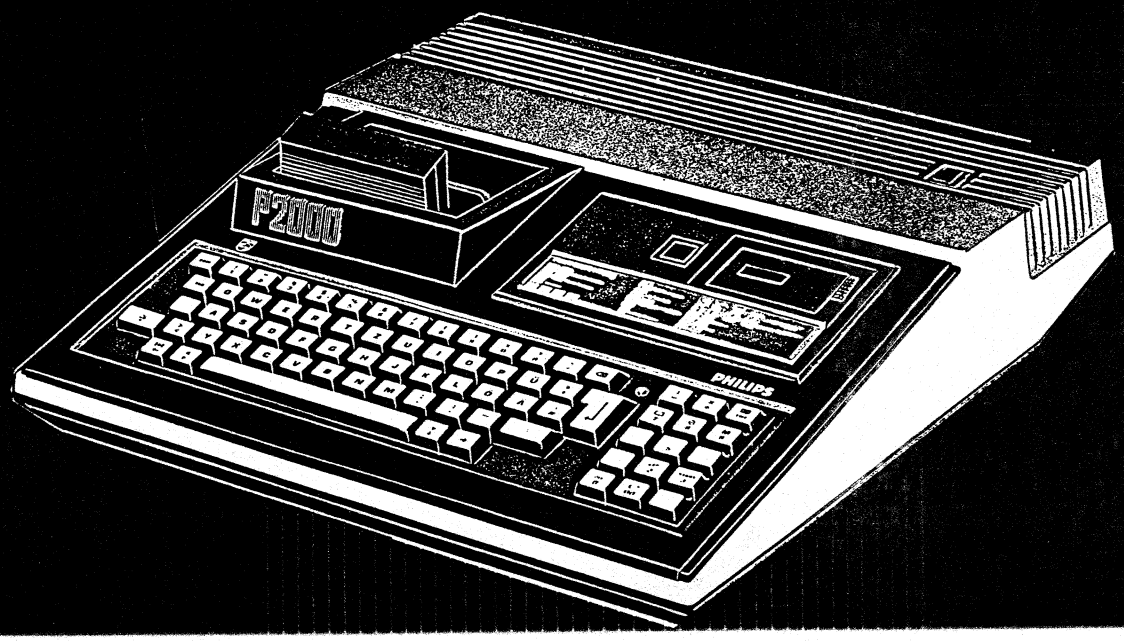
PHILIPS

P2000
SYSTEM T&M
Reference Manual



Data Systems

PHILIPS

P2000

SYSTEM T & M
REFERENCE MANUAL

5103 991 30421

Preface

This manual describes the internal structure and function of the P2000 M and T models. It aims to provide the reader with a description of the units of which the P2000 is composed, the circuits which make up these units, the connections between them, and the basic software, present at all times, (the 'monitor') which controls these units. The manual is intended to be read by all P2000 users, but it does assume a limited knowledge of the principles of computer operation; however, users do not need to be familiar with any particular P2000 application package.

The reasons for which users read this manual will vary from those who are simply interested to know how their P2000 works, to those who intend to implement some special function. (It is hoped that at least some of the first group will, on reading the manual, become members of the second!) The information in the manual is as detailed as seems reasonable, bearing in mind its wide audience. The P2000 is divided into a number of basic units - a description of each unit is given for the less technical reader, backed up by extensive appendices for those who require more detailed information - such as block diagrams showing the relationship between units, the connections and the signals between and within these units. Timing diagrams, however, have been omitted.

The first chapter of the manual gives a brief overview of the P2000 as a whole, and introduces the units of which it is made up. The subsequent chapters each consider one unit in turn, in more detail.

PAGE LAYOUT

The pages in all P2000 manuals are arranged as follows:

