

FORMAT 6, FORMAT 13
Five (Format 6) or six (Format 13) adjacent word pairs make up alternate pixel lines. The 6-pixel line pattern is given twice in each word. An example for Mode 6 is given in Figure 10.21.

FORMAT 7, FORMAT 14
These formats, like 6 and 13, affect alternate pixel lines. They also resemble 5 and 12, in that the 4-bit value refers directly to a color and not to a color look-up table. Figure 10.22 illustrates Format 7.



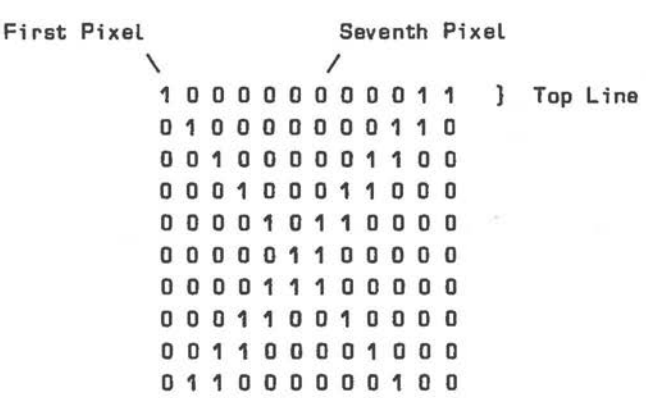


Figure 10.16 Format 1: Example

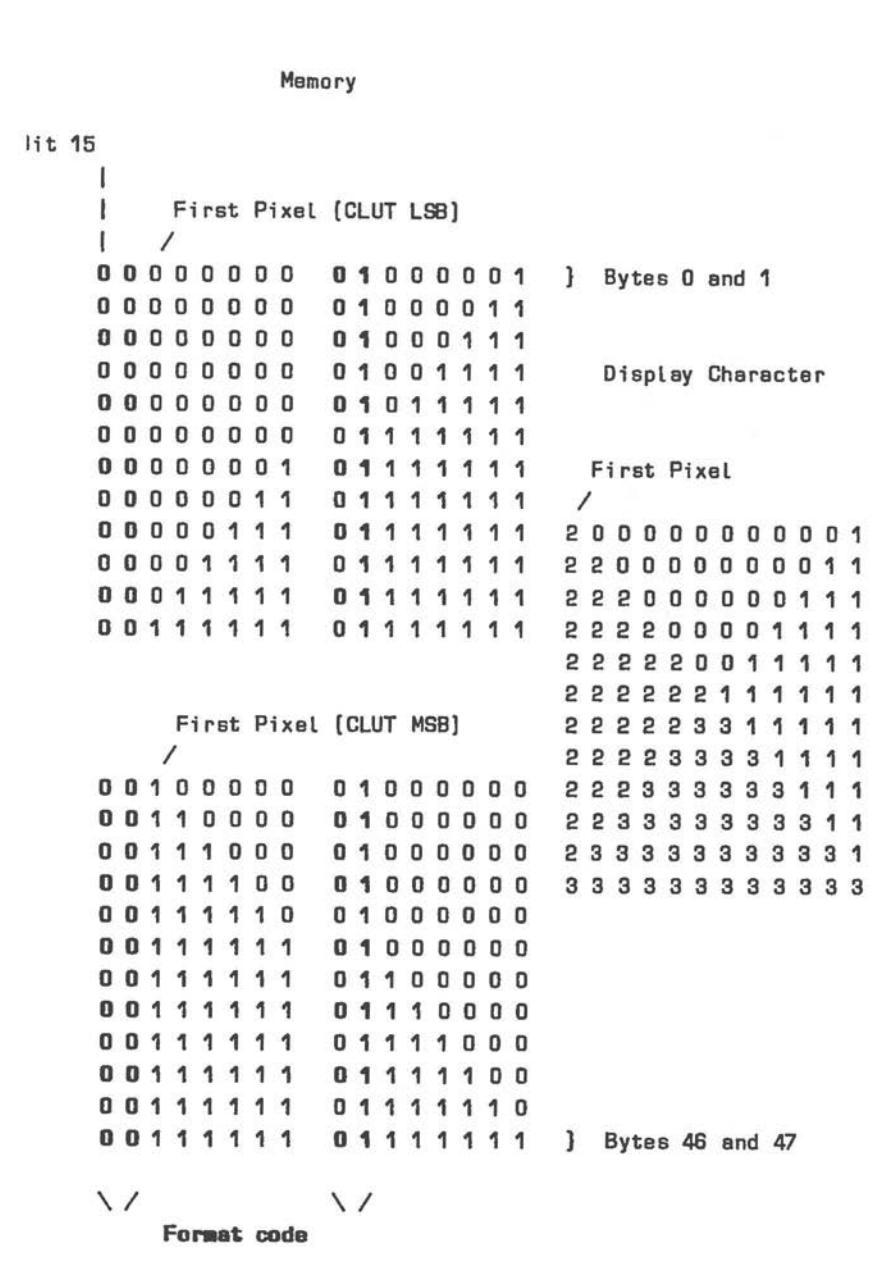
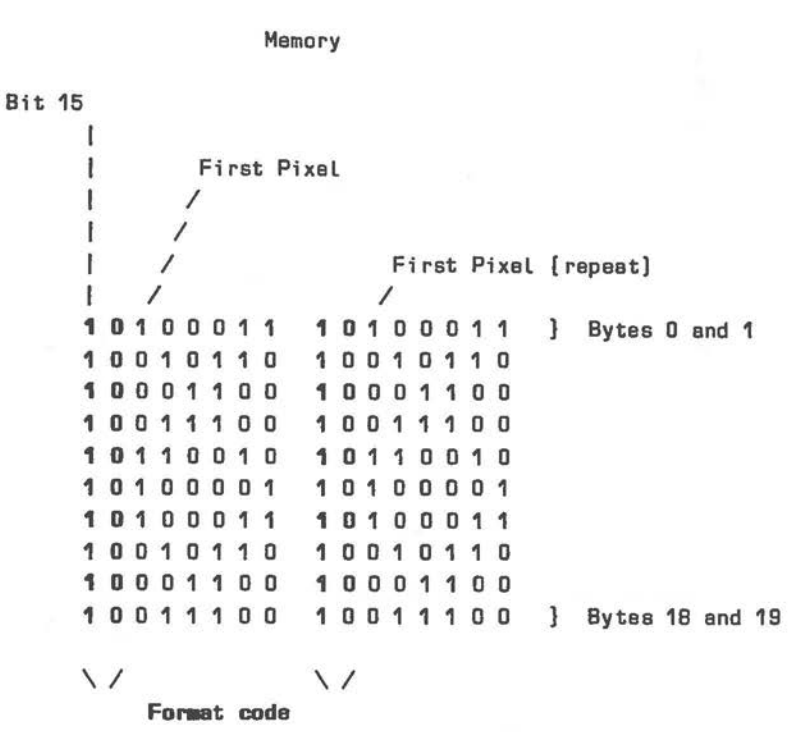


Figure 10.17 Format 9: Example



Display character

```
1 0 0 0 1 1
0 1 0 1 1 0
0 0 1 1 0 0
0 1 1 1 0 0
1 1 0 0 1 0
1 0 0 0 0 1
1 0 0 0 1 1
0 1 0 1 1 0
0 0 1 1 0 0
0 1 1 1 0 0
```

Figure 10.18 Format 3: Example

```
                    Memory

Bit 15
      |
      |        First Pixel (CLUT LSB)
      |       /
      |      /
      |     /                  First Pixel (CLUT MSB)
      |    /                  /
      1 0 1 0 0 0 0 0   0 1 0 0 0 0 0 1   } Bytes 0 and 1
      1 0 1 1 0 0 0 0   0 1 0 0 0 0 1 1
      1 0 1 1 1 0 0 0   0 1 0 0 0 1 1 1
      1 0 1 1 1 1 0 0   0 1 0 0 1 1 1 1
      1 0 1 1 1 1 1 0   0 1 0 1 1 1 1 1
      1 0 1 1 1 1 1 1   0 1 1 1 1 1 1 1
      1 0 1 0 0 0 0 0   0 1 0 0 0 0 0 1
      1 0 1 1 0 0 0 0   0 1 0 0 0 0 1 1
      1 0 1 1 1 0 0 0   0 1 0 0 0 1 1 1
      1 0 1 1 1 1 0 0   0 1 0 0 1 1 1 1   } Bytes 18 and 19

      \ /               \ /
          Format code
```

```
Display character

   1 0 0 0 0 2
   1 1 0 0 2 2
   1 1 1 2 2 2
   1 1 3 3 2 2
   1 3 3 3 3 2
   3 3 3 3 3 3
   1 0 0 0 0 2
   1 1 0 0 2 2
   1 1 1 2 2 2
   1 1 3 3 2 2
```

Figure 10.19 Format 4: Example

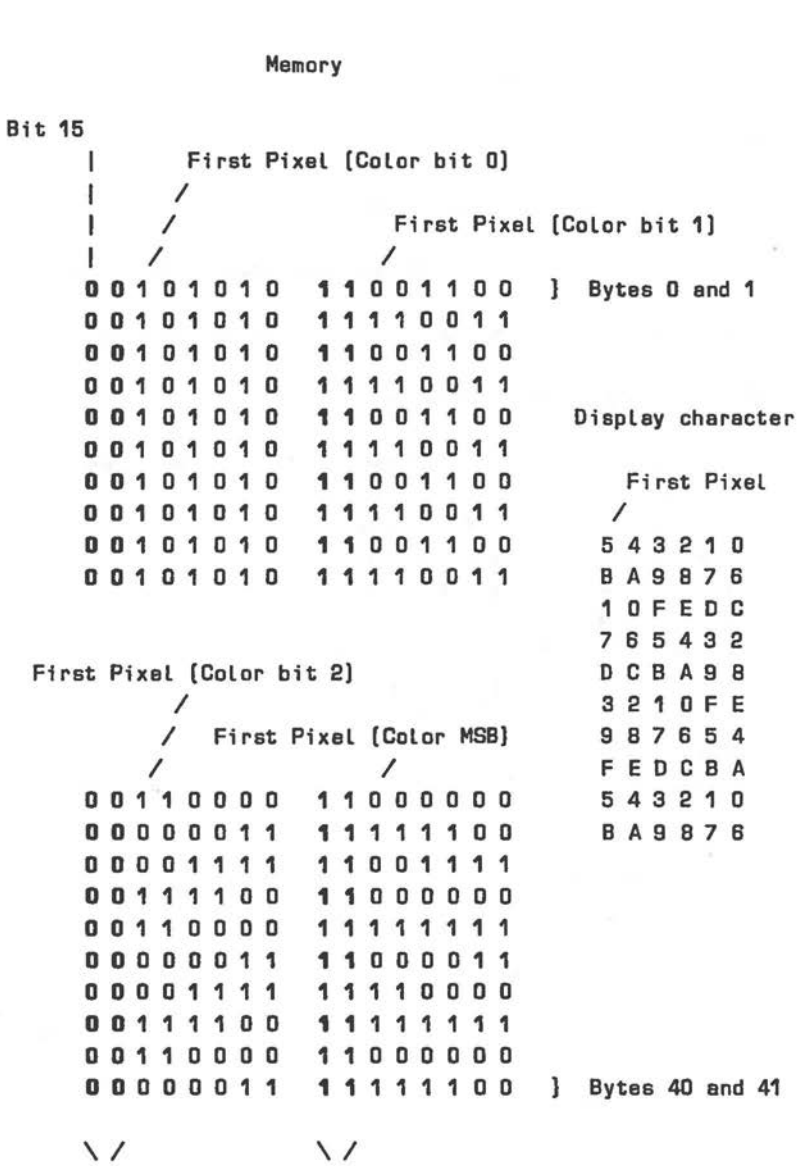
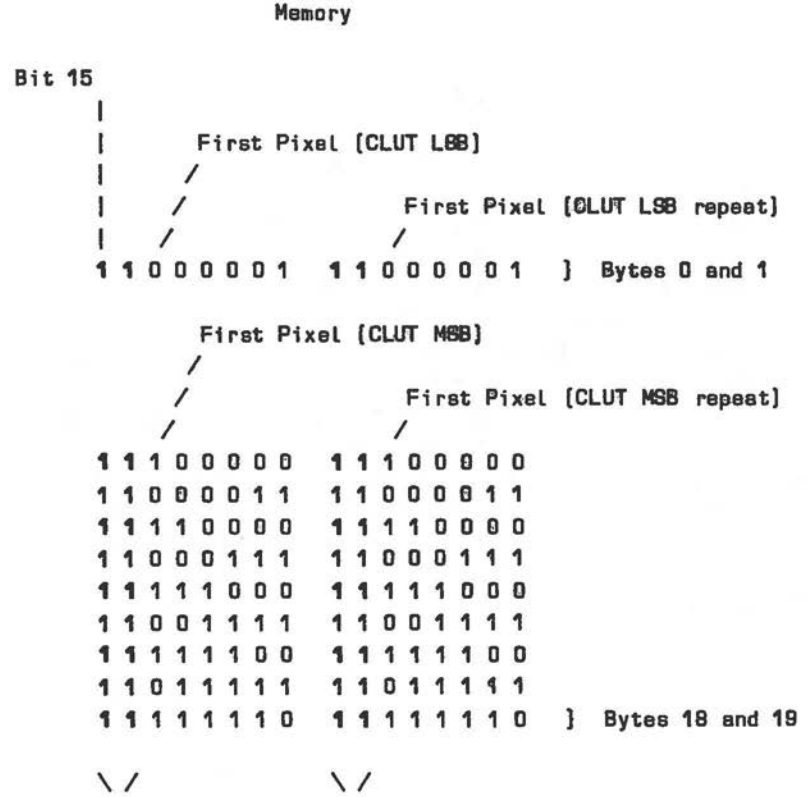


Figure 10.20 Format 5: Example



Display character

Line	0	2	0	0	0	1
	2	2	2	0	0	1 1
	4	2	2	2	1	1 1
	6	2	2	3	3	1 1
	8	2	3	3	3	3 1

Figure 10.21 Format 6: Example

```
                    Memory

Bit 15
      |
      |       First Pixel (Color bit 0)
      |      /
      |     /                 First Pixel (Color bit 1)
      |    /                 /
      1 1 1 0 1 0 1 0   0 1 0 0 1 1 0 0   }  Bytes 0 and 1

             First Pixel (Color bit 2)
            /
           /                  First Pixel (Color MSB)
          /                  /
      1 1 1 1 0 0 0 0   0 1 0 0 0 0 0 0
      1 1 1 0 1 0 1 0   0 1 1 1 0 0 1 1
      1 1 0 0 0 0 1 1   0 1 1 1 1 1 0 0
      1 1 1 0 1 0 1 0   0 1 0 0 1 1 0 0
      1 1 0 0 1 1 1 1   0 1 0 0 1 1 1 1
      1 1 1 0 1 0 1 0   0 1 1 1 0 0 1 1
      1 1 1 1 1 1 0 0   0 1 0 0 0 0 0 0
      1 1 1 0 1 0 1 0   0 1 0 0 1 1 0 0
      1 1 1 1 0 0 0 0   0 1 1 1 1 1 1 1   }  Bytes 18 and 19

      \ /               \ /
          Format code


            Display character

       Line  0   5 4 3 2 1 0

             2   B A 9 8 7 6

             4   1 0 F E D C

             6   7 6 5 4 3 2

             8   D C B A 9 8
```

Figure 10.22 Format 7: Example

CHARACTER ATTRIBUTES

The Videotex adapter contains 1 KB of character code RAM (from address 0D0000H) , and 1 KB of attribute RAM (from address 0D0400H). Attributes can be set in as many as 9 bytes for a single character. This demands economy in the way attributes are applied to video memory characters.

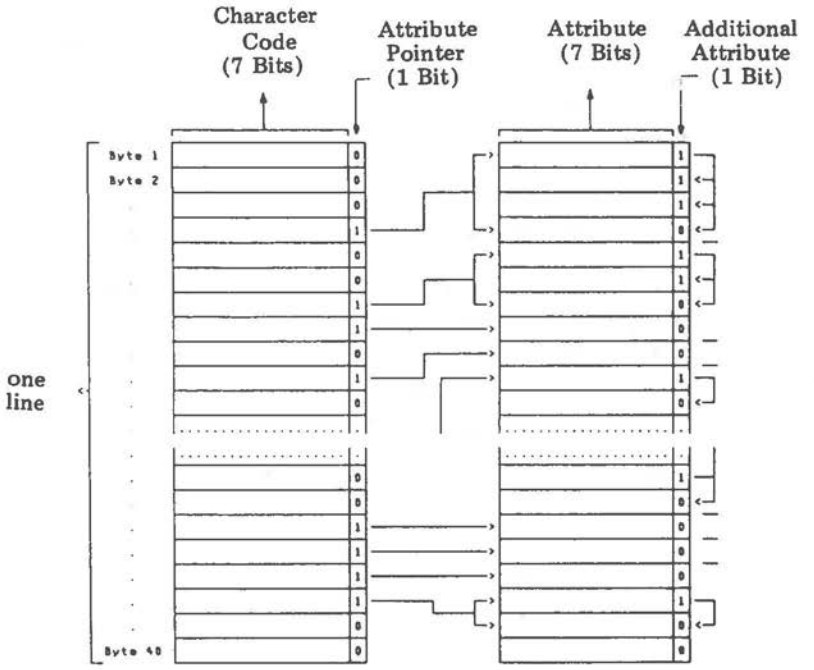


Figure 10.23 Attribute Memory Dynamic Allocation

The Videotex Controller achieves this economy by looking for an attribute pointer bit when writing a character: if bit 7 of the encoded character in video memory is set, the Controller refers to an attribute byte in attribute RAM. If in addition to specifying an attribute, that attribute byte has its bit 7 set, the Controller recognizes that a further attribute byte must be read and so on. This dynamic allocation of attribute memory means that attribute bytes do not lie unused, because less than 9 attributes are required for a particular character. This dynamic allocation is illustrated in Figure 10.23. Up to 40

attribute bytes may be allocated to characters in any one line.

This structure of character and attribute for a display of 40 characters width is described as the "stack mode". It is a mode most conducive to producing a display in accordance with CEPT standards, and is therefore the one described in this document. Other modes like the 80 Character mode and the Explicit Fill mode are described in documents produced by the integrated circuit manufacturer.

An attribute byte is constructed as shown in Figure 10.24. 32 colors are available for both foreground and background. These colors must have already been selected from a range of 4096 colors by programming the Videotex Controller registers (see below).

P is the pointer bit, indicating that at least one more attribute follows for the current character (set), or that this is the last or only attribute (zero) for that character.

The Lock bit is used to set a new default for the character table to which a particular character code refers. If the Lock bit is set, the character table specified in 3 bits of the same byte applies to all further occurences of that character code in the same line, or until a contrary character table attribute byte is specified for that character code. If this attribute byte is issued with the Lock bit zero, the newly specified character table applies only to this one occurrence of the character code. The next occurence of the character code, refers to the most recently declared character table for that code, or, failing such a declaration in the current line, to the default Table 1.

Bit:								Attribute
7	6	5	4	3	2	1	0	
P	0	0	FG4	FG3	FG2	FG1	FG0	Foreground color (0 = transparent)
P	0	1	BG4	BG3	BG2	BG1	BG0	Background color (0 = transparent)
P	1	0	B4	B3	B2	B1	B0	Blinking attributes (see Figure 10.25)
P	1	1	0	L	T2	T1	T0	L = lock bit T0-T2 = character table
P	1	1	1	0	0	H	W	Double Height and/or double Width on character
P	1	1	1	0	1	0	U	Underline (toggle)
P	1	1	1	0	1	1	I	Invert
P	1	1	1	1	0	0	C	Conceal (character witten in background color)
P	1	1	1	1	0	1	W	Window/box: makes screen color transparent at that character
P	1	1	1	1	1	0	x	} reserved
P	1	1	1	1	1	1	x	}

Figure 10.24 Attribute Byte

Bit:					Blinking mode
B4	B3	B2	B1	B0	
0/1	0	0	0	0	No blinking
0/1	0	0	0	1	Blink to right 1 Hz/2 Hz
0/1	0	0	1	0	left 1 Hz/2 Hz
					Blink between color look-up tables
					1/2 or 2/3
0/1	0	1	0	0	50 % of 1 Hz/2 Hz
0/1	0	1	0	1	]
0/1	0	1	1	0	] 33 % of 1 Hz/2 Hz
0/1	0	1	1	1	]
					Blink foreground/background
0/1	1	0	0	0	50 % of 1 Hz/2 Hz
0/1	1	0	0	1	]
0/1	1	0	1	0	] 33 % of 1 Hz/2 Hz
0/1	1	0	1	1	]
					Blink background/foreground
0/1	1	1	0	0	50 % of 1 Hz/2 Hz
0/1	1	1	0	0	]
0/1	1	1	1	0	] 33 % of 1 Hz/2 Hz
0/1	1	1	1	1	]

Figure 10.25 Blinking Attribute

COLOR SELECTION

Color Layers

Colors are selectable for three display layers. In addition to the foreground and background characteristics we associate with computer displays, the Videotex adapter makes use of "line colors". The line color represents the color produced during CRT scanning where no foreground or background color is displayed, that is, where color transparency is specified. This color can be defined separately for each character line (lines 0-24), as well as for the

fringe areas at the top (line 25) and bottom (line -1) of the screen. Each of these lines is controlled by one byte (see "Videotex Memory Organization"). Figure 10.26 illustrates the structure of this byte.

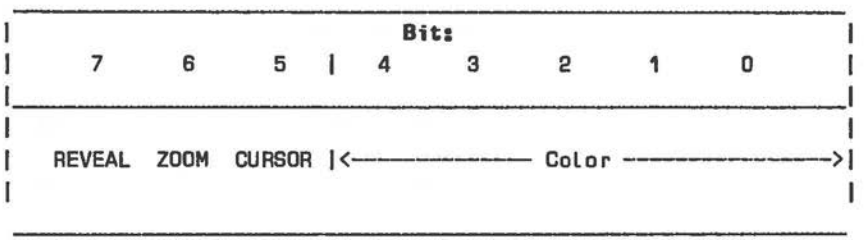


Figure 10.26 Line Characteristics

REVEAL (set) cancels Conceal attribute.
ZOOM effects double height magnification, displacement of following lines.
CURSOR set on the line in which cursor is to be displayed; more than one cursor is allowed, cursor form is set by Videotex Control registers.

Color Coding

Color information for foreground, background, and line is coded in 5 bits (Figure 10.27).

Bit:					Color
4	3	2	1	0	
PALETTE	INTENSITY	BLUE	GREEN	RED	
		0	0	0	Black
		0	0	1	Red
		0	1	0	Green
		0	1	1	Yellow
		1	0	0	Blue
		1	0	1	Magenta
		1	1	0	Cyan
		1	1	1	White

Figure 10.27 Color Coding

Color Look-up Table
As already mentioned, you can select 32 colors from a total number of 4096. This means that the 5-bit value cannot directly control the RGB outputs to the CRT circuitry. Instead, the 5-bit value refers to a palette, also called "color look-up table" (CLUT), in which amplification values in four bits for each of the three color guns are stored. 4096 (2 to the power of 12) colors are available, any one of which can be selected by each of the 32 table pointers. Figure 10.27 describes these bits as Palette, Intensity, Blue, Green, and Red simply because of the CEPT standard. This standard specifies that the first group of eight colors should be black, red, green, yellow, blue, magenta, cyan, and white, while the second group yields the same colors in half-intensity.

These 32 12-bit items of color information are stored in 96 bytes near the top of the 4 KB RAM area at the disposal of the Videotex Controller (see "Videotex Memory Organization"): the lowest byte contains in its four MSBs the red color value for CLUT color 0, the next two bytes contain corresponding values for green and blue; the next three bytes define CLUT color 1, and so on. You can read this 96-byte area in order to ascertain color settings. However, the conventional way of writing the color items is by means of Videotex Controller Registers (see section below).

DRCS Attributes
DRCS characters, like the standard characters, derive their attributes from the dynamically allocatable attribute stack. Where more than one bit per pixel applies,

- the entire character is regarded as foregound color
- the attributes Conceal, Blink and Invert make use of the background color
- the Underline attribute has no effect

- DRCS 4 bit/pixel characters shown in Figures 10.20 and 10.22 do not require an attribute byte in order to specify color. Instead, the 4-bit color code refers directly to one of 16 colors in the color look-up table. Whether the lower or upper group of sixteen is being addressed depends on a single-bit palette switch which comprises one of the Videotex Controller registers.

Transparency

Instead of specifying a color, it is possible to regard a character or pixel (in DRCS 4 bit/pixel display) as transparent. Transparency on the screen layer (line color, see "Color Layers") means that input from a composite video signal can be accepted on that layer. In principal, this signal can be derived from any suitable broadcasting device. The VLP interface is designed to supply just such a signal.

Transparency for a particular layer is achieved by specifying the color attribute "half-intensity black" (all guns off), where this color is referred to by color look-up code 0. Figure 10.28 illustrates the origin of display signal used by the Videotex Controller for invert and color permutations of the attribute byte.

Attribute Settings			Layer from which color of resultant display is taken	
Foreground/ Background	Foreground	Background	Active dot color	Inactive color
NORMAL	normal	normal	F	B
		transparent	F	line
	transparent	normal	line	B
		transparent	line	line
INVERT	normal	normal	B	F
		transparent	line	F
	transparent	normal	B	line
		transparent	line	line

Figure 10.28 Layer transparency

B - Background
F - Foreground
line - the illuminated CRT area controlled by the 27 line colors

VIDEOTEX CONTROLLER REGISTERS

A number of Videotex Controller initialization and other parameters are determined by data written to its internal registers 0 - 58 mapped (in that ascending order) in the memory area between 0DC000H and 0DC03AH. The Videotex Controller internally uses copies of these registers in the RAM address area 0D1000-0D103F. You should not normally write into

these copies, but instead write directly into the registers in the higher memory area.
The registers are summarized in Figure 10.29. This section also includes more detailed notes on the use of certain registers, together with values as initialized by NCR software.

R0 SYNC REGISTER Initialization value 0F4H; VLP requires 0F3H

Bit 7 is set to denote that the synchronization signal is external.

Bit 6 is set to provide a 1 MHz squarewave from the controller.

Bit 5 set results in an active low pulse that coincides with the "broad pulses" of composite sync.

Bit 4 is set to indicate that the Videotex Controller is not a slave in a configuration of more than one controller.

Bit 3 is zero to determine a character display height of 10 lines (otherwise 12 lines).

Bit 2 zero disables the phase lock for display interlace when VLP input is taking place. Otherwise, this bit is set.

Bit 1 controls the signal source for the internal field sync generator: 0 = text, 1 = video (VLP).

Bit 0 controls the text composite sync: 0 = display is non-interlaced, generating even fields of 312 lines; 1 = display is interlaced to broadcast reception standard.

R1 TEST REGISTER Initialization value 04

Bits 0 and 1 are used for functional test purposes only.

Bit 2 if set, enables data transfer to the color map to take place at any time, even outside display blanking.

R2 DMA WARNING TIME Initialization value 16H

This is the length of time in which the Bus Request signal is asserted, prior to the system bus being taken over. The value specified must not exceed 23 (microseconds).
Note that EUROM memory access is inhibited until this register is explicitly written.

R3 DISPLAY COMMAND 1 Initialization value 62H

Bit 7 enables (1) and disables (0) VLP.

Bit 6 enables (1) and disables (0) text display.

Bit 5 enables and disables the Window attribute in text mode as well as the Box attribute in superimpose mode (VLP).

Bit 1 selects color look-up table (0 = colors 0-15, 1 = colors 16-31).

Bit 0 if zero, inhibits the effect of any flash attributes.

R4 DISPLAY COMMAND 2 Initialization value 0

Bit 4 Chapter "A" pointer incrementing (R7) enable (1) or disable (0). This incrementing in step with the horizontal scan need be enabled in "Full Field DRCS" mode only. This would require more than the 2 KB of DRCS memory available.

Bit 3 if set, restores all attributes (except Lock bit) to default values: white foreground, transparent background, not blinking, no double height/width, not underlined, not inverted, not concealed, no window/box.

Bit 2 enable (0) or disable (1) the effect of the Conceal attribute.

Bit 1 is zero to set Stack and not Explicit Fill mode.

Bit 0 is zero to specify 40 and not 80 characters per line.

R5 DISPLAY START Initialization value 1BH

This register determines the vertical start position of text on the CRT. In this way, the character display can be vertically centred. Adjustment of this value can be used to achieve "soft" scrolling.

R6 DISPLAY PAGE CHAPTER Initialization value 0

Memory addressable by the Videotex Controller can be considered as consisting of 64 Chapters, each of 1024 16-bit words (Stack mode).

R7 DRCS CHAPTER "A" POINTER Initialization value 1

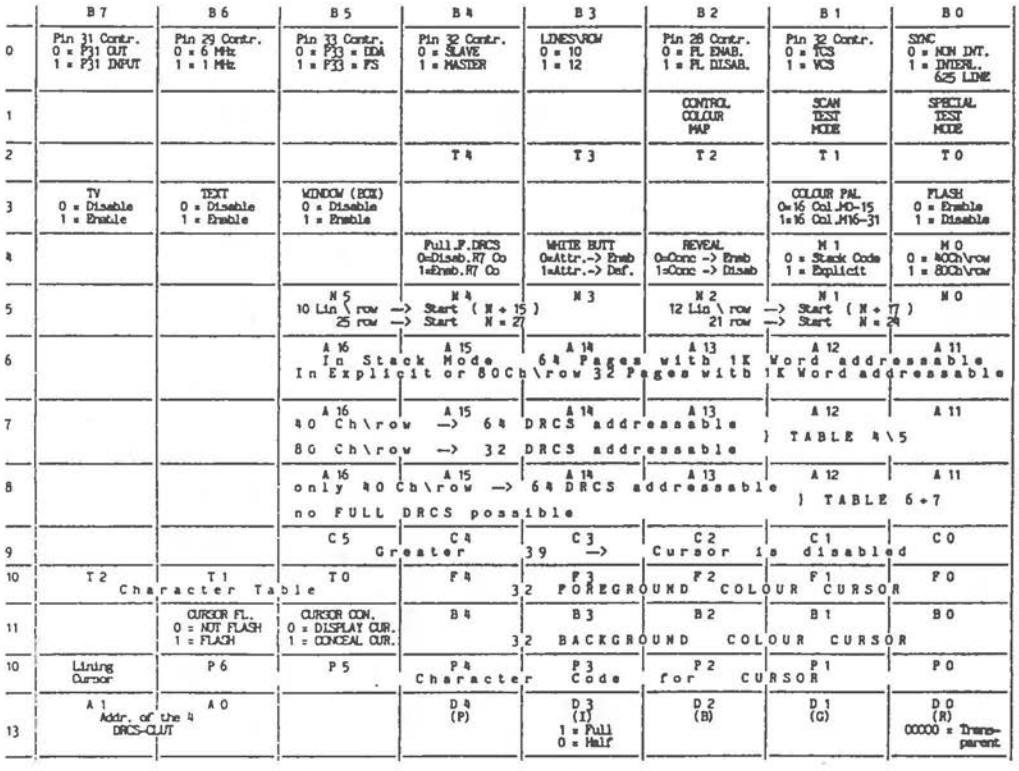
This value represents in effect the start address offset of DRCS Tables 5 and 6 (Tables 1-4 are fixed character set) in terms of register contents times 1024 bytes (Stack mode).

R8 DRCS CHAPTER "B" POINTER Initialization value 2

As R7, except that this register applies to Tables 7 and 8 (RAM is not installed).

R9 CURSOR COLUMN

Defines the character column that is to be used for cursor display. Cursor lines are defined in the line color definition bytes (see "Color Layers").



	B 7	B 6	B 5	B 4	B 3	B 2	B 1	B 0	
0	Pin 31 Contr. 0 = P31 OUT 1 = P31 INPUT	Pin 29 Contr. 0 = 6 MHz 1 = 1 MHz	Pin 33 Contr. 0 = P33 = DDA 1 = P33 = FS	Pin 32 Contr. 0 = SLAVE 1 = MASTER	LINES\ROW 0 = 10 1 = 12	Pin 28 Contr. 0 = PL ENAB. 1 = PL DISAB.	Pin 32 Contr. 0 = TCS 1 = VCS	SYNC 0 = NON INT. 1 = INTERL. 625 LINE	SYNC REG.
1						CONTROL COLOUR MAP	SCAN TEST MODE	SPECIAL TEST MODE	TEST REG.
2				T 4	T 3	T 2	T 1	T 0	DMA WARKING TIME (Tin us)
3	TV 0 = Disable 1 = Enable	TEXT 0 = Disable 1 = Enable	WINDOW (BOX) 0 = Disable 1 = Enable				COLOUR PAL 0=16 Col.N0-15 1=16 Col.N16-31	FLASH 0 = Enable 1 = Disable	Displ. Con. Reg. 1
4				Full.F.DRCS 0=Disab.R7 Co 1=Enab.R7 Co	WHITE BUTT 0=Attr.-> Enab 1=Attr.-> Def.	REVEAL 0=Conc -> Enab 1=Conc -> Disab	M 1 0 = Stack Code 1 = Explicit	M 0 0 = 40Ch\row 1 = 80Ch\row	Display Con. Reg. 2
5			N 5 10 Lin \ row —> Start (N + 15) 25 row —> Start N = 27	N 4	N 3	N 2 12 Lin \ row —> Start (N + 17) 21 row —> Start N = 24	N 1	N 0	Display Start (N in Lines) (Min = 3)
6			A 16 A 15 A 14 A 13 A 12 A 11 In Stack Mode 64 Pages with 1K Word addressable In Explicit or 80Ch\row 32 Pages with 1K Word addressable						Display Page Chapter Pointer (Ext. Addr.) Start Pos.
7			A 16 A 15 A 14 A 13 A 12 A 11 40 Ch\row —> 64 DRCS addressable } TABLE 4\5 80 Ch\row —> 32 DRCS addressable						DRCS Chapter A (Ext. Addr.) Full DRCS-Mode
8			A 16 A 15 A 14 A 13 A 12 A 11 only 40 Ch\row —> 64 DRCS addressable } TABLE 6+7 no FULL DRCS possible						DRCS Chapter B (Ext. Addr.) Start Position
9			C 5	C 4	C 3	C 2	C 1	C 0	CURSOR COLUMN (0-39)
			Greater		39 —>	Cursor is disabled			
10	T 2	T 1	T 0	F 4	F 3	F 2	F 1	F 0	CURSOR
	Character Table				32 FOREGROUND COLOUR CURSOR				
11		CURSOR FL. 0 = NOT FLASH 1 = FLASH	CURSOR CON. 0 = DISPLAY CUR. 1 = CONCEAL CUR.	B 4	B 3	B 2	B 1	B 0	CURSOR
					32 BACKGROUND COLOUR CURSOR				
10	Lining Cursor	P 6	P 5	P 4	P 3	P 2	P 1	P 0	CURSOR
				Character	Code	for CURSOR			
13	A 1 Addr. of the 4 DRCS-CLUT	A 0		D 4 (P)	D 3 (I) 1 = Full 0 = Half	D 2 (B)	D 1 (G)	D 0 (R) 00000 = Trans-parent	DRCS CLUT

Figure 10.29 Videotex Controller Registers (1 of 2)

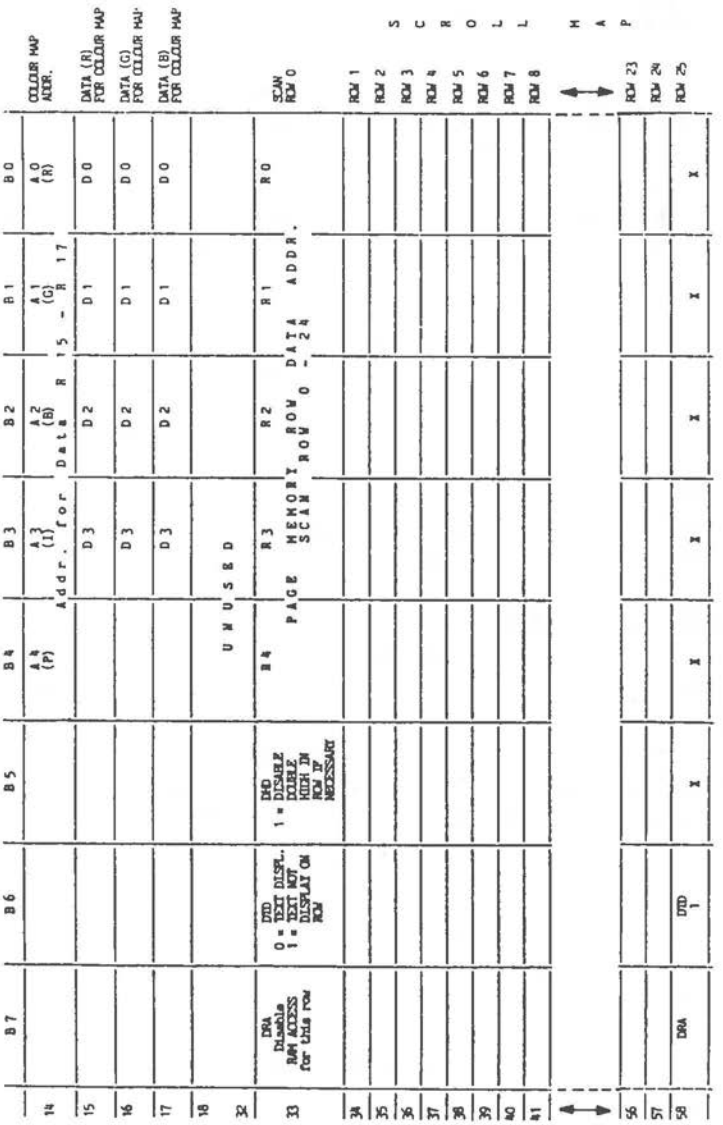


Figure 10.29 Videotex Controller Registers (2 of 2)

R10 CURSOR CHARACTER TABLE AND FOREGROUND COLOR
Initialization value 4EH
Determines the character table [0 = Table 1] and foreground color to be used for cursor display [Stack mode]. The contents of this register, or even the presence of a cursor, do not affect the rest of the cursor line.

R11 CURSOR ATTRIBUTES AND BACKGROUND COLOR
Initialization value 48H
Bits 5 and 6 enable/disable for the cursor the corresponding attributes currently in force. The contents of this register, or even the presence of a cursor, do not affect the rest of the cursor line.

R12 CURSOR LINING AND CHARACTER MODE Initialization value 7FH
Bits 0-6 specify the code of the cursor character in the Table specified in R10.
Bit 7 can be used to effect the Underlining attribute.

R13 DRCS CLUT
Bits 6-7 select one of the 4 color look-up table locations.
The remaining bits define the contents of that CLUT entry.

R14 COLOR MAP ADDRESS
Writes one of the 32 color map addresses with the data already held in R15, R16, and R17. To avoid display corruption, this command is best executed during non-display time (see R1, bit 2).

R15 (red), R16 (green), R17 (blue) COLOR MAP DATA
These three 4-bit values make up any one of 4096 colors. Only the 4 LSBs of these bytes are significant.

SCROLL MAP REGISTERS

SCROLLING

Videotex Controller registers R33 - R57 are dedicated to the Videotex scroll map. When writing page data from video character code memory to the screen, the controller looks at bits 0-4 of these registers, in order to see in what sequence the lines of data are to be written when the screen is next scanned.

Each of the scroll map registers should contain a unique value in the range 0 - 24 (a greater value would cause address wrap-round within the display Chapter).

The scroll map is useful if you wish to exchange the positions of lines on the screen. It saves you the software overhead of block moving in video character code memory. A complete re-organization of screen lines requires a maximum of 25 single byte write operations into the scroll map registers.

TEXT DISPLAY AND RAM ACCESS

Bit 5 set in a scroll map register disables any Double Height attribute for the line concerned. This facility would be used where the following line belongs to a scroll area, or where the line concerned is designated as the bottom display line.

Bit 6 set disables the text display for the line concerned. Any number of text rows may be displayed, provided that they form a contiguous block of display lines. (The visual effect of non-contiguous display lines can be achieved by use of the transparency.)
Bit 6 also has an effect on display colors: the line color of the lower screen fringe area (see "Color Layers") applies from the first line for which bit 6 is set (including this line), down to the bottom of the screen. The scroll map registers for these lines are then ignored.

Bit 7, the Disable Ram Access bit, determines, if set, that the Videotex Controller is not to fetch character and attribute information. This saves the Controller RAM access time for that line.

If bit 7 = 1 and bit 6 = 0 for the same line, that line is re-written on the next screen scan, even though RAM access for that line has been disabled. This is possible because the Videotex Controller holds screen output data in internal intermediate buffers, before finally transmitting them to the CRT. Note, however, that DRCS characters cannot be displayed without RAM access being enabled.

Videotex Controller register R58 is significant only as its bit 6 must be set, and bit 7 has an enabling/disabling effect regarding bit 6 of the registers R33 - R57: if bit 7 of R58 is set, the adoption of the lower screen fringe area color does not apply.

NCR SOFTWARE

The NCR Videotex package includes a number of executable files providing a user friendly interface conforming to the requirements of the specific country.
One of the files supplied provides an interface to useful Videotex Controller, VLP, and modem functions. This program, like the others in the Videotex package, is designed to run under the disk operating system. Once loaded, it remains in memory as a resident process, providing a library of routines which can be called from subsequently loaded processes. Refer to your Videotex User's document for the actual filename for your country.
The functions are executed in much the same way as disk operating system function calls: loading certain CPU registers, issuing an interrupt, and accepting return parameters.

* The function number is placed in BX

* The interrupt issued is type 75H

Other parameters are function-specific. Where 8-bit parameters are passed in a 16-bit CPU register, the upper 8 bits of the register are zero.

BX	Function	Entry/Return Parameters
0	Initialize to CEPT standard Clear out data of this process Fill character display memory with 20H Fill attribute and DRCS memory with 0 Locate cursor top left Decoder mode is serial (see function 1B)	
1	Return status of modem (automatic dialling)	Return: AX = 0 not ready 1 link established
2	Return type of modem (manual dialling)	Return: AX = 0 half duplex 1 full duplex
3	Discontinue modem connection	
4	Switch to CEPT standard	
5	Switch to PRESTEL standard	
6	Toggle line zoom (Double Height) in cursor line	
7	Switch to display interlace	
8	Switch to non-interlaced display	
9	Toggle attribute enable/disable (R4, bit 3)	
A	Toggle Conceal enable/disable (R4, bit 2)	
B	Enable cursor (R11, bit 5)	

BX	Function	Entry/Return Parameters
C	Disable cursor [R11, bit 5]	
D	Enable display, provided that Videotex mode is enabled [R3, bit 7]	
E	Disable Videotex display [R3, bit 7]	
F	Switch to Videotex mode [bit 7 0DC101H]	
10	Switch to PERSONAL COMPUTER mode, retaining Videotex register/ RAM status [bit 7 at 0DC101H]	
11	Toggle scan frequency	
12	Write a controller register	Entry: AX = register number DX = new register value
13	Read a controller register	Entry: AX = register number DX = register contents
14	Decode a CEPT or PRESTEL standard character	Entry: AX = character [see CEPT or PRESTEL spec.]
15	Transmit one character via modem	Entry: AX = character
16	Read one byte from Videotex RAM	Entry: AX = RAM address offset to paragraph 0D000H Return: AX = RAM byte

BX	Function	Entry/Return Parameters
17	Write one byte to Videotex RAM	Entry: AX = RAM location offset to paragraph 0D000H DX = byte for RAM
18	Set line (screen) color	Entry: AX = CLUT color number
19	Read cursor line number	Return: AX = line number
1A	Read number of current column	Return: AX = column number
1B	Read current decoder mode Parallel: attributes for a particular character code valid until changed for an occurence of that character Serial: attributes valid for all characters from current pos. to end of line, or until new attributes defined, or until parallel mode selected	Return: AX = 1 parallel 0 serial
1C	Read number of lowest line in scroll area	Return: AX = 0-18H line number > 18H no scroll area
1D	Read number of uppermost line in scroll area	Return: AX = 0-18H line number > 18H no scroll area

BX	Function	Entry/Return Parameters
1E	Define cursor as character from standard or DRCS table	Entry: DX = character table [0 - table 1, etc.] AX = character code 0-7FH or 80H: line cursor CX = foreground color no. in CLUT SI = background color no. in CLUT
1F	Transmit one character via video interface	Entry: AX = character Return: AX = 0 successful > 0 error
20	Read USART status for video interface	Return: AX = status byte
21	Set cursor position. Display attributes are unaffected	Entry: AX = line number DX = column number Return: AX = 0 successful > 0 error
40	Set address of modem status byte [Status is 1 = online, or 0 = offline]	Entry: DS:AX = segment:offset of 16-bit word
41	Read modem interface, returning character, if available	Entry: DS:AX = address of byte to receive char. Return: AX = 1 character read to specified address 0 no character read

BX	Function	Entry/Return Parameters
42	Disable display, load up to 4608 bytes into Videotex RAM, re-enable screen	Entry: DS:AX = address of first byte to be loaded DX = Videotex RAM address to which first byte to be loaded (offset to paragraph 0D000H) CX = number of bytes (max. 4608) Return: Bytes at specified Videotex RAM address
43	Read up to 4608 bytes from Videotex RAM	Entry: DS:AX = address to which first byte is to be loaded DX = Videotex RAM address of first byte to be copied (offset to paragraph 0D000H) CX = number of bytes Return: Bytes at specified PC memory address
44	Read one character from video interface	Entry: DS:AX = address of byte to receive char. Return: AX = 0 character read to specified address > 1 error

BX	Function	Entry/Return Parameters
45	Read character code and attributes for one character position. 12 bytes are read: 0 - character code 1..9 attribute bytes as detailed in Figure 10.24 10 - marked area 11 - protected area	Entry: AX = character position 40 * line + column or AX = 0FFFFH current cursor pos. DS:DX = destination addr. of first byte Return: 12 bytes starting at specified address
80	Write 40 text characters to line 25, also specifying "stack" mode attributes	Entry: DS:AX = address of first character DS:DX = address of first attribute Return: Text written to line 25
81	Scroll screen up one line, filling vacated line with 40 text characters according to "stack" mode attributes	Entry: DS:AX = address of first character DS:DX = address of first attribute CX = uppermost line of scrolled area SI = lowermost line of scrolled area Return: Display scrolled and new text written to vacated line
82	As function 81, except scroll is downwards	

PRINTING VIDEOTEX RAM

The Videotex program of your NCR software contains routines to dump Videotex RAM to an attached printer. Pressing Control-F1 from the main menu offers a further selection of print items. Using this selection, it is possible to print one or more of

* Character code memory
* Attribute Memory
* DRCS memory
* The Videotex Controller registers, including the scroll map registers
* The 32 12-bit RGB color values
* The line (screen) color as defined for each line

This facility is possible because switching from Videotex to the NCR PERSONAL COMPUTER mode of display does not reset the Videotex system. Therefore, the contents of Videotex RAM is not disturbed.

RS-232-C INTERFACES

The modem and video interfaces of the Videotex adapter are controlled by two separate 2651 USART integrated circuits, used in the asynchronous receiver/transmitter mode.
The registers of each of these Programmable Communications Interfaces (PCIs) are memory mapped in the address ranges 0DC200H-0DC207H (modem) and 0DC300H-0DC307H (video). The precise memory map for the modem interface is given below. Corresponding addresses for the video interface are mapped in the higher area.

Address (hex)	USART Register
DC200	Read Receive Holding Register
DC201	Read Status Register
DC202	Read Mode Register 1/2
DC203	Read Command Register
DC204	Write Transmit Holding Register
DC205	Write SYNC1, SYNC2, DLE Register
DC206	Transmitter/Receiver Mode Register 2
DC207	Write Command Register

The remaining sections of this Chapter deal with the programmable features of the PCI, followed by an explanation of its use by the Videotex and modem interfaces.

RS-232-C SOFTWARE CONTROL

The following asynchronous communications features can be programmed with the 2651 integrated circuit:

* 5, 6, 7, or 8-bit data characters
* 1, 1 1/2, or 2 stop bits
* Odd, even, or no parity
* Baud rates at -

50	1800
75	2000
110	2400
134.5	3600
150	4800
300	7200
600	9600
1200	19200

* Line break signal generation and detection
* False start bit detection
* Parity overrun, and framing error detection
* Internal diagnostics

The 2651 Programmable Communications Interface (PCI) serializes parallel data provided on the system data bus for transmission. At the same time, incoming serial data can be converted into parallel form for transmission to the system. PIC features are:

* Baud rate generator, generating up to 16 different baud rates
* Differentiation between communication technique information and data on both transmitter and receiver (TxD and RxD) lines
* Allows handshaking and status information exchange between microprocessor and connected device

PCI Receiver

Condition for receiving data is that the CD/ input line is low and that the RxEN bit in the Command Register is set. In asynchronous mode, the receiver waits for a high to low transition on RxD. One and a half bit lengths after this transition, RxD is sampled again. If it is now high the receiver reverts

to the waiting state. If RxD is low the receiver assumes that a valid start bit has been encountered.

The line is now sampled at one bit intervals until the pre-determined number of bits (data with parity and stop bits) have been received. The data part of this information is held in a Receive Holding Register (unused upper bits for a character length of less than eight are zero), the RxRDY bit in the Status Register is set, and the RxRDY/ line to the device goes low. Only after RxD has gone high again does the receiver start to wait for the next character.

The Status Register reflects errors of parity, framing, or overrun. If RxD is low all the time during character and stop bit receiving time, all bits zero are returned to the Data Holding Register and the Status Register recognizes a framing error.

PCI Transmitter

Data transmission can take place when the CTS/ input signal is low and the TxEN bit in the Status register is set. The PCI indicates to the microprocessor by means of the TxRDY/ signal and the TxRDY bit in the Status Register that it is ready to accept a character into its Transmit Holding Register for transmission. Having buffered the single character from the system data bus, the PCI resets both signal and status bit and holds them in that condition until the character has been serialized and transmitted. The sequence of asynchronous transmission bits is as follows:

One start bit
Pre-selected number of data bits, LSB first
Optional odd or even parity bit
Pre-selected number of stop bits

If a new character is then not available, the TxD signal remains high and the TxEMT/DSCHG bit in the Status Register is set. An all low (data, parity, and stop zero) transmission can be achieved by setting the Break bit in the Command Register.

Note: The 2651 synchronous transmitter/receiver mode cannot be utilized within the hardware configuration of the Videotex adapter.

Modem Interrupts

Interrupts are normally provided on the IRQ line selected by switch X6 (see Figure 10.8) whenever an incoming character has been assembled by the PCI and is available for reading by modem control software.

This hardware interrupt can be extended by means of switch X10, so that an interrupt is also issued whenever the PCI has completed serializaton and transmission of a character. This means that controller software need not continuously poll PCI status in order to determine whether the interface can accept the next character for transmission. (This facility is not required by NCR software, therefore X10 should not normally be closed.)

Programming the PCI

Communication between the CPU and 2651 registers takes place via memory addresses as set out in the memory map above. Figure 10.30 summarizes the Command, Status, and Mode Registers. The sequence of PCI programming is outlined in the following paragraphs.

MODE REGISTER 1

7	6	5	4	3 2	1	0
		Parity Type	Parity Control	Character Length	Mode and Baud Rate Factor	
ASYNCH: STOP BIT LENGTH 00 = Invalid 01 = 1 Stop bit 10 = 1 1/2 Stop bits 11 = 2 Stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 Bits 01 = 6 Bits 10 = 7 Bits 11 = 8 Bits	00 = Synchronous 1X rate 01 = Asynchronous 1X rate 10 = Asynchronous 16X rate 11 = Asynchronous 64X rate	
SYNCH: NUMBER OF SYN CHAR 0 = Double syn 1 = Single syn	SYNCH: TRANS-PARENCY CONTROL 0 = Normal 1 = Transparent					
Note (asynchronous mode) Baud rate factor applies only if external clock is selected. Factor is 16X if internal clock is selected.						

MODE REGISTER 2

7	6	5	4	3	2	1	0
0	0	Transmitter Clock	Receiver Clock	Baud Rate Selection			
		0 = External 1 = Internal	0 = External 1 = Internal	0000 = 50 Baud 0001 = 75 0010 = 110 0011 = 134.5 0100 = 150 0101 = 300 0110 = 600 0111 = 1200		1000 = 1800 Baud 1001 = 2000 1010 = 2400 1011 = 3600 1100 = 4800 1101 = 7200 1110 = 9600 1111 = 19,200	

Figure 10.30 PCI Registers (1 of 2)

COMMAND REGISTER

7 6	5	4	3	2	1	Ø
Operating Mode	**Request to Send**	**Reset Error**		**Receive Control (RxEN)**	**Data Terminal Ready**	**Transmit Control (TxEN)**
00 = Normal operation 01 = Asynch: atutomatic echo mode Synch: SYN and /or DLE stripping mode 10 = Local loop 11 = Remote loop	0 = Force $\overline{\text{RTS}}$ Output High 1 = Force $\overline{\text{RTS}}$ Output low	0 = Normal 1 = Reset error flag in status reg (FE, OE, PE/DLE detect)	**ASYNCH: FORCE BREAK** 0 = Normal 1 = Force break **SYNCH: SEND DLE** 0 = Normal 1 = Send DLE	0 = Disable 1 = Enable	0 = Force $\overline{\text{DTR}}$ Output High 1 = Force $\overline{\text{DTR}}$ Output low	0 = Disable 1 = Enable

STATUS REGISTER

7	6	5	4	3	2	1	Ø
Data Set Ready	**Data Carrier Detect**	**FE/SYN Detect**	**Overrun**	**PE/DLE Detect**	**TxEMT/DSCHG**	**RxRDY**	**TxRDY**
0 = $\overline{\text{DSR}}$ input is high 1 = $\overline{\text{DSR}}$ input is low	0 = $\overline{\text{DCD}}$ input is high 1 = $\overline{\text{DCD}}$ input is low	**ASYNCH:** 0 = Normal 1 = Framing error **SYNCH:** 0 = Normal 1 = SYN char detected	0 = Normal 1 = Overrun error	**ASYNCH:** 0 = Normal 1 = Parity error **SYNCH:** 0 = Normal 1 = Parity error or DLE char received	0 = Normal 1 = Change in $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$, or transmit shift register is empty	0 = Receive holding reg empty 1 = Receive holding reg has data	0 = Transmit holding reg busy 1 = Transmit holding reg empty

Figure 10.30 PCI Registers (2 of 2)

Programming Asynchronous Communications

1. Load Mode Register 1.
   The frequency multiplier in the two LSBs is significant only if Mode Register 2 selects an external clock. (If these bits are zero, value 2 for stop bit length defaults to 1 stop bits for transmission.)

2. Load Mode Register 2.
   This step may be omitted, if external clock signals are being used.

3. Load Command Register.
   Disabling the transmitter (bit 0) does not prevent output of character currently in the Transmit Shift Register. Disabling the receiver (bit 2) results in the loss of a character in the Receive Holding Register which has not yet been completely assembled.
   Setting bit 3 forces the TxD line low and holds it low until this bit is reset (space after a transmitted character). TxD then goes high for at least one bit time before transmission of the next character.
   Bit 4 (reset error flags in Status Register) is automatically reset after command execution.
   Bits 6 and 7 govern operational mode. 00 is normal (transmitter and receiver independent of one another). Automatic Echo means that data received, including the first character of a break condition, is echoed on the TxD line , provided that transmitter and receiver clock are identical (bit 2 must be set, but bit 0 need not be set, as CPU transmit commands are ignored).

4. Check Status Register (optional, see below).

5. Issue new command, without loading Mode Register(s)

   or

Issue command to disable receiver and transmitter bits 0 and 2 of Command Register, and return to step 1. for a new communications set-up.

Diagnostics

Bits 6 and 7 in the Command Register can be used for communications diagnostics.
Local Loop sets up the following configuration:

Transmitter output is connected to receiver input
DTR/ is connected to CD/
RTS/ is connected to DTS/
DTR/, RTS/, and TxD/ are held high
CTS/, CD/, DSR/, and RxD/ are ignored
Receive clock = transmit clock

provided that Command Register bits 0, 1, and 5 are set (bit 2 is ignored).

Remote Loop has the following effect:

Data assembled in the Receive Hold Register is placed in the Transmit Hold Register and transmitted on the TxD line
No data transfer from the PCI to system data bus, but parity, overrun, and framing errors are detected
RxRDY/, TxRDY, and TxEMT/DSCHG/ (not connected) are held high
Command Register bit 0 is ignored
Other signals operate normally
Transmit clock = receive clock.

Status Register
The significance of the Status Register bits:

0 - When set, data for transmission can be accepted from the system data bus and the TxRDY/ line is low.
In Automatic Echo and Remote Loop modes this bit is not set and the TxRDY/ line is held high.

1 - When set a complete character in the PCI is waiting to be read and RxRDY/ is low.
This bit is zero when the character is read, or when Command Register bit 2 disables the receiver.

2 - When set DSR/ or CD/ has changed state, or following transmission of a character there is no new character waiting in the Transmit Holding Register.
This bit is zero when the transmitter is enabled by Command Register bit 0, or when the Status Register is read.

3 - Assuming parity is enabled, this bit set indicates a parity error.
This bit is zero following disabling of the receiver or issue of a command with Command Register bit set.

4 - An overrun error means that a new character was accepted by the Receive Holding Register, although the previously assembled character had not yet been read by the microprocessor.
This bit is zero following disabling of the receiver or issue of a command with Command Register bit set.

5 - If this bit is set, a received character in asynchronous mode was not framed by the specified number of stop bits (If 1 1/2 stop bits were specified, only the first stop bit is checked).

6 - Reflects the state of the CD/ line.

7 - Reflects the state of the DSR/ line.

ADAPTER INITIALIZATION

This section outlines initialization of the 2651 and the basic transmit/receive interfaces, as used by the NCR Videotex software.

MODEM INTERFACE

The modem interface is interrupt driven, using the interrupt request line 2 of the Programmable Interrupt Controller of the NCR PERSONAL COMPUTER (interrupt type 0AH). The interrupt service routine checks for overrun, parity, and framing errors by reading the Status Register. If any of the corresponding bits (3, 4, or 5) is set, an error condition is recognized, otherwise the routine proceeds to read a character from the interface.

Initializing Mode Register 1: 0100 1110 B --> 0DC206H

* Mode is asynchronous
* 1 Stop Bit
* No Parity
* Character length is 8 bits
* External baud rate is 16 x

Initializing Mode Register 2: 00?1 0111 B --> 0DC206H

* Transmitter clock is external (bit 5 zero), where baud rate is 75
* Transmitter clock is internal (bit 5 set), where baud rate is 1200
* Receiver clock is internal
* Internal baud rate is 1200

Initialize Command Register: 00?0 0111 B --> 0DC207H

* Normal operation (no echo, no diagnostics)
* RTS/ (bit 5) is set (active) for 1200 transmission baud rate, otherwise zero
* Reset error is normal
* Asynchronous force break is normal
* Receive control is enabled
* DTR/ is active
* Transmit control is enabled

Receiving one byte from interface (in response to interrupt):

* Read byte from Receive Holding Register <— 0DC200H

Transmitting one byte to modem:

* Disable all other IRQ lines to Programmable Interrupt Controller 0000 0100 B —> port 21H
  Modem transmission can now be interrupted only by a modem service request
* Check 2651 Status Register: <— 0DC201H
  If modem data is waiting to be read (bit 1 set), check status for error (bits 3, 4, 5); read character or process error
  Otherwise, check that the Transmit Holding Register is available (empty, bit 0 set). If a timeout situation is evident, check the on-line status of the modem by means of the DSR/ bit (7). This bit reflects the status of the modem M1 (manual dialling modem) or S (automatic dialling modem) signal: modem is not active if this bit is set
* Assuming no error and that the modem is available, write the byte to the Transmit Holding Register —> 0DC204H
* Re-enable IRQ lines 1111 1111 B --> port 21H

Error consequences:

* If error is non-availability of modem, close down communication by disabling receiver and transmitter, and forcing DTR/ and RTS/ high 0000 0000 B --> 0DC203H
  Restore interrupt type 0AH to its former function.

* If error is parity/framing/overrun, handle error as appropriate. There is normally no need to reset the error status bits (3, 4, 5) explicitly (see Command Register, bit 4)

Remember that the Videotex protocol includes not only displayable characters, but also single control characters and control sequences. The codes for control characters and sequences are set out in the Videotex specification available for your country.

VIDEO INTERFACE

The interface procedures for the video (video recorder, VLP) interface are similar to those controlling the modem interfaces. The main difference is that the video interface does not make use of interrupt handling.

Initializing Mode Register 1: 0100 1110 B --> 0DC306H

* as for modem interface

Initializing Mode Register 2: 0011 0111 B --> 0DC306H

* Transmitter clock is internal
* Receiver clock is internal
* Baud rate is 1200

Initializing Command Register: 0010 0111 B --> 0DC307H

* Normal operation (no echo, no diagnostics)
* RTS/ is active
* Reset error is normal
* Asynchronous force break is normal
* Receive control is enabled
* DTR/ is active
* Transmit control is enabled

Receiving one byte from video interface:

* Check Status Register bits 1, 3, 4, and 5 for byte unavailability/parity/overrun framing errors <— 0DC301H
* Assuming no error, read byte <— 0DC300H

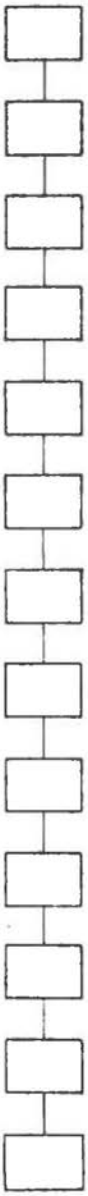
Transmitting one byte to video interface:

* Check Status Register bit 0 to see if the Transmit Holding Register is available (empty) <— 0DC300H
* Include a timeout check on the availabilty of this register
* When Transmit Holding Register is available, write one byte to this register —> 0DC304H

Details of the control/data protocol with the video unit vary according to the hardware characteristics of that device. If you are intending to design your own interface software, you should refer to relevant documents of the video unit.

SWITCHES AND REGISTERS

Figures 10.31 to 10.37 illustrate the sequence of software commands for Videotex switching and register access operations.



DC100/7 = 1 Screen Enable

DC101/7 = 1 Video Controller to Scree

DC103/7 = 0 Monitor 15 kHz H-Sync.

DC000/7 = 1 VCS Input on Pin 31

DC000/6 = 1 1MHz on Pin 29

DC000/5 = 1 FS/Output Pin 33

DC000/4 = 1 Sync Master

DC000/2 = 0 PLL enable

DC000/1 = 1 VCS is used Pin 31

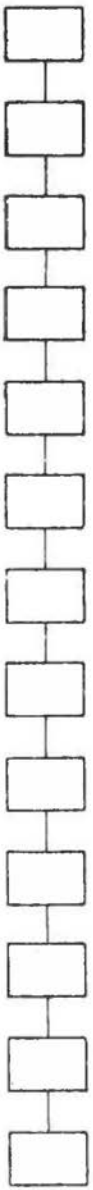
DC000/0 = 1 Interlace

DC003/5 = 1 WBEN

DC003/6 = 1 TEXT Enable

DC105/7 = 1 Enable VLP

Figure 10.31 Switch To Videotex - 50 Hz



DC100/7 = 1 Screen Enable

DC101/7 = 1 Videotex Controller to Screen

DC103/7 = 1 Monitor 18 kHz H-Sync

DC000/7 = 1 VCS Input on Pin 31

DC000/6 = 1 1MHz on Pin 29

DC000/5 = 1 FS/Output on Pin 33

DC00004 = 1 Sync Master

DC000/2 = 0 PLL Enable

DC000/1 = 1 VCS is used on Pin 31

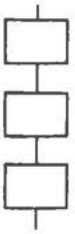
DC000/0 = 0 non Interlace

DC003/5 = 0 WBEN

DC003/6 = 1 TEXT Enable

DC105/7 = 0 Disable VLP

Figure 10.32 Switch To Videotex - 60 Hz

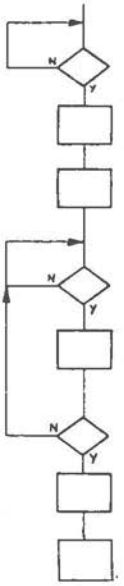


DC101/7 = 0 Switch to IBM

DC003/7 = Monitor 15 kHz H-Sync

DC103/7 = 0 Disable VLP

Figure 10.33 Switch to NCR PERSONAL COMPUTER mode



DC100/7 = 1 Buffer empty?

DC100/7 = 0 Output Disable

DC003/6 = 0 Text Disable

DC100/7 = 1 Buffer empty?

D0000- Write all Page

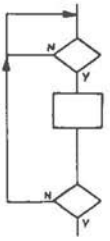
D3FFF- Memory data

last data

DC100/7 = 1 Output Enable

DC003/6 = 1 Text Enable

Figure 10.44 Fast Write To Page Memory

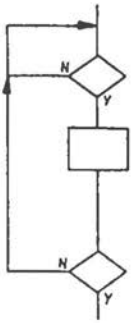


DC100/7 = 1 Buffer empty

Write Page Memory data

last Write?

Figure 10.35 Slow Write To Page Memory

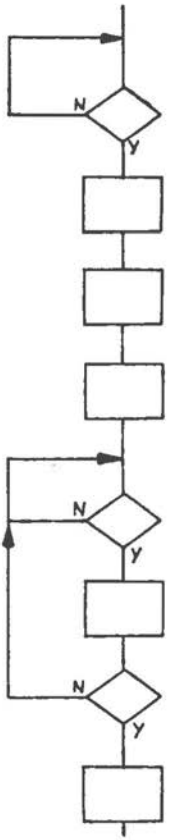


DC100/7 = 1 Buffer empty?

DC000 to
DC01F Write Register

last Write?

Figure 10.36 Write Videotex Controller Register



DC100/7 = 1 Buffer empty?

DC104/7 = 1 Wait Field Sync.

Write first Scroll Reg.

DC104/7 = 0 Don t wait Field Sync.

DC100/7 = 1 Buffer empty?

DC020 to
DC03A Scroll Reg.

last Byte

DC104/7 = 1 Wait Field Sync.

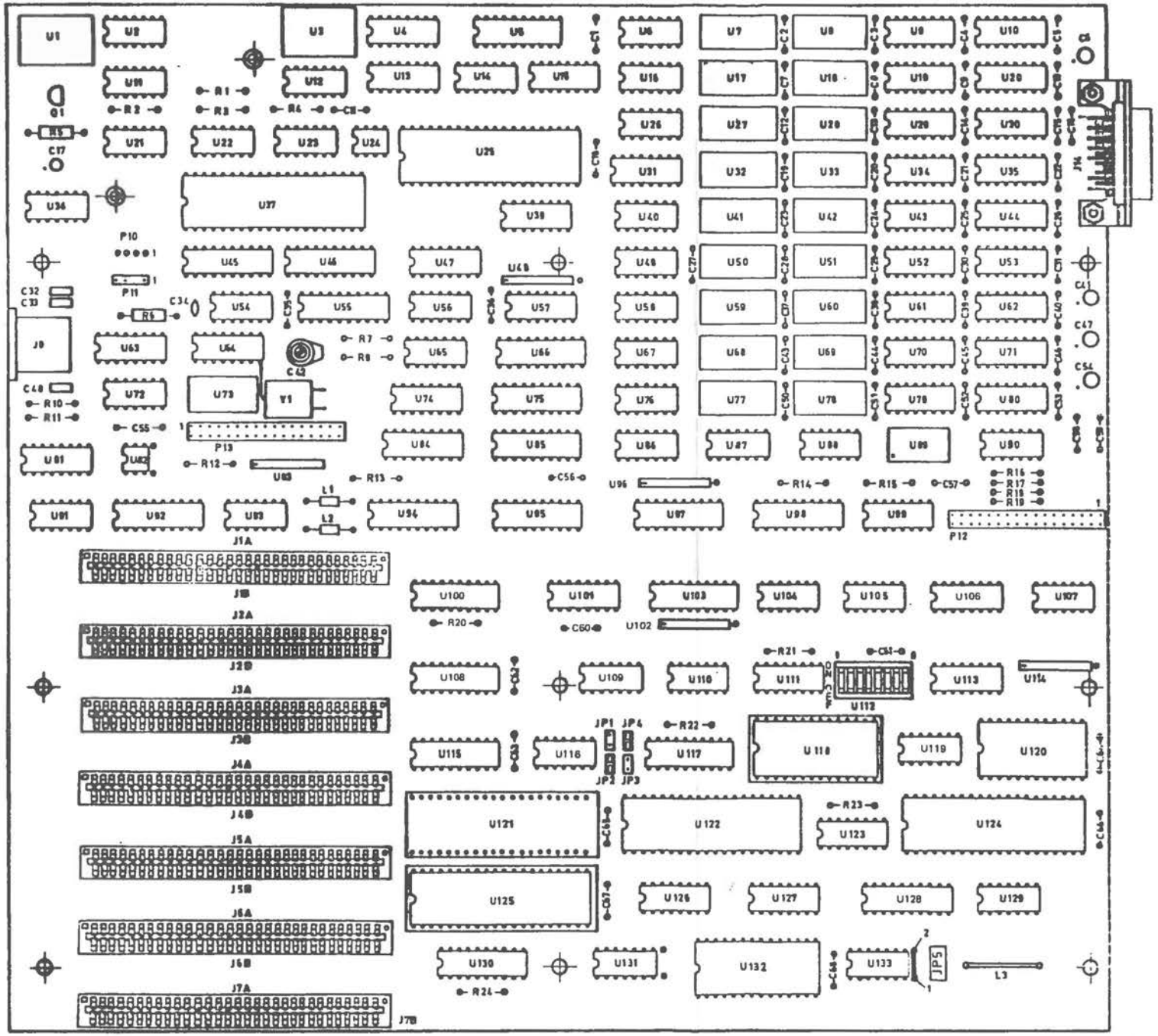
Figure 10.37 Write Scroll Map Register

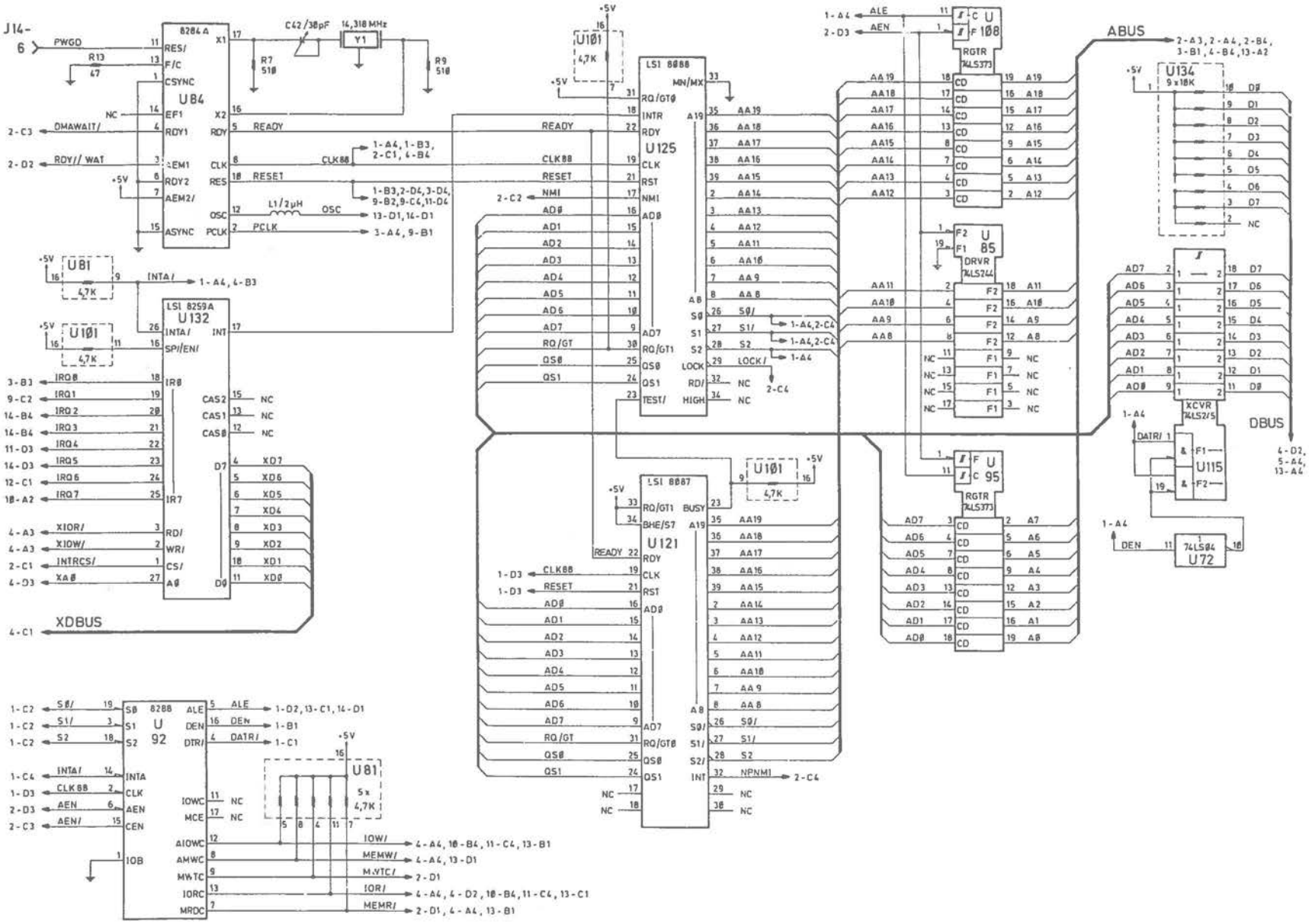
Index

LAYOUT AND SCHEMATICS

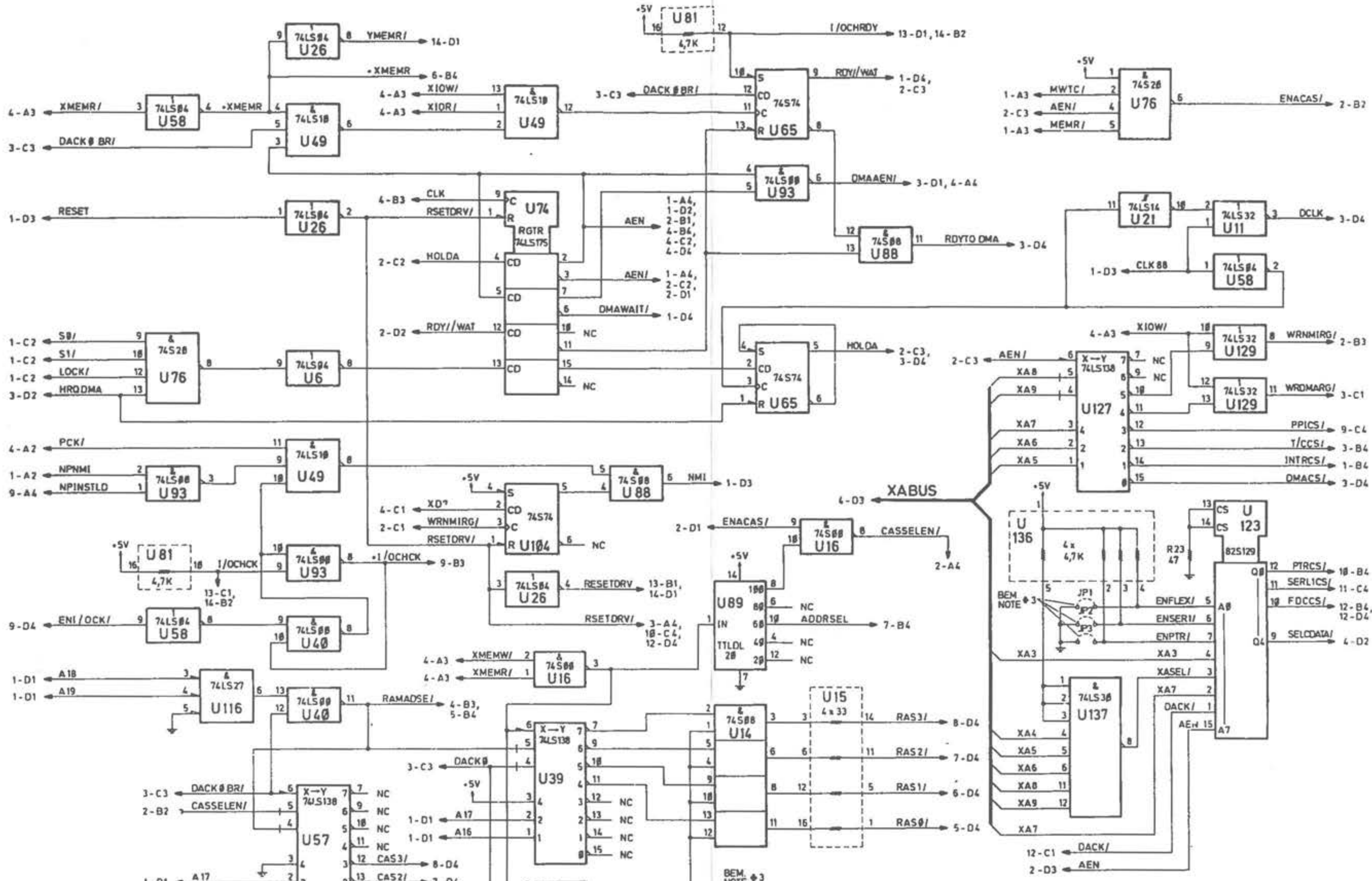
CONTENTS

Early-type Main Processor Board





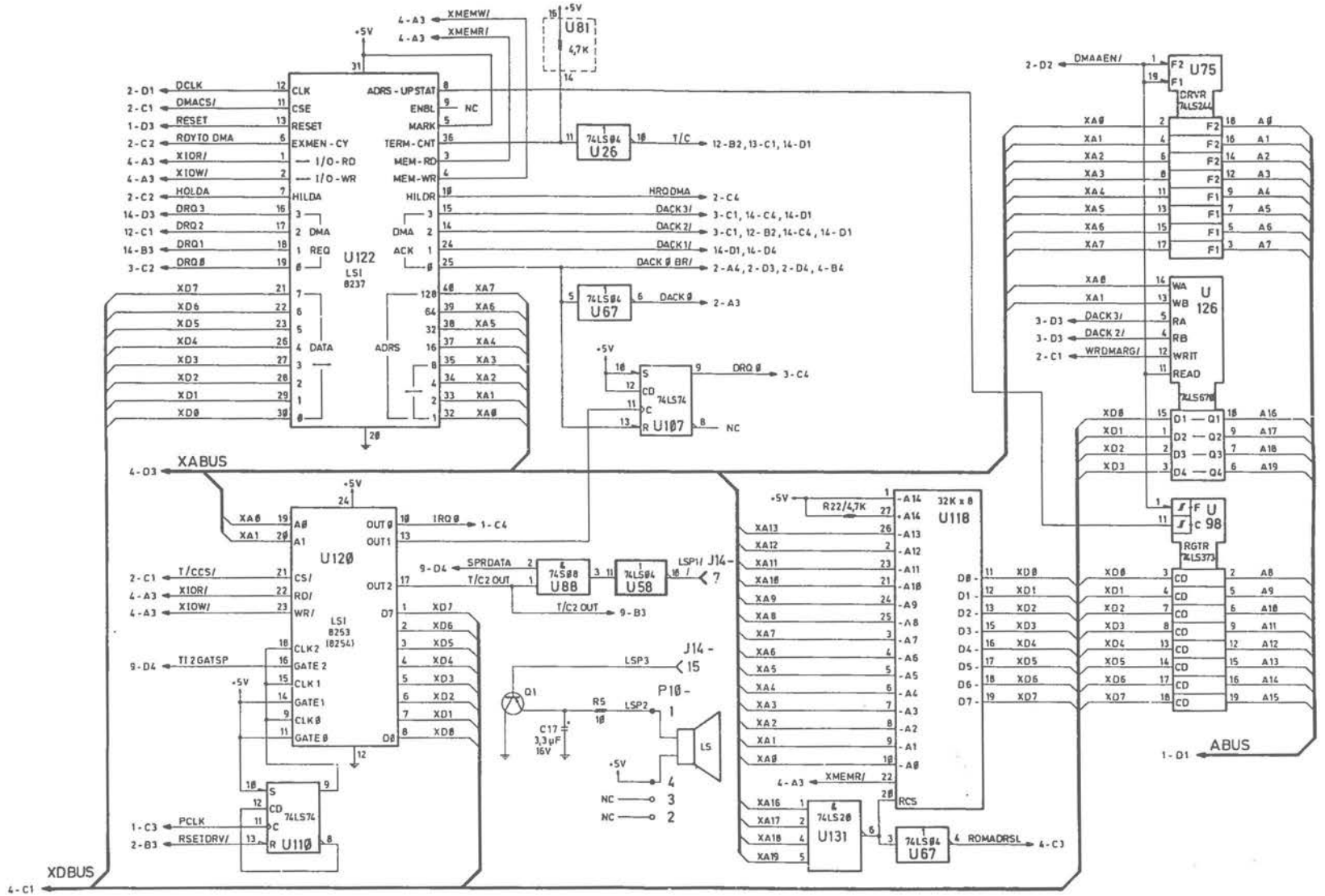
Early-type Main Processor Board (1 of 14)



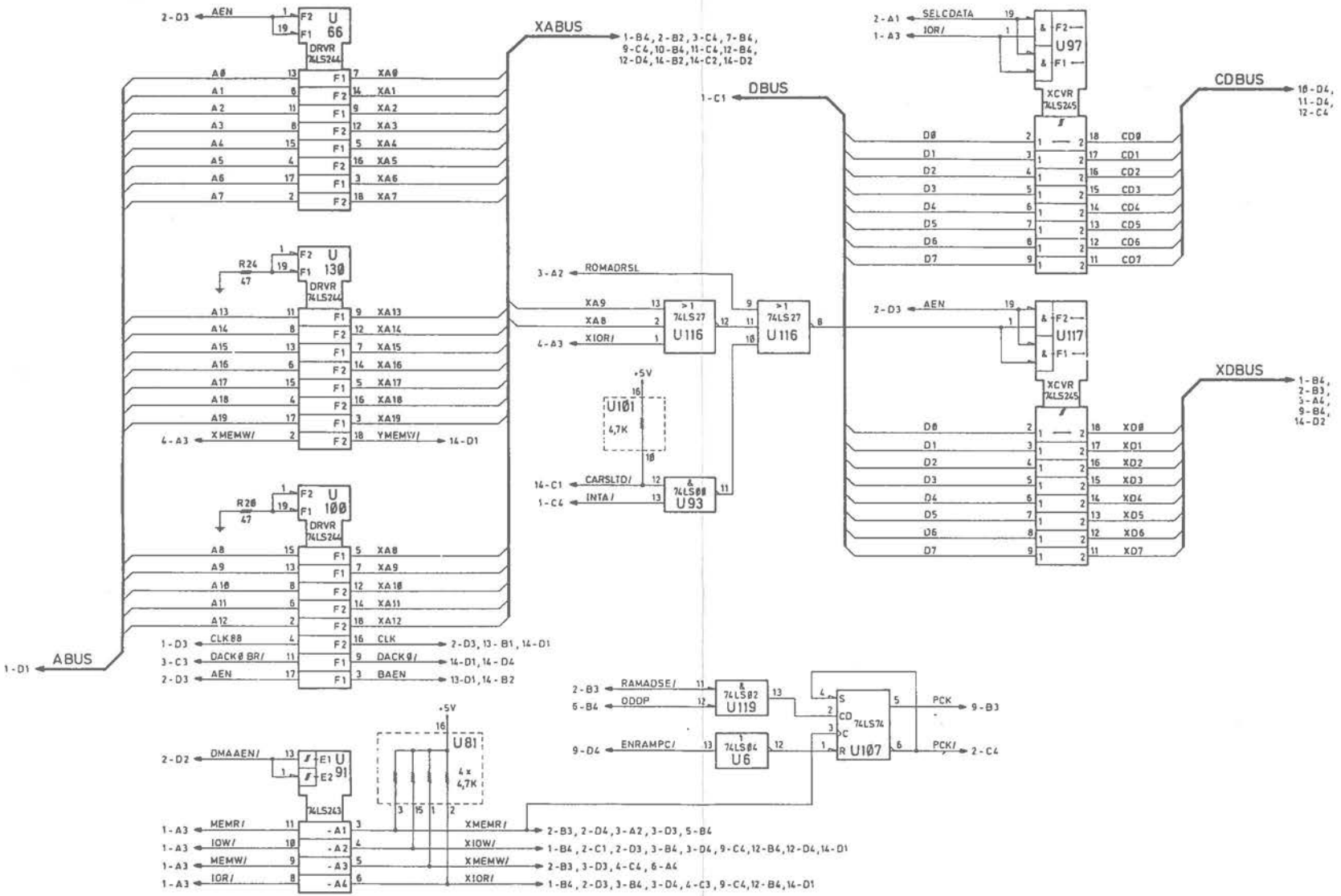
BEM. NOTE ♦3

	CLOSED		
	JP1	JP2	JP3
FLEX DISK	X		
SERIAL I/F		X	
PRINTER I/F			X

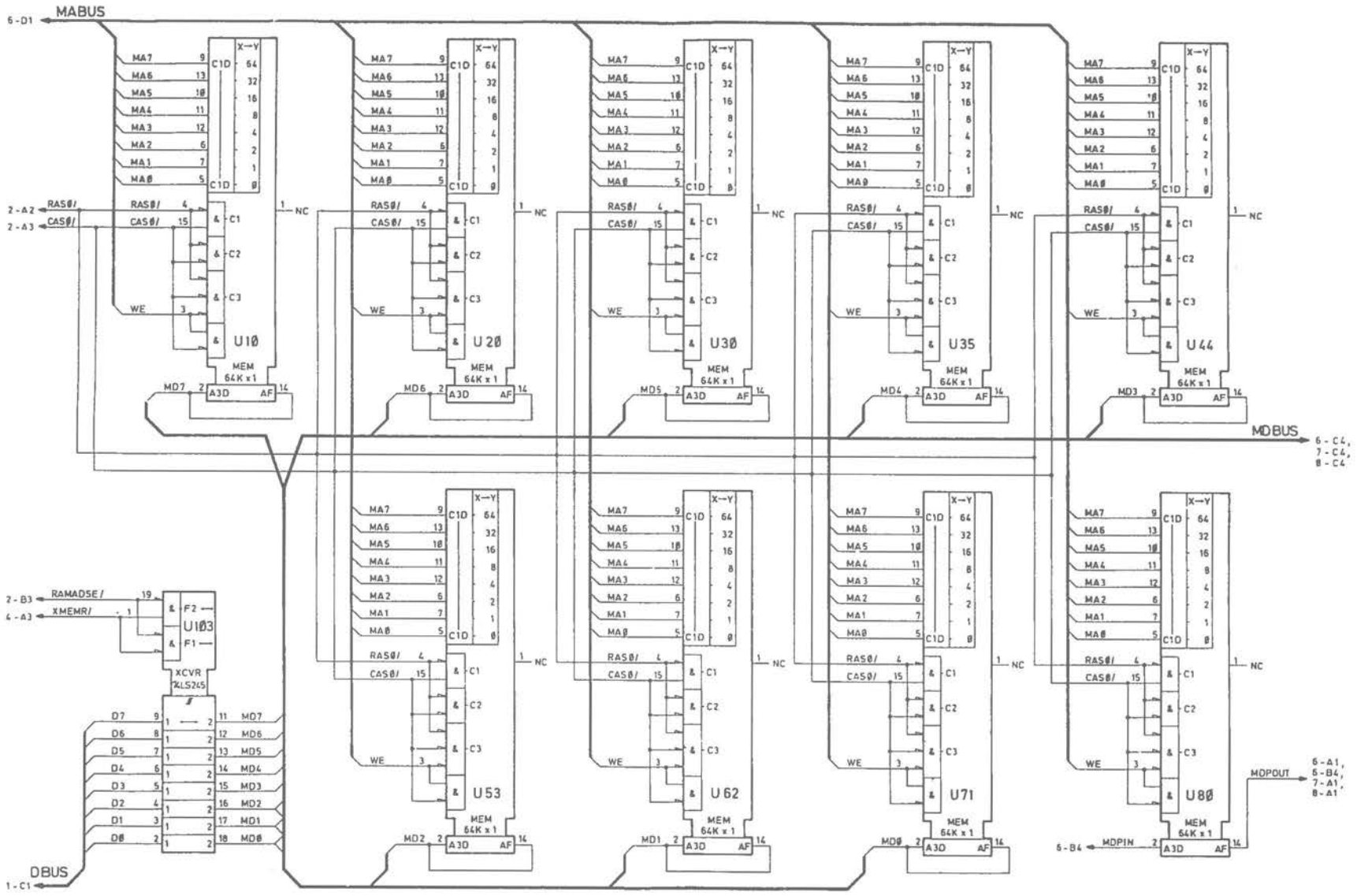
Early-type Main Processor Board (2 of 14)



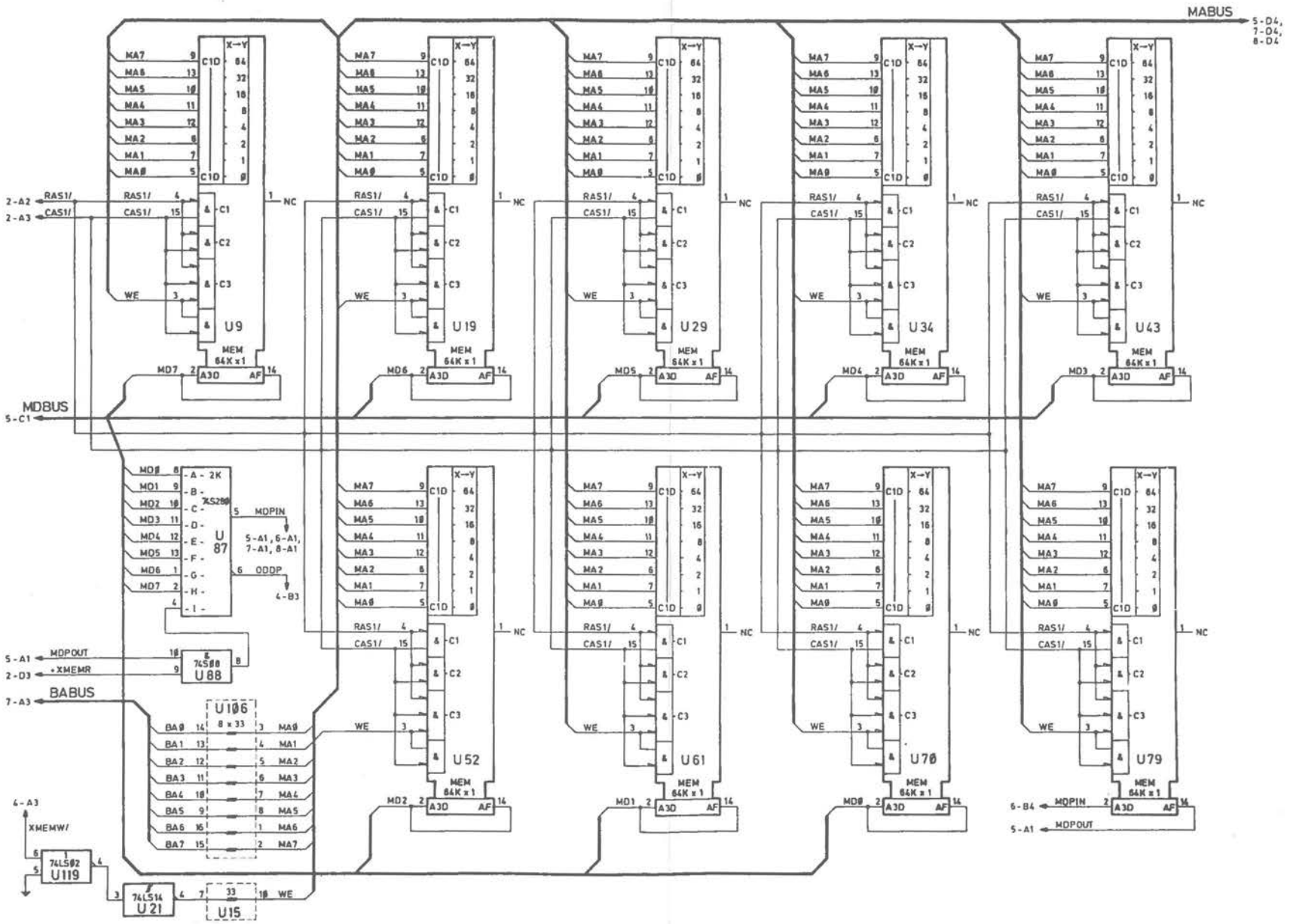
Early-type Main Processor Board (3 of 14)



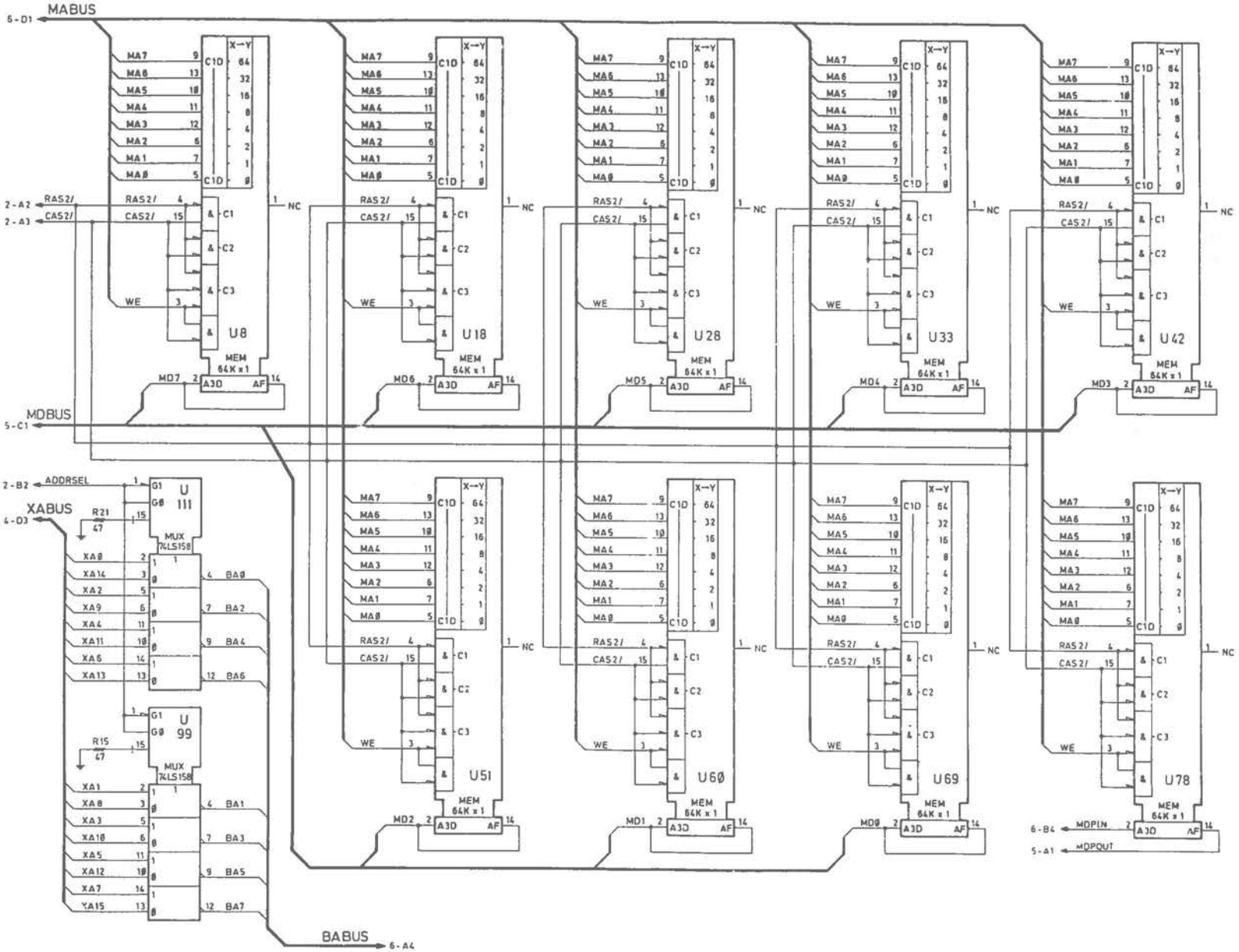
Early-type Main Processor Board (4 of 14)



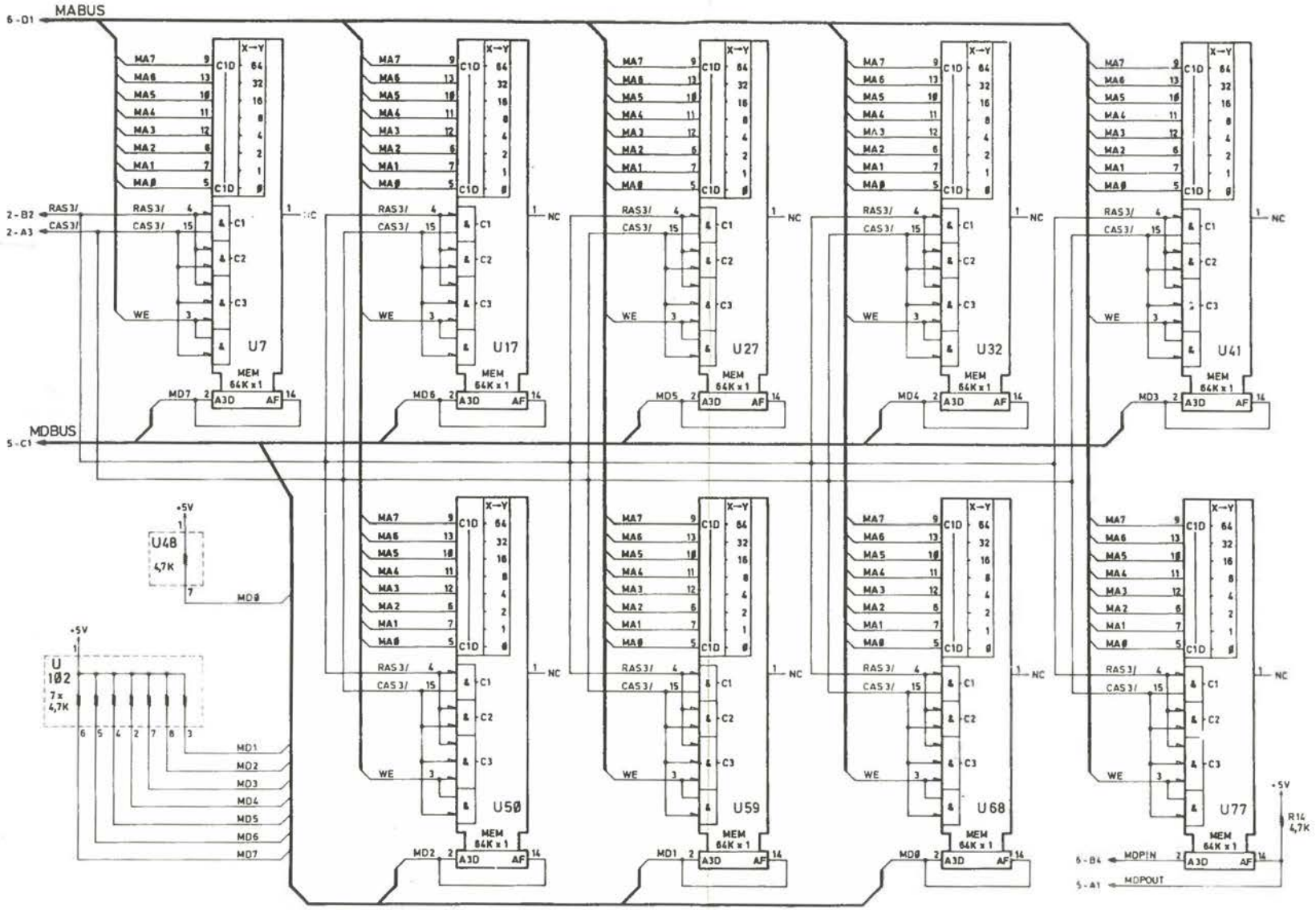
Early-type Main Processor Board (5 of 14)



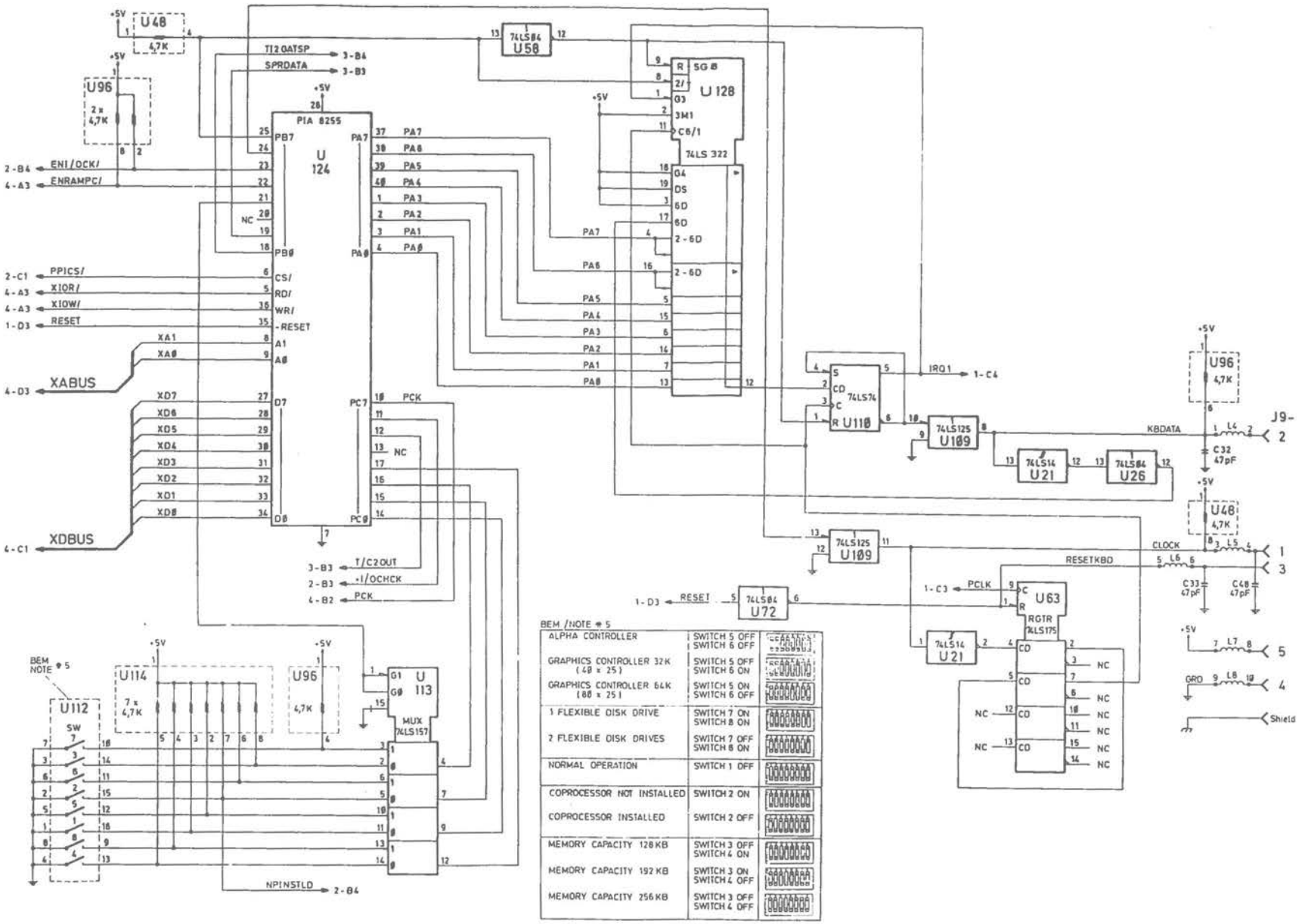
Early-type Main Processor Board (6 of 14)



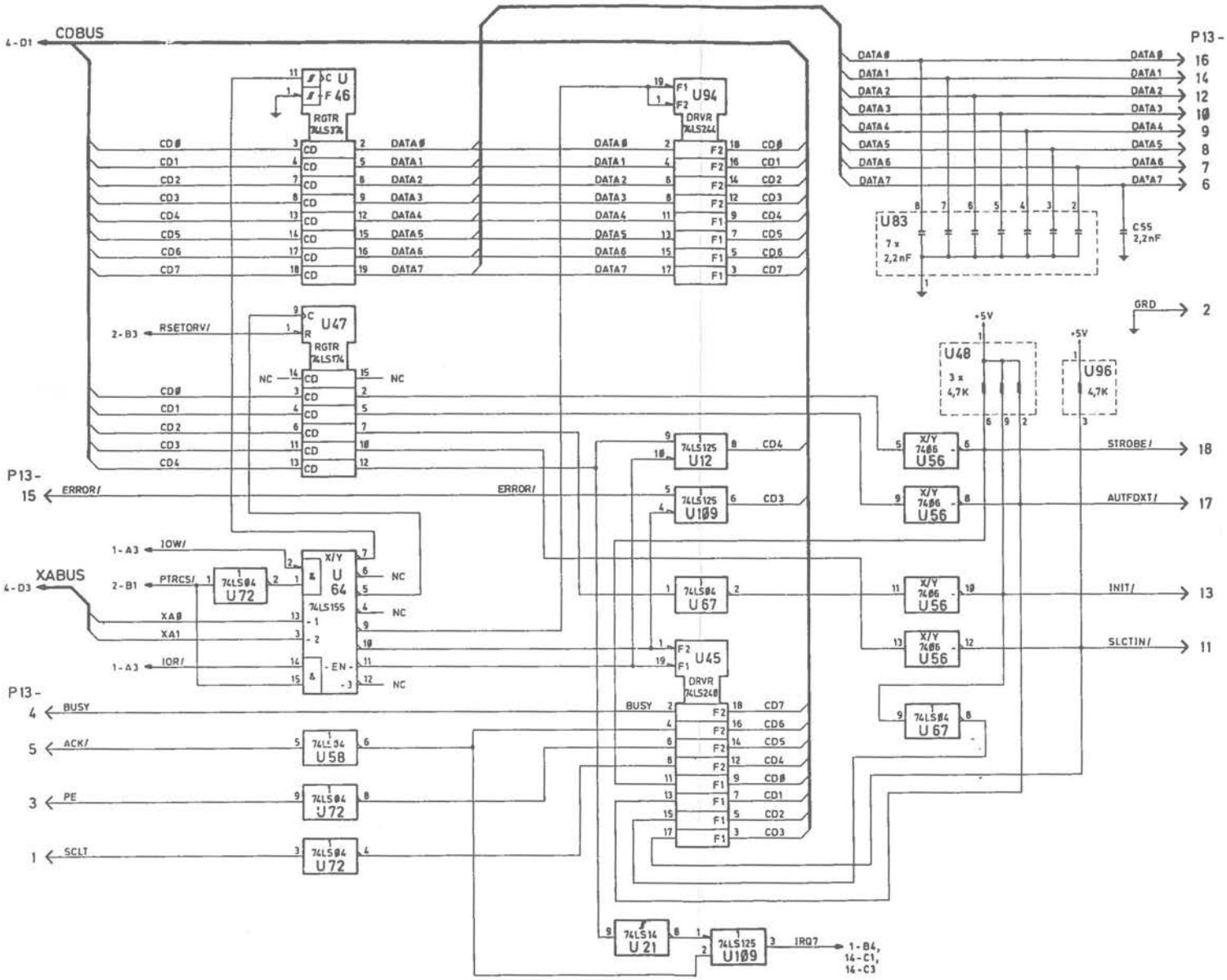
Early-type Main Processor Board (7 of 14)



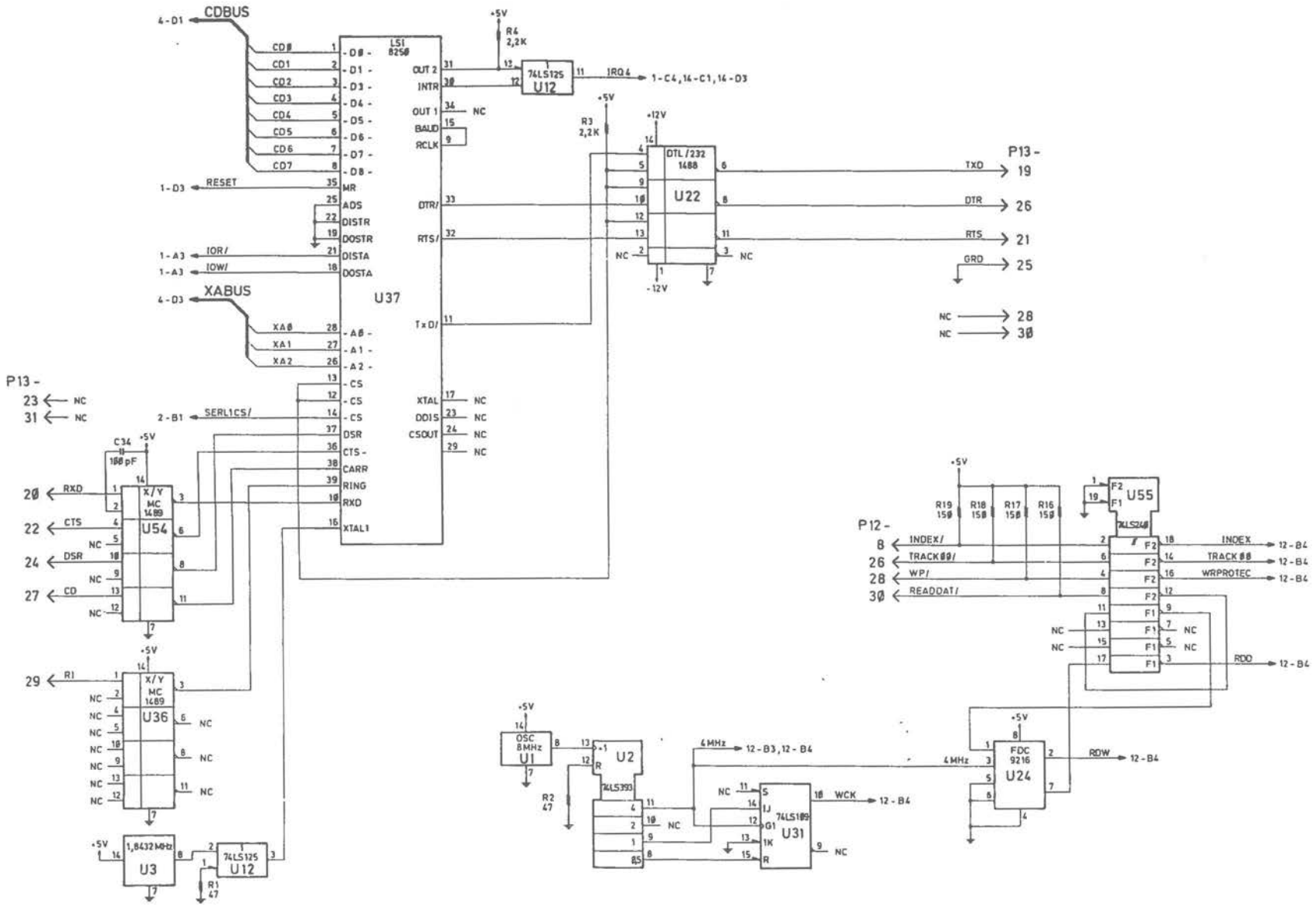
Early-type Main Processor Board (8 of 14)



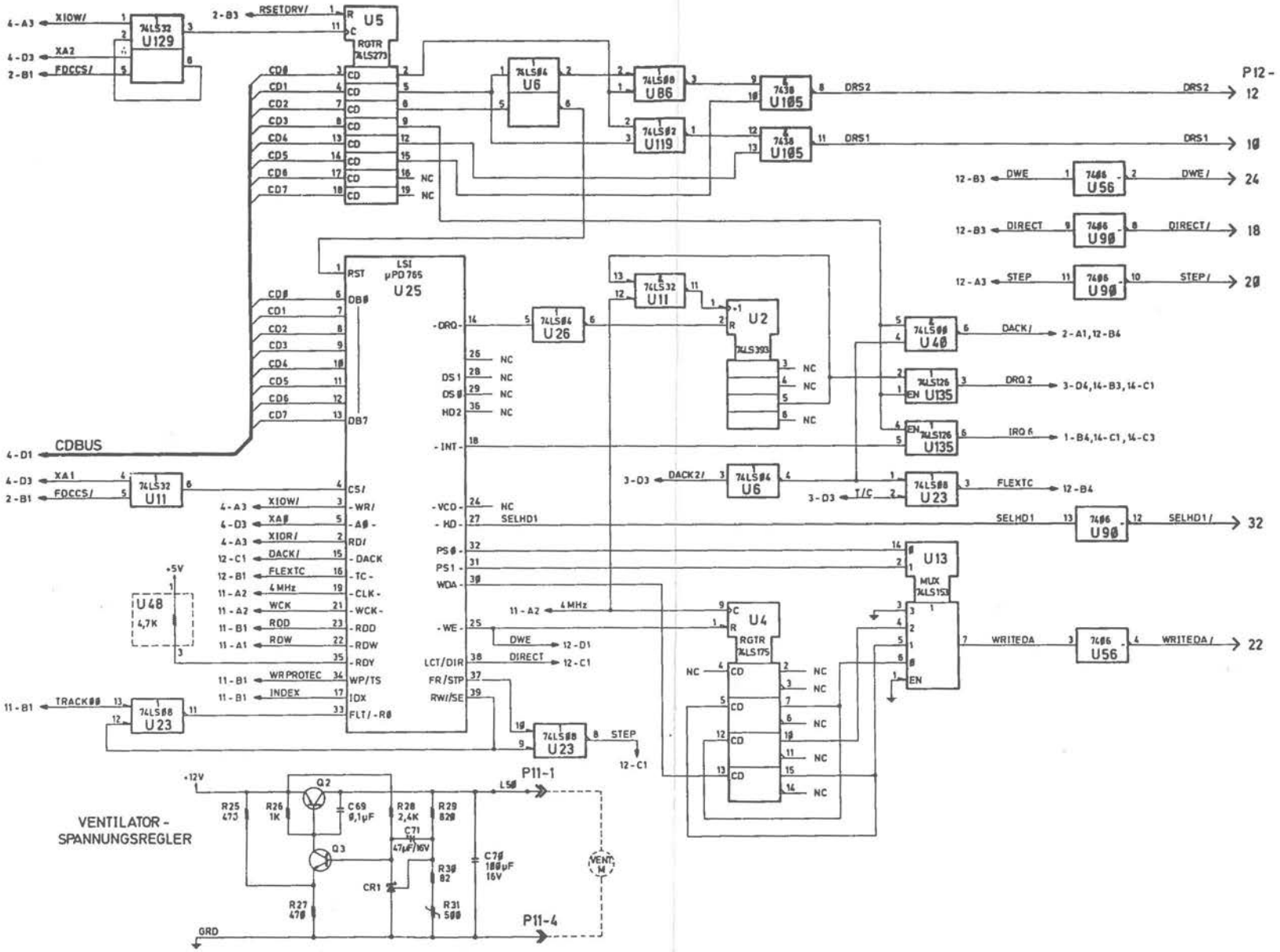
Early-type Main Processor Board (9 of 14)



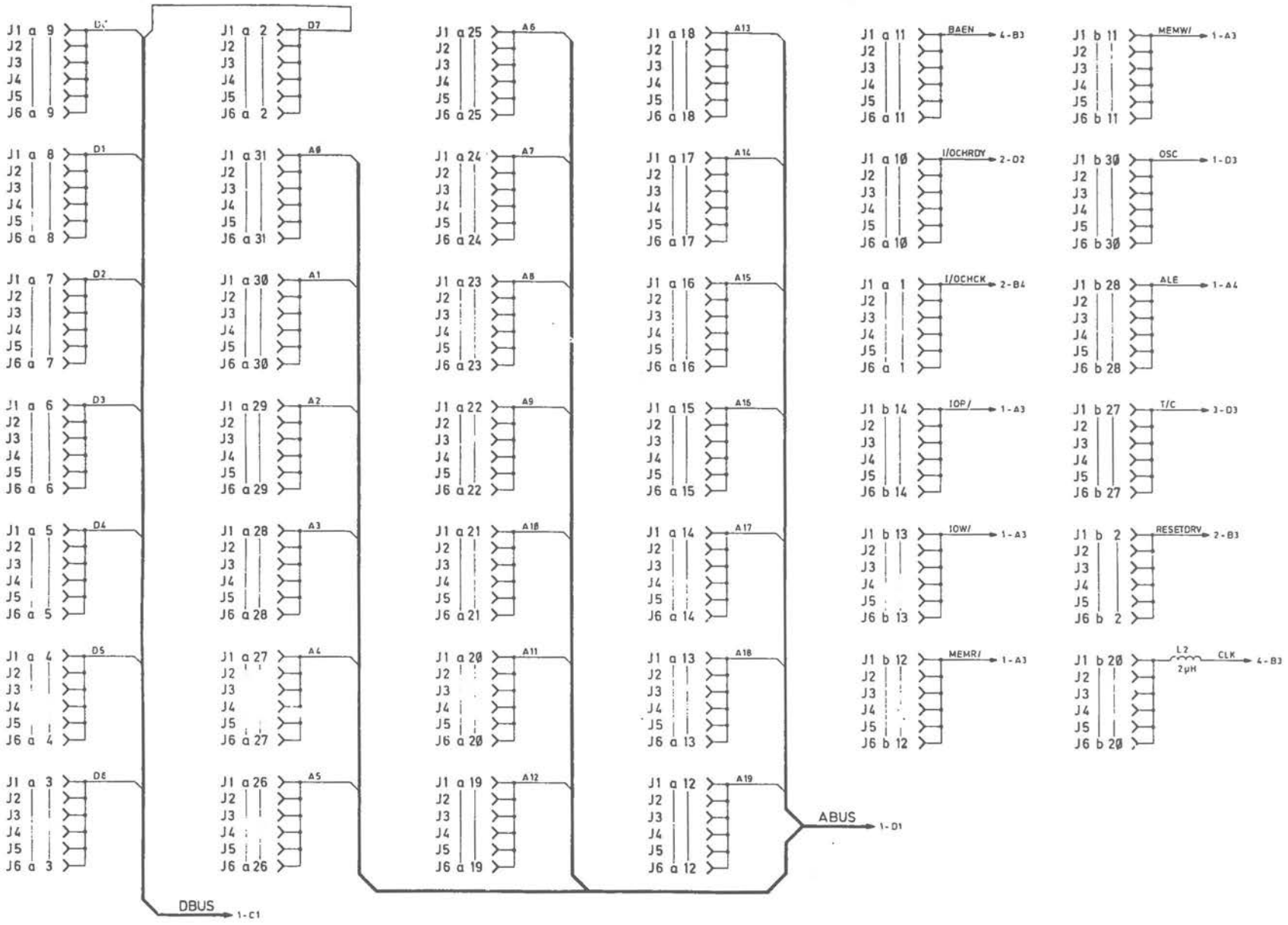
Early-type Main Processor Board (10 of 14)



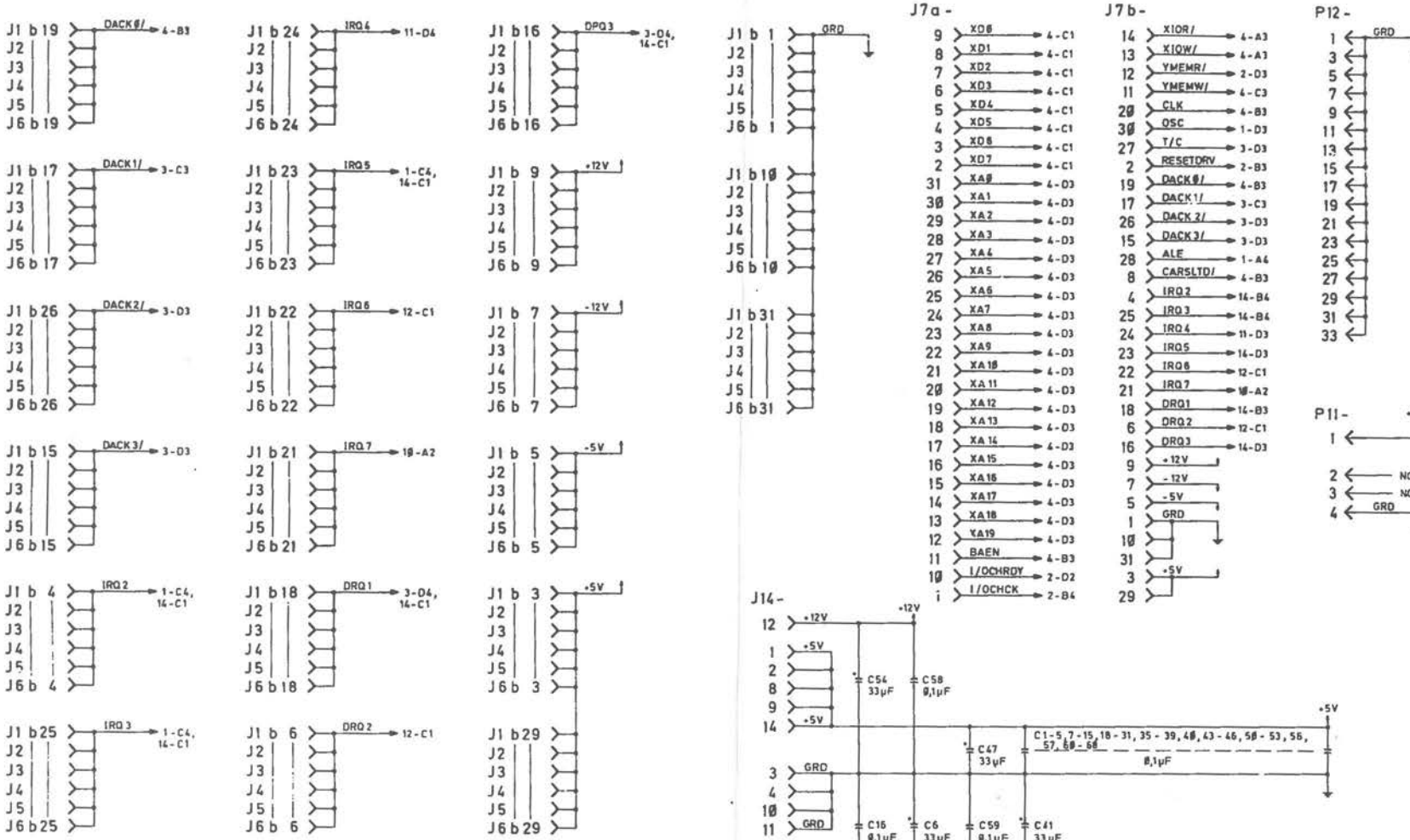
Early-type Main Processor Board (11 of 14)



Early-type Main Processor Board (12 of 14)

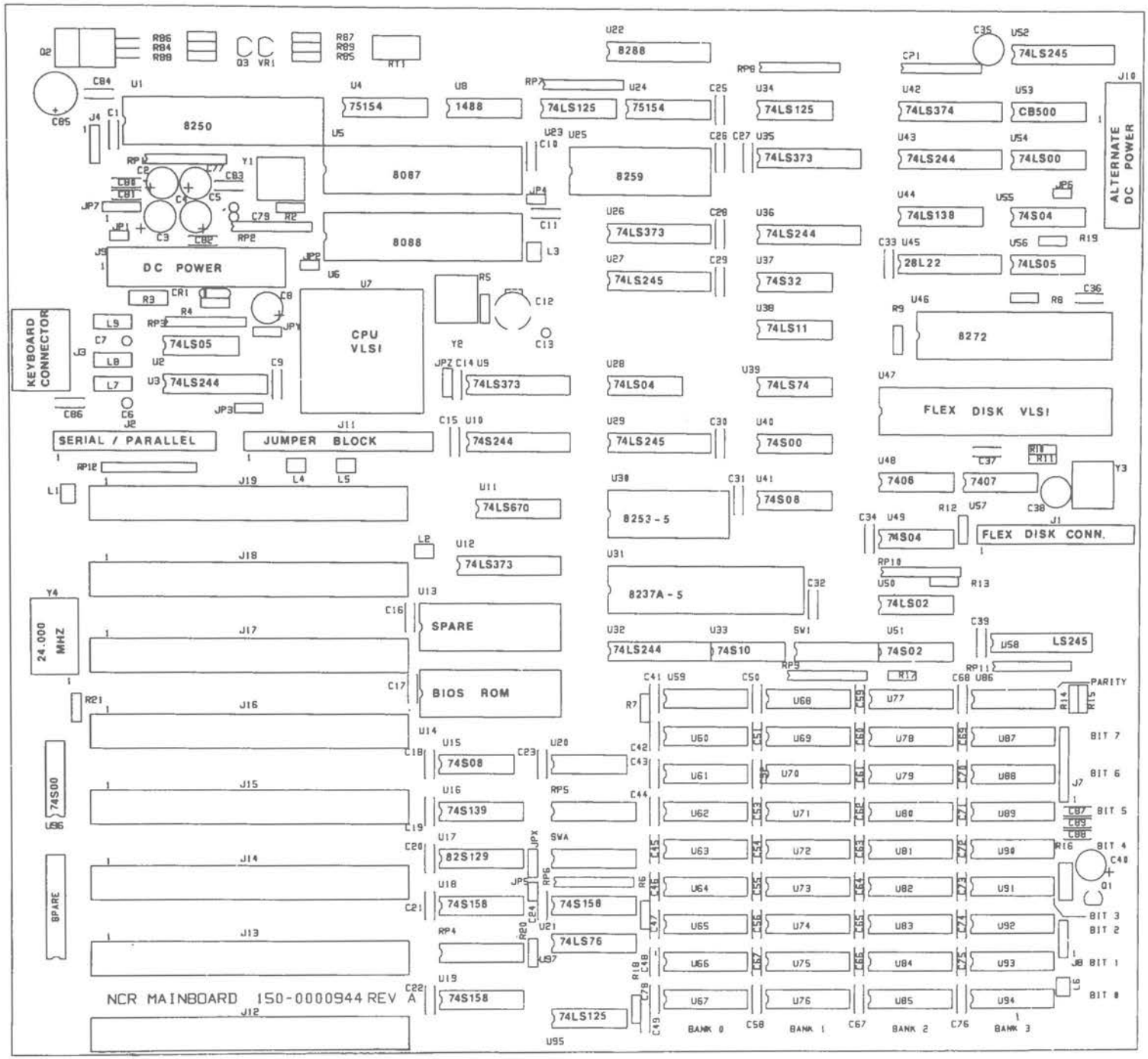


Early-type Main Processor Board (13 of 14)



Early-type Main Processor Board (14 of 14)

Alternate Main Processor Board



LOGIC DIAGRAMS

U NUMBER	PIN NUMBERS +5	GND	UNUSED
1	40	20	
2	14	7	
3	20	10	
4	15	8	
5	40	1,20	
6	40	1,20	
7	30 59	26,9,43	
8	—	—	+12V - PIN 14 -12V - PIN 1
9	20	10	
10	20	10	
11	14	7	
12	20	10	
13	28	14	
14	28	14	
15	14	7	
16	16	8	
17	16	8	
18	16	8	
19	16	8	
20	14	7	
21	16	8	
22	20	10	
23	14	7	
24	15	8	
25	28	14	
26	20	10	
27	20	10	
28			
29	20	10	
30	24	12	
31	31	20	
32	20	10	
33	14	7	
34	14	7	
35	20	10	
36	20	10	

U NUMBER	PIN NUMBERS +5	GND	UNUSED
37	14	7	
38	14	7	
39	14	7	
40	14	7	
41	14	7	
42	20	10	
43	20	10	
44	16	8	
45	20	10	
46	40	20	
47	19	42	
48	14	7	
49	14	7	
50	14	7	
51	14	7	
52	20	10	
53	14	7	
54	14	7	
55	14	7	
56	14	7	
57	14	7	
58	20	10	
59	8	16	
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72	8	16	

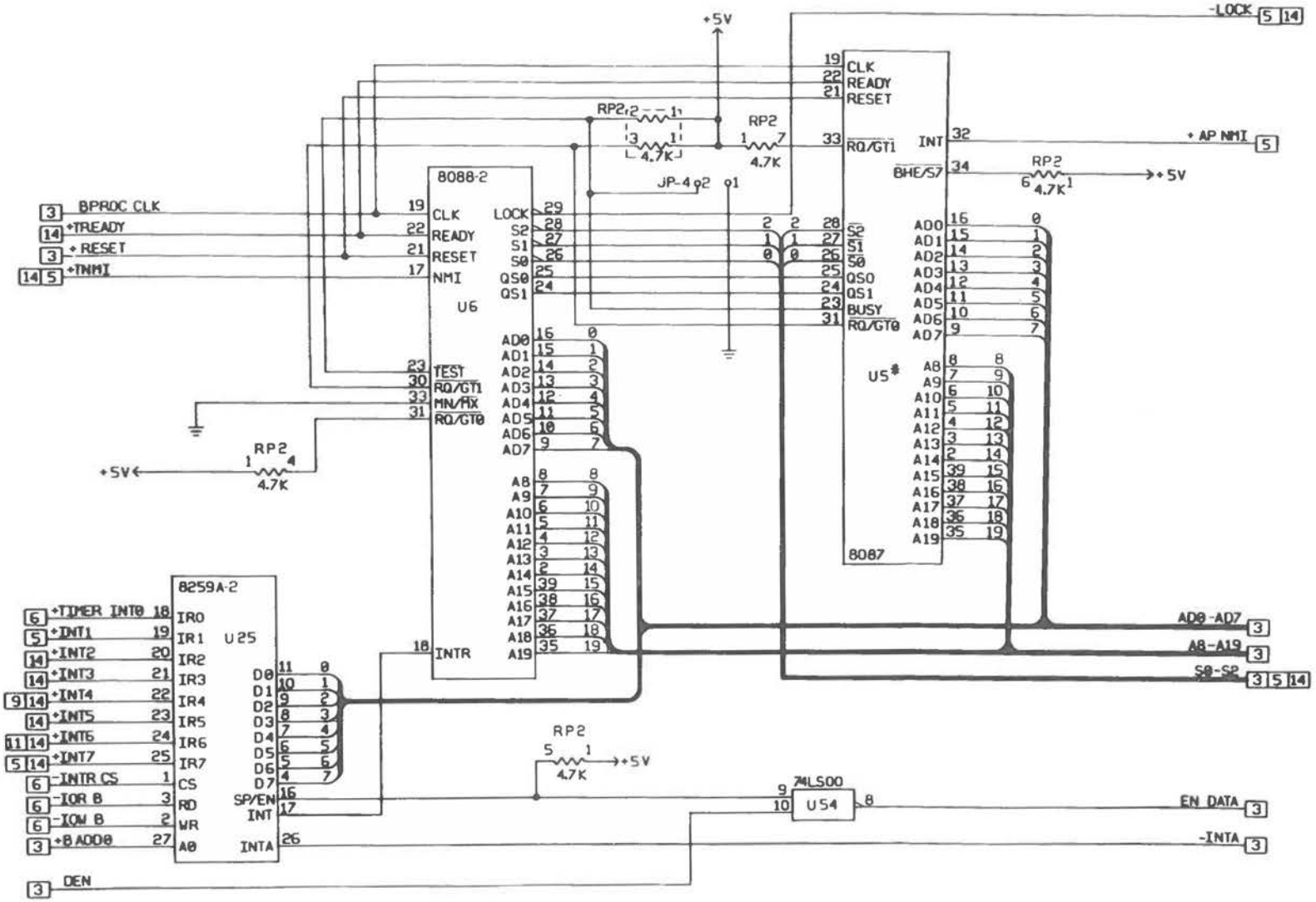
U NUMBER	PIN NUMBERS +5	GND	UNUSED
73	8	16	
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94	8	16	
95	14	7	
96	14	7	
97	5	13	
98			
99			
100			
101			
102			
103			
104			
105			
106			
107			
108			

U NUMBER	PIN NUMBERS +5	GND	UNUSED
109			
110			
111			
112			
113			
114			
115			
116			
117			
118			
119			
120			
121			
122			
123			
124			
125			
126			
127			
128			
129			
130			
131			
132			
133			
134			
135			
136			
137			
138			
139			
140			
141			
142			
143			
144			

U NUMBER	PIN NUMBERS +5	GND	UNUSED
145			
146			
147			
148			
149			
150			
151			
152			
153			
154			
155			
156			
157			
158			
159			
160			
161			
162			
163			
164			
165			
166			
167			
168			
169			
170			
171			
172			
173			
174			
175			
176			
177			
178			
179			
180			

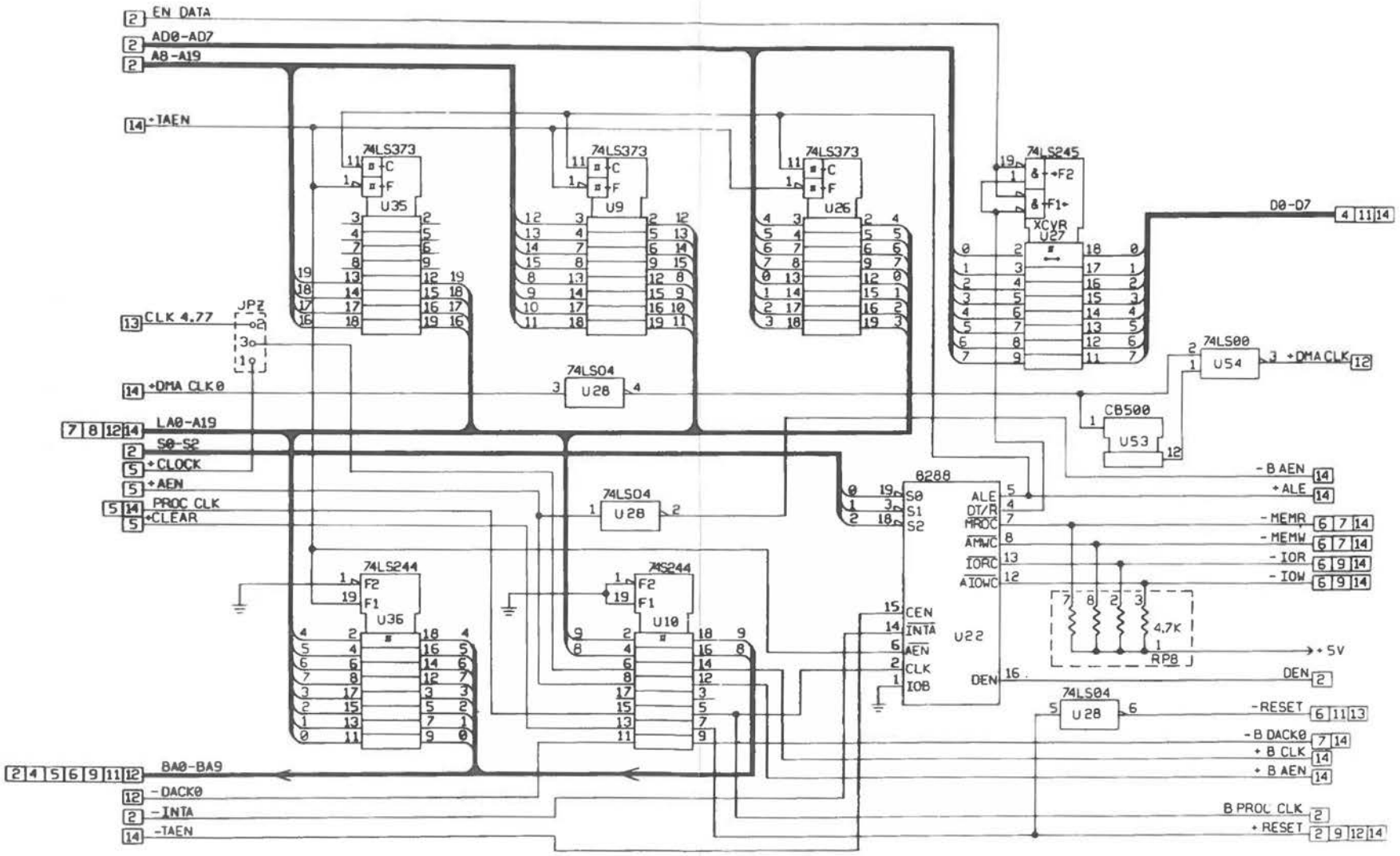
Alternate Processor Mainboard (1 of 14)

Logic Diagram



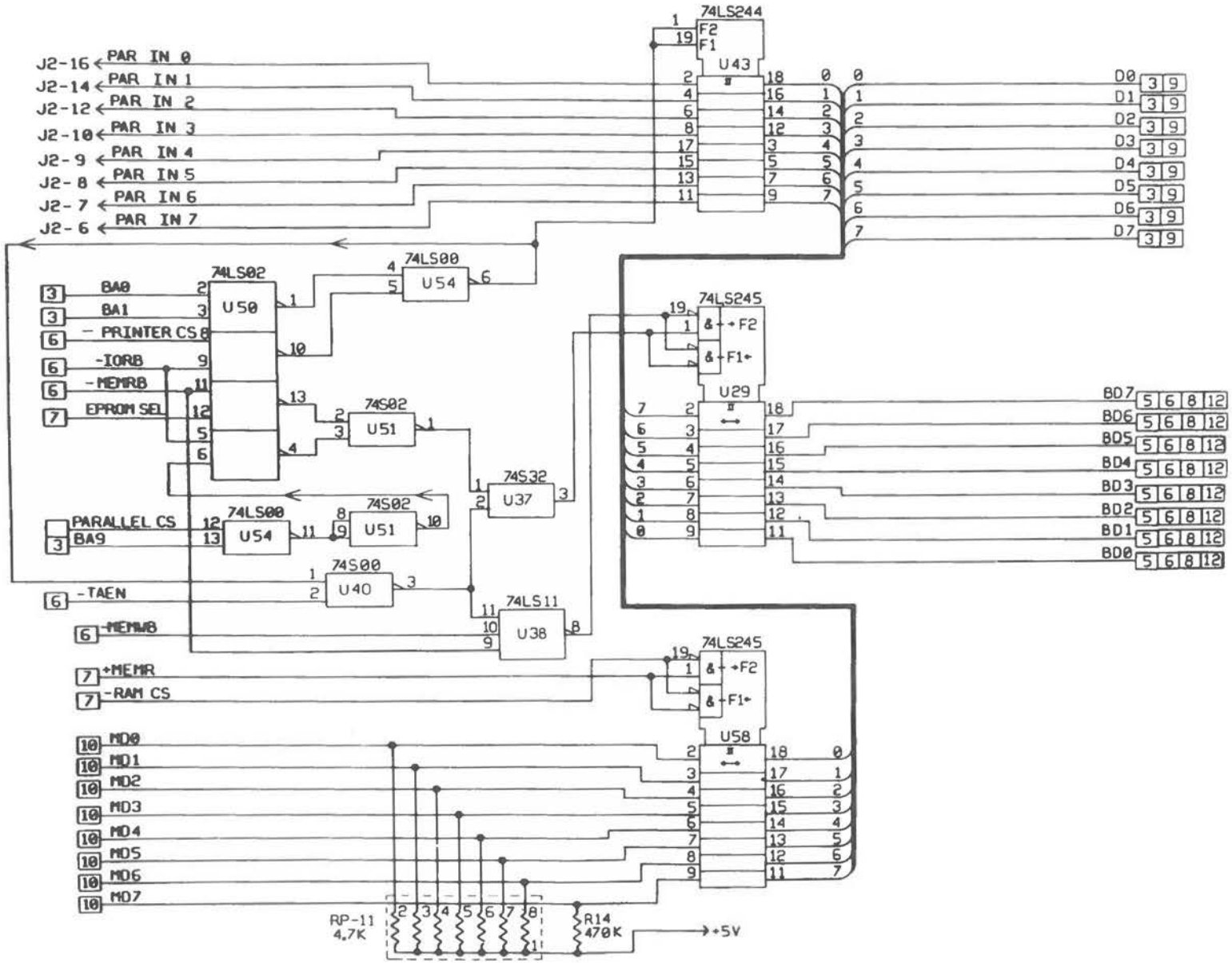
Alternate Processor Mainboard [2 of 14]

Logic Diagram



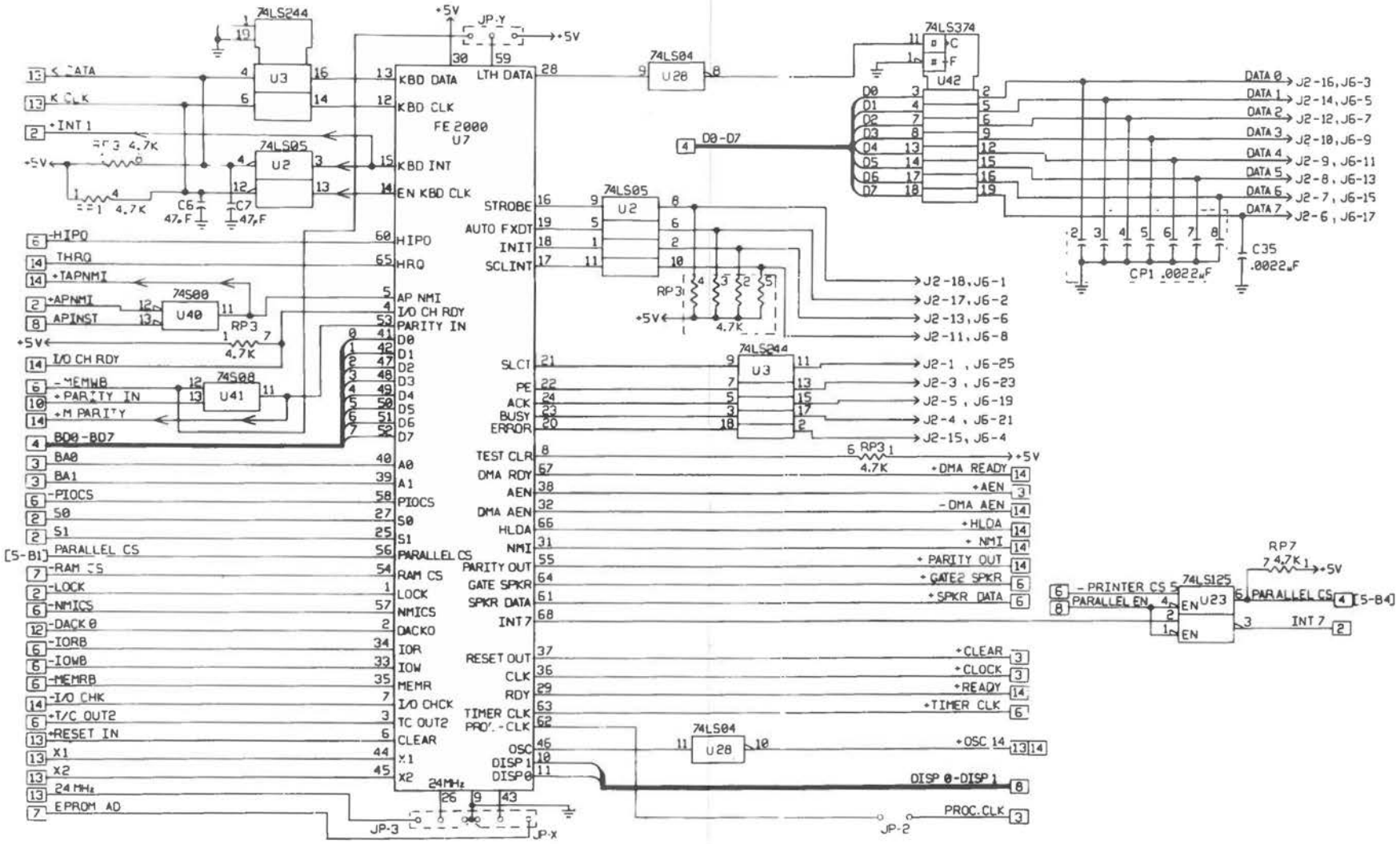
Alternate Main Processor Board (3 of 14)

Logic Diagram



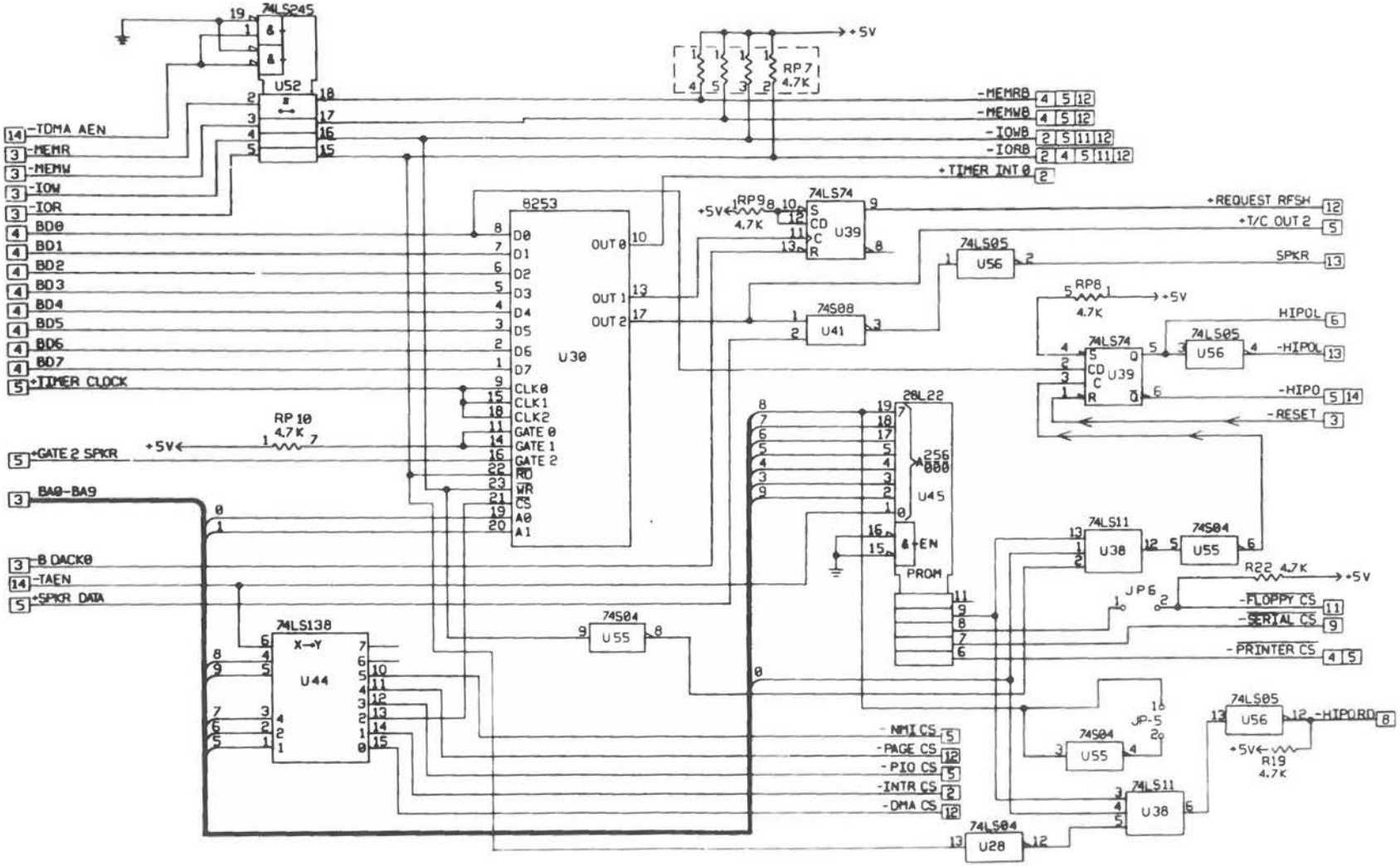
Alternate Main Processor Board [4 of 14]

Logic Diagram



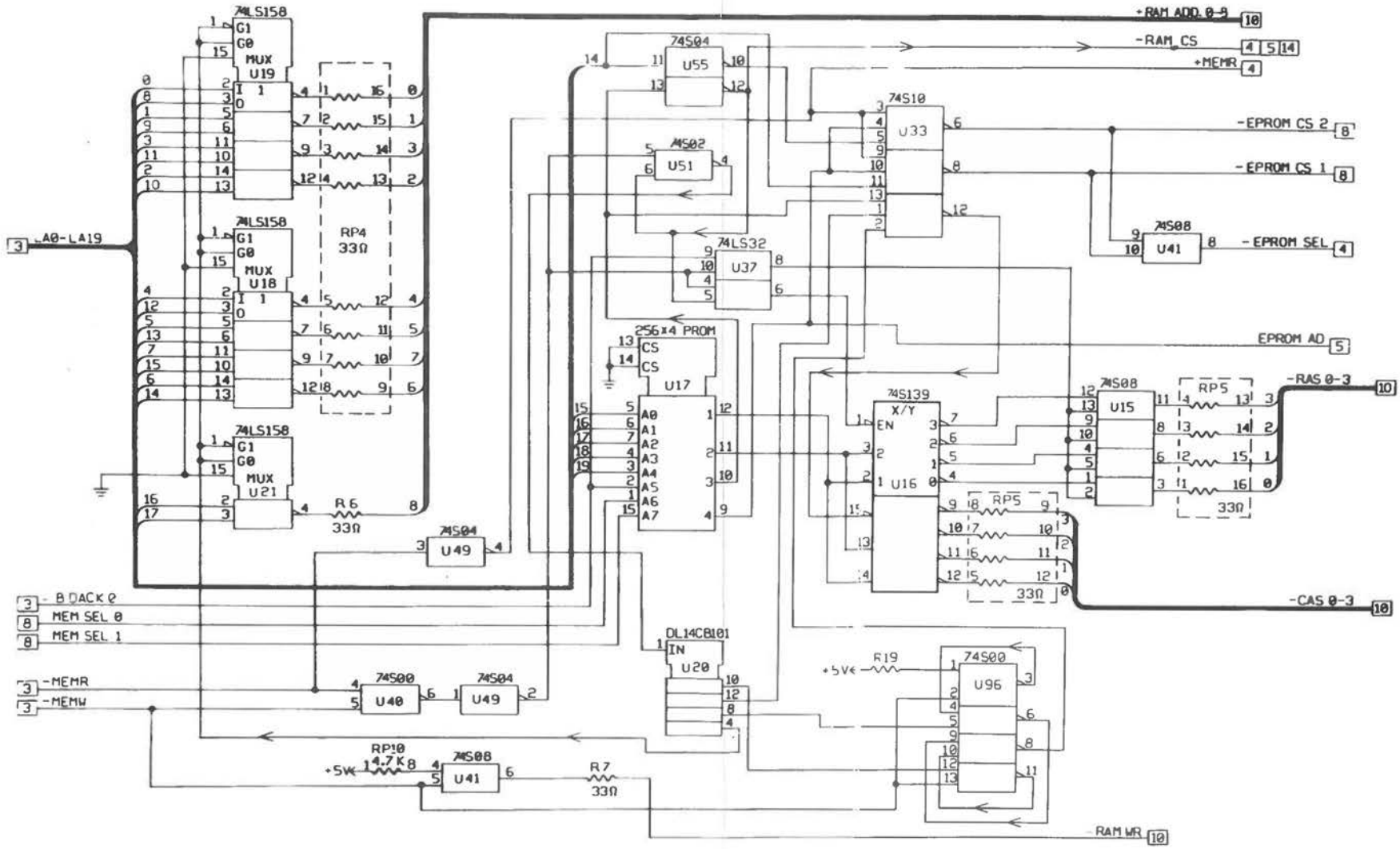
Alternate Main Processor Board (5 of 14)

Logic Diagram



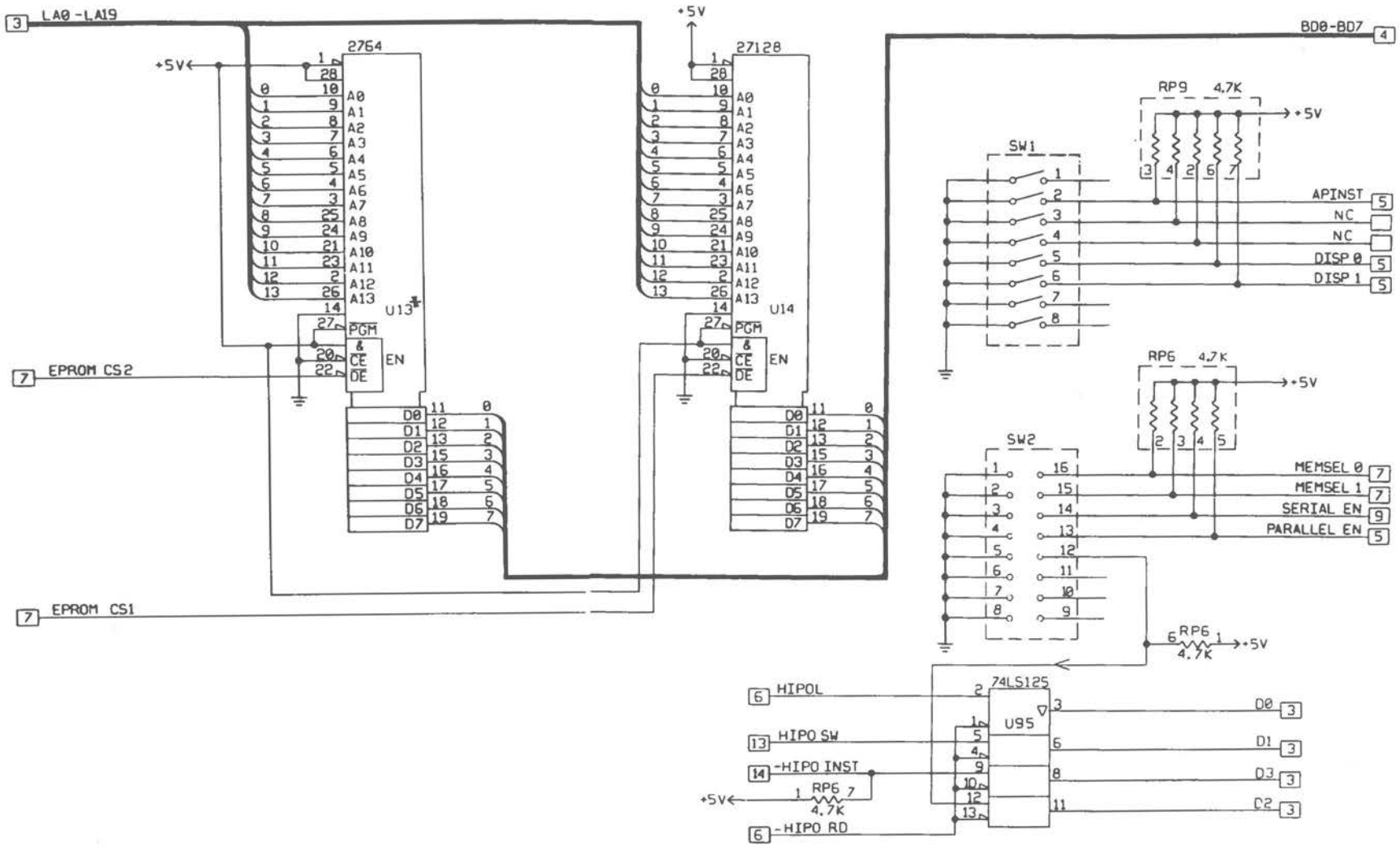
Alternate Main Processor Board (6 of 14)

Logic Diagram



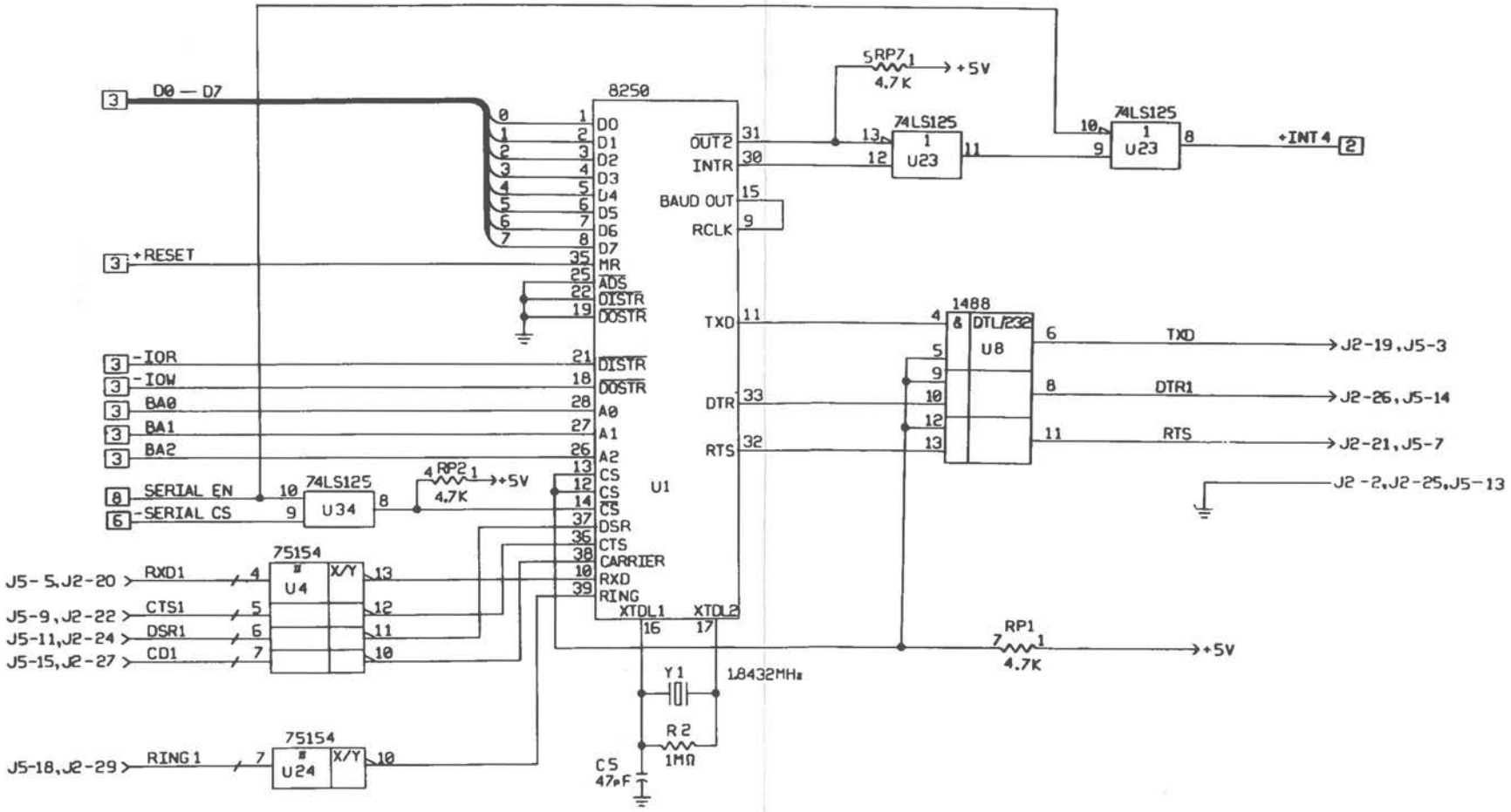
Alternate Main Processor Board (7 of 14)

Logic Diagram



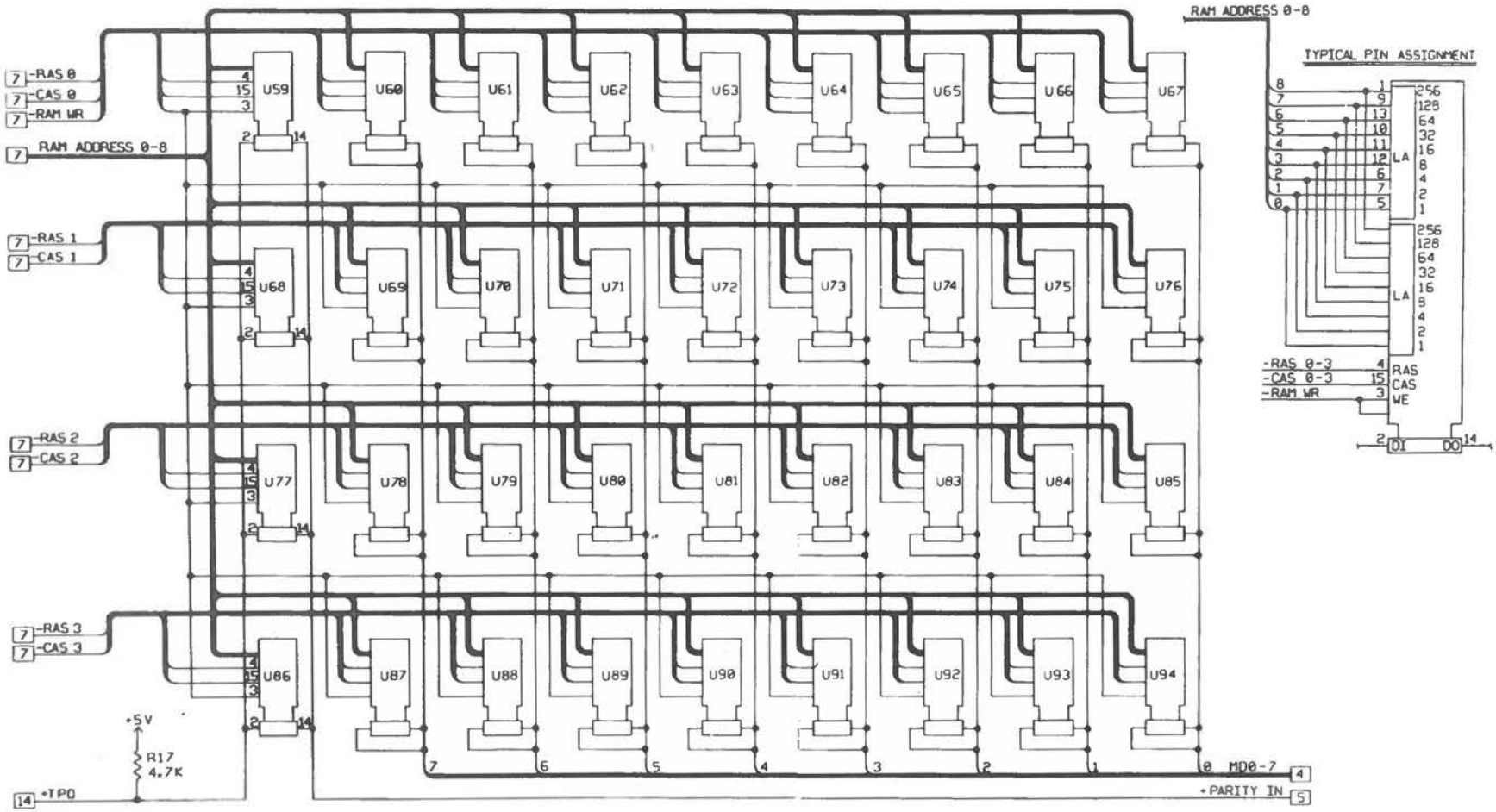
Alternate Main Processor Board (8 of 14)

Logic Diagram



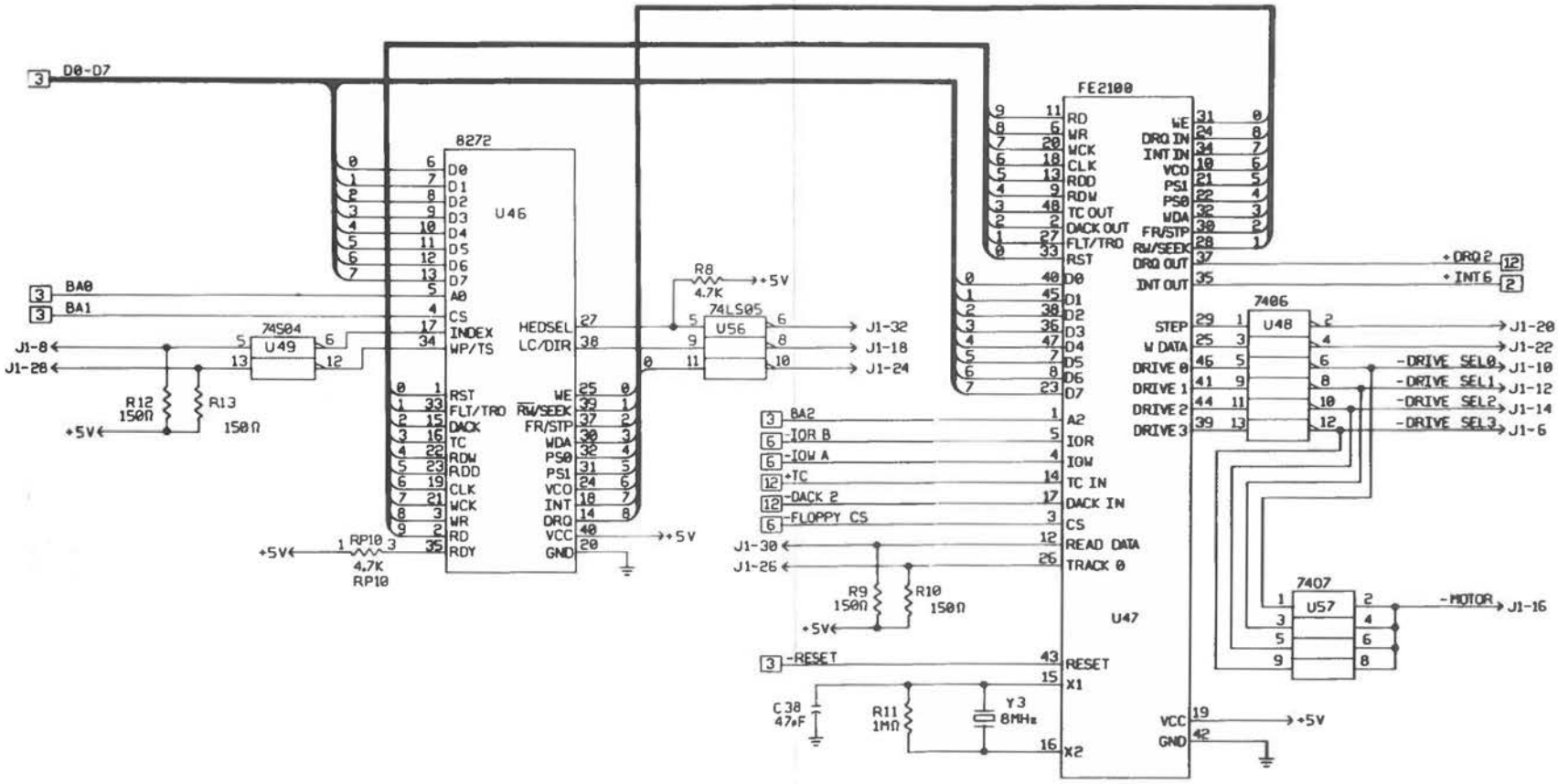
Alternate Main Processor Board (9 of 14)

Logic Diagram



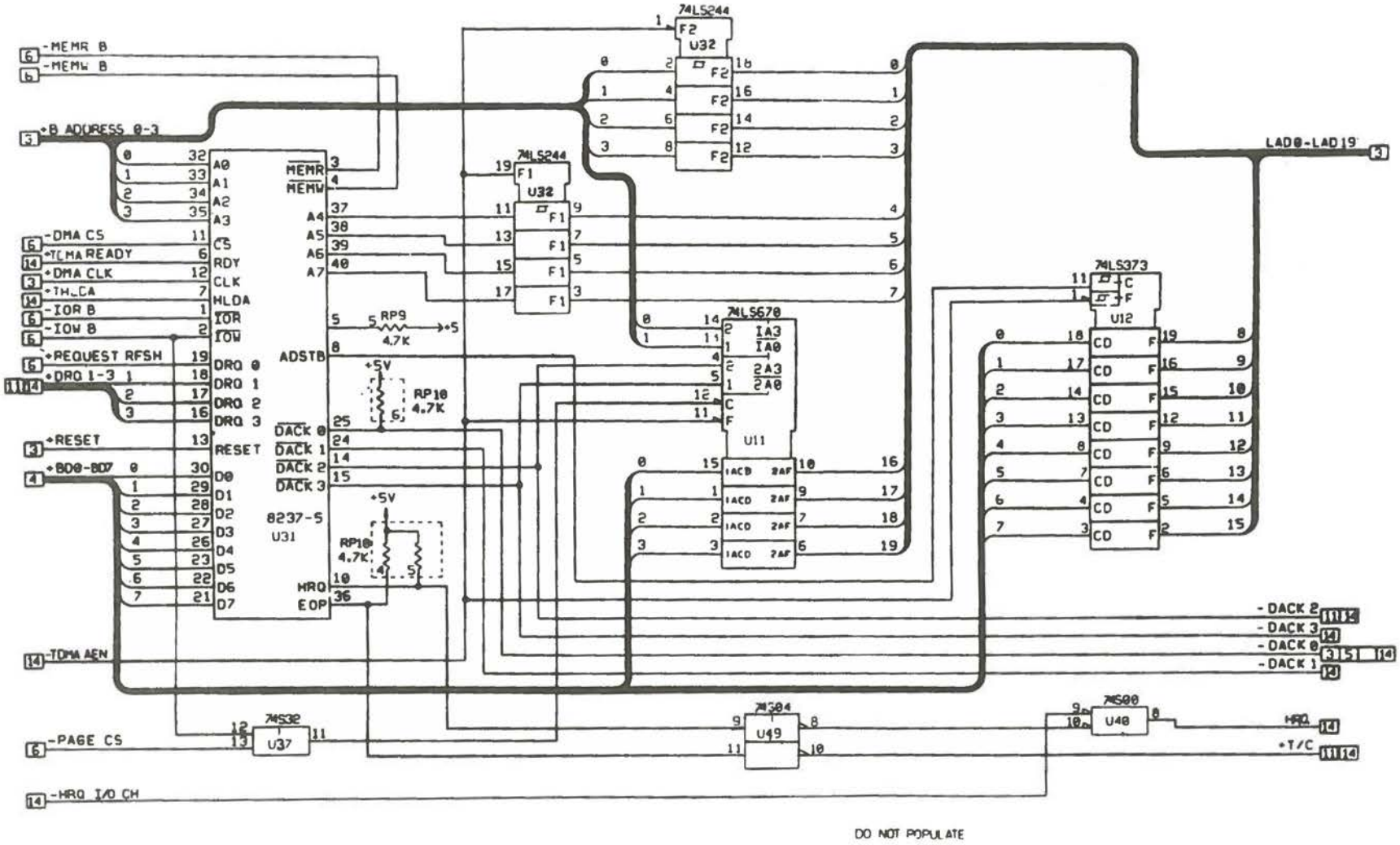
Alternate Main Processor Board (10 of 14)

Logic Diagram



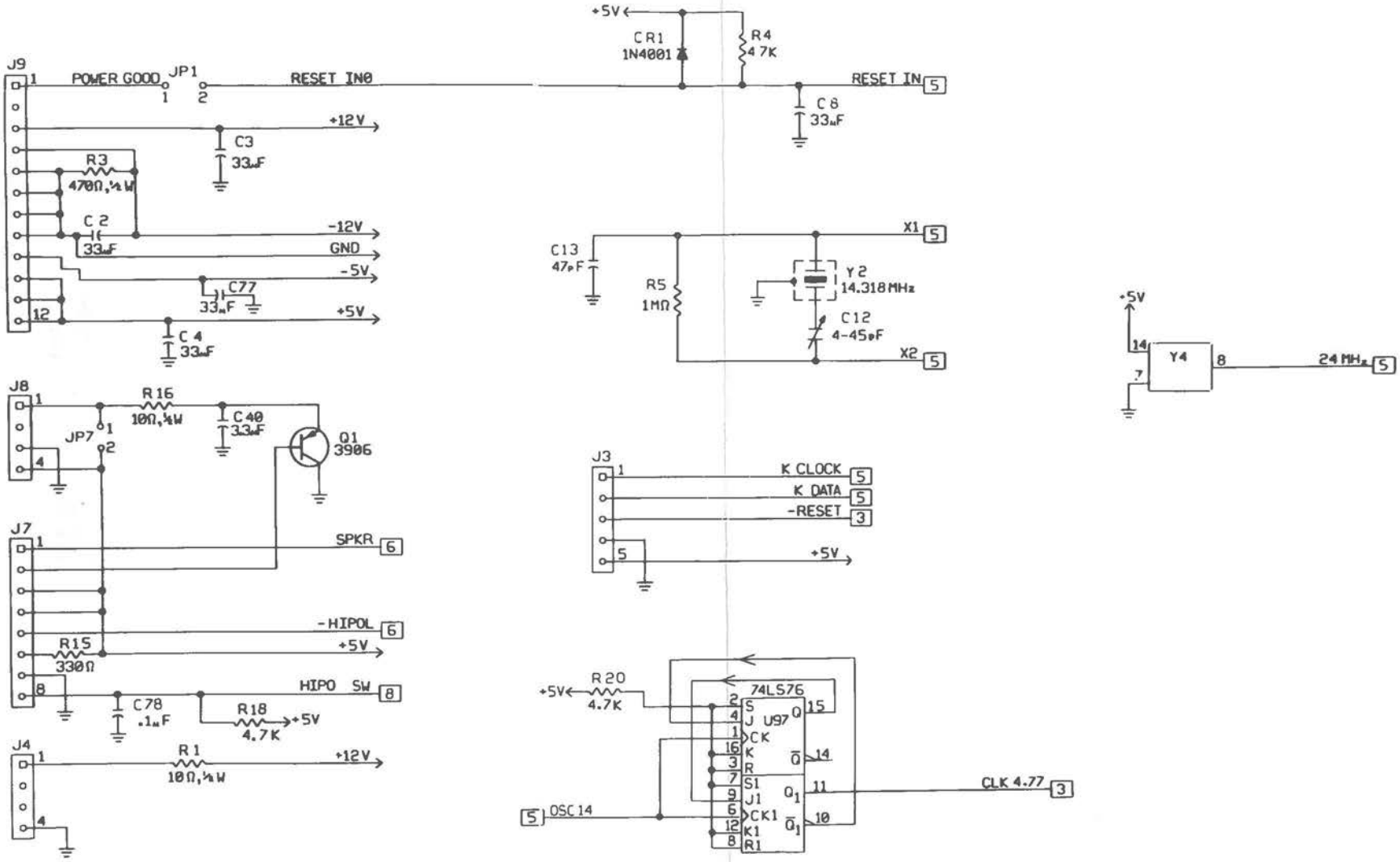
Alternate Main Processor Board (11 of 14)

Logic Diagram



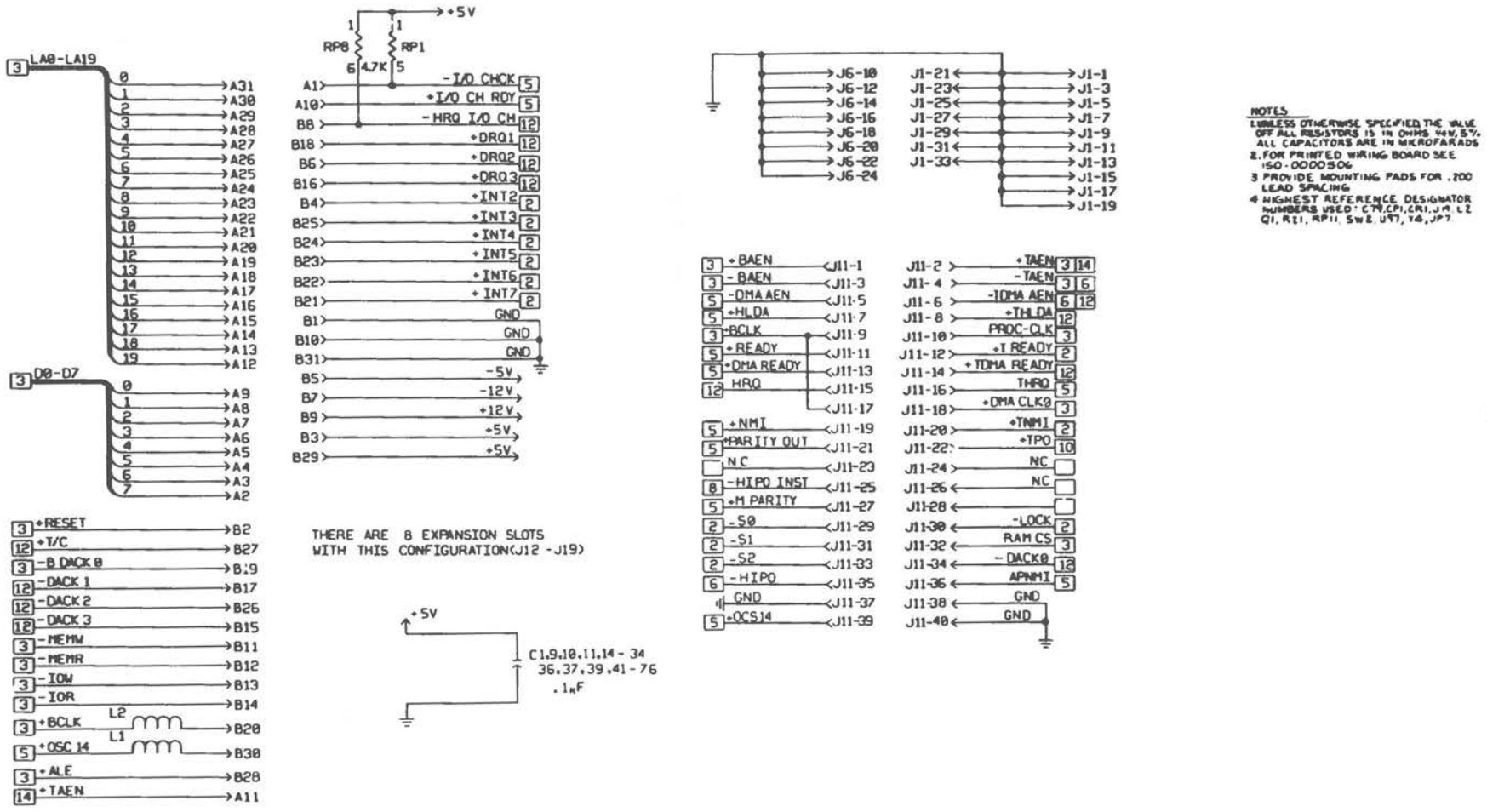
Alternate Main Processor Board (12 of 14)

Logic Diagram



Alternate Main Processor Board (13 of 14)

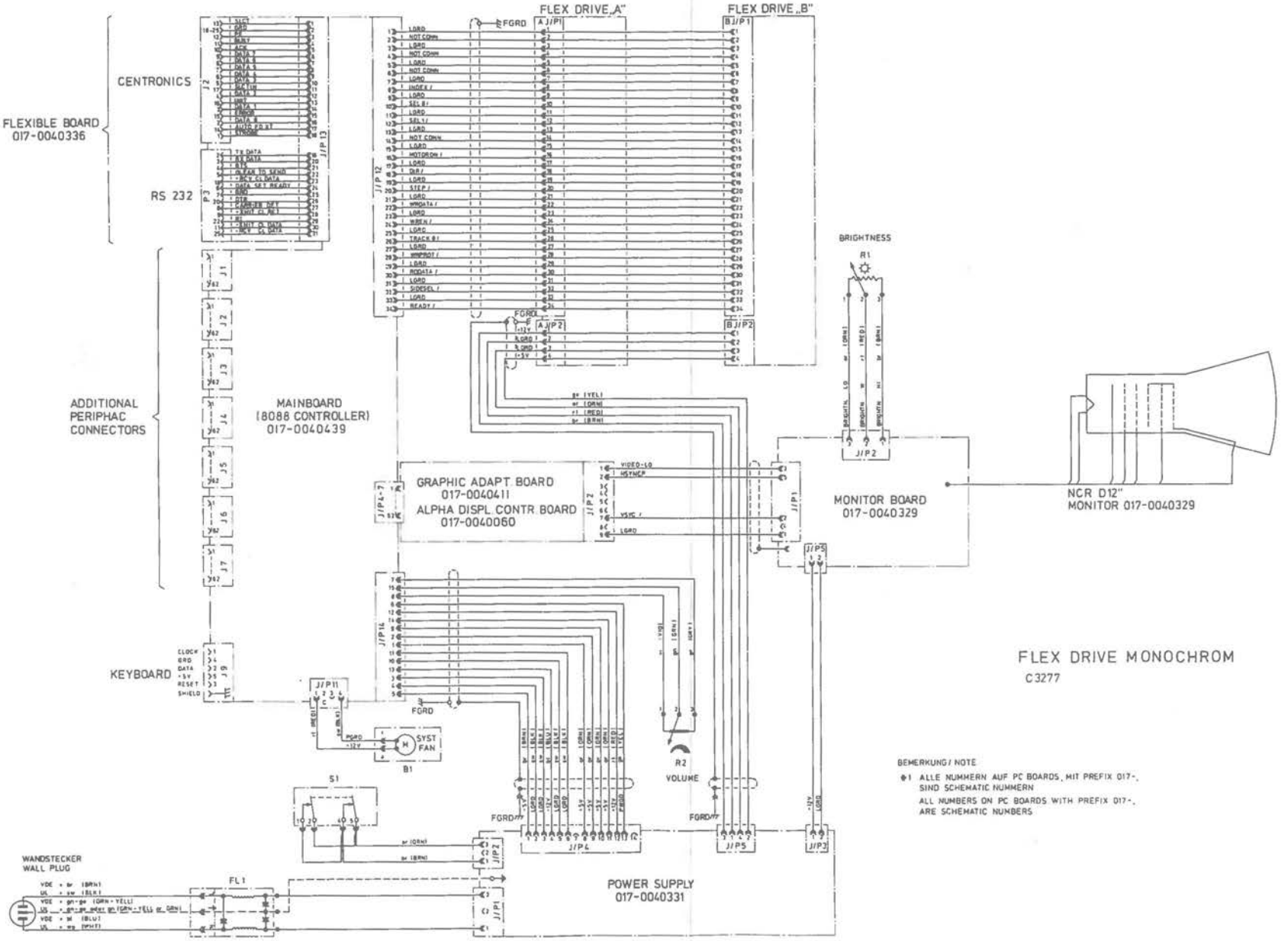
Logic Diagram



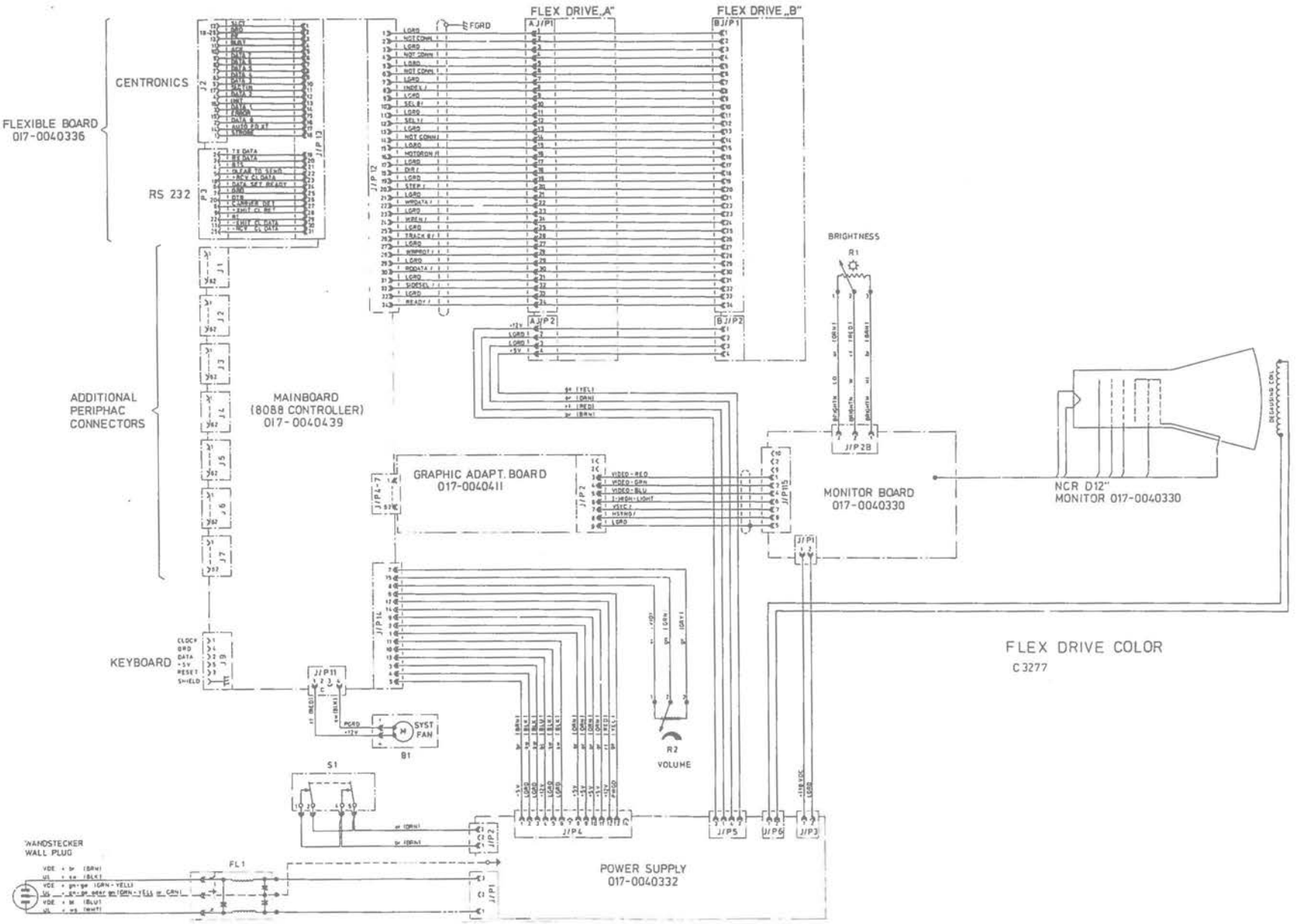
NOTES:
1. UNLESS OTHERWISE SPECIFIED THE VALUE OF ALL RESISTORS IS IN OHMS, 1/4W, 5%. ALL CAPACITORS ARE IN MICROFARADS
2. FOR PRINTED WIRING BOARD SEE 150-0000506
3. PROVIDE MOUNTING PADS FOR .200 LEAD SPACING
4. HIGHEST REFERENCE DESIGNATOR NUMBERS USED: C79, CP1, CR1, J4, L2, Q1, R11, RP11, SW2, U97, Y4, JP7

Alternate Main Processor Board (14 of 14)

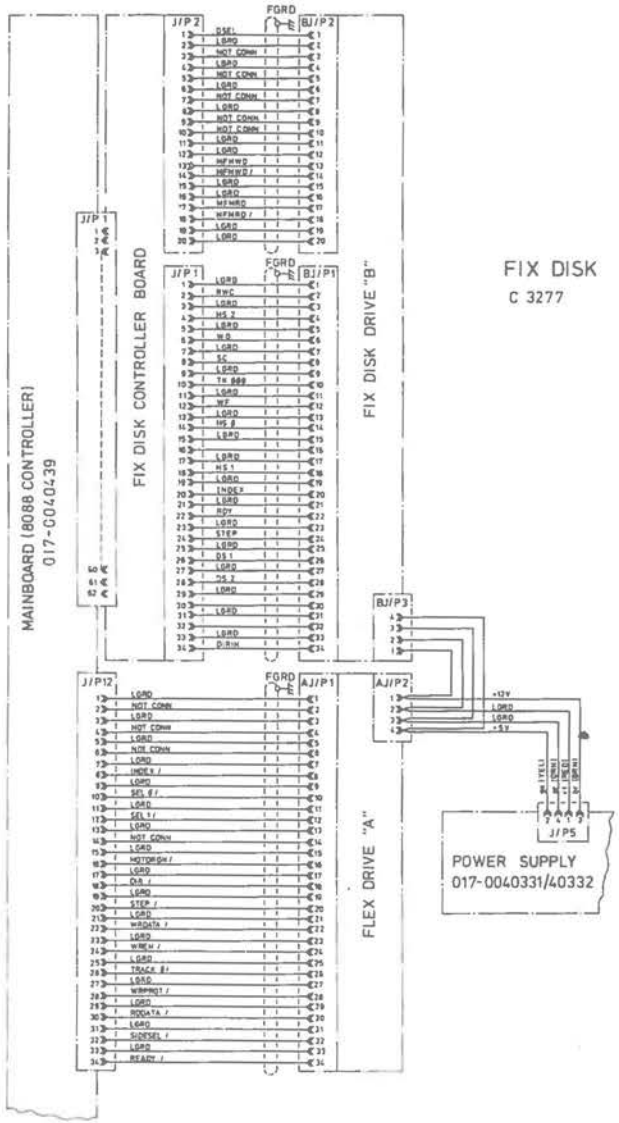
Wiring Diagram



Wiring Diagram (1 of 3)

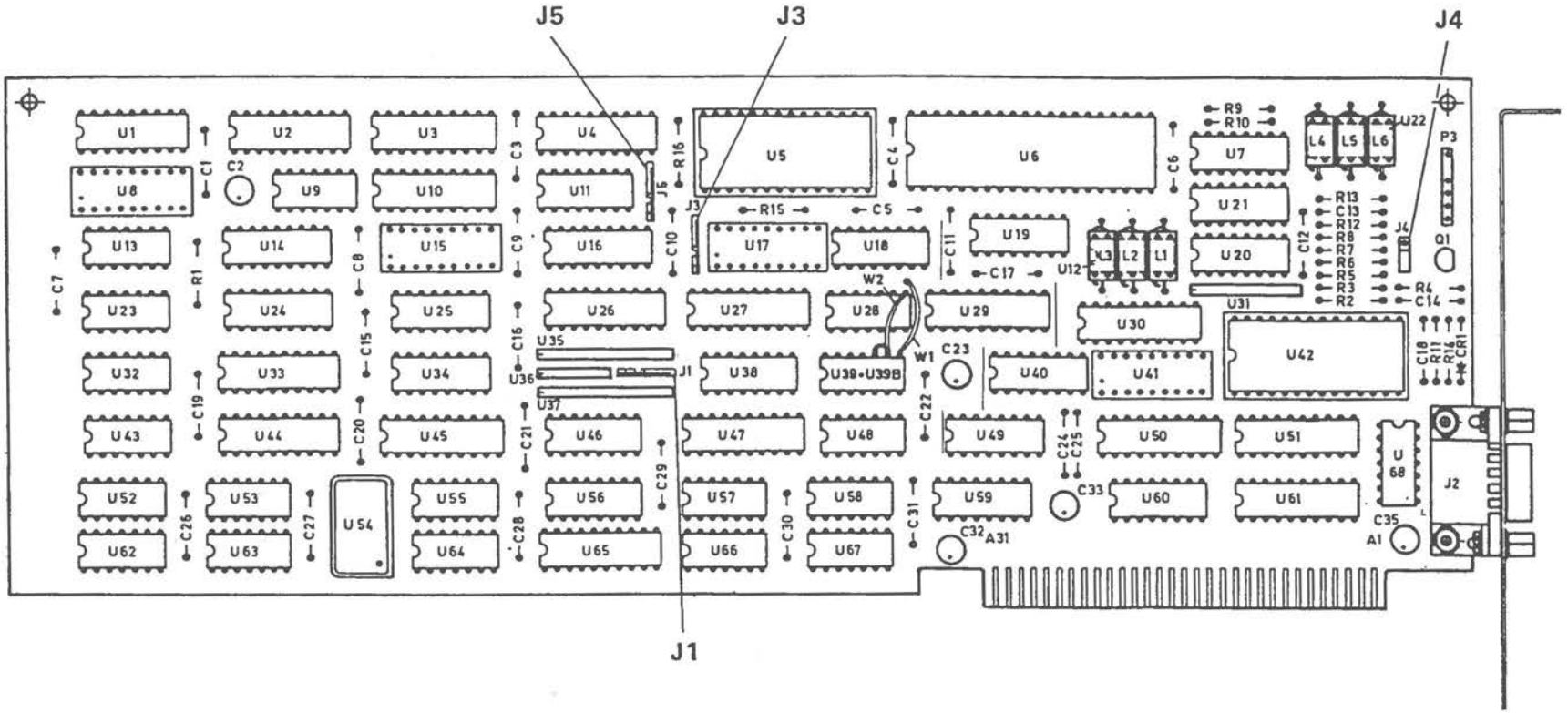


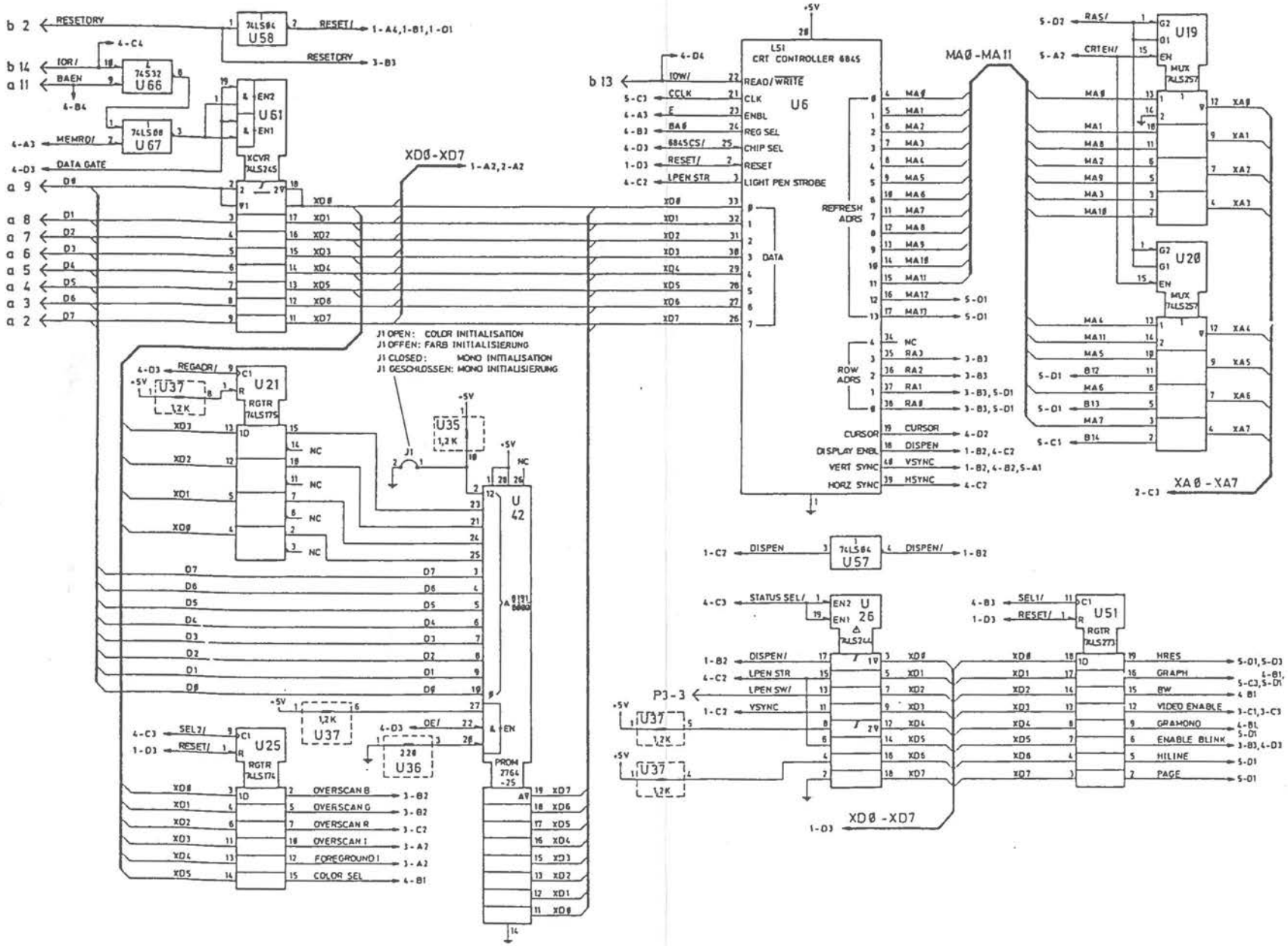
Wiring Diagram (2 of 3)



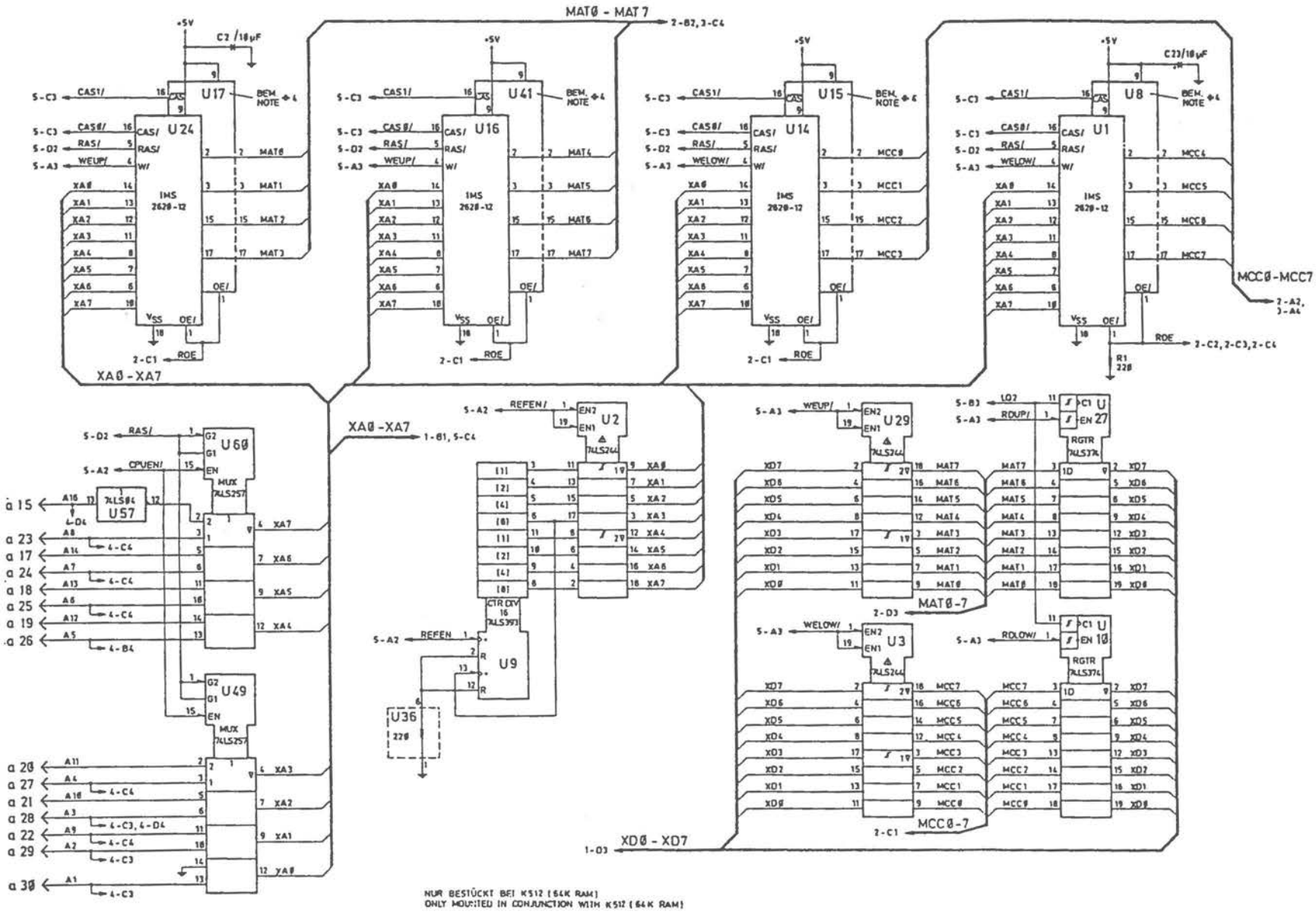
Wiring Diagram (3 of 3)

Graphics Controller Board

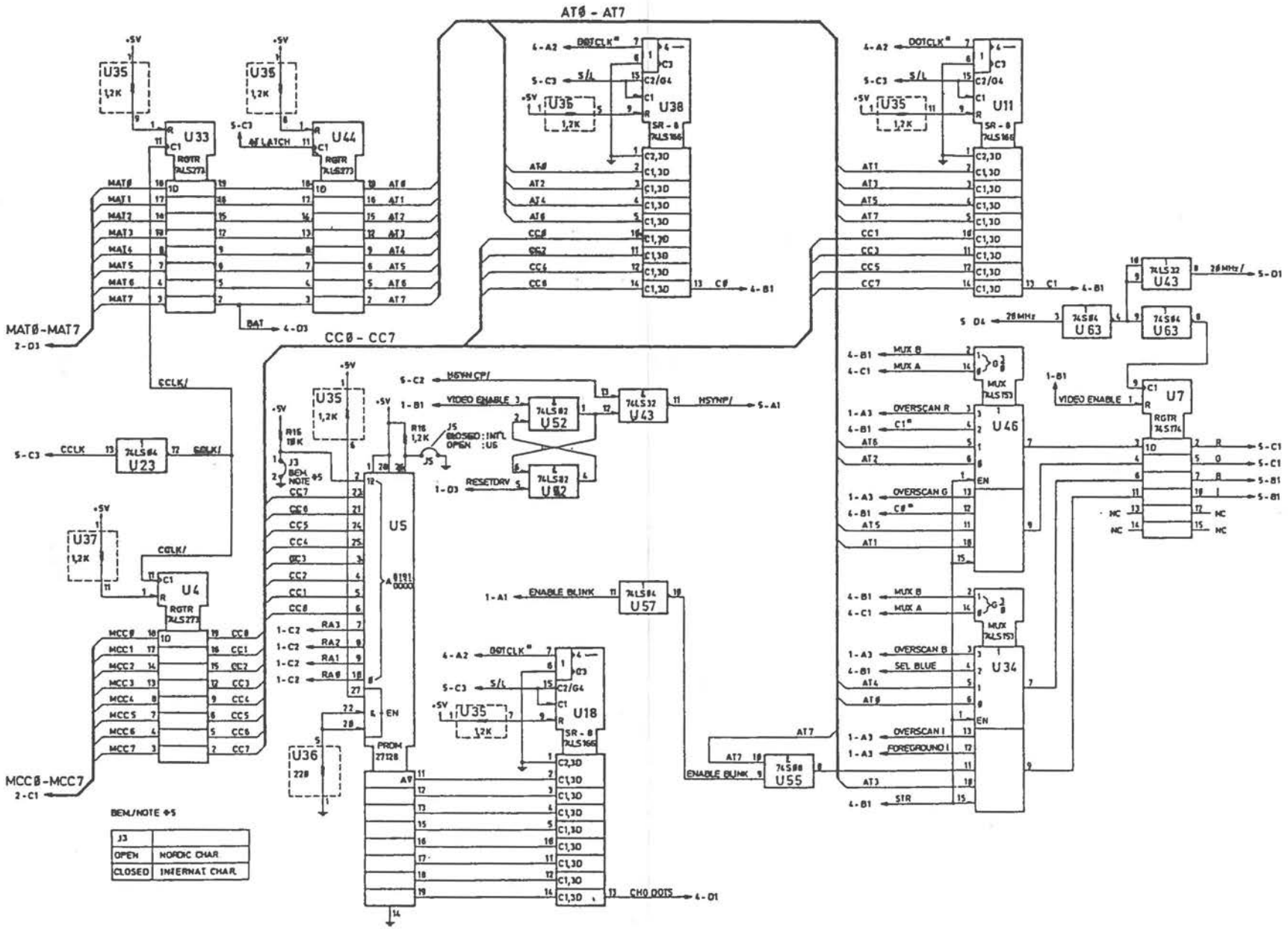




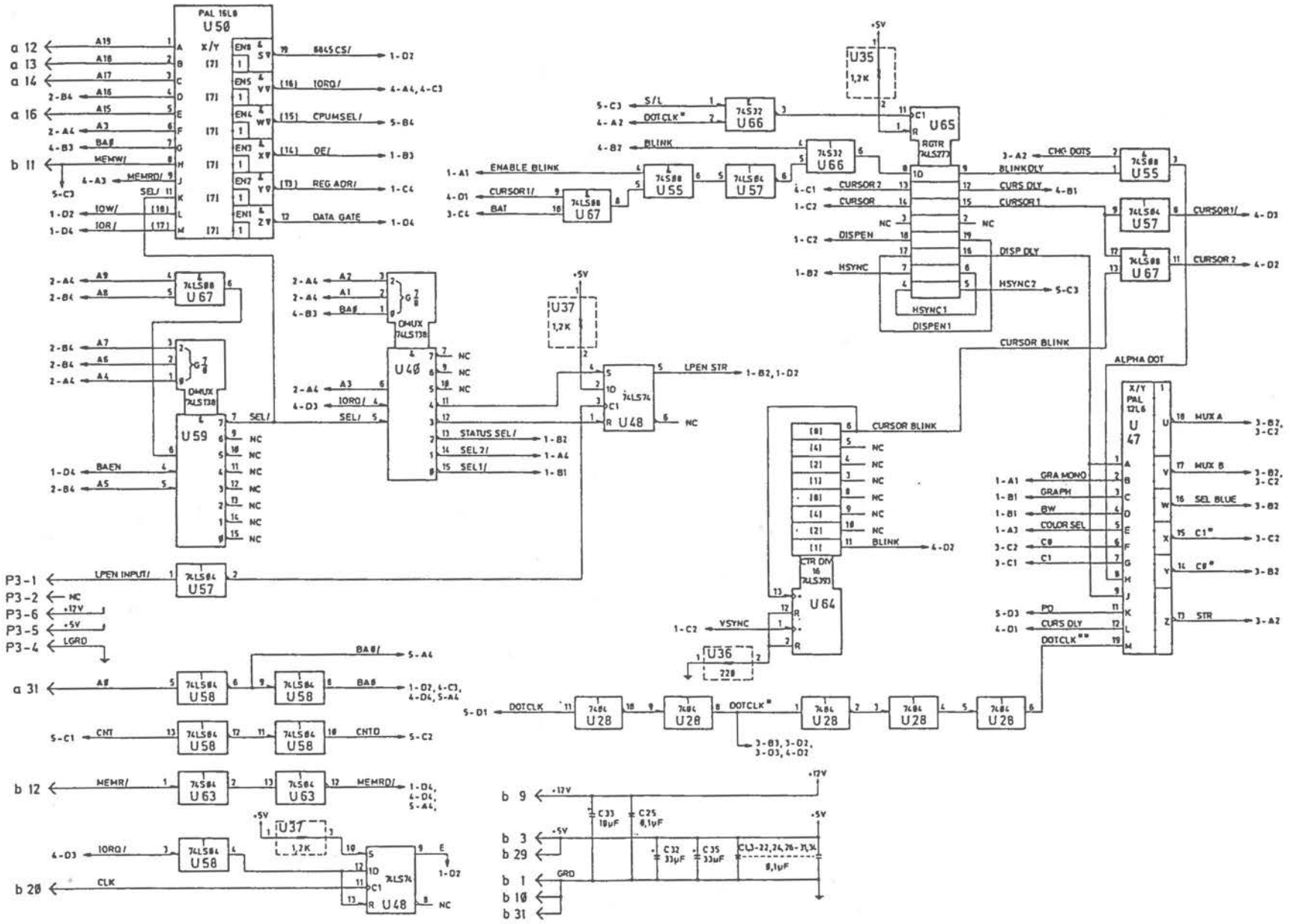
Graphics Controller Board (1 of 5)



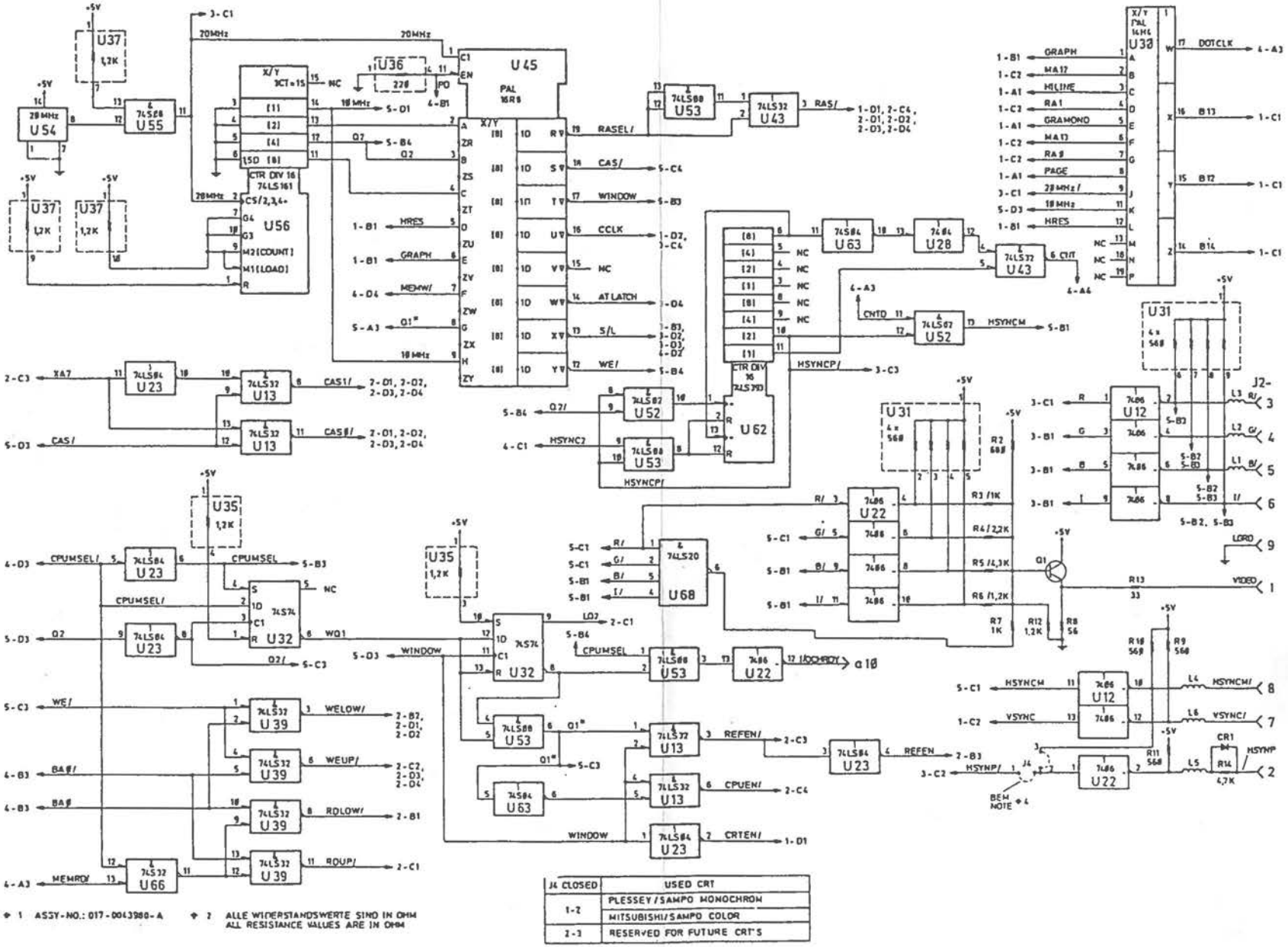
Graphics Controller Board (2 of 5)



Graphics Controller Board (3 of 5)



Graphics Controller Board (4 of 5)



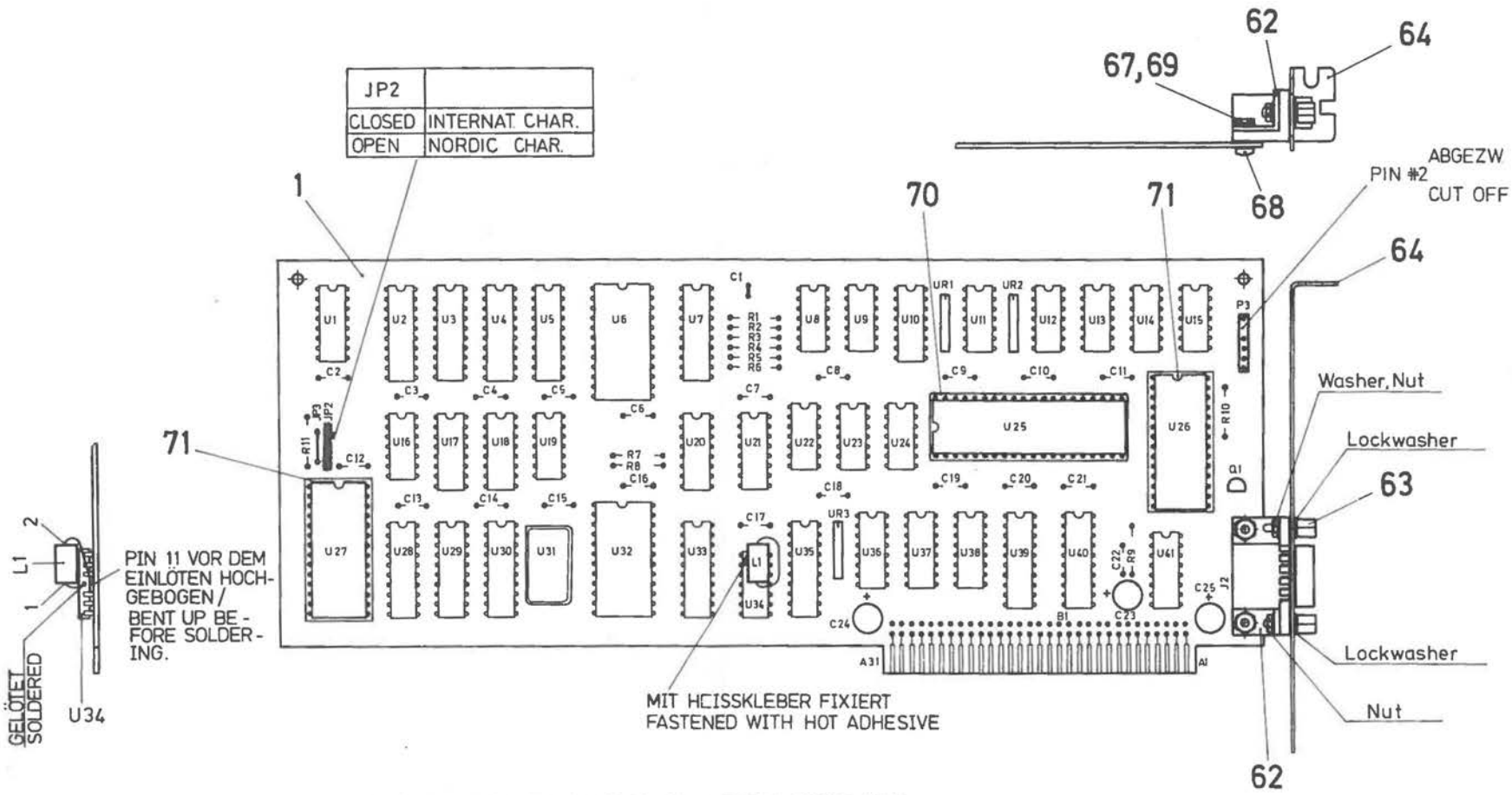
J4 CLOSED	USED CRT
1-2	PLESSEY/SAMPO MONOCHROM
	MITSUBISHI/SAMPO COLOR
2-3	RESERVED FOR FUTURE CRT'S

* 1 ASSY-NO.: 017-0043980-A

* 2 ALLE WIDERSTANDSWERTE SIND IN OHM
ALL RESISTANCE VALUES ARE IN OHM

Graphics Controller Board (5 of 5)

Alpha Controller Board

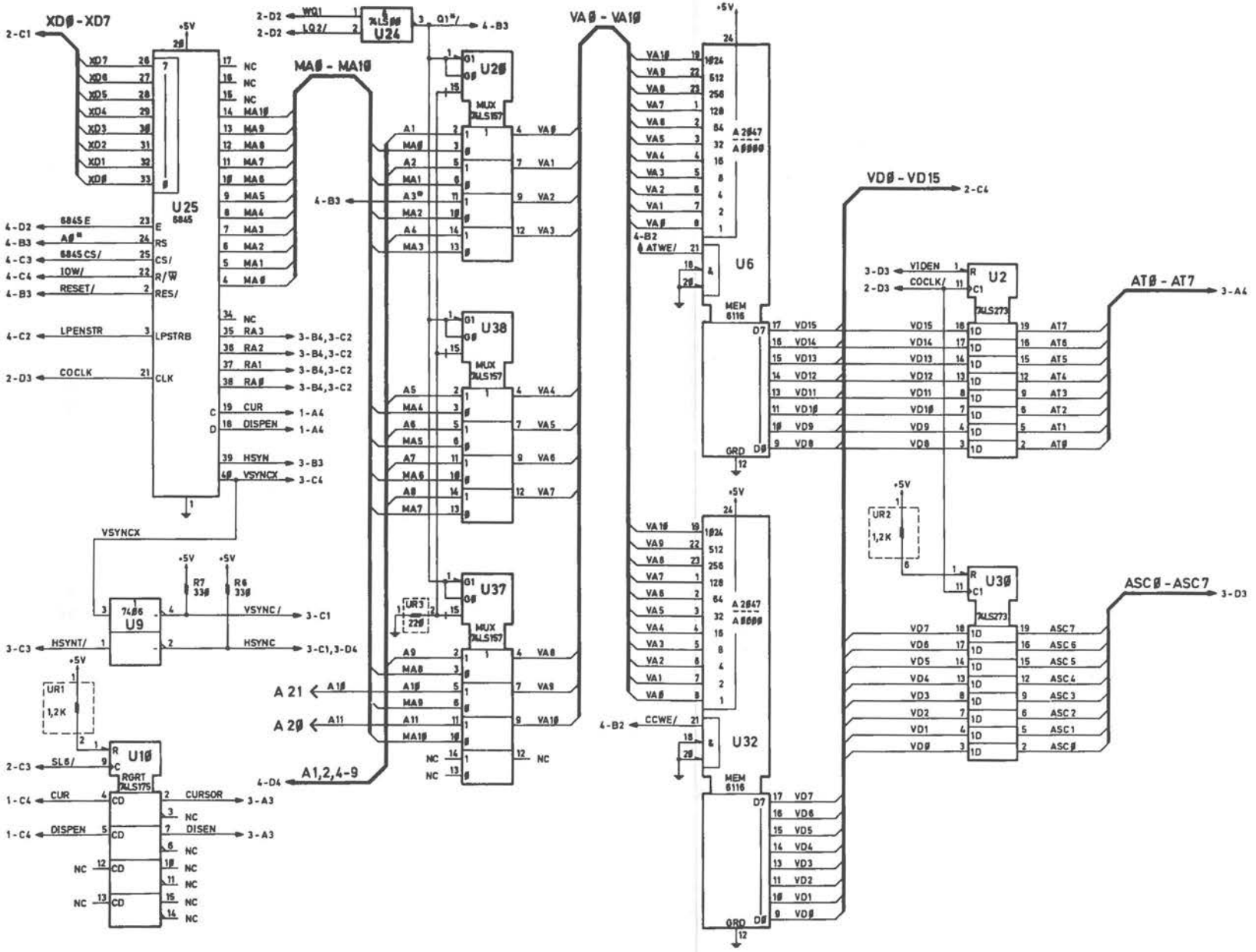


LABEL „SCHM. 017-0040060-F" AUFGEKLEBT/GLUED

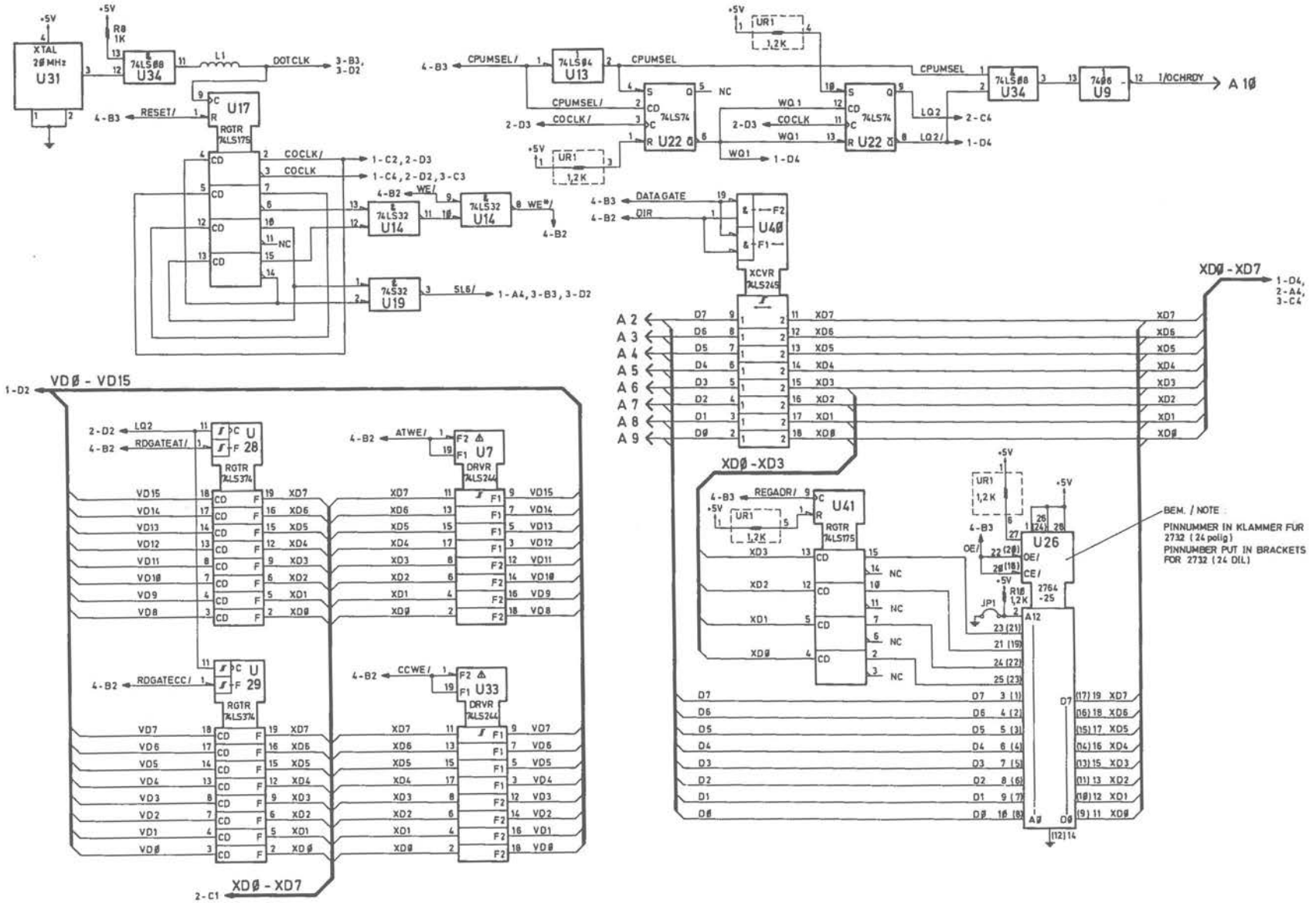
3 ITEM 77 GLUED ON SOLDERSIDE

2 NICHT BESTÜCKT : C1, JP1, JP3, P2
NOT MOUNTED : C1, JP1, JP3, P2

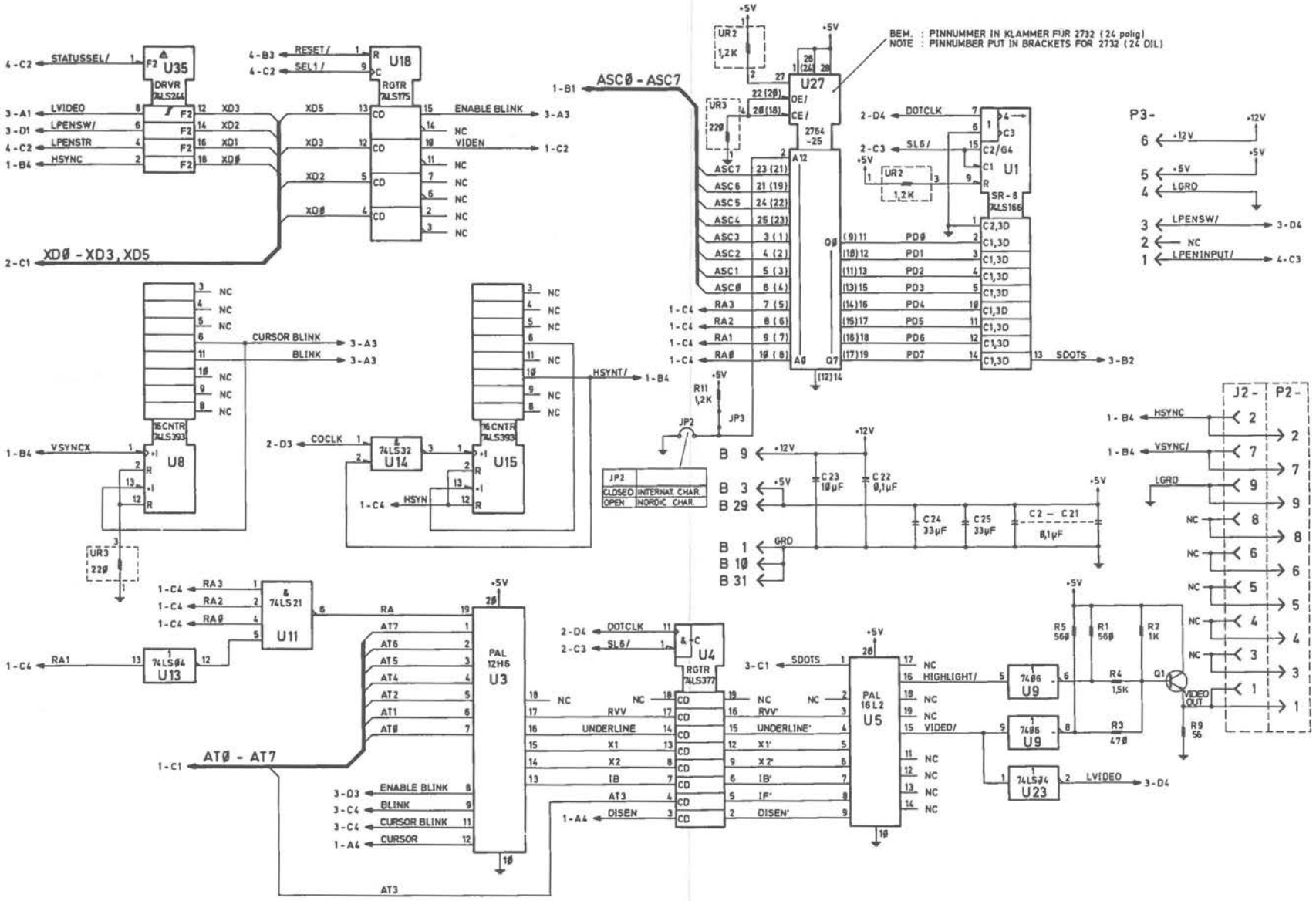
1 SCHEMATIC NO.: 017-0040060-F



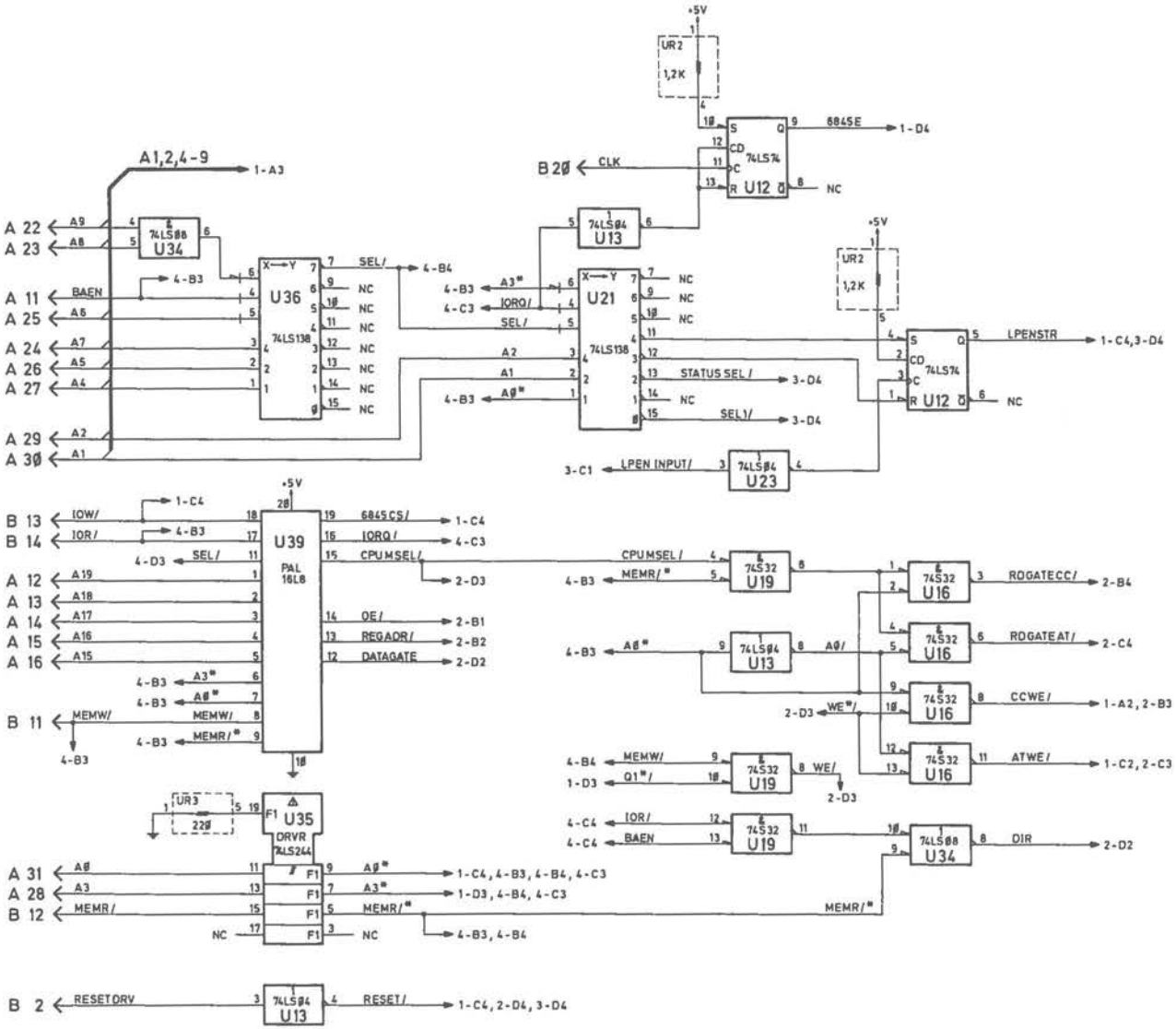
Alpha Controller Board (1 of 4)



Alpha Controller Board (2 of 4)



Alpha Controller Board (3 of 4)

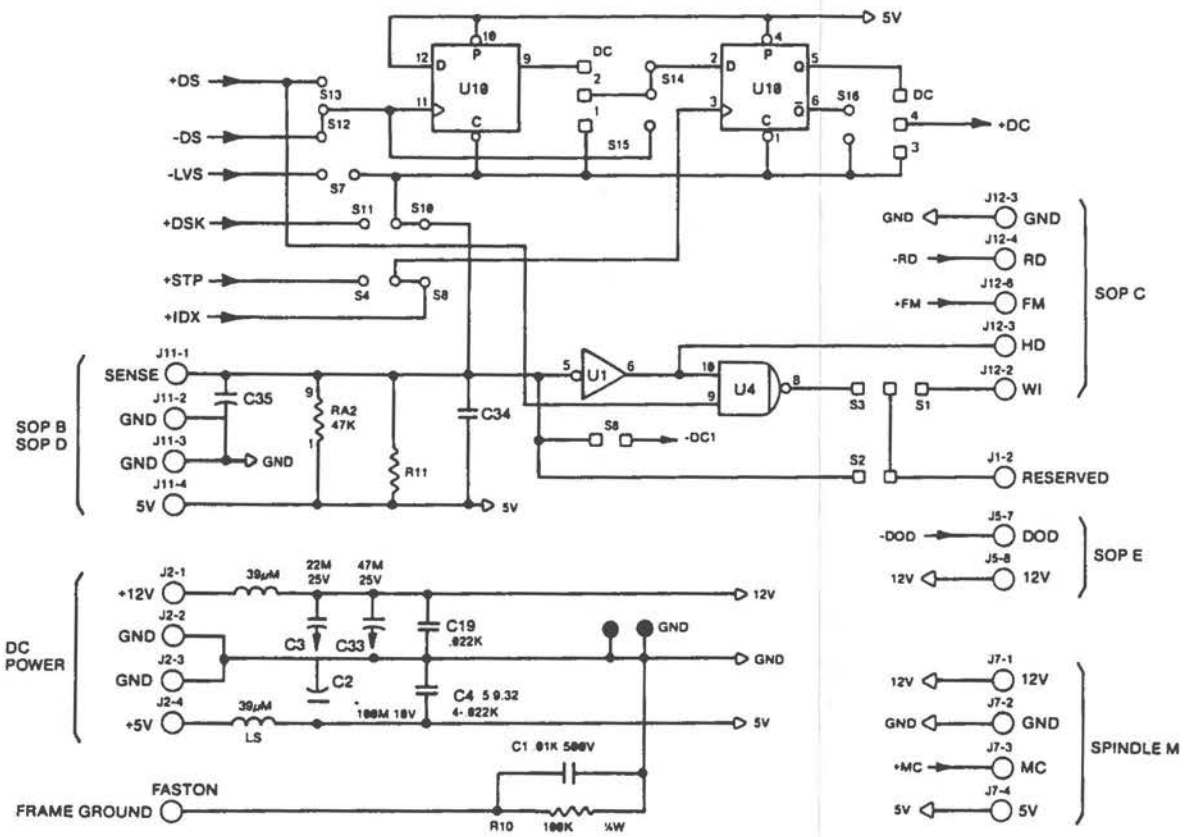


✦ 2 ALLE WIDERSTANDSWERTE SIND IN OHM
ALL RESISTANCE VALUES ARE IN OHM

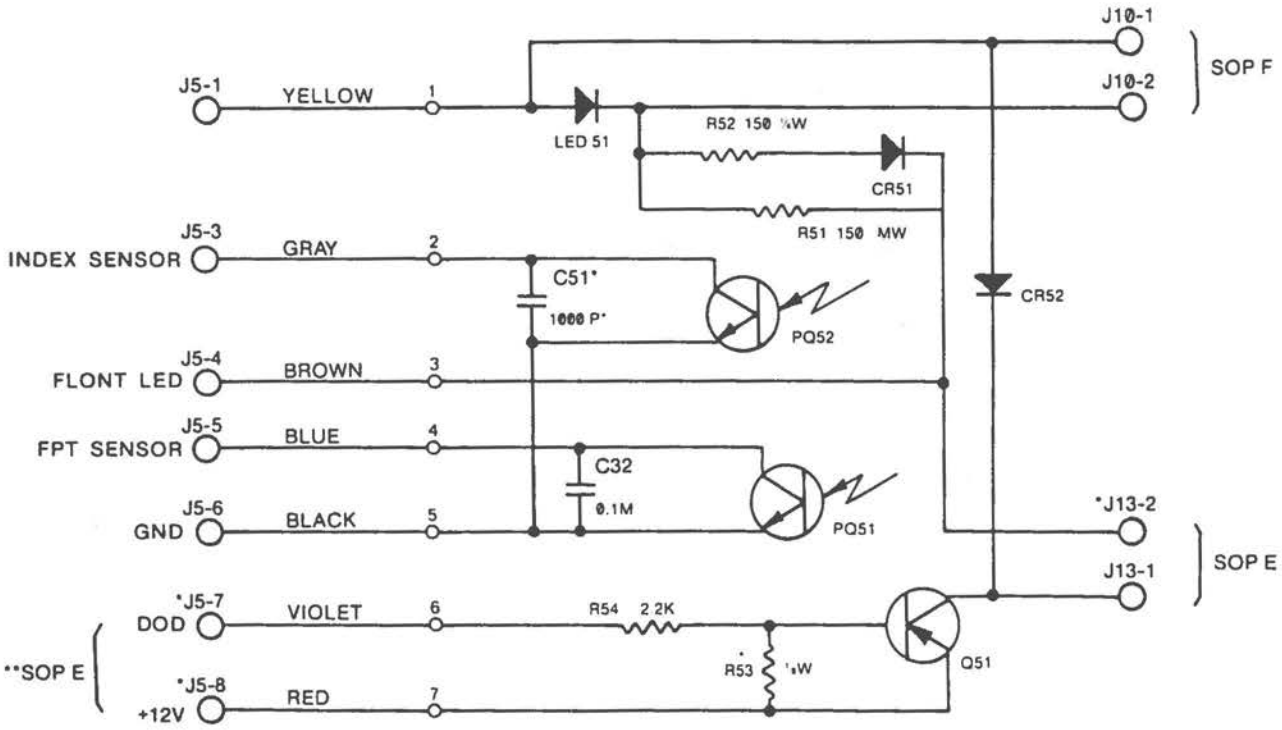
✦ 1 ASSY-NO.: 017-0040424 -E

Alpha Controller Board (4 of 4)

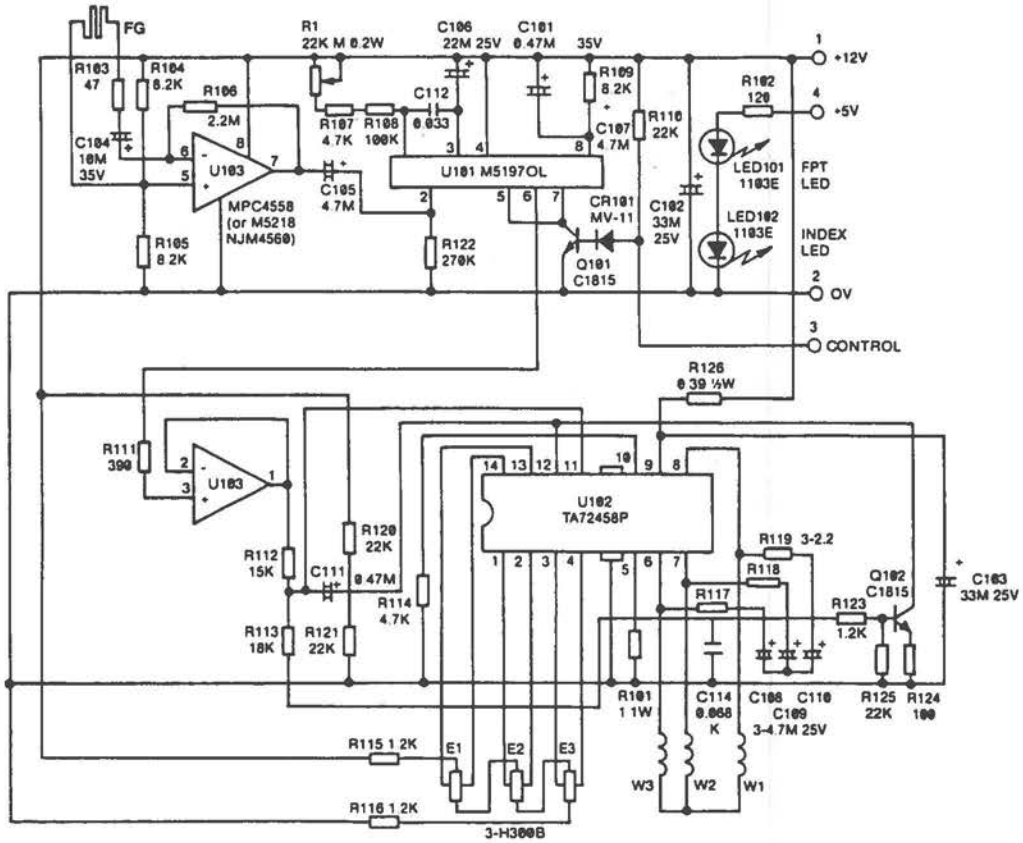
Flexible Disk Drive



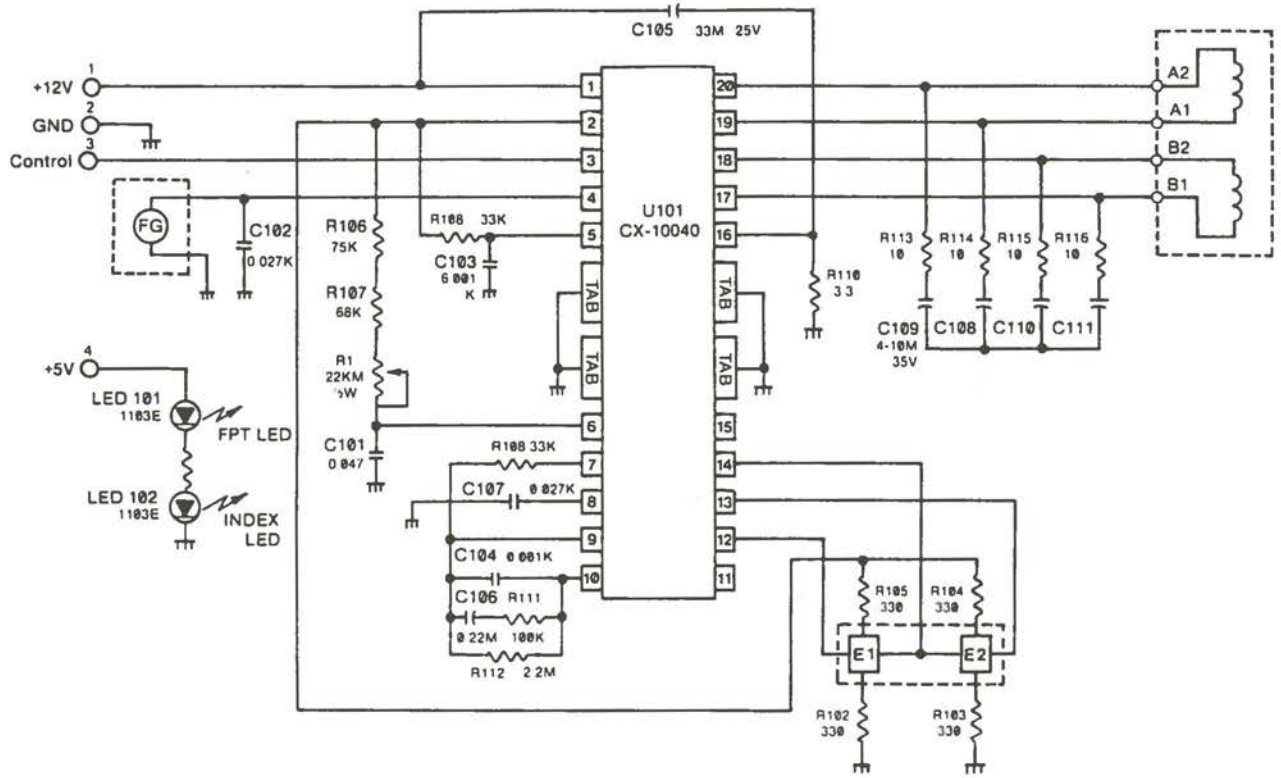
Flexible Disk Drive (1 of 8)



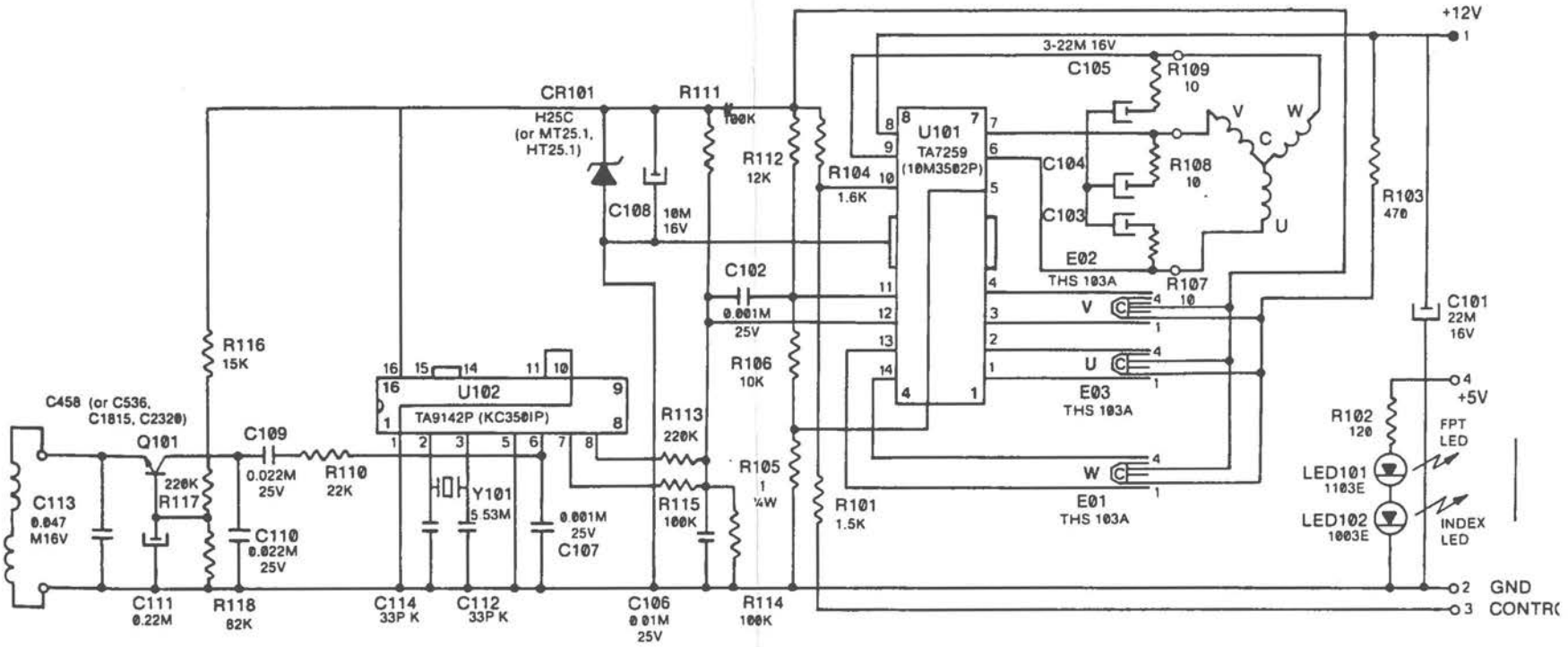
Flexible Disk Drive (2 of 8)



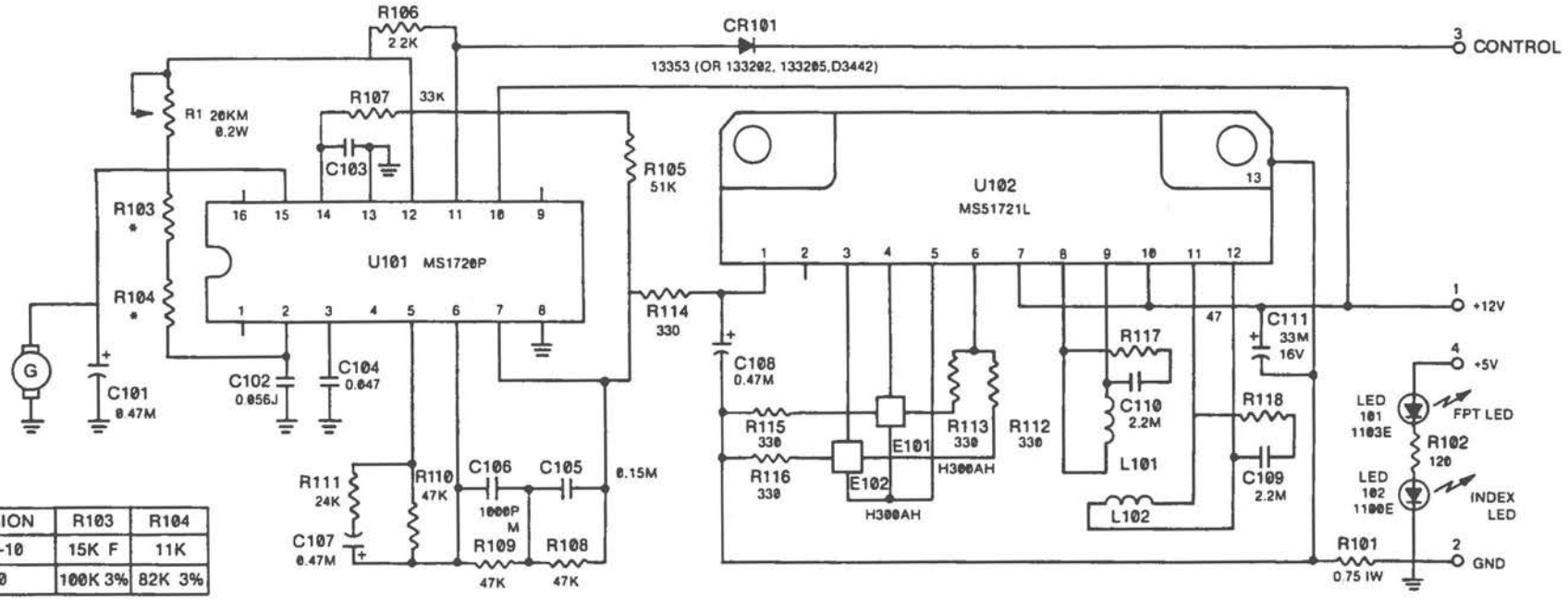
Flexible Disk Drive (3 of 8)



Flexible Disk Drive (4 of 8)



Flexible Disk Drive (5 of 8)

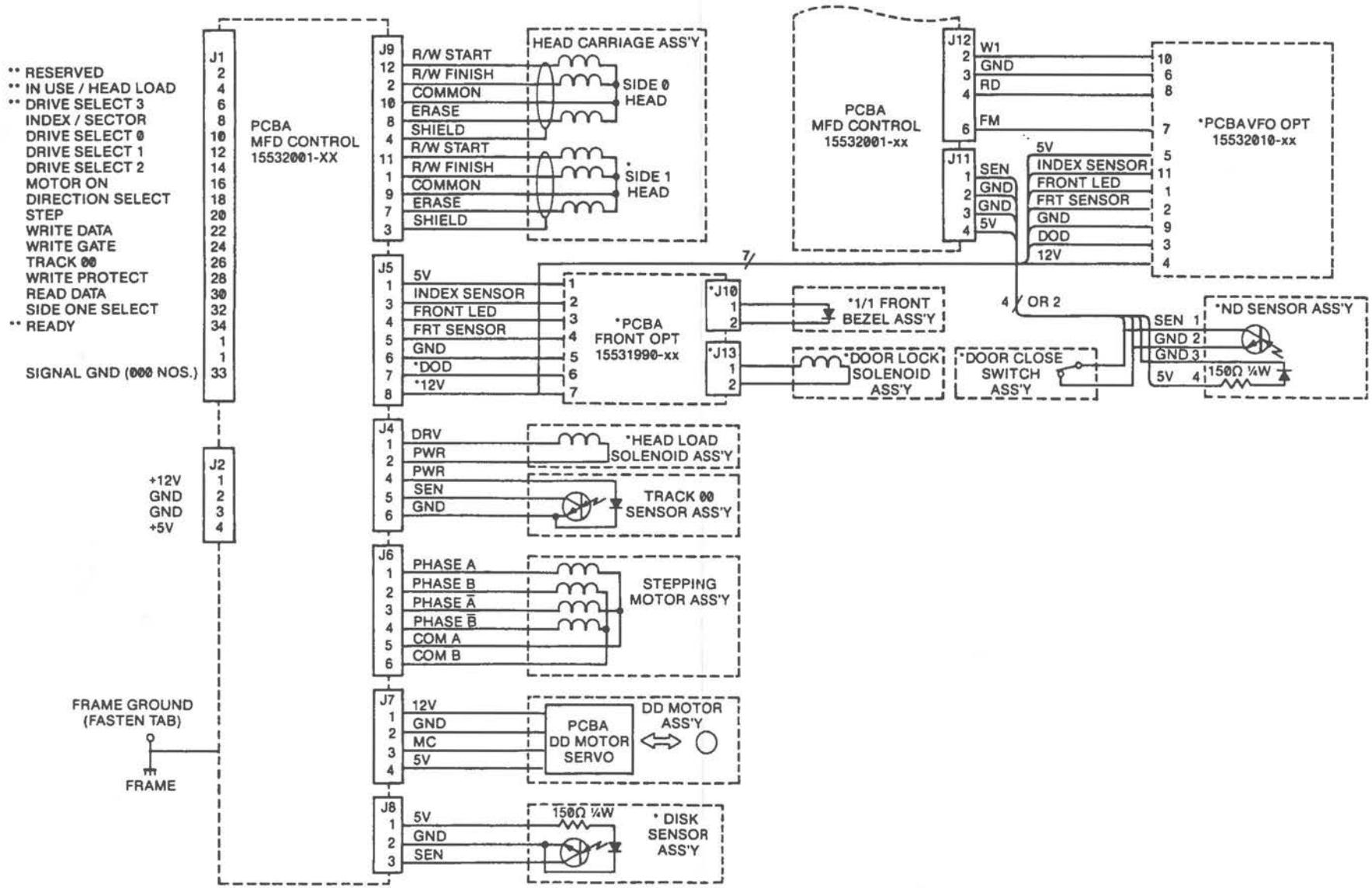


VERSION	R103	R104
-00 -10	15K F	11K
-50	100K 3%	82K 3%

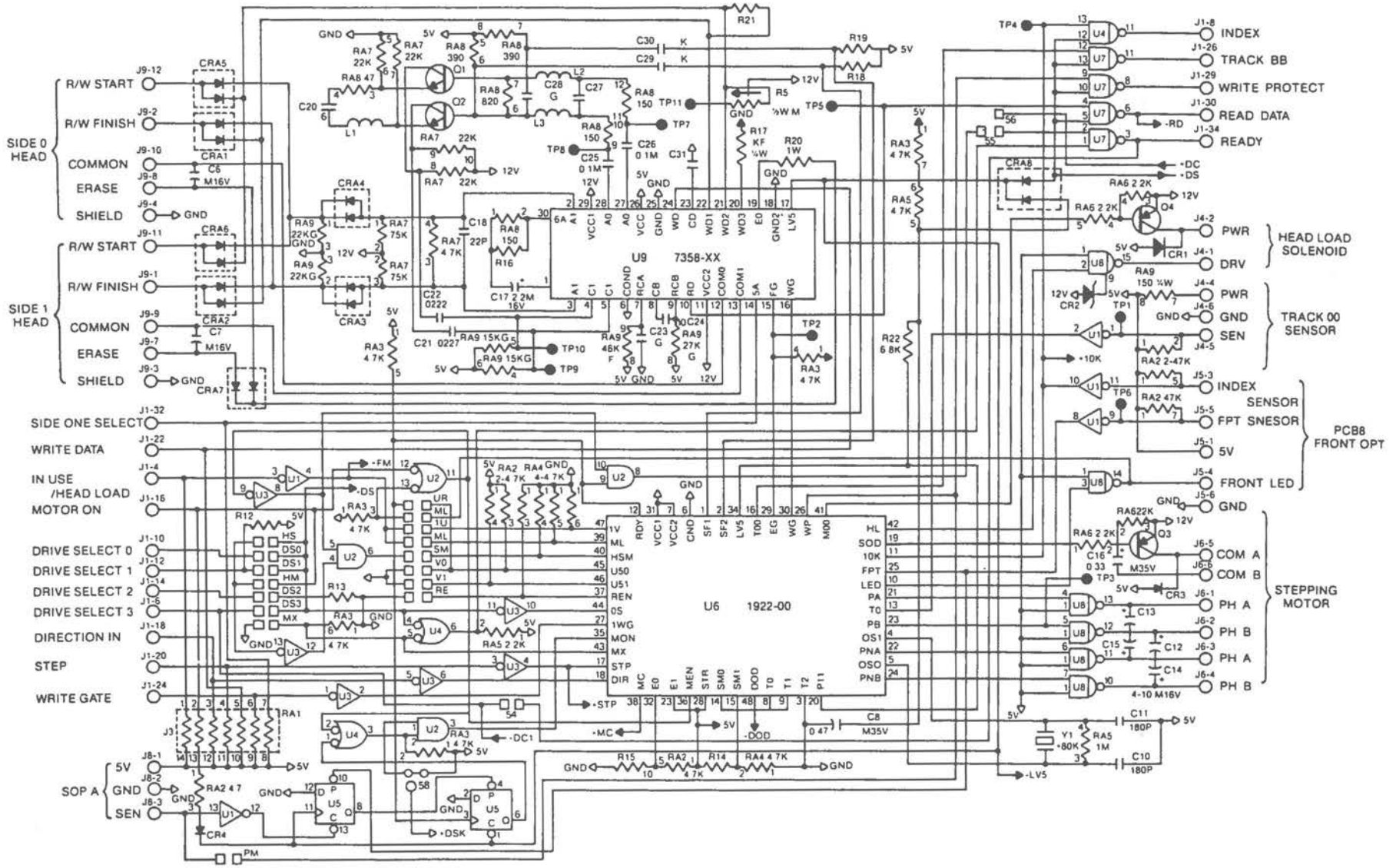
VERSION TABLE

NOTE: RESISTORS ARE ¼W, UNLESS OTHERWISE SPECIFIED.

Flexible Disk Drive (6 of 8)

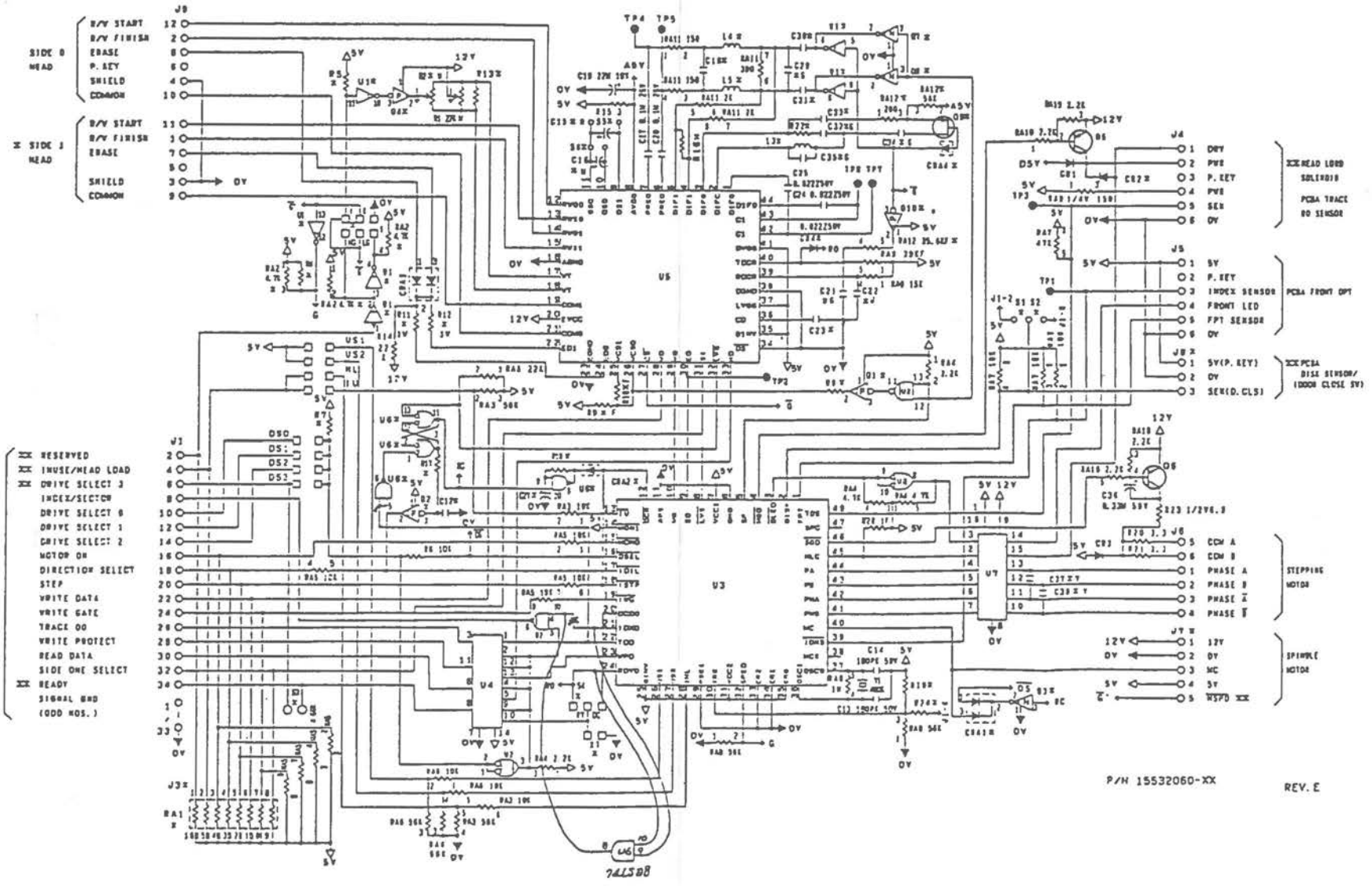


Flexible Disk Drive (7 of 8)

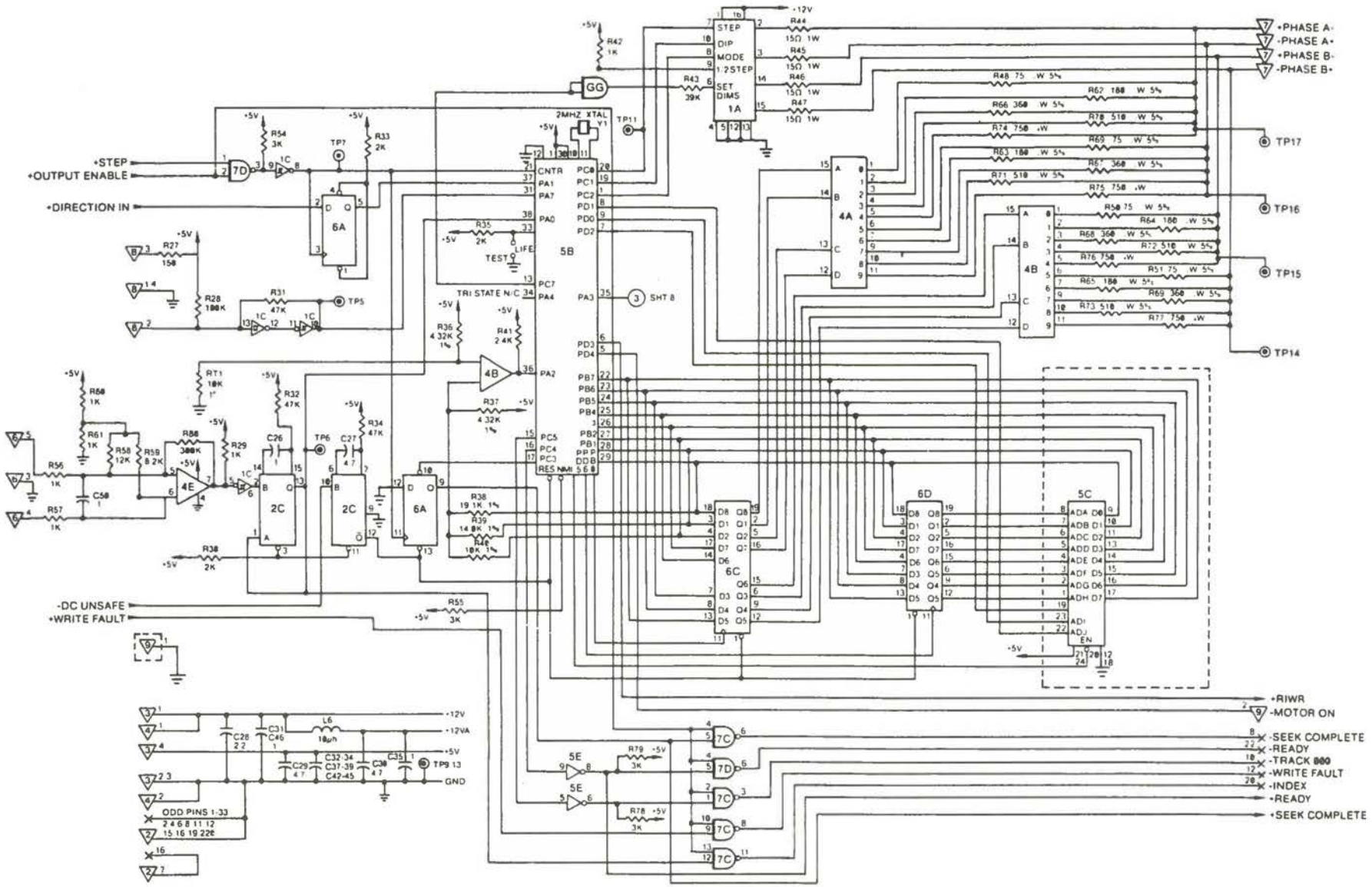


Flexible Disk Drive (8 of 8)

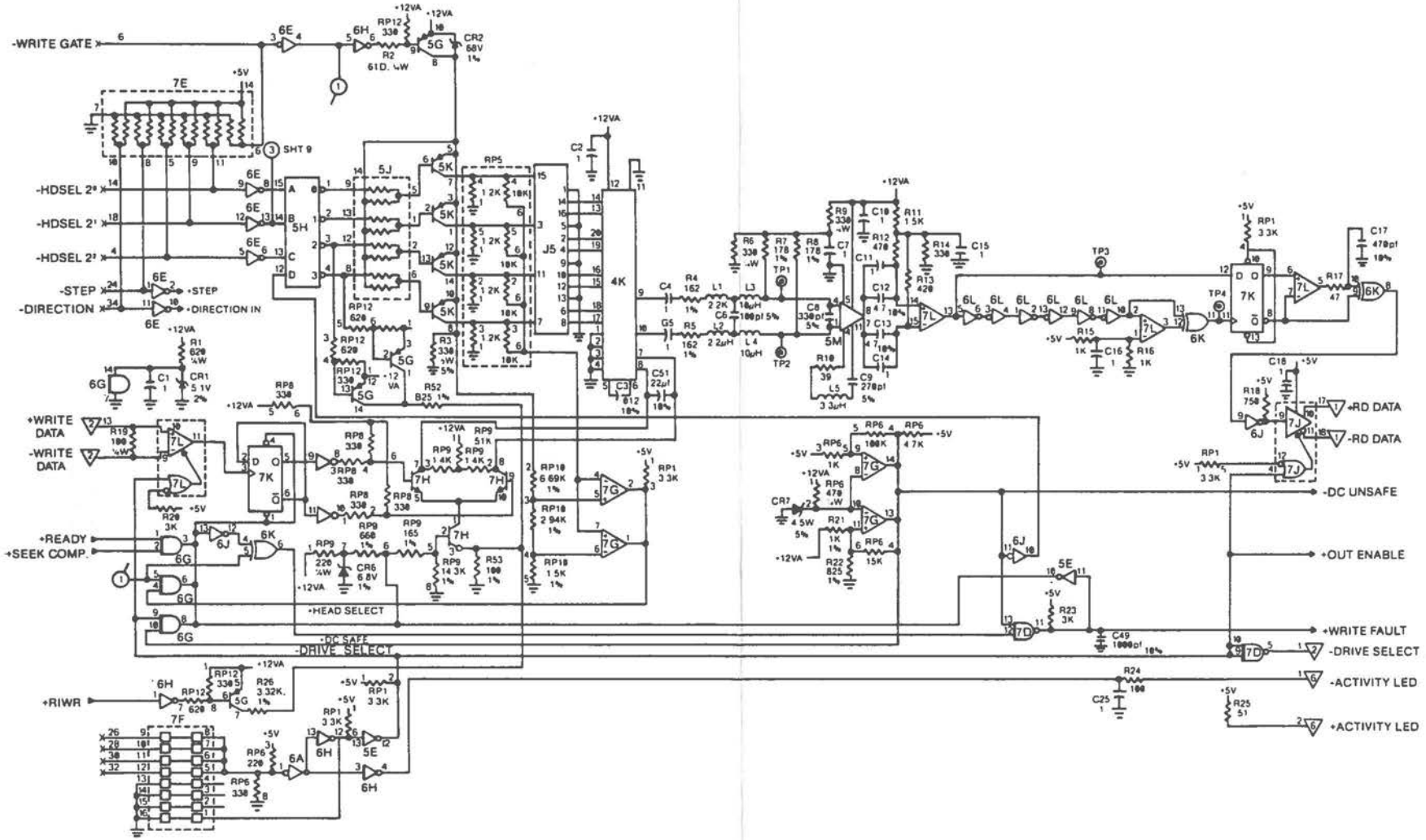
Flexible Disk Drive 1.2 MB



Fixed Disk Drive

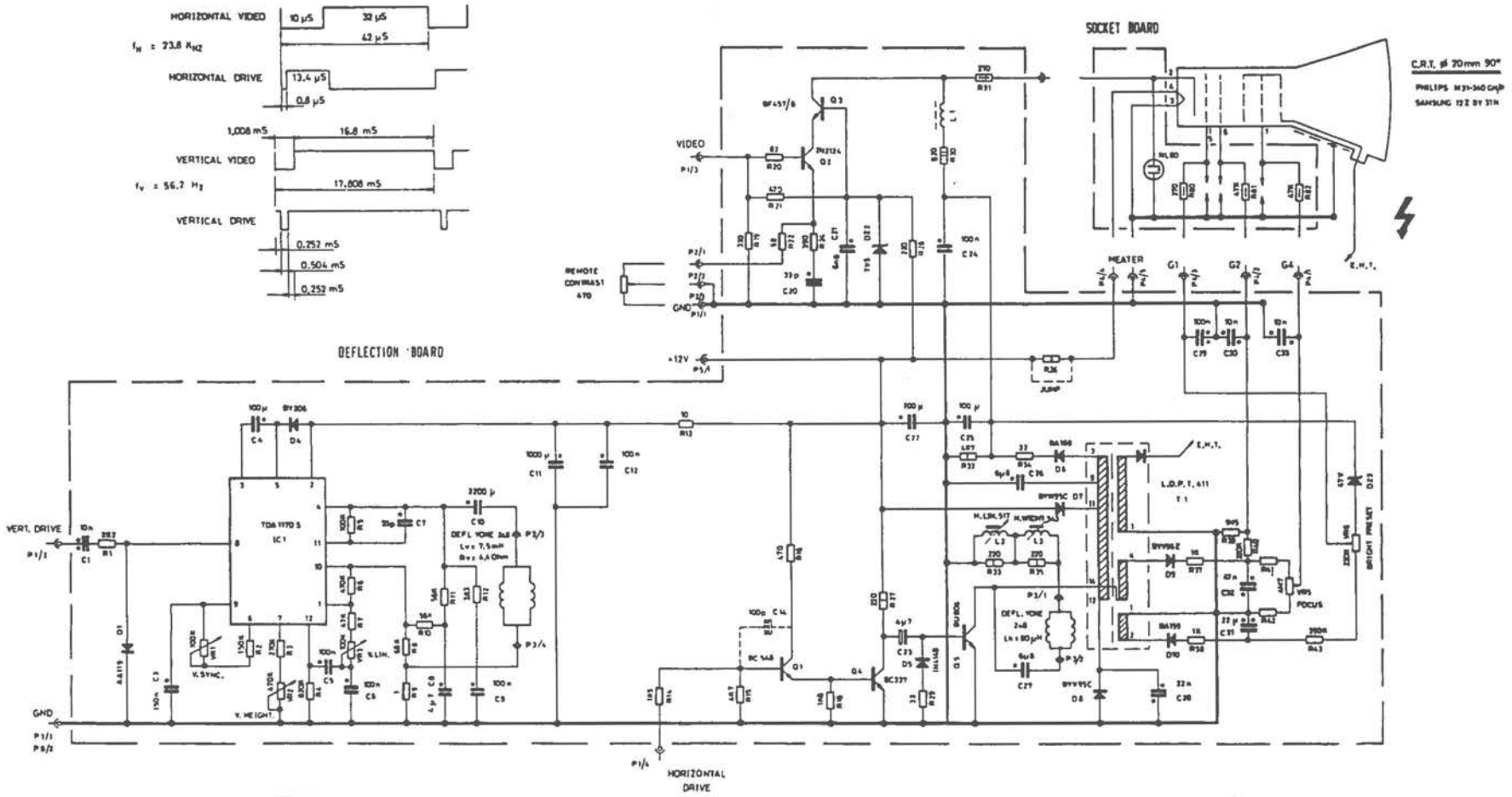


Fixed Disk Drive (1 of 2)



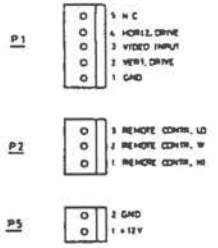
Fixed Disk Drive (2 of 2)

Monochrome CRT



WOUNDS COMPONENTS

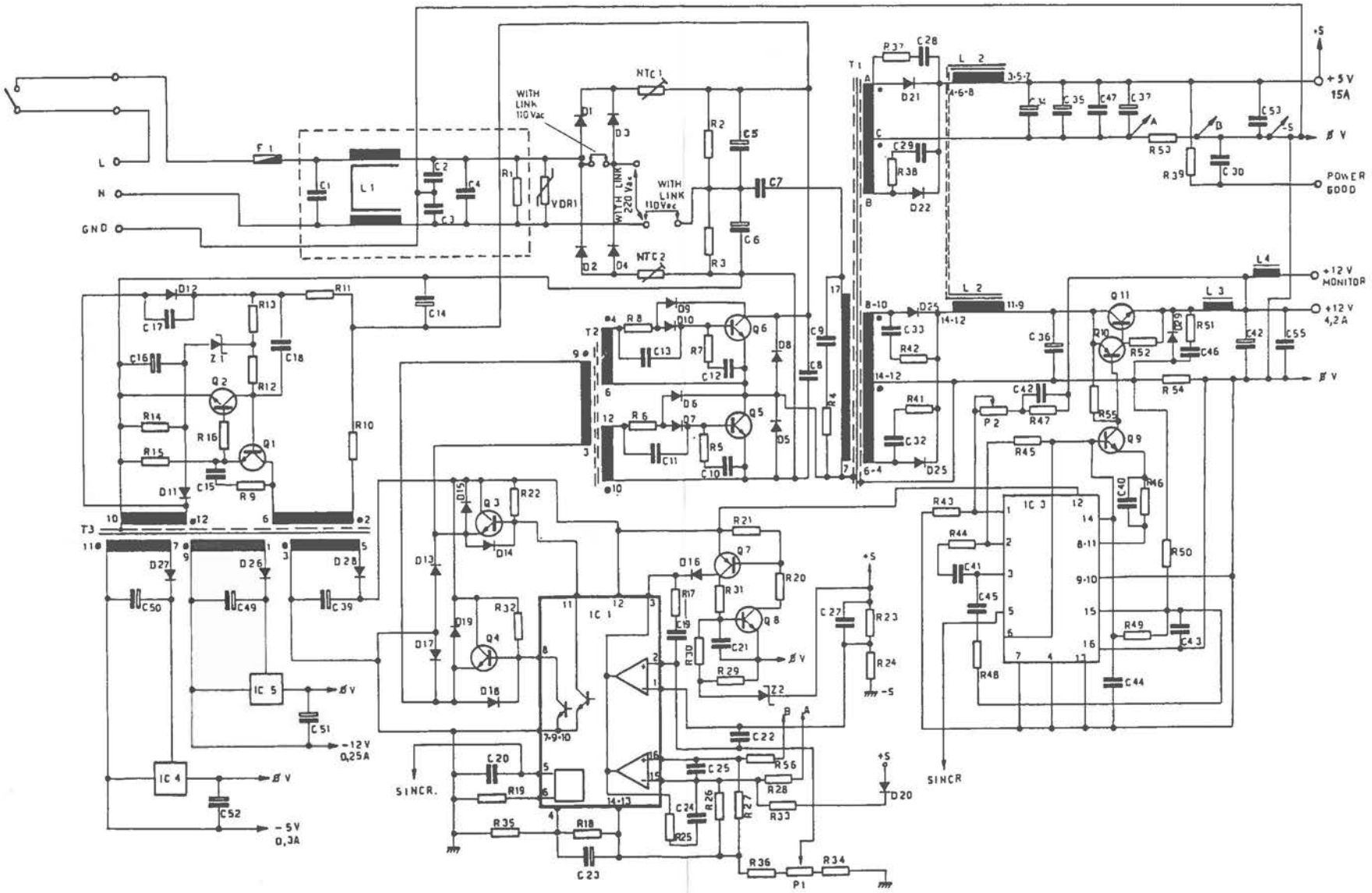
TYPE	CODE
L.O.P.T.	411
LINEARITY COIL	517
AMPLITUDE COIL	543
DEFLECTION YOKE	248



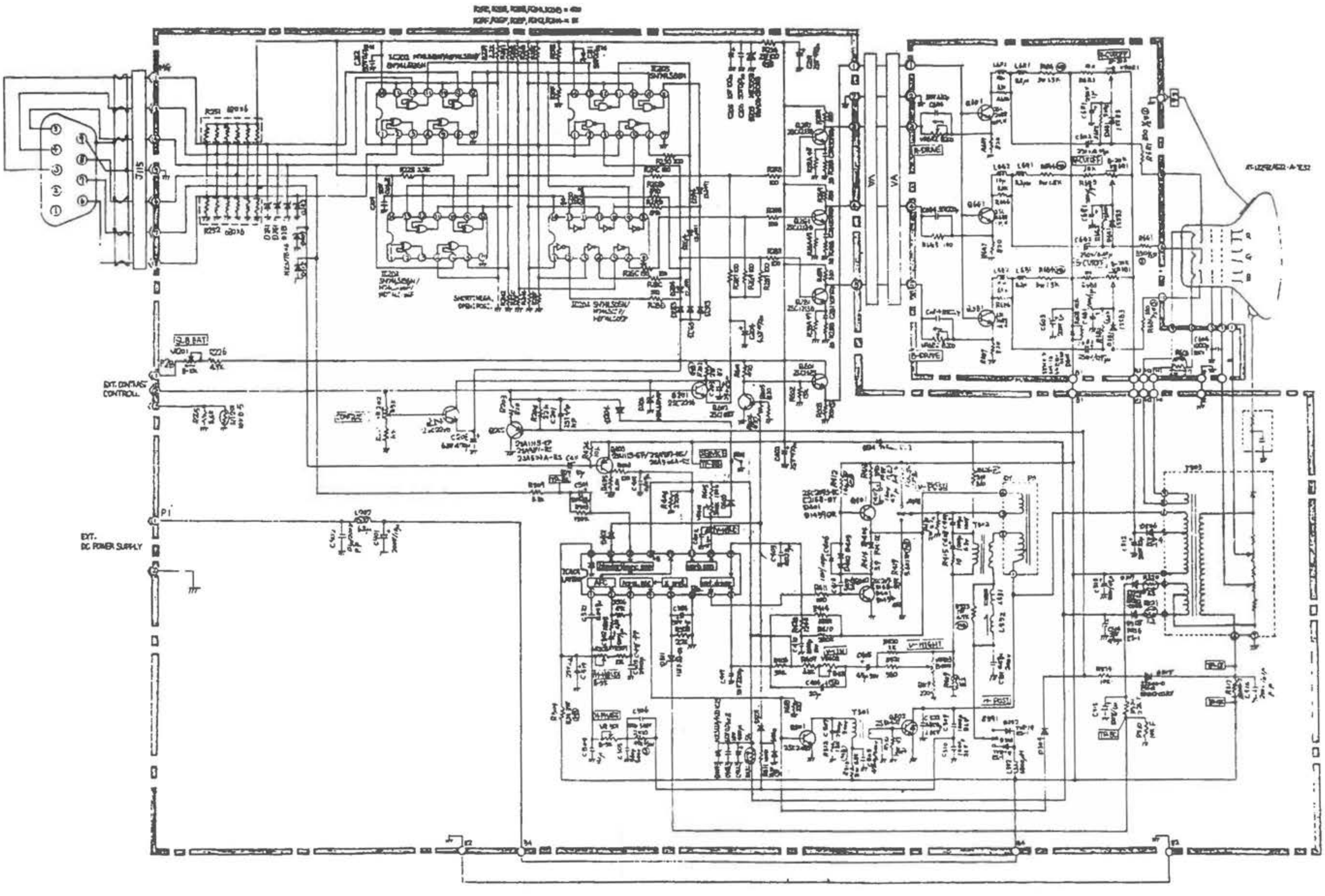
	C.R.T.	
	PHILIPS	SAMSUNG
R 36	0	0
R 41	1M5	2M7
R42	1M2	1004

	CAPACITORS				RESISTORS	
VOLTAGE	ELECTROL.	CERAMIC	POLYEST	POLYPROP		WATT
100 n						1/4
150 n 250						1/2
300–900						1
1000 n						2
						3
						4
						5

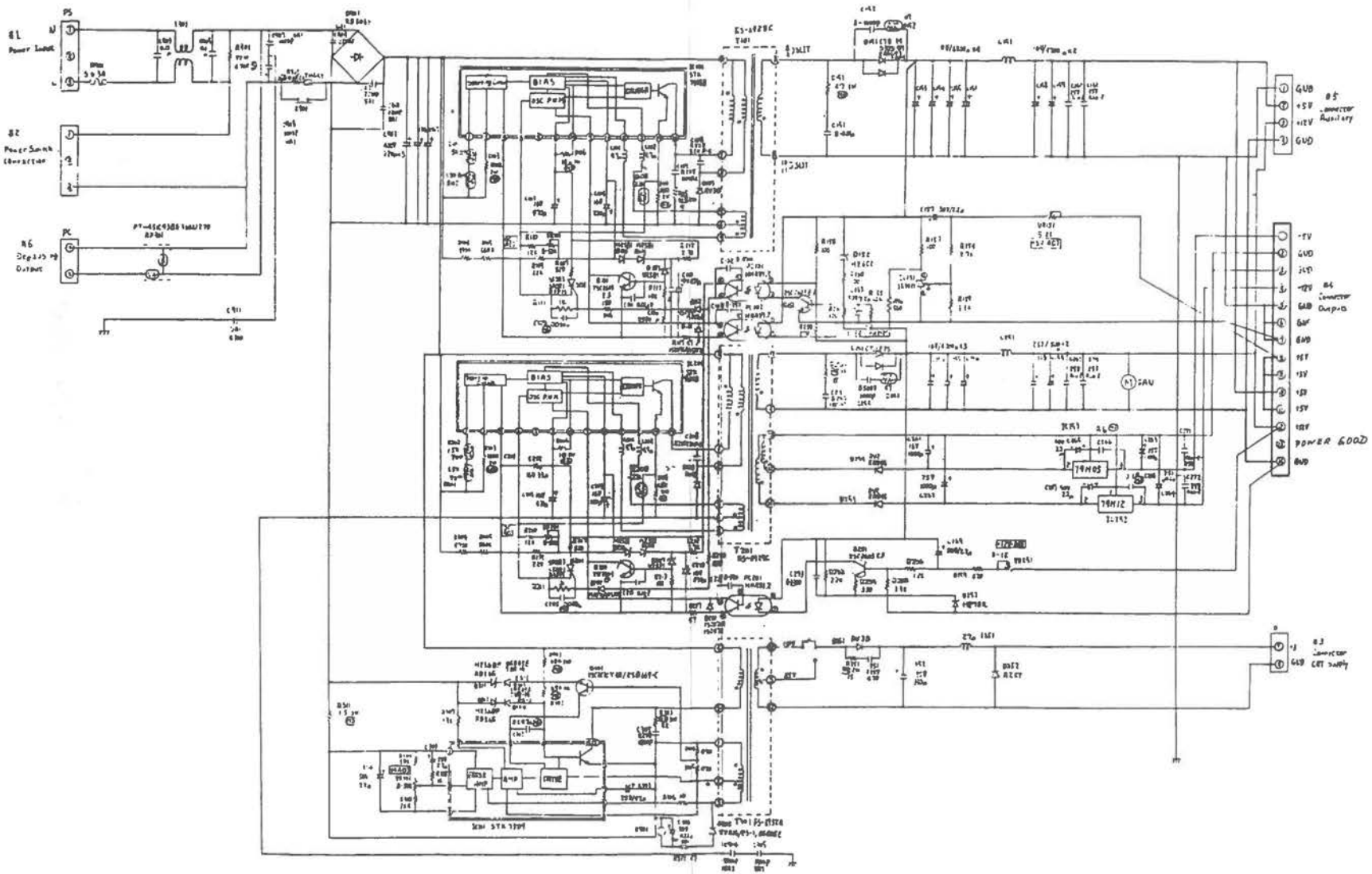
Monochrome Power Supply

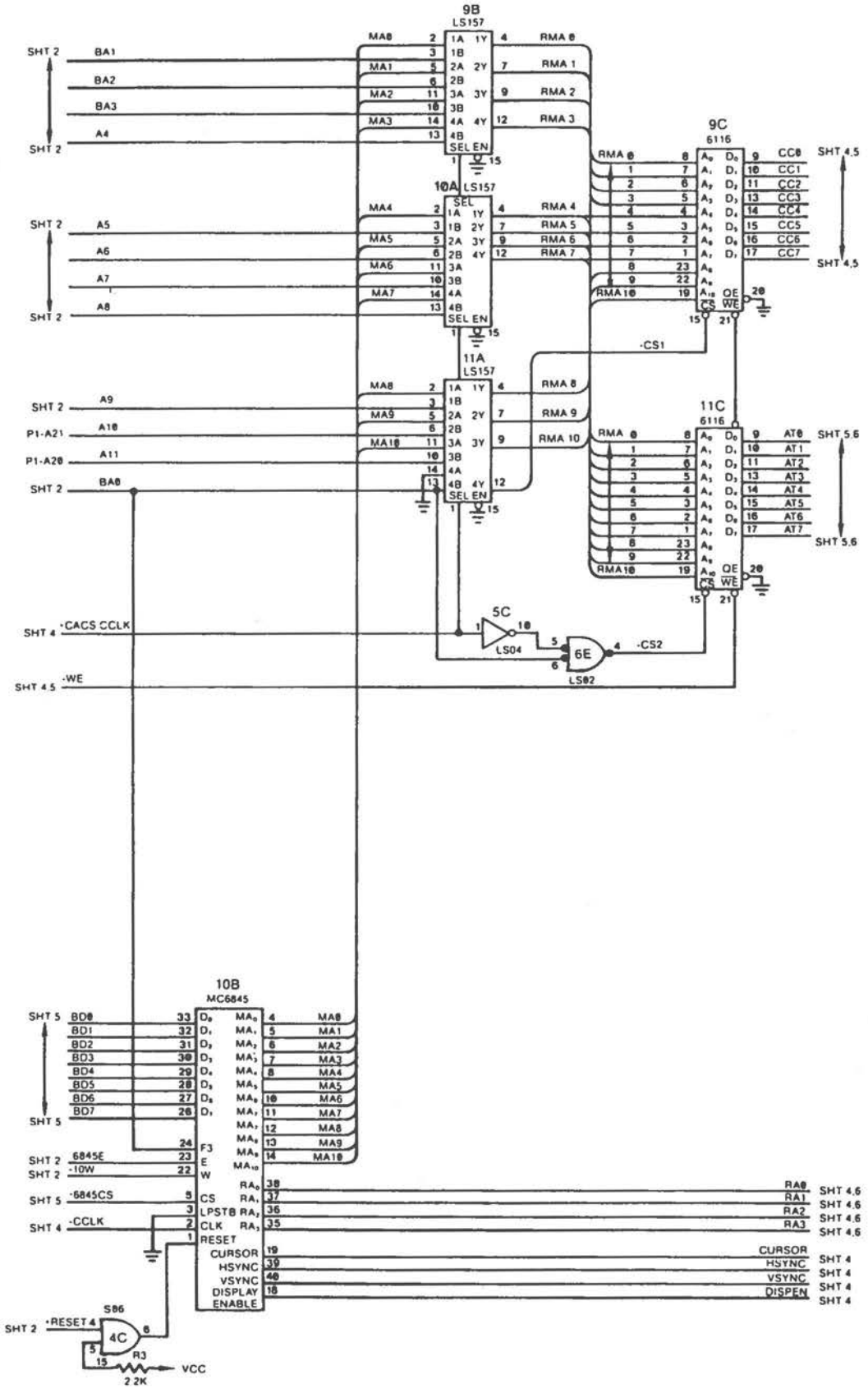


Color CRT

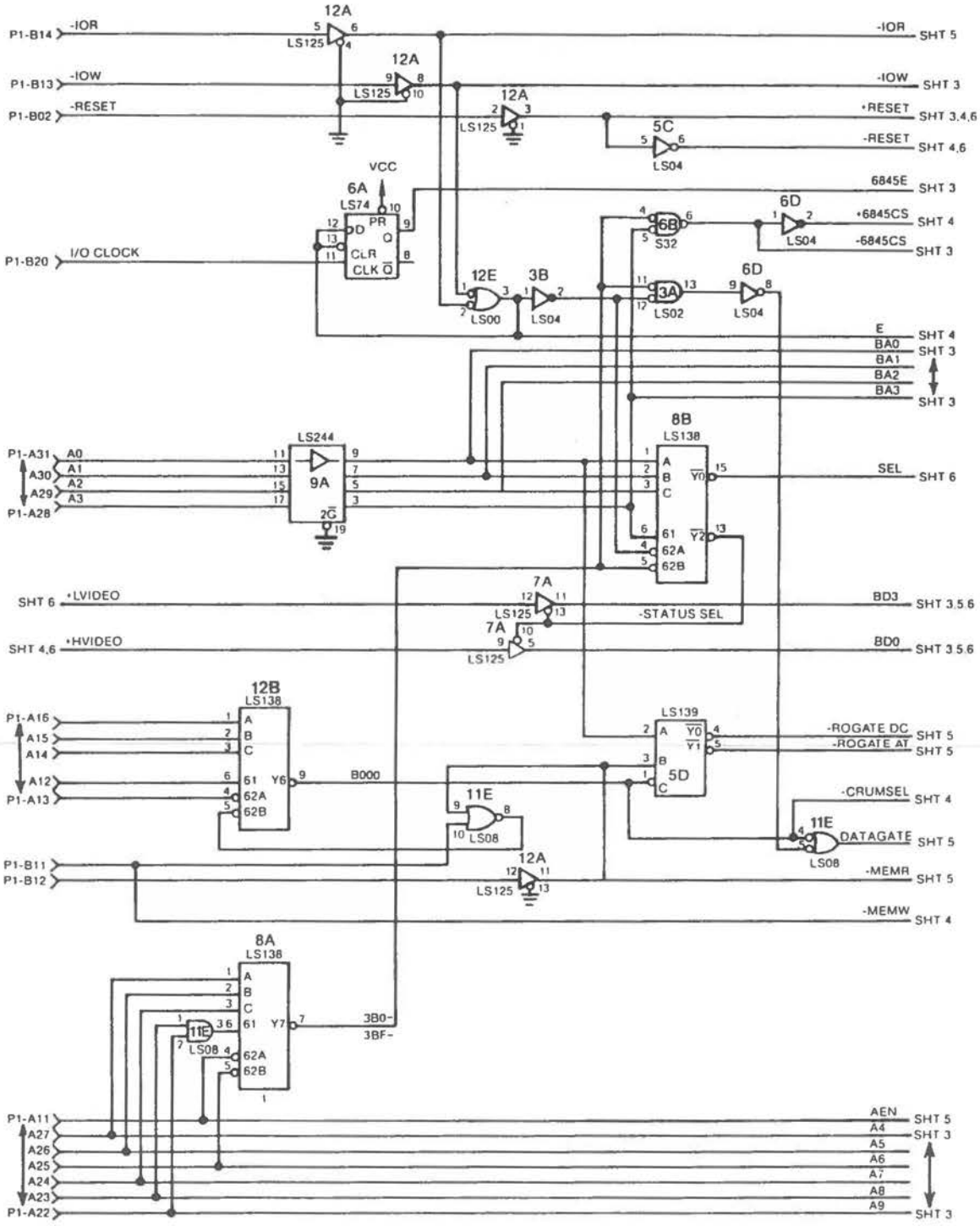

J115
R251
R252
IC202
IC204
IC205
Q201
Q301
Q801
VA
VA
B-DRIVE
R-DRIVE
Q601
Q661
Q631
R
G
B
EXT. CONTRAST
CONTROL
EXT.
DC POWER SUPPLY
P1
AFC
T301
T902
V-HIGHT
D601
D602
D603
D604
D901
TP-Q2
TP-Q1
TP-9C

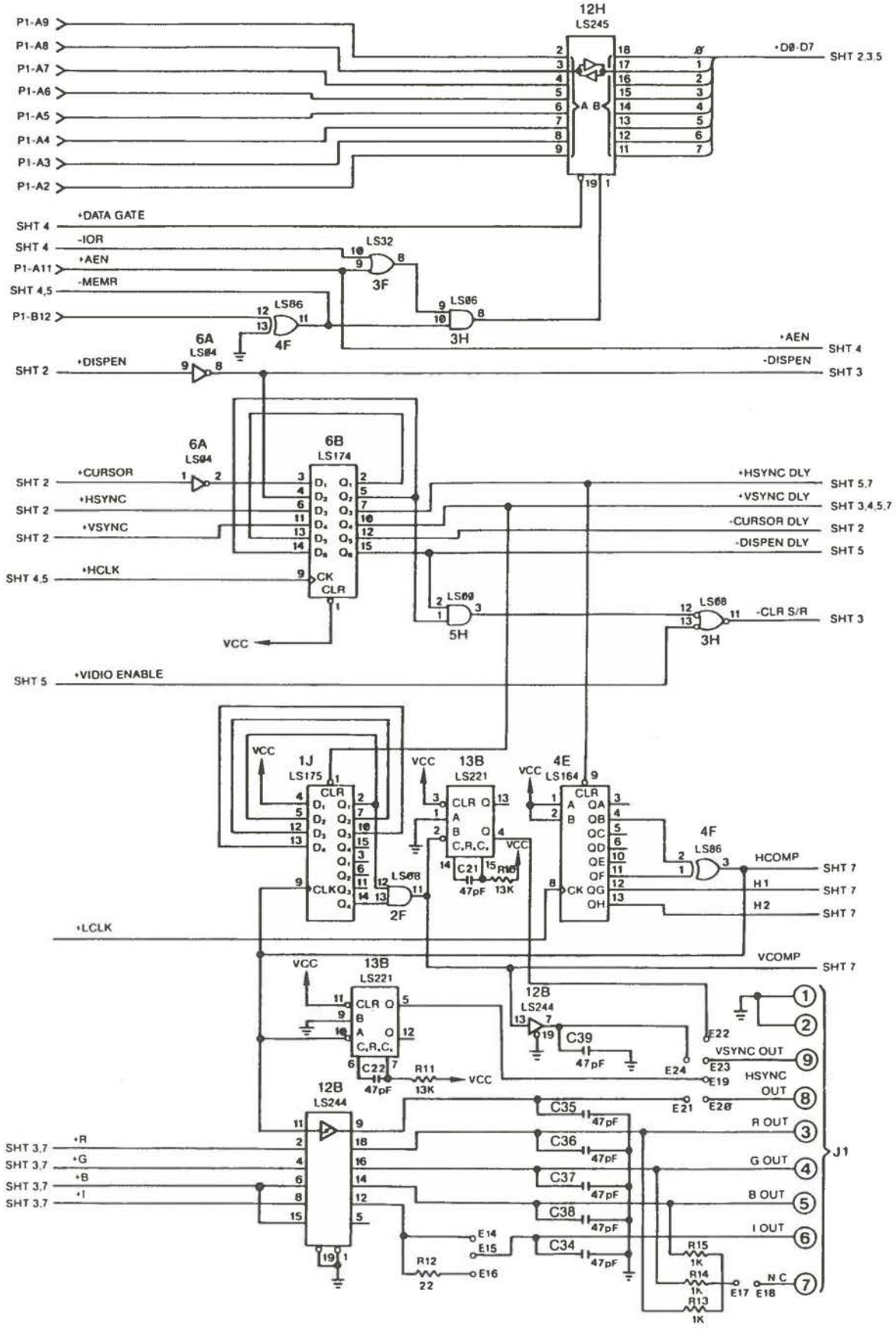
Color Power Supply

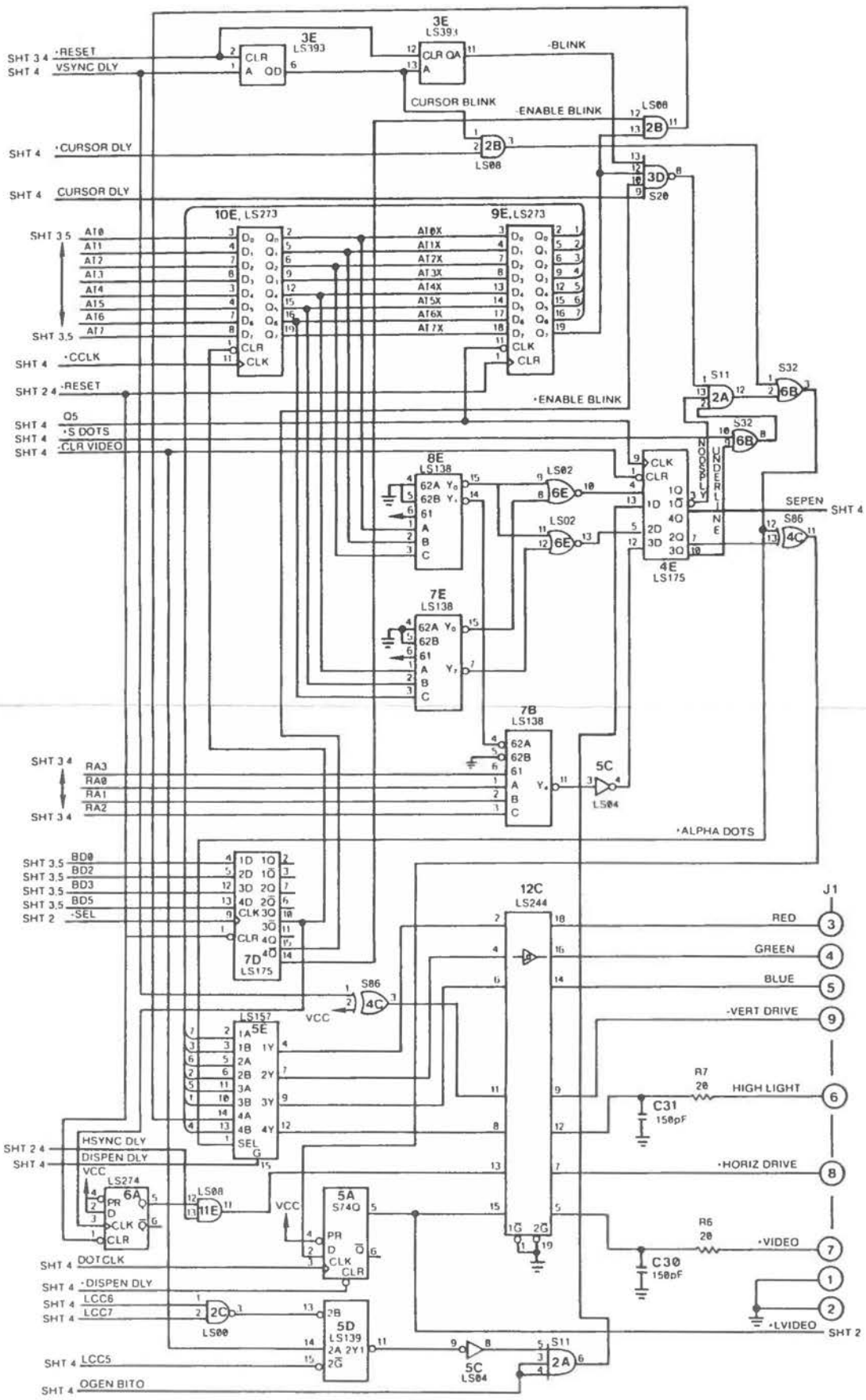


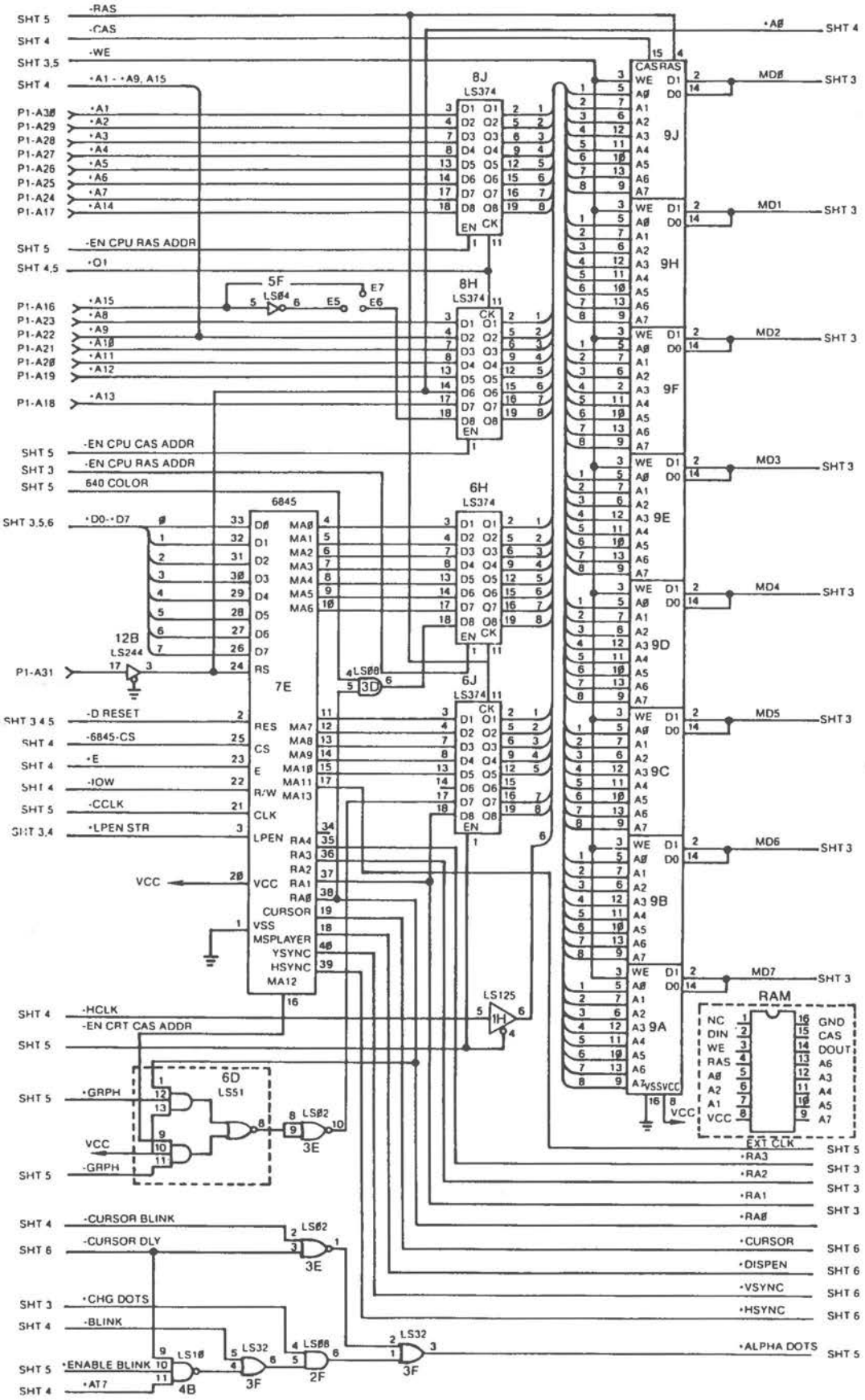


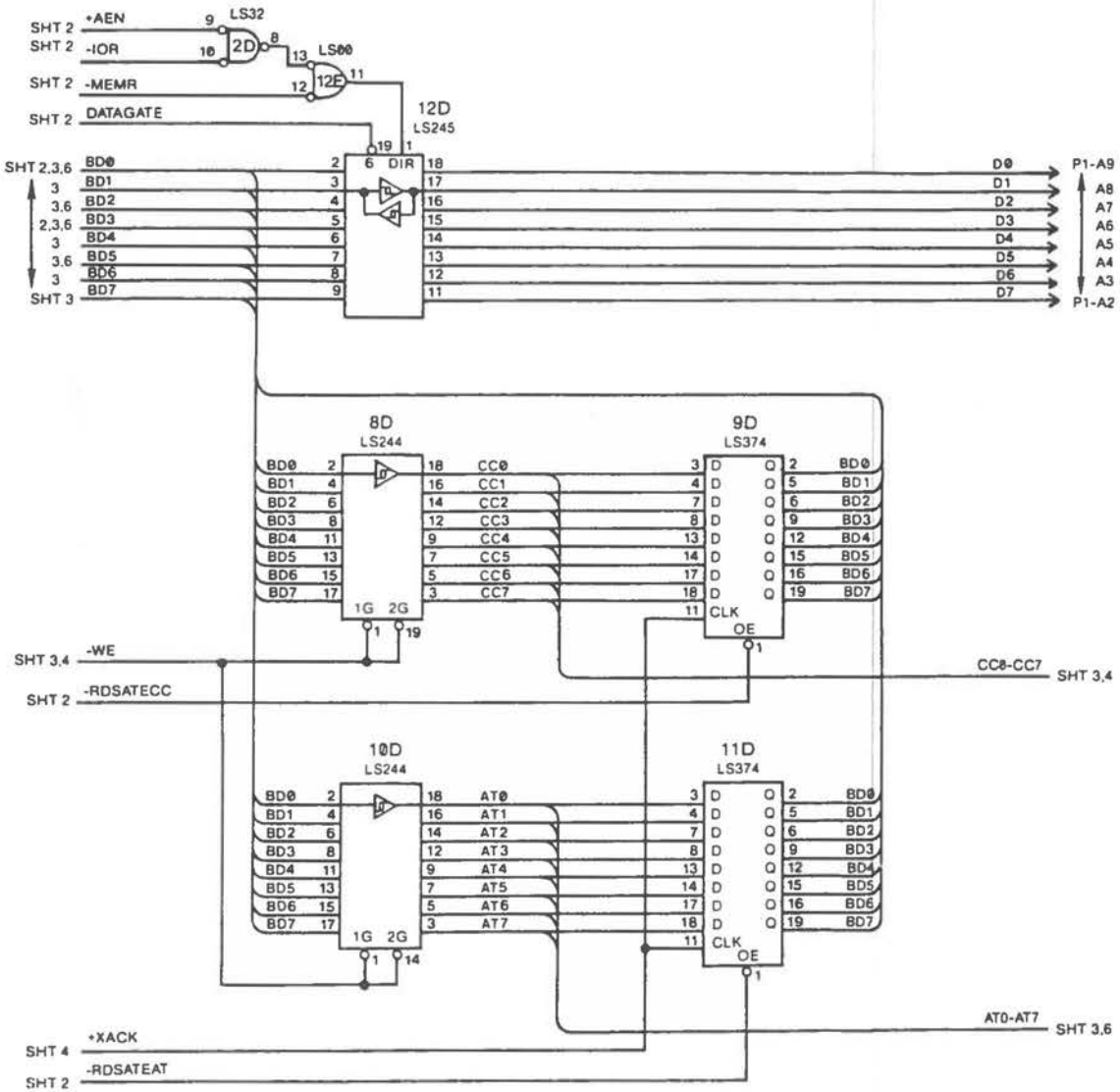
CRT CONTROLLER CHARACTER SET


P1-B14 -IOR
P1-B13 -IOW
P1-B02 -RESET
P1-B20 I/O CLOCK
12A LS125
5C LS04
6A LS74
VCC
6B S32
6D LS04
12E LS00
3B LS04
3A LS02
-IOR SHT 5
-IOW SHT 3
*RESET SHT 3,4,6
-RESET SHT 4,6
6845E SHT 3
+6845CS SHT 4
-6845CS SHT 3
E SHT 4
BA0 SHT 3
BA1
BA2
BA3
SHT 3
P1-A31 A0
A30 A1
A29 A2
P1-A28 A3
LS244 9A
8B LS138
SEL SHT 6
SHT 6 *LVIDEO
SHT 4,6 *HVIDEO
7A LS125
-STATUS SEL
BD3 SHT 3,5,6
BD0 SHT 3,5,6
12B LS138
P1-A16 A15 A14 A12 P1-A13
B000
LS139 5D
-ROGATE DC SHT 5
-ROGATE AT SHT 5
-CRUMSEL SHT 4
11E LS08
DATAGATE SHT 5
P1-B11 P1-B12
-MEMR SHT 5
-MEMW SHT 4
8A LS138
3B0-
3BF-
P1-A11 A27 A26 A25 A24 A23 P1-A22
AEN SHT 5
A4 SHT 3
A5
A6
A7
A8
A9
SHT 3

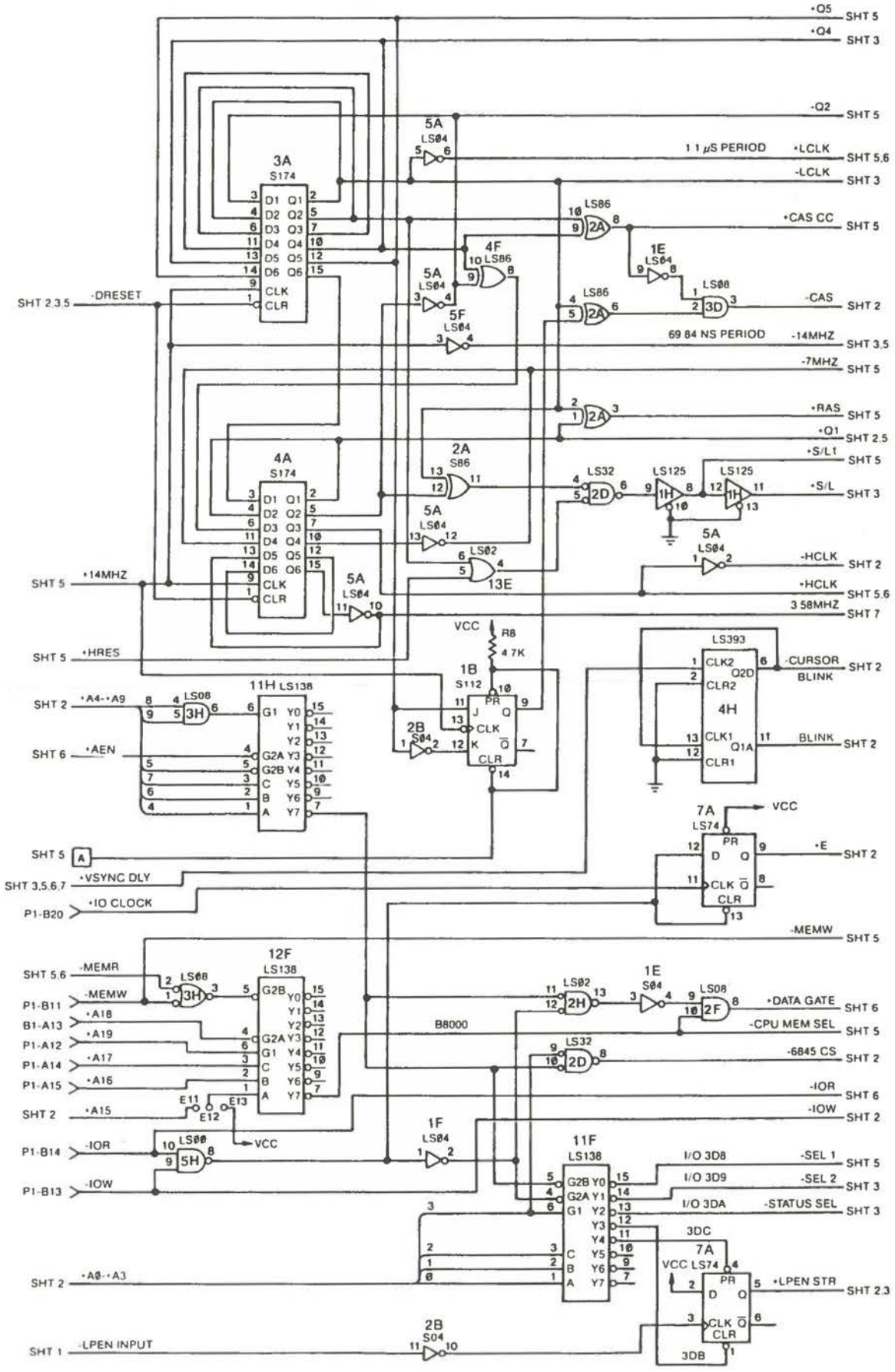

12H
LS245
P1-A9
P1-A8
P1-A7
P1-A6
P1-A5
P1-A4
P1-A3
P1-A2
+D0-D7
SHT 2,3,5
SHT 4
+DATA GATE
SHT 4
-IOR
SHT 4
+AEN
P1-A11
SHT 4,5
-MEMR
P1-B12
LS32
3F
LS86
4F
LS86
3H
6A
LS04
+AEN
SHT 4
SHT 2
+DISPEN
-DISPEN
SHT 3
6A
LS04
6B
LS174
SHT 2
+CURSOR
SHT 2
+HSYNC
SHT 2
+VSYNC
SHT 4,5
+HCLK
+HSYNC DLY
SHT 5,7
+VSYNC DLY
SHT 3,4,5,7
-CURSOR DLY
SHT 2
-DISPEN DLY
SHT 5
LS09
5H
LS08
3H
-CLR S/R
SHT 3
VCC
SHT 5
+VIDIO ENABLE
1J
LS175
13B
LS221
4E
LS164
4F
LS86
LS08
2F
HCOMP
SHT 7
H1
SHT 7
H2
SHT 7
+LCLK
VCOMP
SHT 7
13B
LS221
12B
LS244
C39
47pF
C22
47pF
R11
13K
VCC
12B
LS244
C35
47pF
C36
47pF
C37
47pF
C38
47pF
C34
47pF
SHT 3,7
+R
SHT 3,7
+G
SHT 3,7
+B
SHT 3,7
+I
R12
22
R15
1K
R14
1K
R13
1K
E14
E15
E16
E17
E18
E19
E20
E21
E22
E23
E24
VSYNC OUT
HSYNC OUT
R OUT
G OUT
B OUT
I OUT
N C
J1

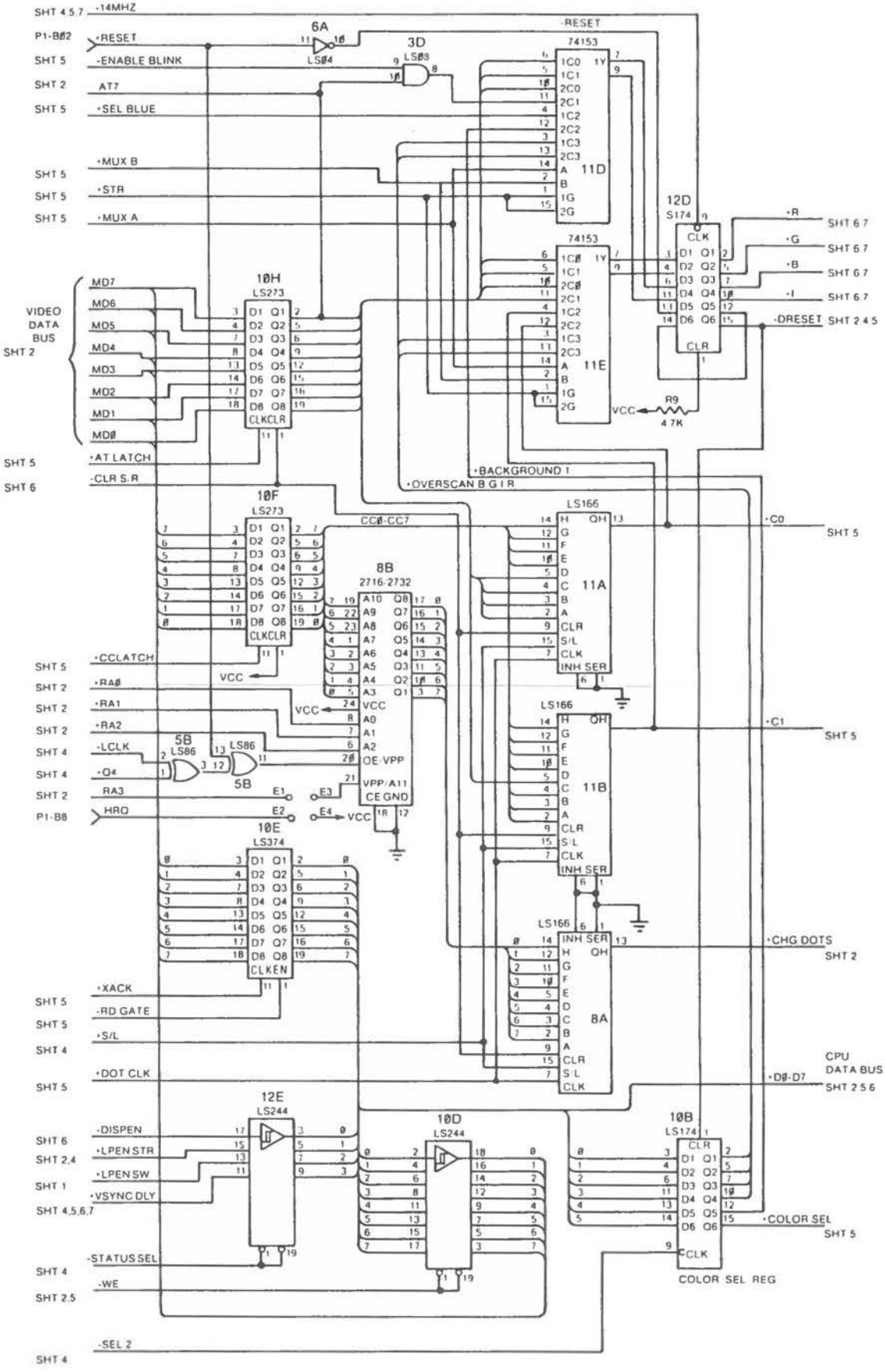

3E LS393
3E LS391
SHT 3 4 ·RESET
SHT 4 VSYNC DLY
·BLINK
CURSOR BLINK
·ENABLE BLINK
LS08
SHT 4 ·CURSOR DLY
SHT 4 CURSOR DLY
S20
10E, LS273
9E, LS273
SHT 3 5 AT0 AT1 AT2 AT3 AT4 AT5 AT6 AT7
AT0X AT1X AT2X AT3X AT4X AT5X AT6X AT7X
SHT 4 ·CCLK
SHT 2 4 ·RESET
·ENABLE BLINK
S11
S32
SHT 4 O5
SHT 4 ·S DOTS
SHT 4 ·CLR VIDEO
8E LS138
7E LS138
7B LS138
LS02
4E LS175
SEPEN
SHT 4
S86
5C LS04
SHT 3 4 RA3 RA0 RA1 RA2
·ALPHA DOTS
SHT 3 5 BD0 BD2 BD3 BD5
SHT 2 ·SEL
7D LS175
12C LS244
J1
RED
GREEN
BLUE
·VERT DRIVE
·HORIZ DRIVE
·VIDEO
·LVIDEO
SHT 2
R7 20 HIGH LIGHT
C31 150pF
R6 20
C30 150pF
VCC
LS157 5E
SHT 2 4 HSYNC DLY
SHT 4 DISPEN DLY
LS274 6A
LS08 11E
5A S74Q
SHT 4 DOTCLK
SHT 4 ·DISPEN DLY
SHT 4 LCC6
SHT 4 LCC7
LS00
5D LS139
SHT 4 LCC5
SHT 4 OGEN BIT0

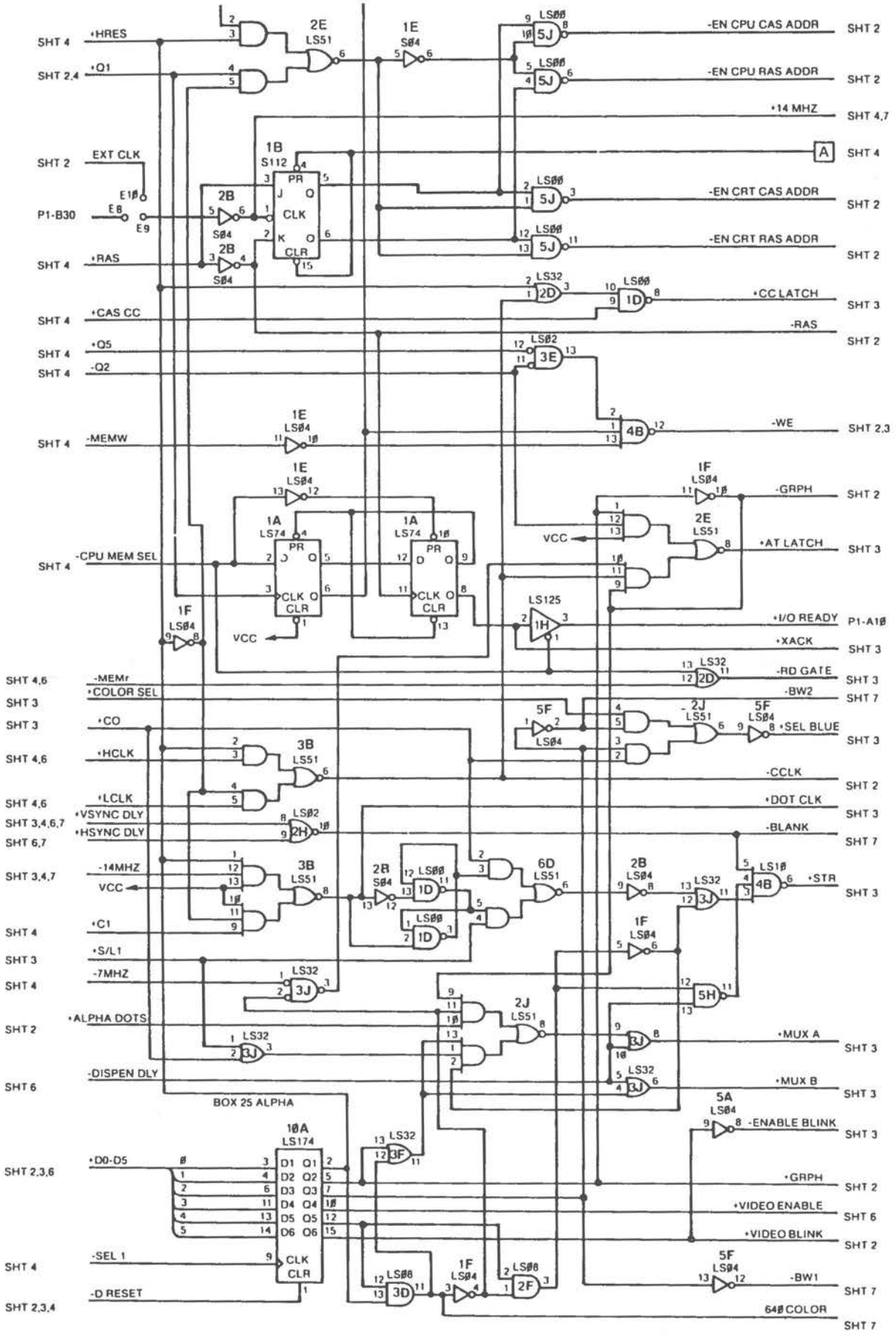

-RAS
-CAS
-WE
•A1 - •A9, A15
•A1
•A2
•A3
•A4
•A5
•A6
•A7
•A14
-EN CPU RAS ADDR
•Q1
•A15
•A8
•A9
•A10
•A11
•A12
•A13
-EN CPU CAS ADDR
-EN CPU RAS ADDR
640 COLOR
•D0-•D7
-D RESET
-6845-CS
•E
-IOW
-CCLK
•LPEN STR
VCC
6845
7E
12B
LS244
8J
8H
6H
6J
LS374
5F
LS04
LS00
LS08
LS125
9J
9H
9F
9E
9D
9C
9B
9A
MD0
MD1
MD2
MD3
MD4
MD5
MD6
MD7
RAM
•A0
•RA3
•RA2
•RA1
•RA0
•CURSOR
•DISPEN
•VSYNC
•HSYNC
•ALPHA DOTS
-HCLK
-EN CRT CAS ADDR
•GRPH
-GRPH
6D
LS51
LS02
3E
-CURSOR BLINK
-CURSOR DLY
•CHG DOTS
-BLINK
•ENABLE BLINK
•AT7
LS10
LS32
LS86
3F
2F
4B

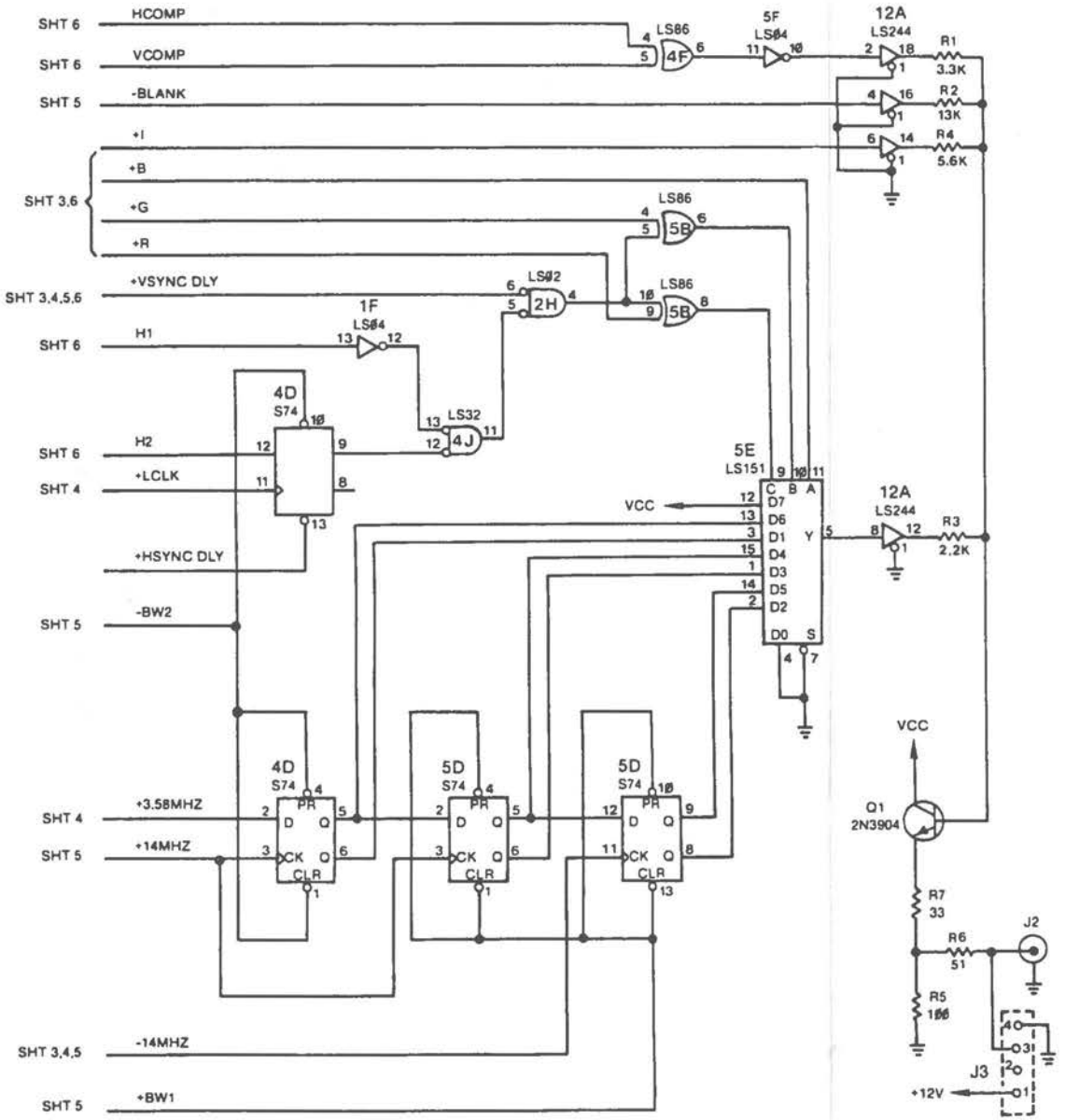


External Color Monitor Controller (2 of 6)


•Q5 SHT 5
•Q4 SHT 3
-Q2 SHT 5
1.1 µS PERIOD •LCLK SHT 5,6
-LCLK SHT 3
•CAS CC SHT 5
-CAS SHT 2
69.84 NS PERIOD -14MHZ SHT 3,5
-7MHZ SHT 5
•RAS SHT 5
•Q1 SHT 2,5
•S/L1 SHT 5
•S/L SHT 3
-HCLK SHT 2
•HCLK SHT 5,6
3.58MHZ SHT 7
-CURSOR BLINK SHT 2
BLINK SHT 2
•E SHT 2
-MEMW SHT 5
•DATA GATE SHT 6
-CPU MEM SEL SHT 5
-6845 CS SHT 2
-IOR SHT 6
-IOW SHT 2
-SEL 1 SHT 5
-SEL 2 SHT 3
-STATUS SEL SHT 3
•LPEN STR SHT 2,3
SHT 2,3,5 -DRESET
SHT 5 •14MHZ
SHT 5 •HRES
SHT 2 •A4-•A9
SHT 6 •AEN
SHT 5 A
SHT 3,5,6,7 •VSYNC DLY
P1-B20 •IO CLOCK
SHT 5,6 -MEMR
P1-B11 -MEMW
B1-A13 •A18
P1-A12 •A19
P1-A14 •A17
P1-A15 •A16
SHT 2 •A15
P1-B14 -IOR
P1-B13 -IOW
SHT 2 •A0-•A3
SHT 1 -LPEN INPUT
B8000
VCC
3A S174
4A S174
5A LS04
1E LS04
5F LS04
2A S86
4F LS86
LS86 2A
LS08 3D
LS32 2D
LS125 1H
LS125 1H
5A LS04
13E LS02
R8 4.7K
1B S112
2B S04
LS393 4H
7A LS74
11H LS138
LS08 3H
12F LS138
LS08 3H
LS08 5H
1F LS04
11F LS138
LS02 2H
1E S04
LS08 2F
LS32 2D
3DC
7A LS74
2B S04
3DB

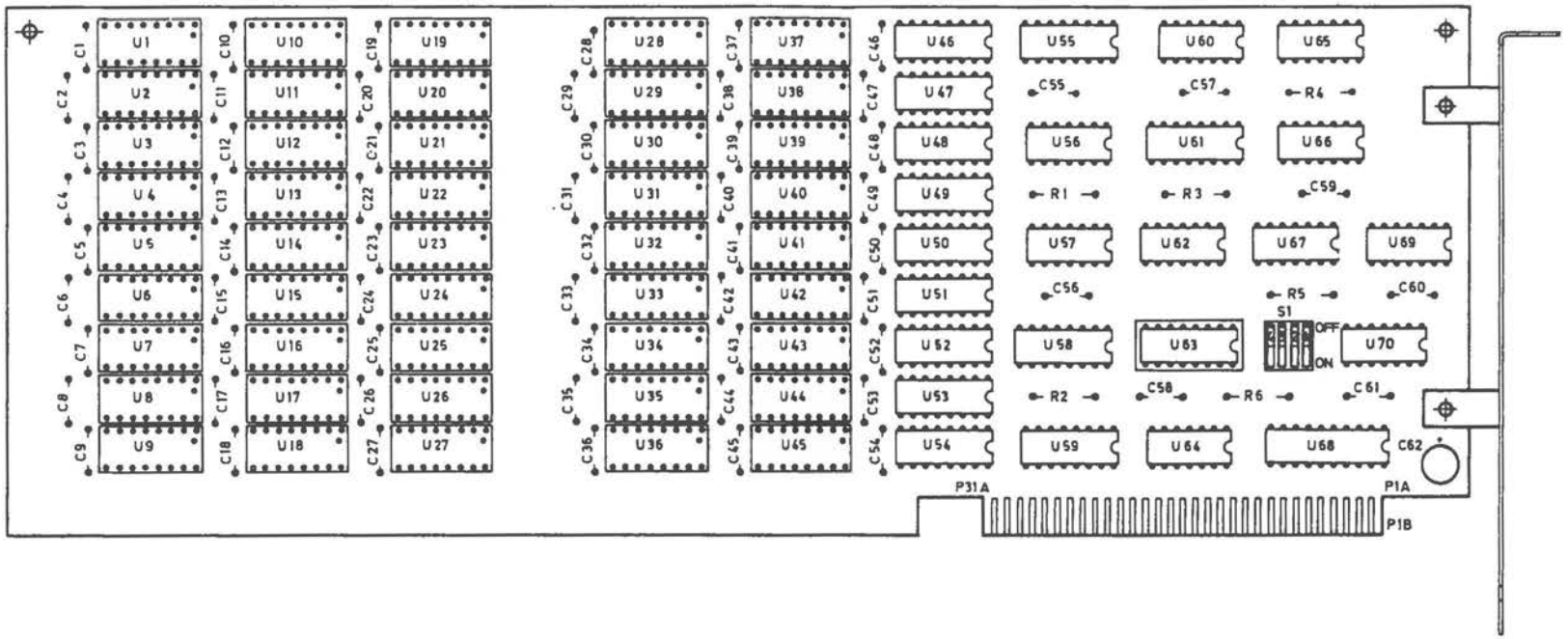

SHT 4,5,7 -14MHZ
P1-B02 *RESET
SHT 5 -ENABLE BLINK
SHT 2 AT7
SHT 5 *SEL BLUE
SHT 5 *MUX B
SHT 5 *STR
SHT 5 *MUX A
6A
LS04
3D
LS08
-RESET
74153
11D
74153
11E
12D
S174
CLK
CLR
*R
*G
*B
*I
*DRESET
SHT 6,7
SHT 2,4,5
R9
VCC
4.7K
VIDEO DATA BUS
SHT 2
MD7
MD6
MD5
MD4
MD3
MD2
MD1
MD0
10H
LS273
CLKCLR
SHT 5 *AT LATCH
SHT 6 -CLR S.R
*BACKGROUND 1
*OVERSCAN B G I R
10F
LS273
CLKCLR
CC0-CC7
8B
2716/2732
SHT 5 *CCLATCH
SHT 2 *RA0
SHT 2 *RA1
SHT 2 *RA2
SHT 4 -LCLK
SHT 4 -O4
SHT 2 RA3
P1-B8 HRQ
5B
LS86
VCC
E1
E2
E3
E4
10E
LS374
CLKEN
SHT 5 *XACK
SHT 5 -RD GATE
SHT 5 *S/L
SHT 4
SHT 5 *DOT CLK
LS166
11A
LS166
11B
LS166
8A
*C0
*C1
*CHG DOTS
SHT 5
SHT 2
CPU DATA BUS
*D0-D7
SHT 2,5,6
12E
LS244
10D
LS244
10B
LS174
CLR
CLK
COLOR SEL REG
*COLOR SEL
SHT 5
SHT 6 -DISPEN
SHT 2,4 *LPEN STR
SHT 1 *LPEN SW
*VSYNC DLY
SHT 4,5,6,7
-STATUS SEL
SHT 4
SHT 2,5 -WE
-SEL 2
SHT 4

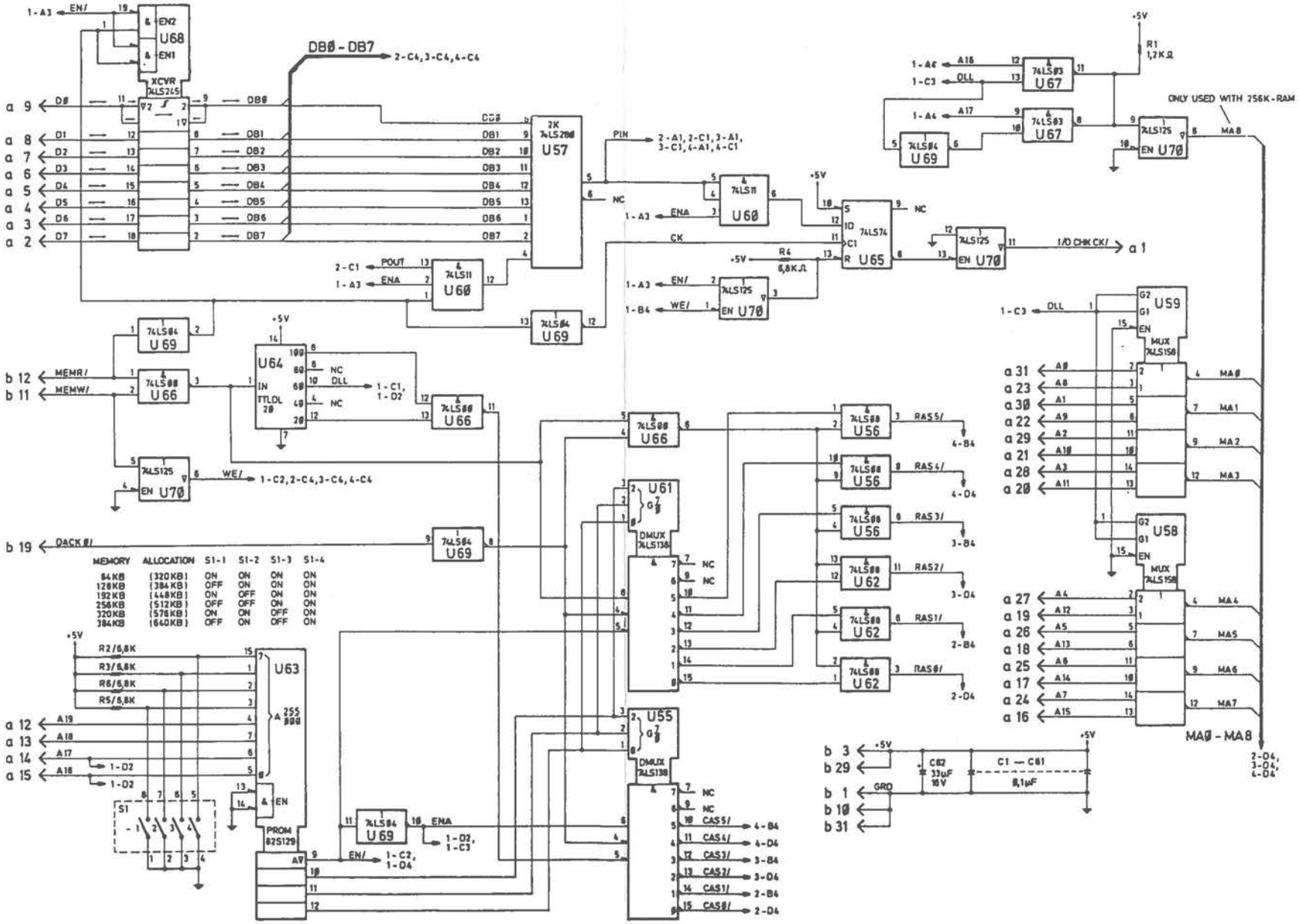

SHT 4 +HRES
SHT 2,4 +Q1
SHT 2 EXT CLK
P1-B30
SHT 4 +RAS
SHT 4 +CAS CC
SHT 4 +Q5
SHT 4 -Q2
SHT 4 -MEMW
SHT 4 -CPU MEM SEL
SHT 4,6 -MEMr
SHT 3 +COLOR SEL
SHT 3 +CO
SHT 4,6 +HCLK
SHT 4,6 +LCLK
SHT 3,4,6,7 +VSYNC DLY
SHT 6,7 +HSYNC DLY
SHT 3,4,7 -14MHZ
VCC
SHT 4 +C1
SHT 3 +S/L1
SHT 4 -7MHZ
SHT 2 +ALPHA DOTS
SHT 6 -DISPEN DLY
BOX 25 ALPHA
SHT 2,3,6 +D0-D5
SHT 4 -SEL 1
SHT 2,3,4 -D RESET
-EN CPU CAS ADDR SHT 2
-EN CPU RAS ADDR SHT 2
+14 MHZ SHT 4,7
A SHT 4
-EN CRT CAS ADDR SHT 2
-EN CRT RAS ADDR SHT 2
+CC LATCH SHT 3
-RAS SHT 2
-WE SHT 2,3
-GRPH SHT 2
+AT LATCH SHT 3
+I/O READY P1-A1Ø
+XACK SHT 3
-RD GATE SHT 3
-BW2 SHT 7
+SEL BLUE SHT 3
-CCLK SHT 2
+DOT CLK SHT 3
-BLANK SHT 7
+STR SHT 3
+MUX A SHT 3
+MUX B SHT 3
-ENABLE BLINK SHT 3
+GRPH SHT 2
+VIDEO ENABLE SHT 6
+VIDEO BLINK SHT 2
-BW1 SHT 7
64Ø COLOR SHT 7



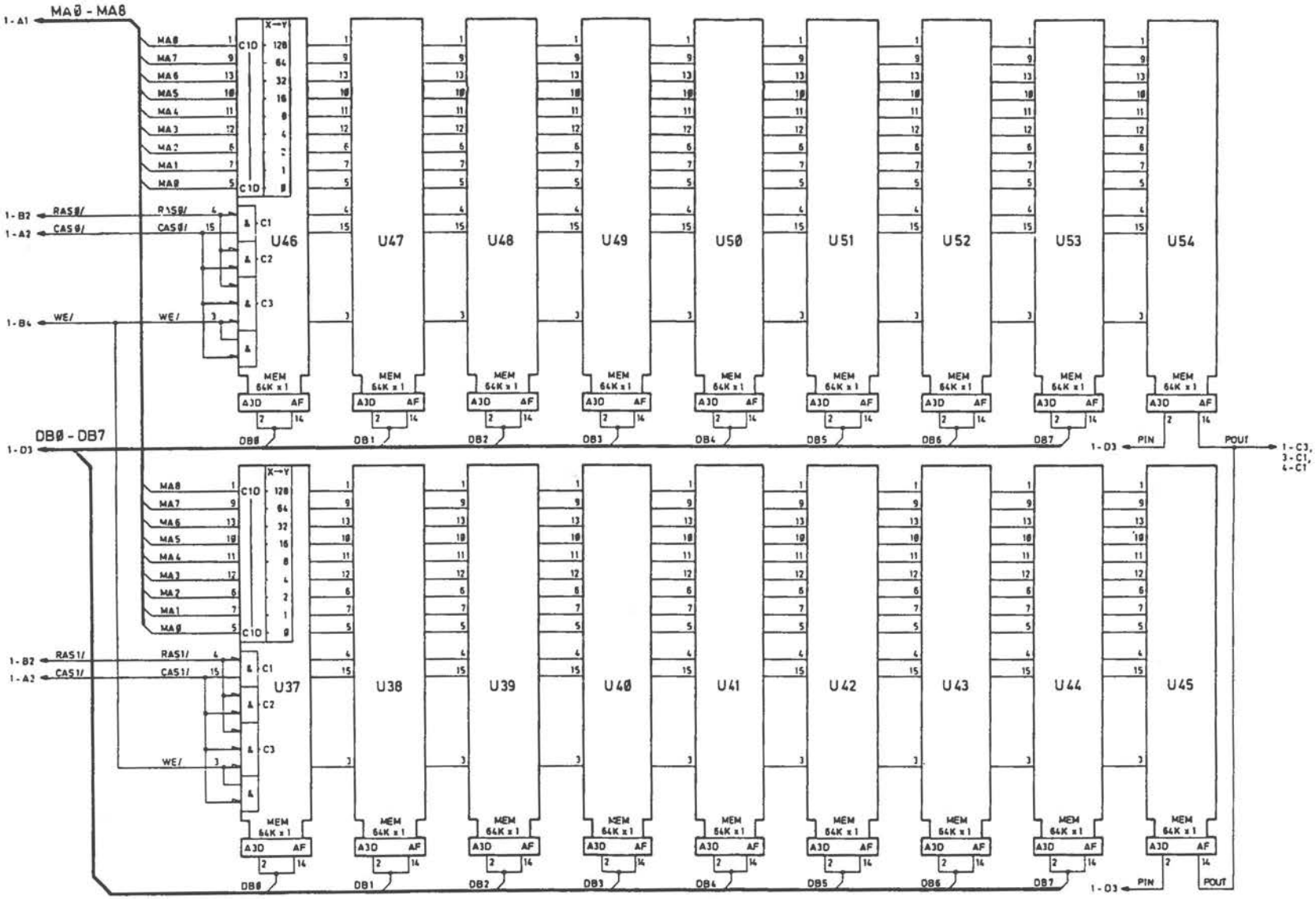
External Color Monitor Controller (6 of 6)

Memory Expansion Board

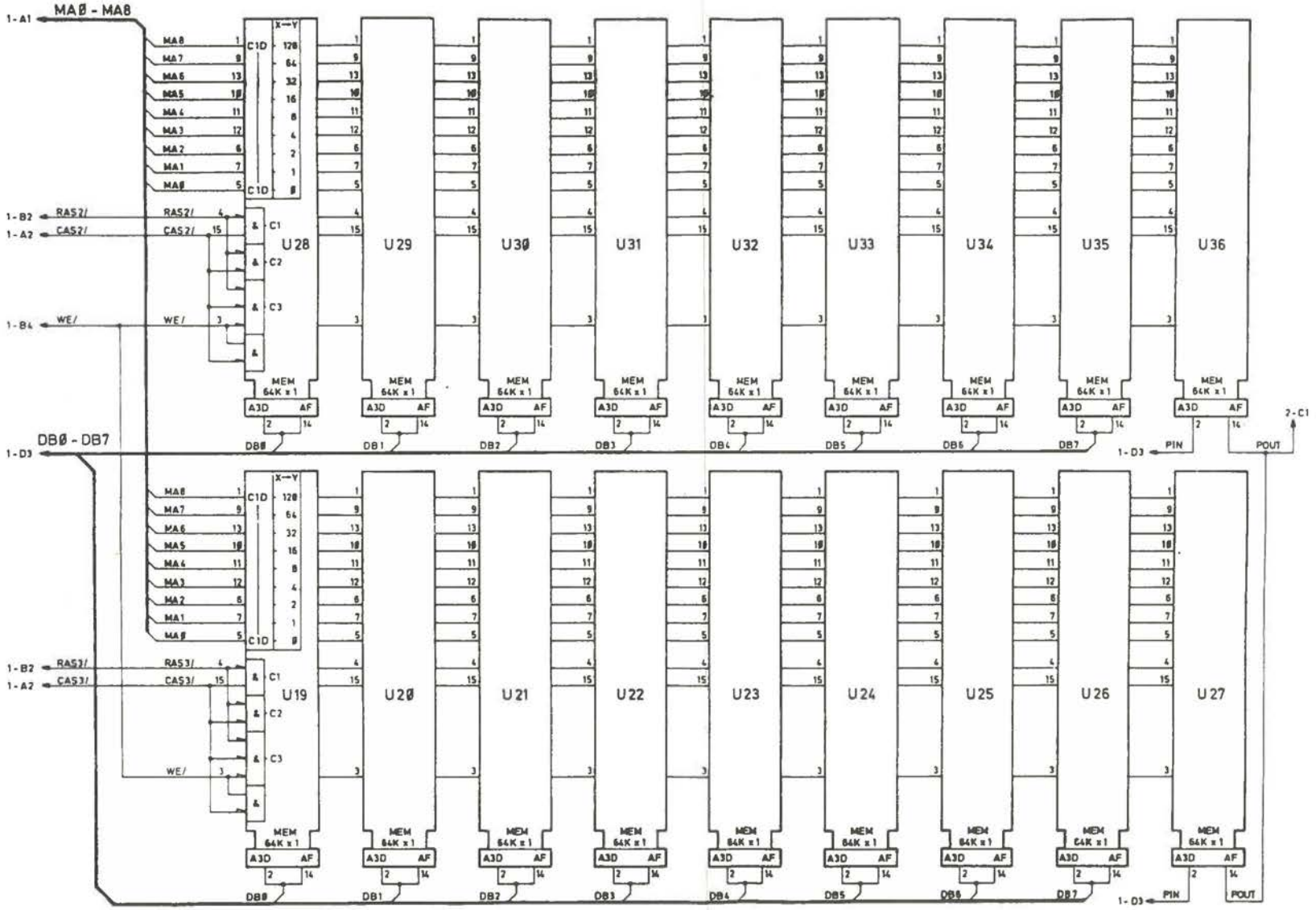




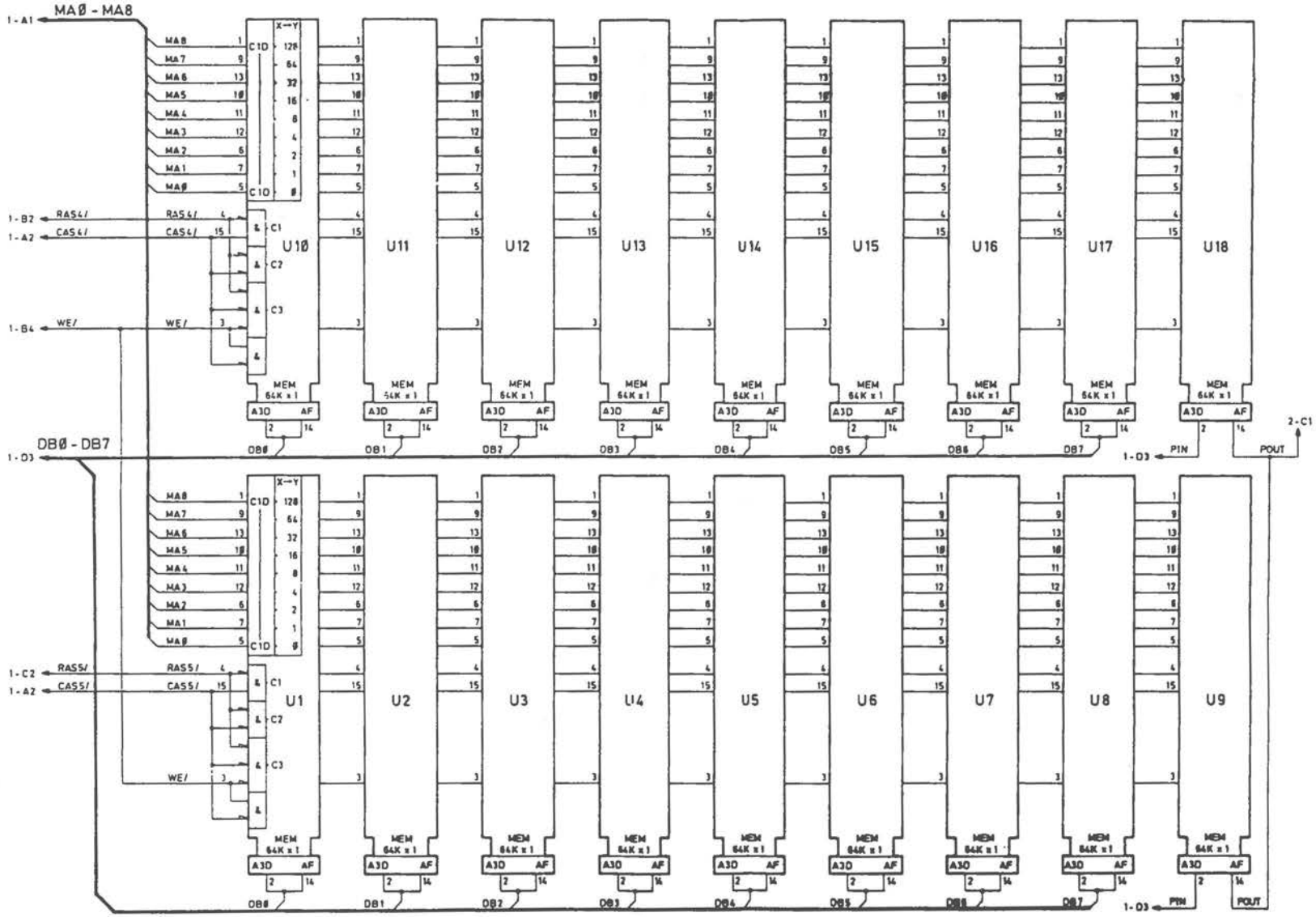
Memory Expansion Board <1 of 4]



Memory Expansion Board (2 of 4)

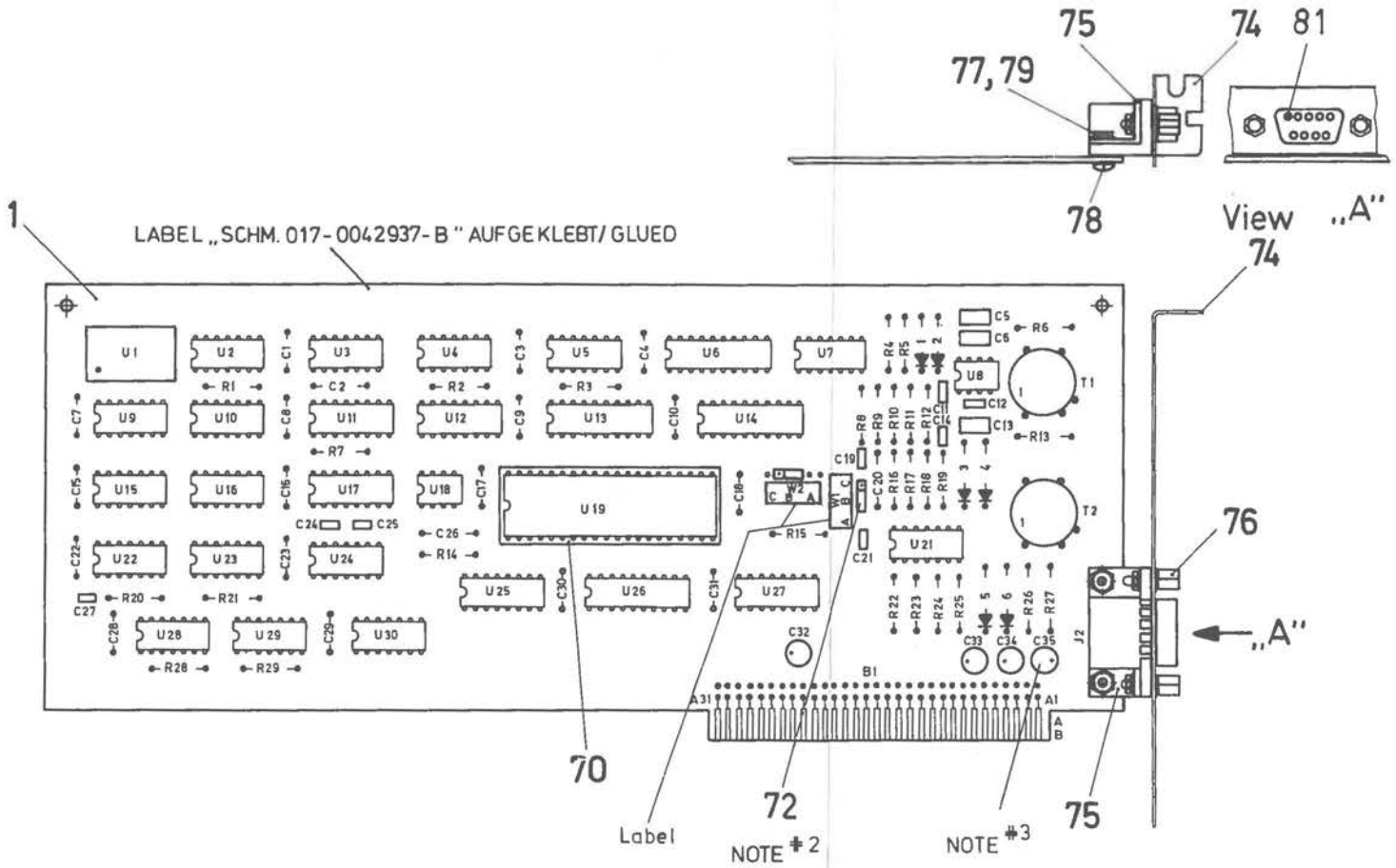


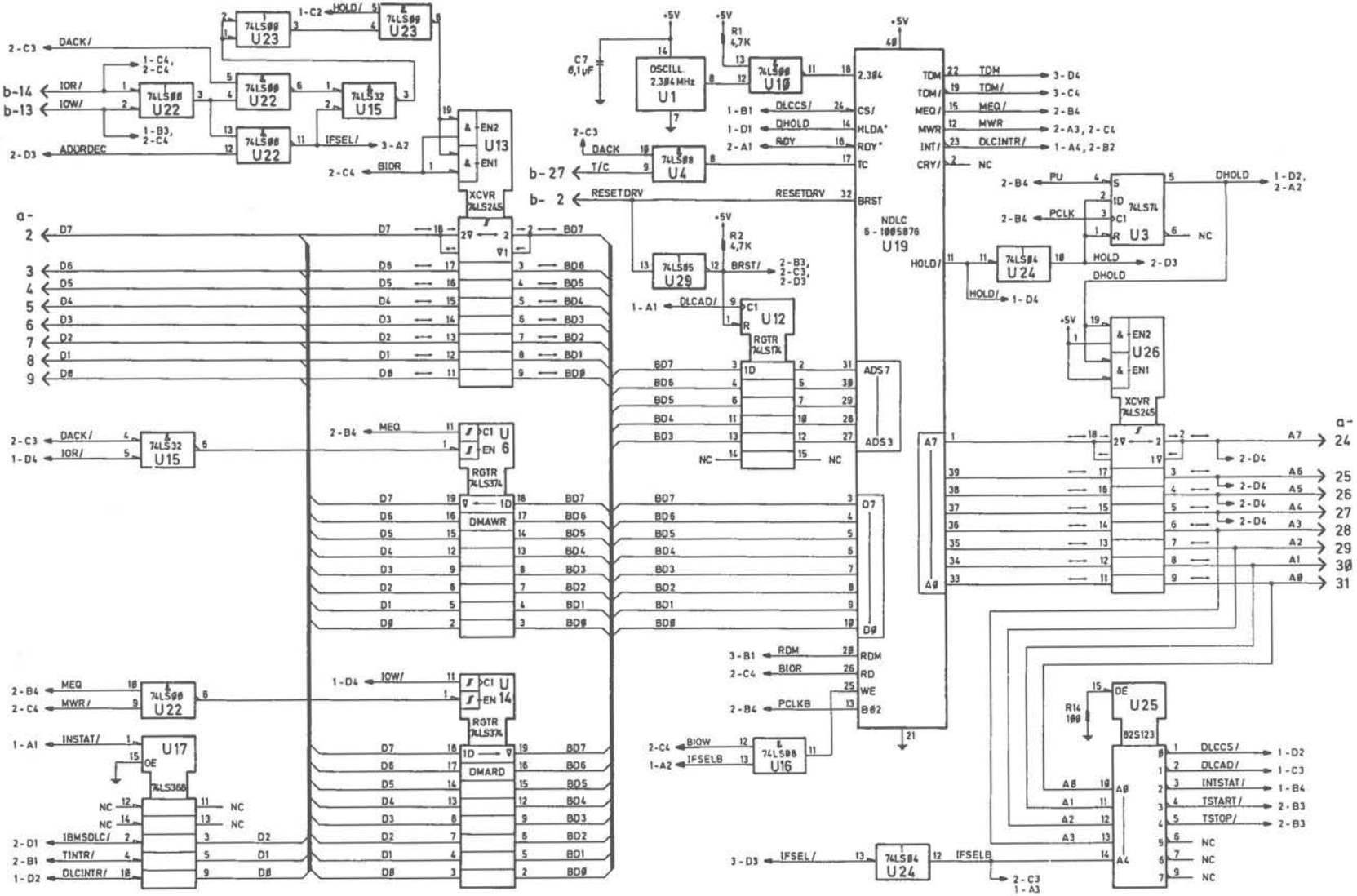
Memory Expansion Board (3 of 4)



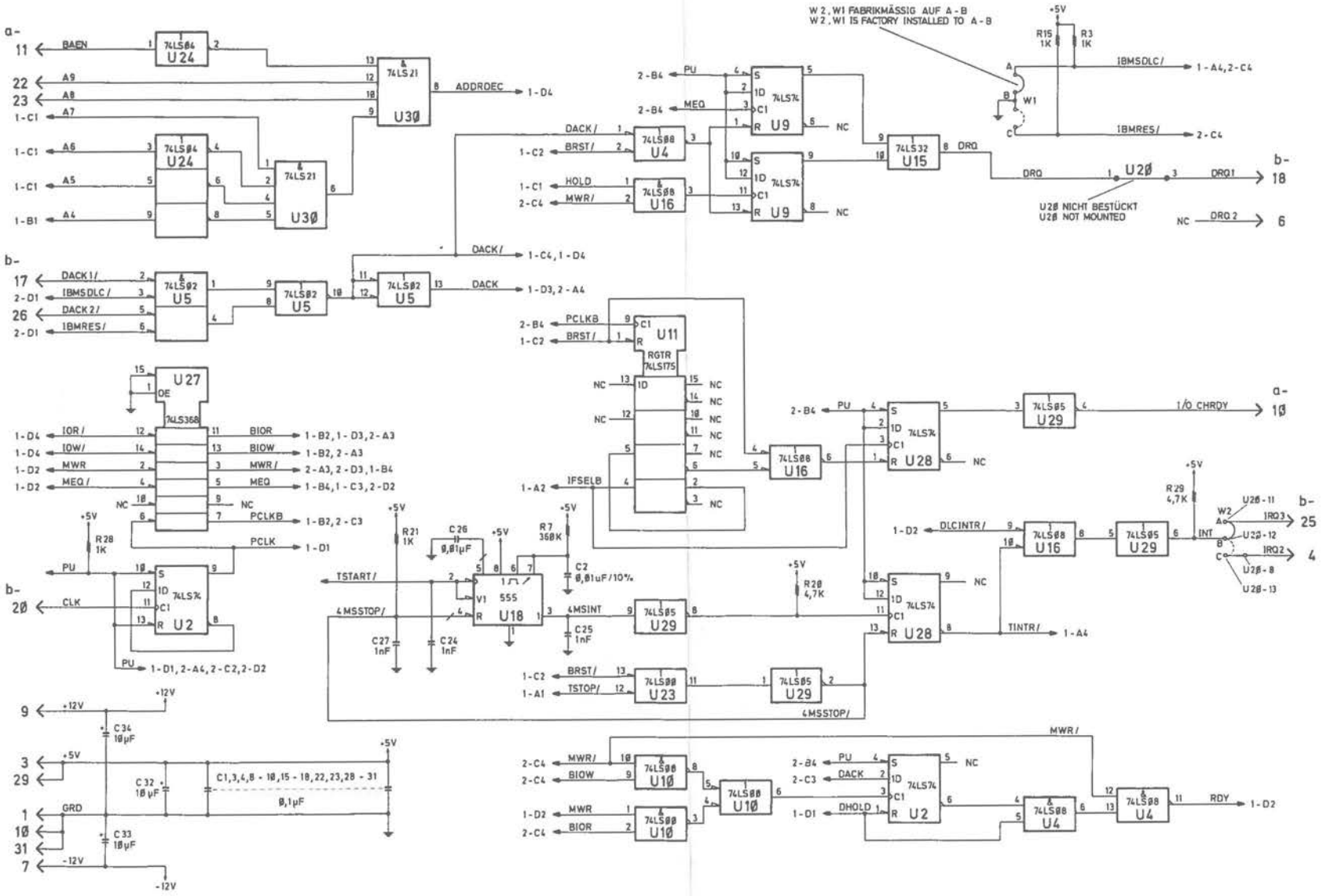
Memory Expansion Board (4 of 4)

DLC Inhouse I/F Board

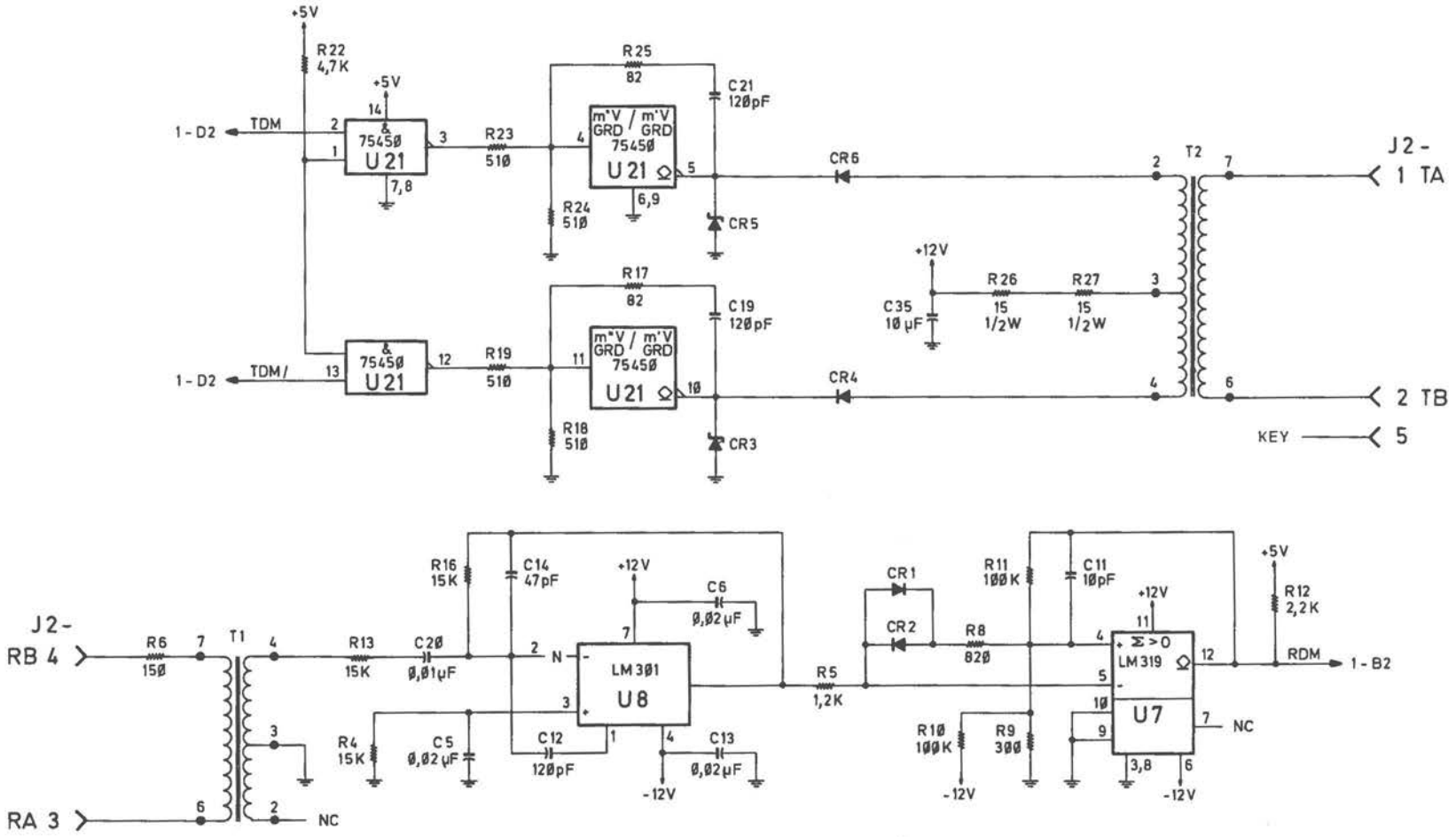




DLC Inhouse I/F Board (1 of 3)

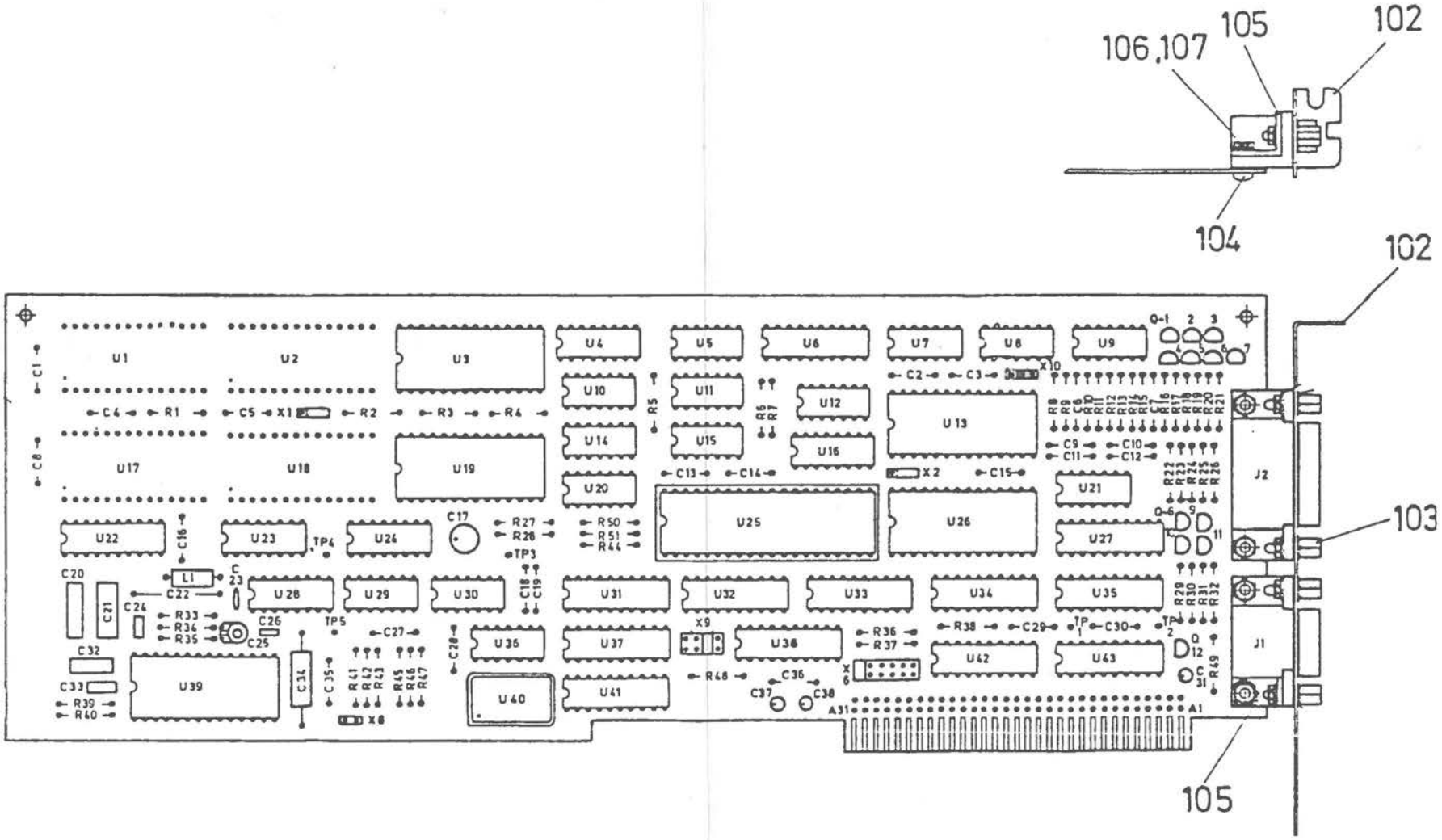


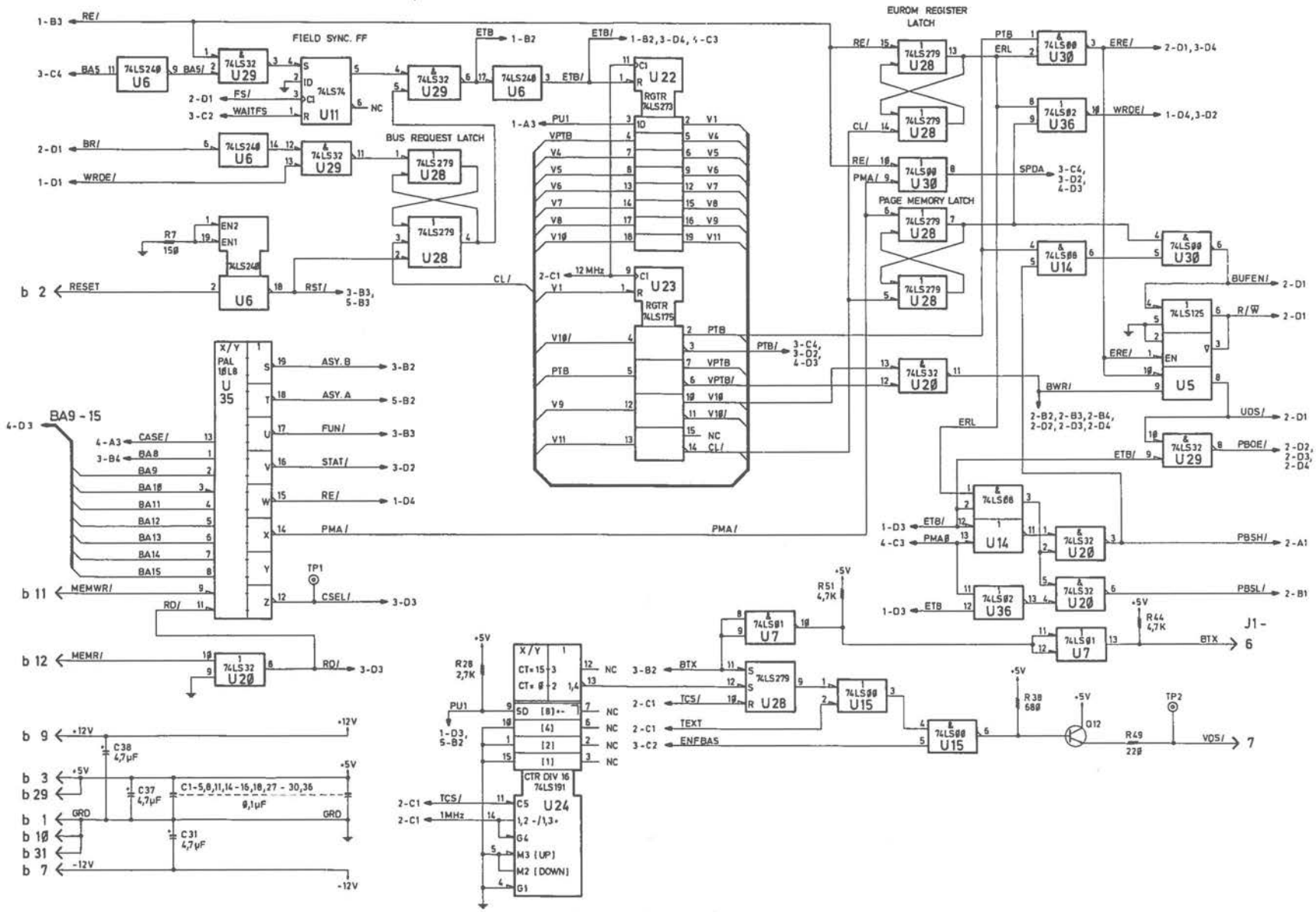
DLC Inhouse I/F Board (2 of 3)



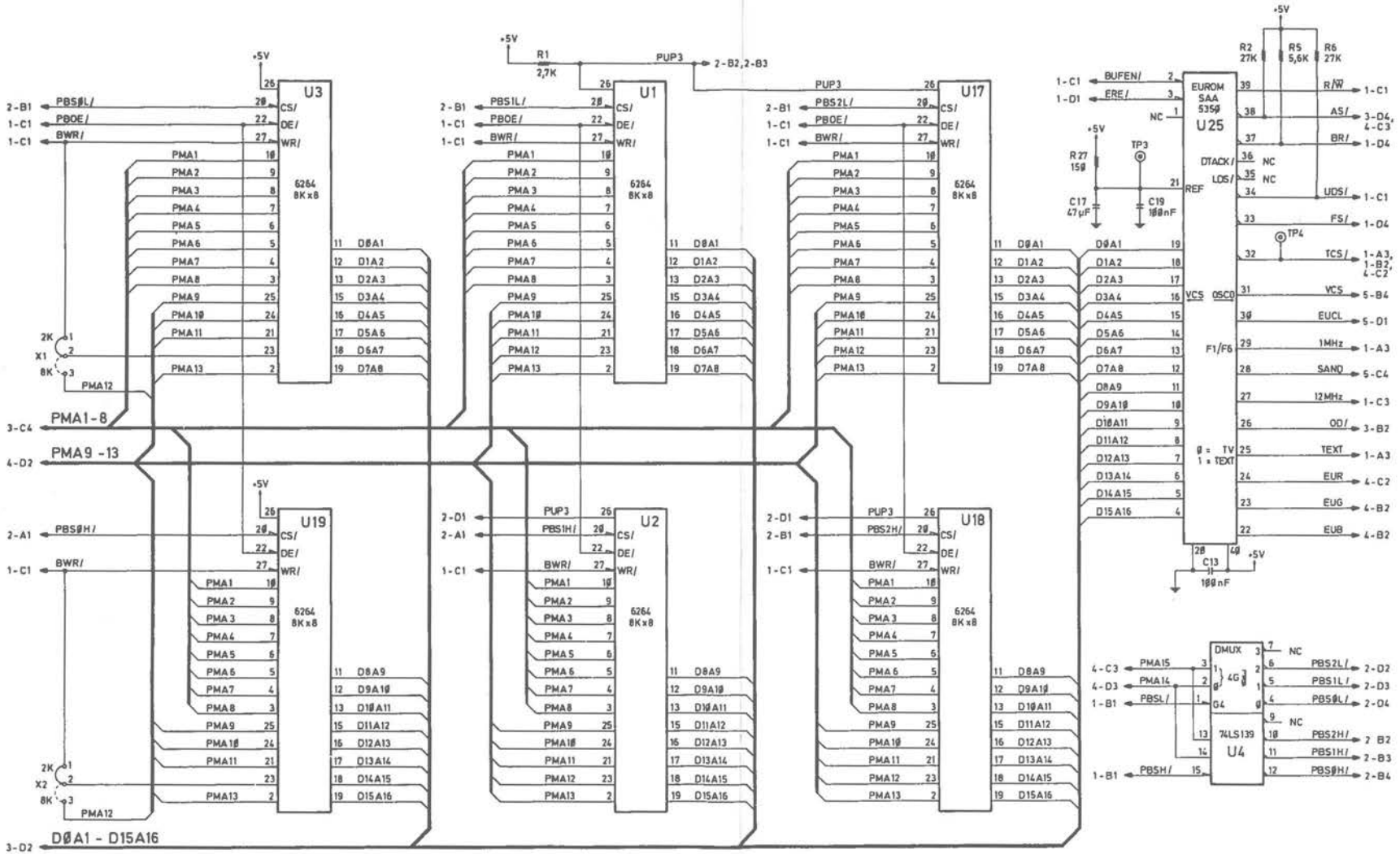
DLC Inhouse I/F Boerd [3 of 3]

Videotex Adapter Board

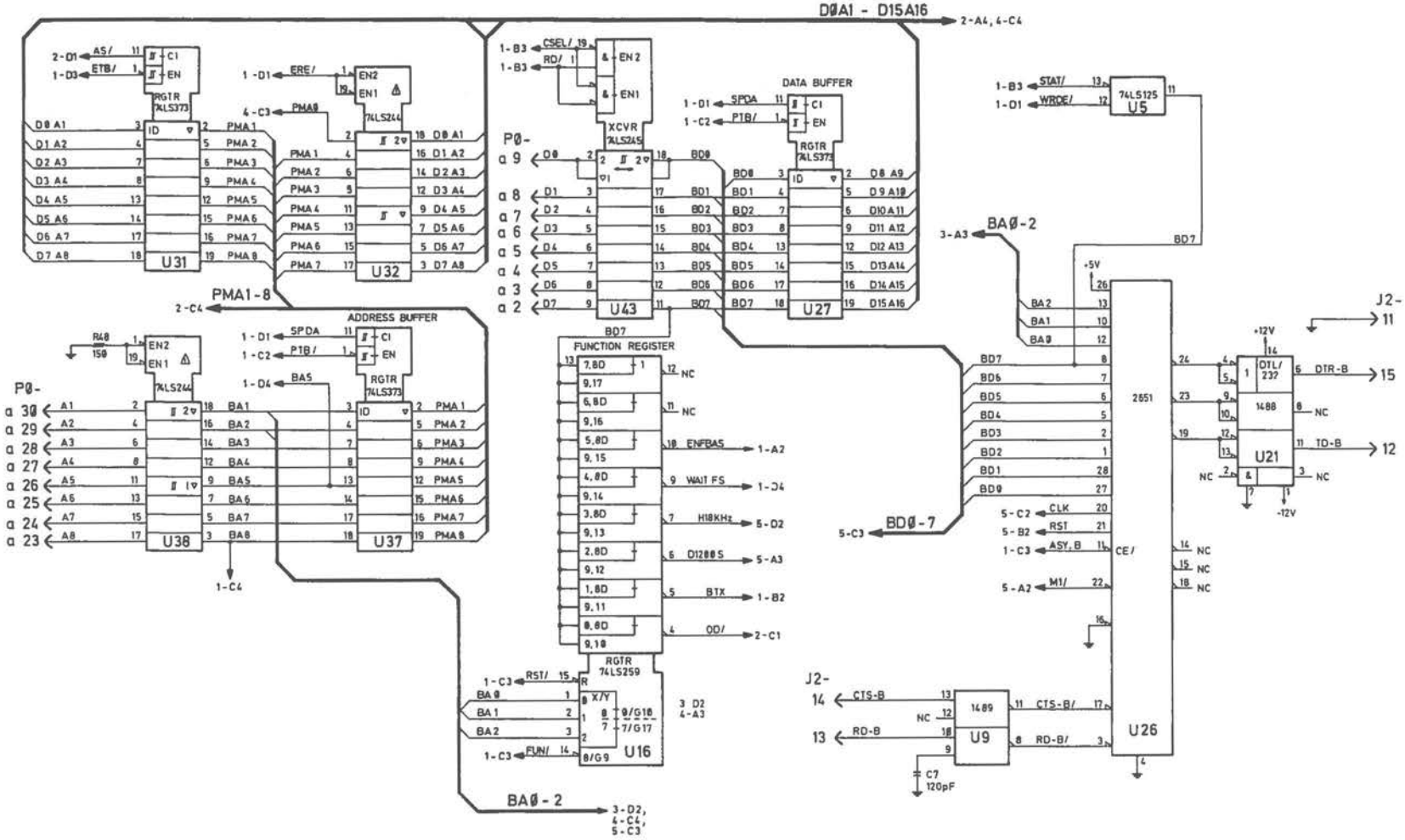




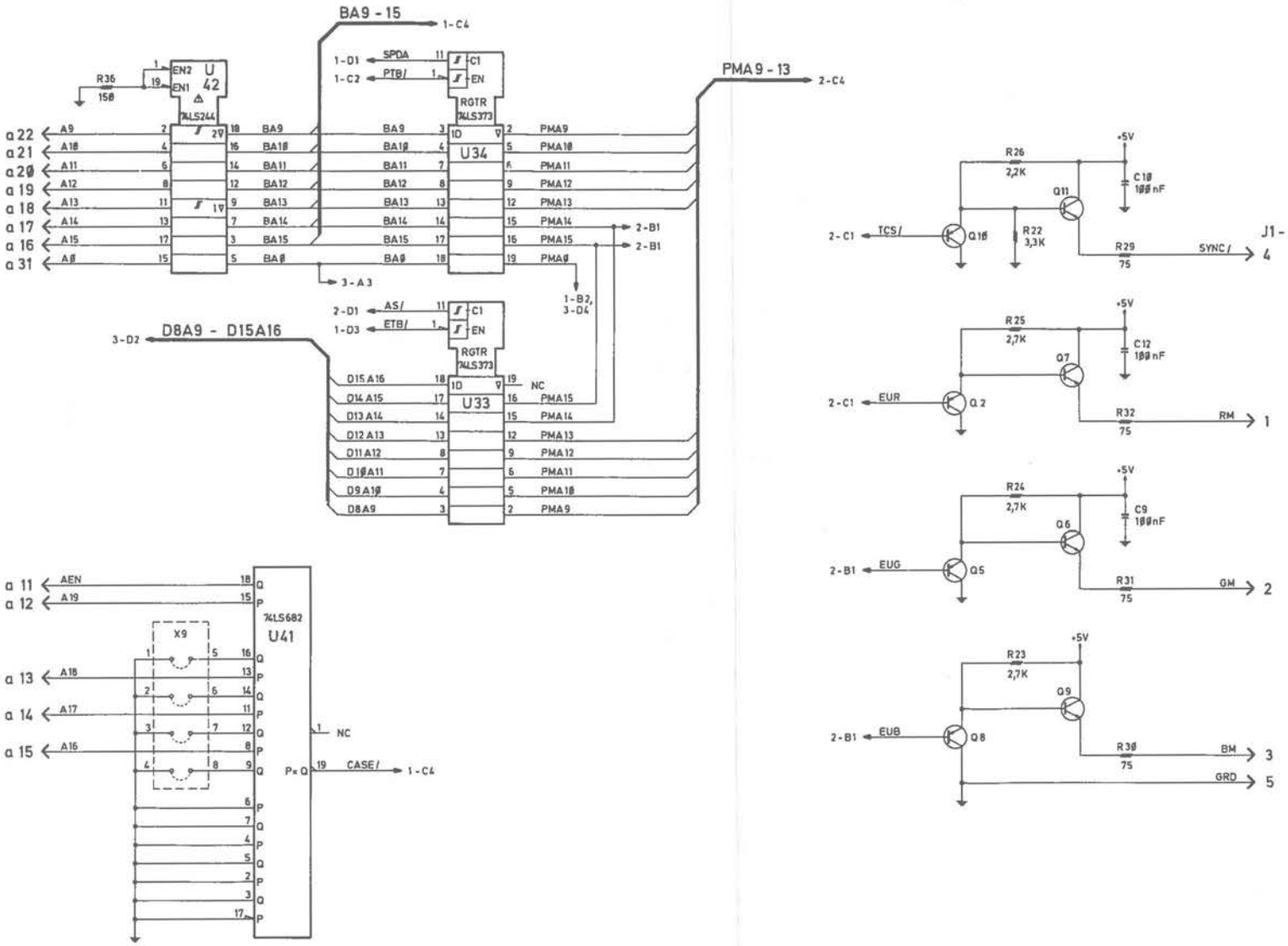
Videotex Adapter Board (1 of 5)



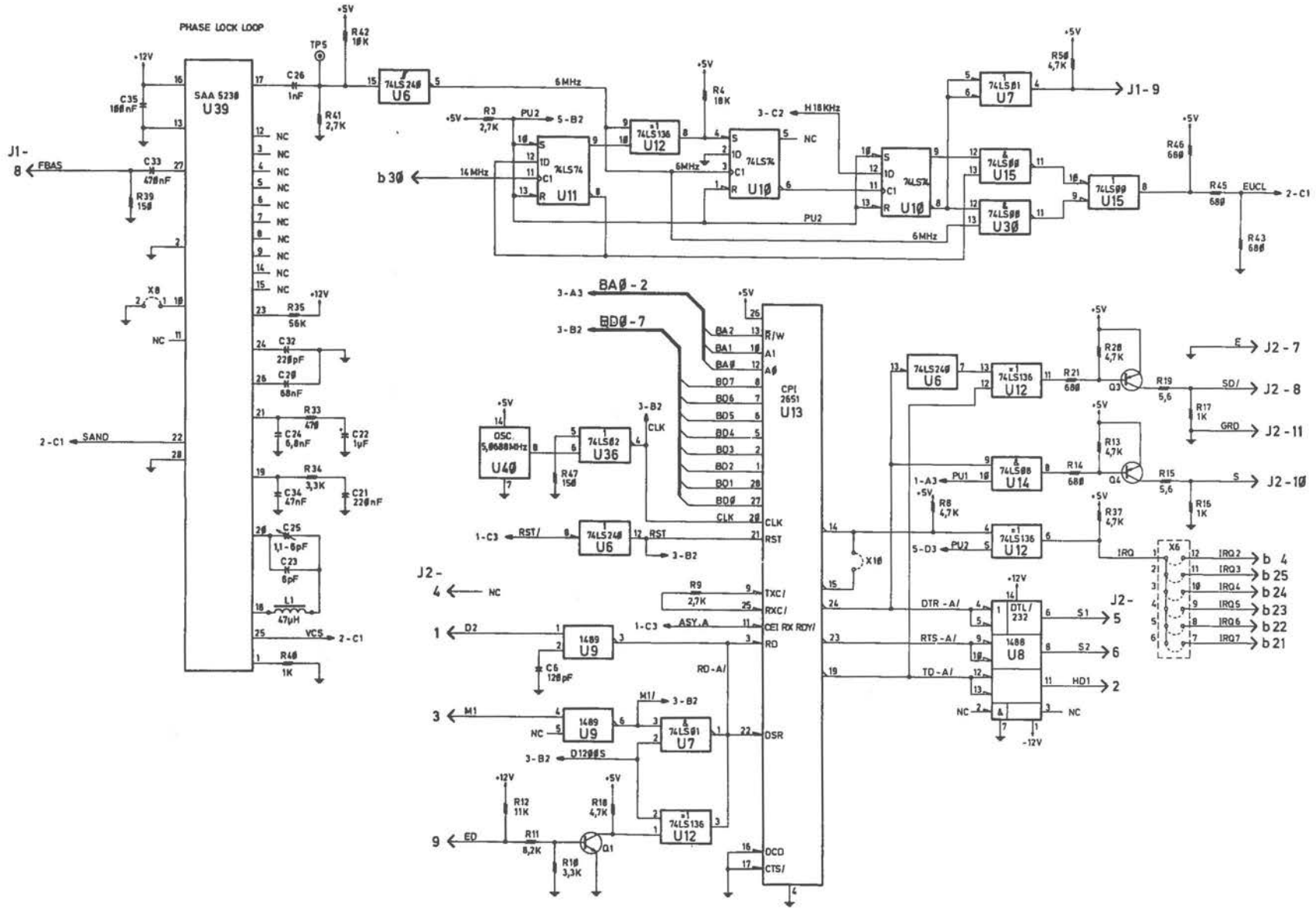
Videotex Adapter Board (2 of 5)



Videotex Adapter Board (3 of 5)

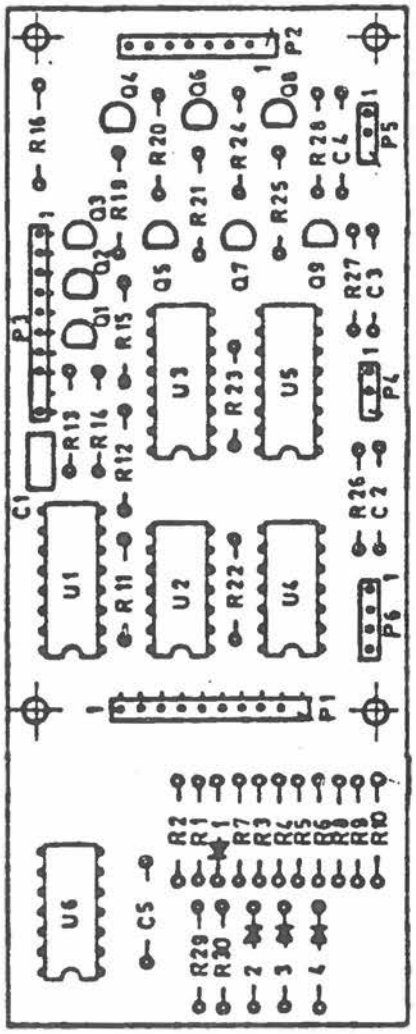


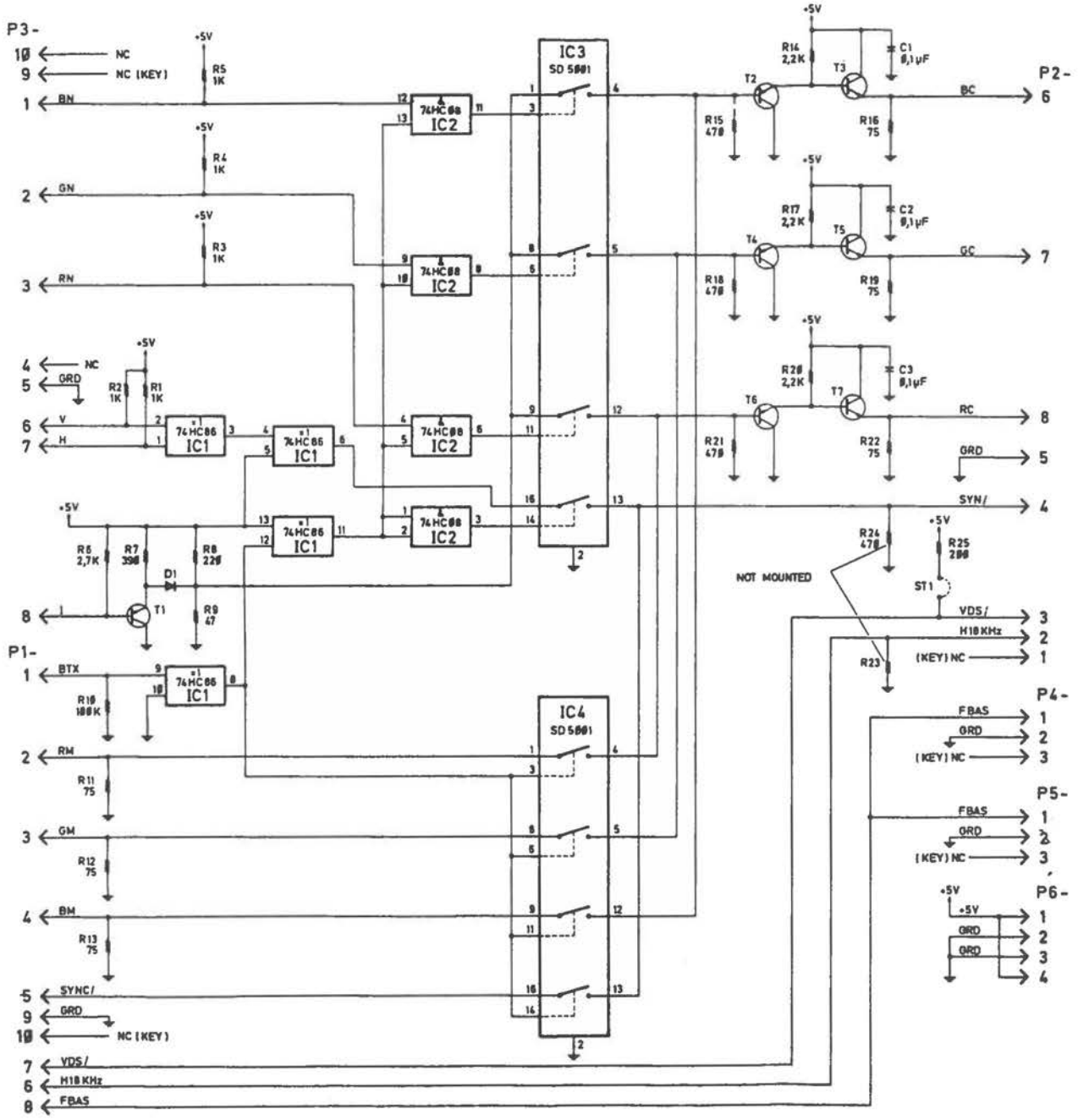
Videotex Adapter Board (4 of 5)



Videotex Adapter Board (5 of 5

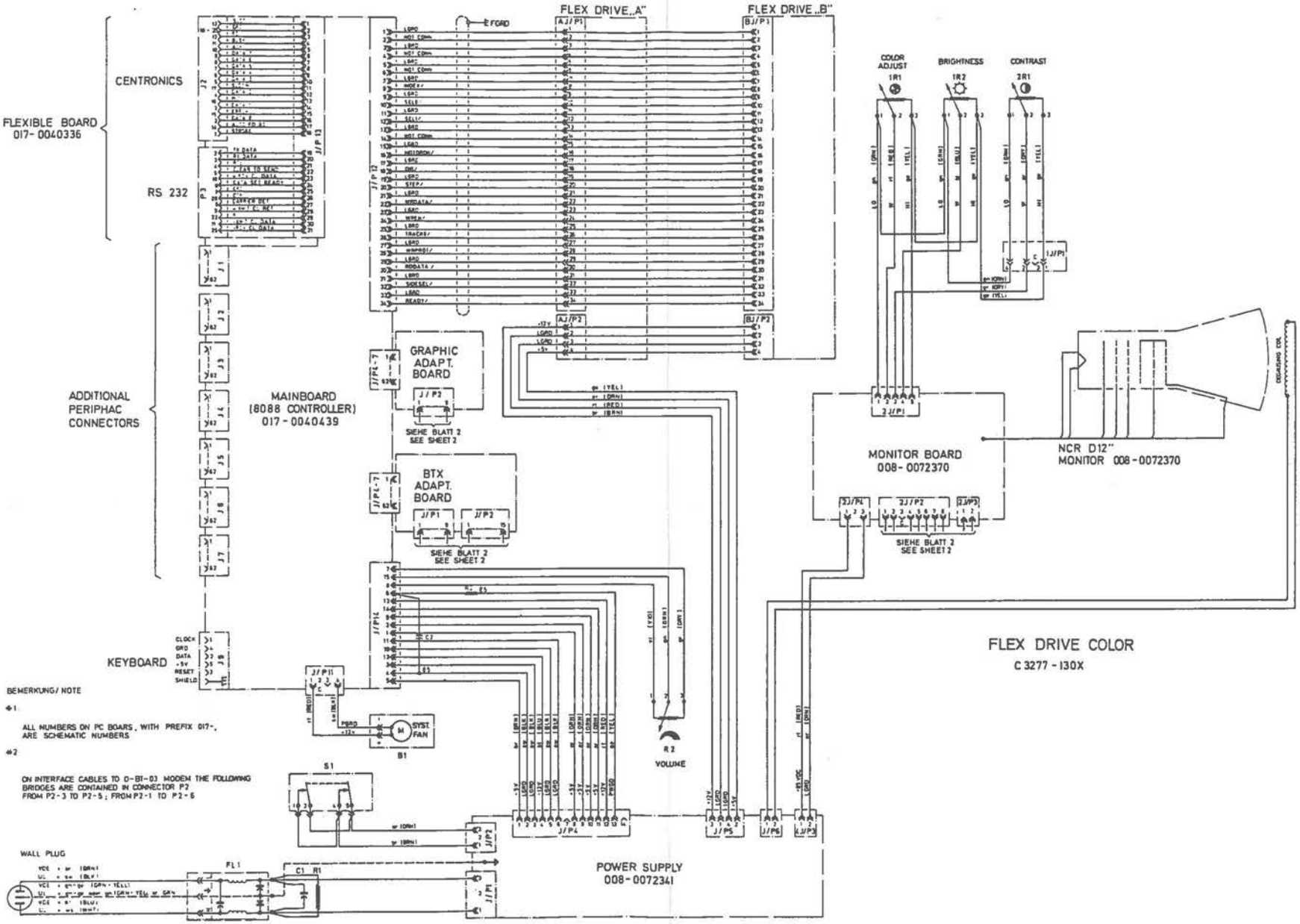
Videotex Switchboard



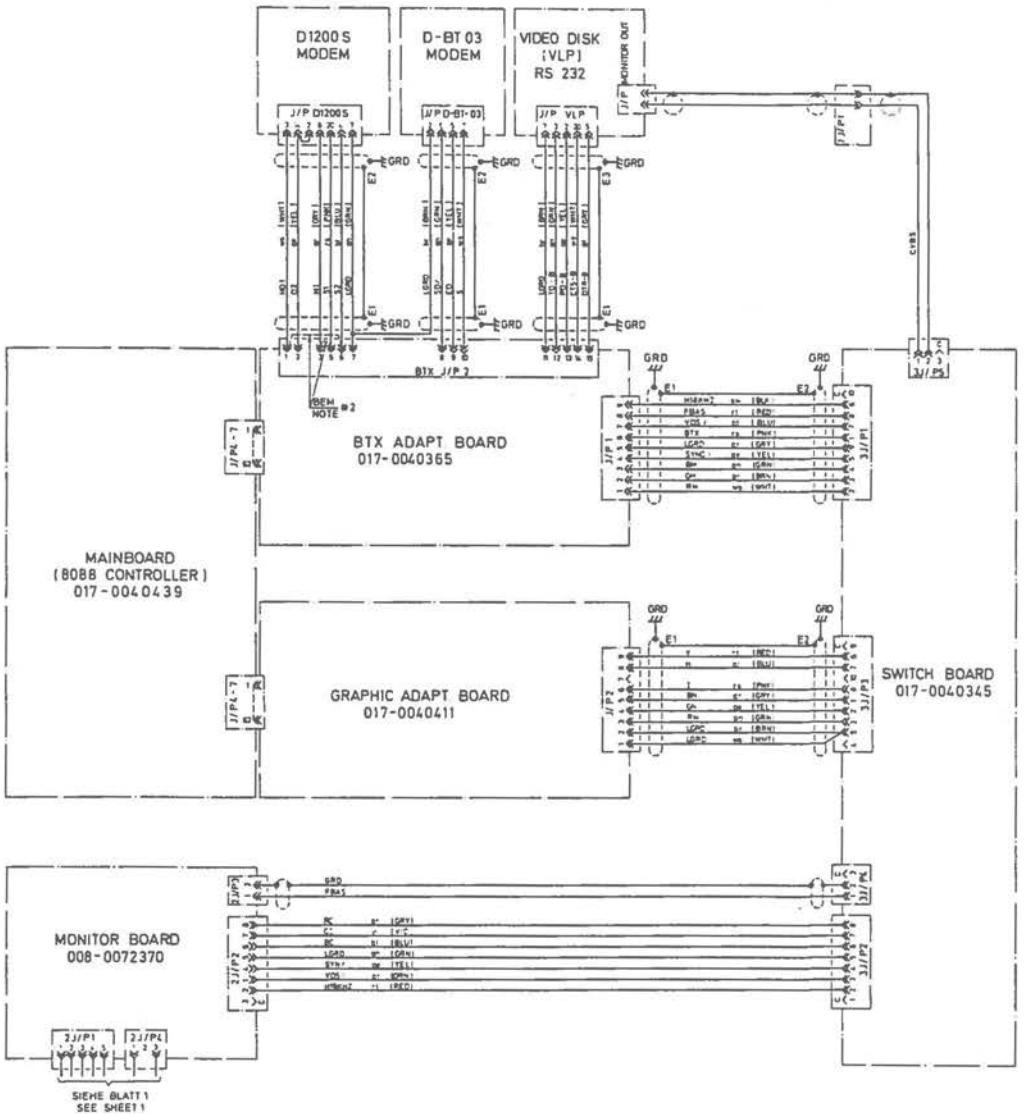


Videotex Switchboard

Wiring Diagram Videotex

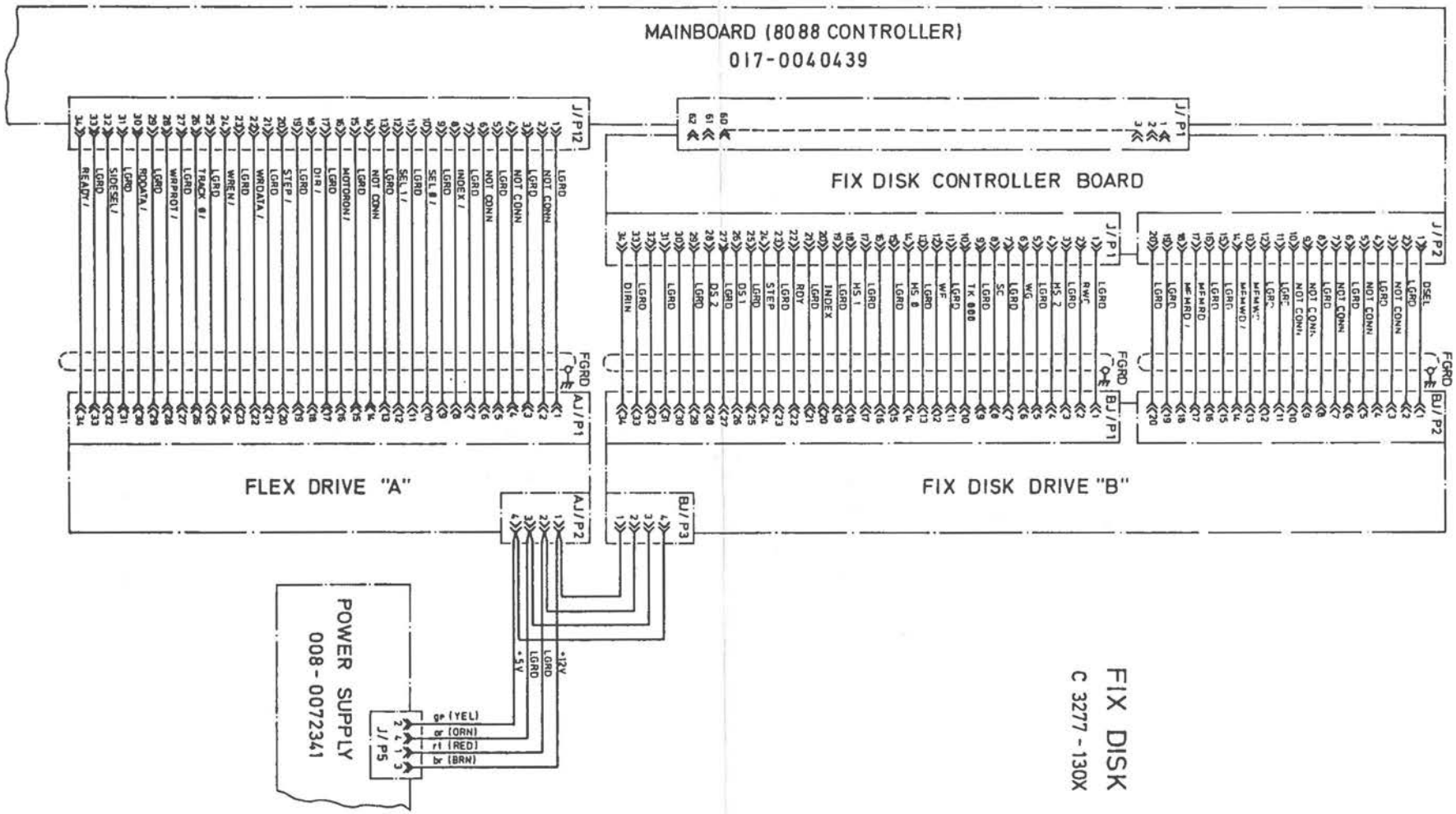


Wiring Diagram Videotex (1 of 3)



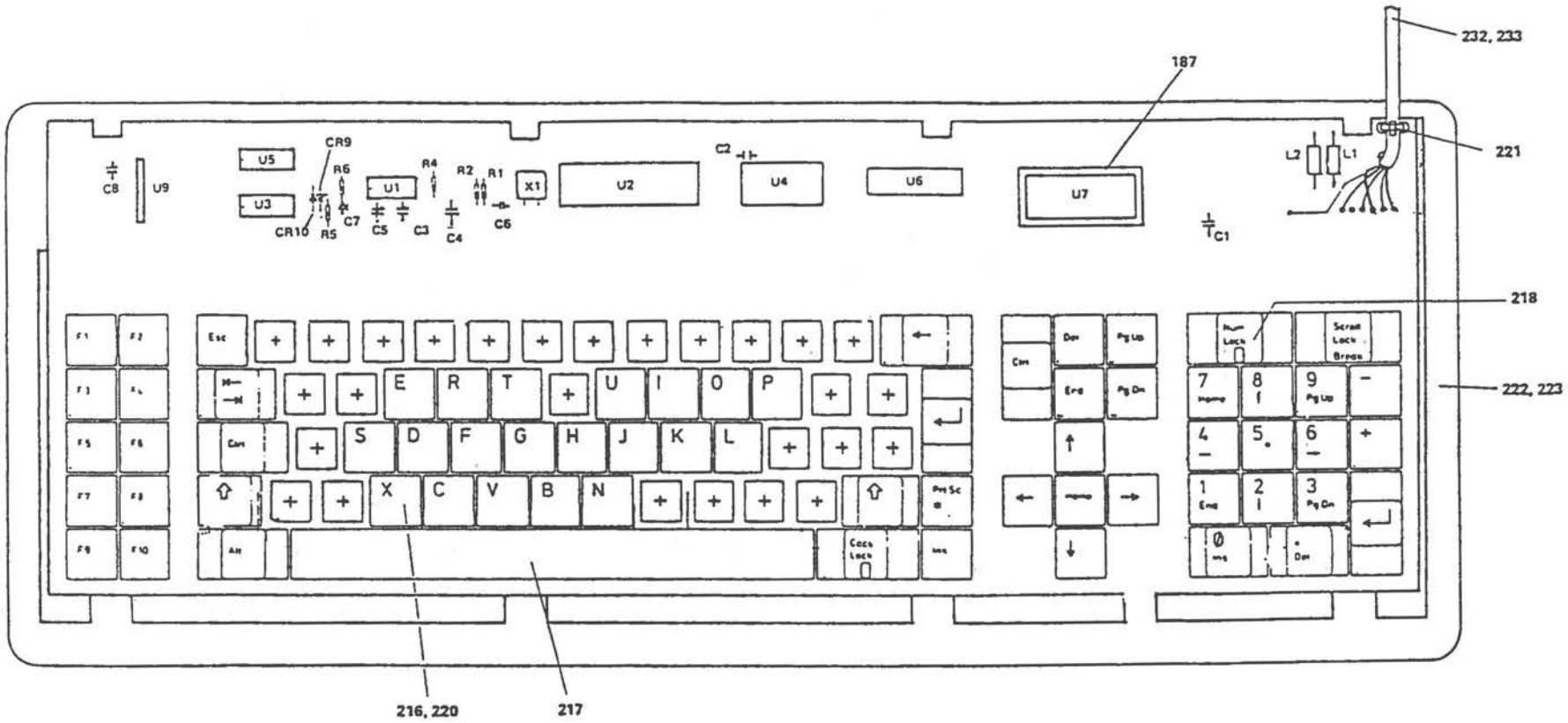
MODEM D1200 S; MODEM D-BT 03; VIDEO, DISK (VLP);
C-3277-130X

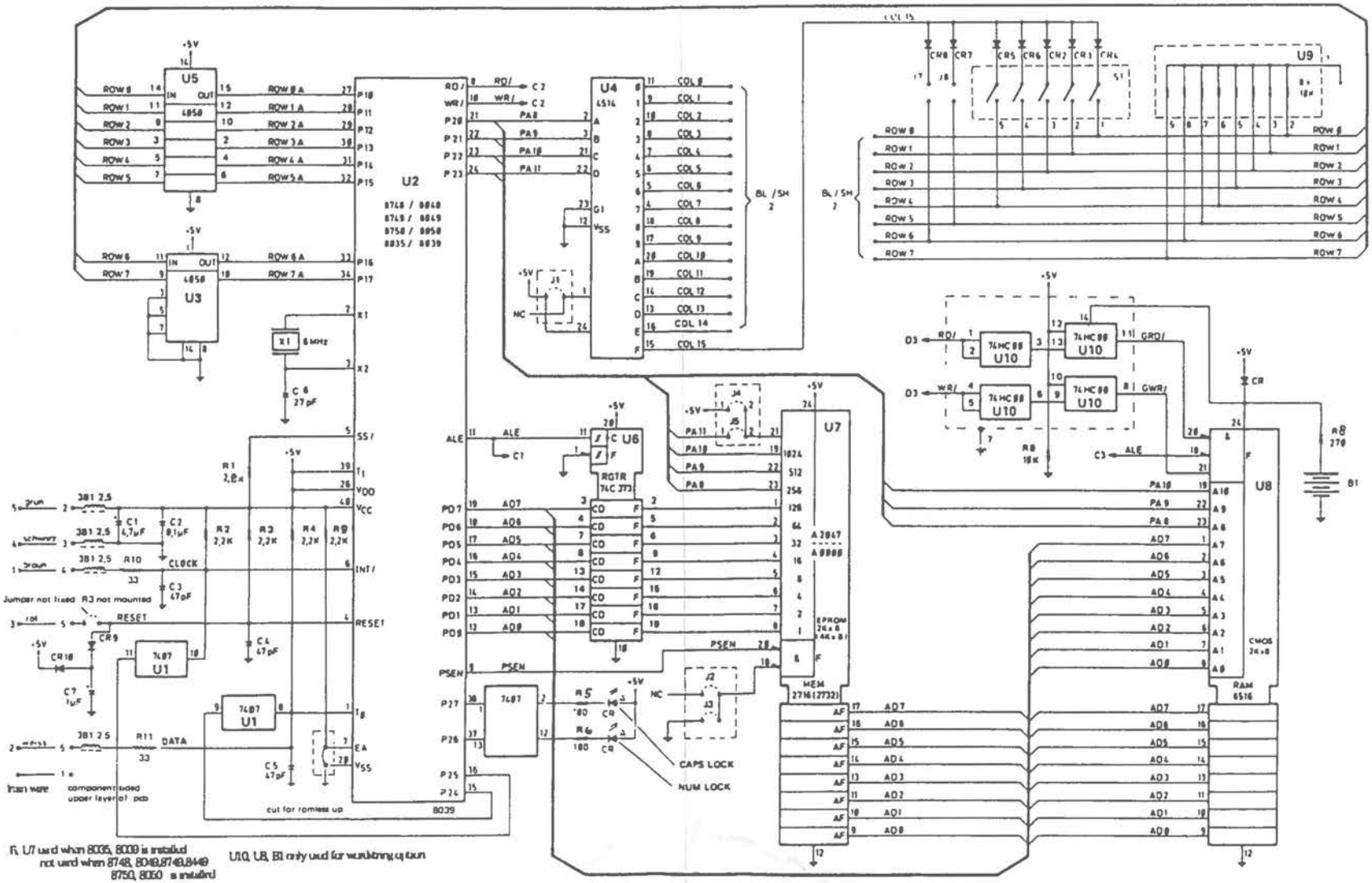
Wiring Diagram Videotex (2 of 3)



Wiring Diagram Videotex [3 of 3]

Keyboard


232, 233
187
221
218
222, 223
216, 220
217
U1
U2
U3
U4
U5
U6
U7
U9
X1
C1
C2
C3
C4
C5
C6
C7
C8
CR9
CR10
R1
R2
R4
R5
R6
L1
L2



R, U7 used when 8035, 8039 is installed
not used when 8748, 8049, 8749, 8449
8750, 8050 is installed

U10, U8, B1 only used for wordstory upgrade

Keyboard Schematic