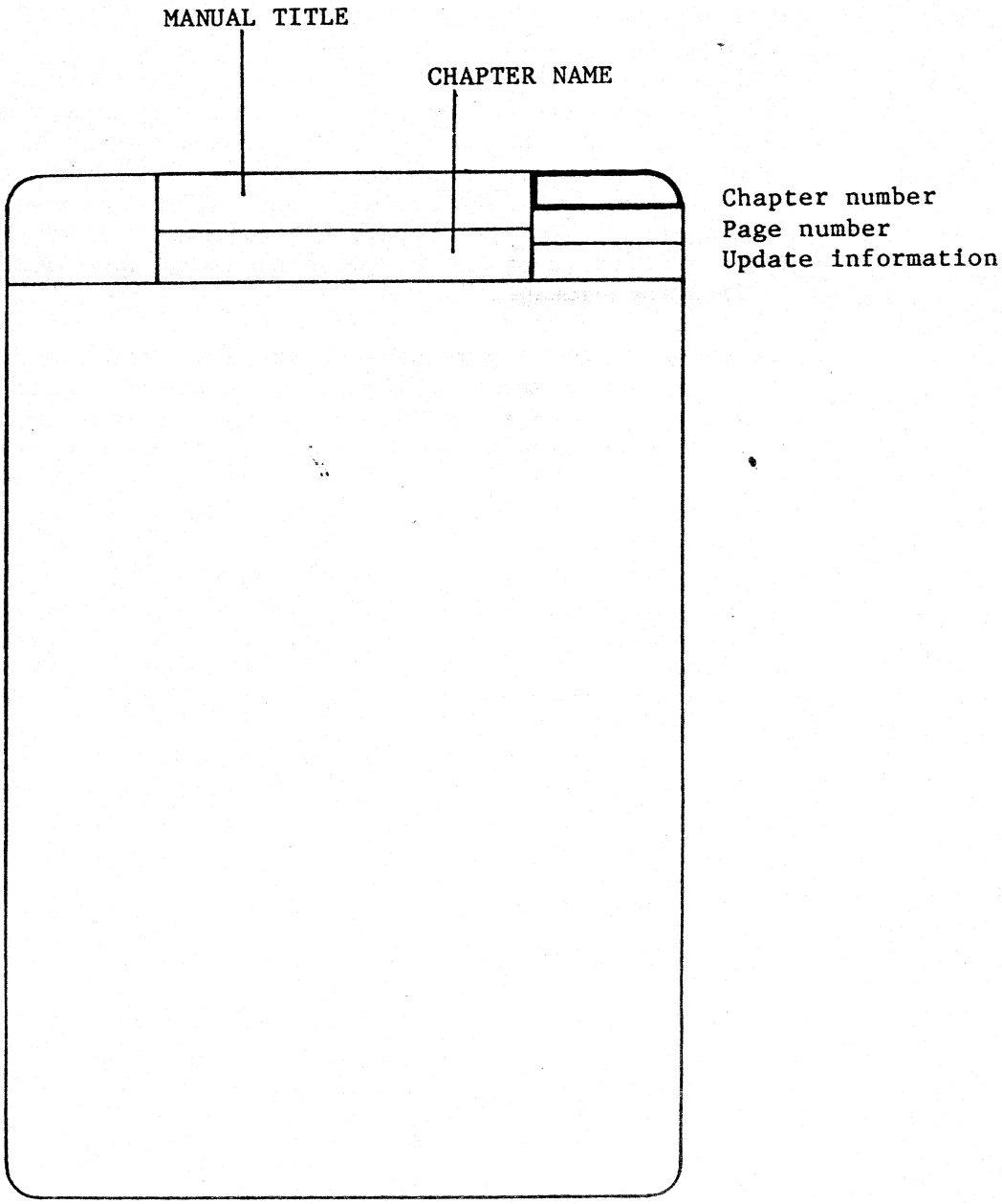


TABLE OF CONTENTS

		page
	Preface	i
	Page Layout	ii
	Table of Contents	iii
1	STRUCTURE OF THE P2000	1-1
1.1	Overview	1-1
	Block diagram	1-2A
1.2	Central Processing Unit	1-3
1.3	Flexible Disks	1-5
1.4	Keyboard	1-6
1.5	I/O Slot	1-7
1.6	Memory	1-8
1.7	Mini-digital Cassette	1-10
1.8	Printer	1-11
1.9	Video	1-12
2	FLEXIBLE DISKS	2-1
2.1	Disk Drive Units	2-1
2.2	Disk Format	2-3
2.3	Disk Control Software	2-4
3	KEYBOARD	3-1
3.1	Keyboard Unit	3-1
3.2	Monitor Keyboard Control	3-2
	3.2.1 Memory Areas	3-2
	3.2.2 Keyboard Initialise Routine	3-3
	3.2.3 Reading Keyboard	3-4
3.3	Application Keyboard Access	3-6
	3.3.1 Keyboard Status Routine	3-6
	3.3.2 Keyboard Input Routine	3-8
	3.3.3 Keyboard Clear Routine	3-9
	3.3.4 Keyboard Disable Routine	3-9
	3.3.5 Keyboard Enable Routine	3-9

4	MEMORY AND MONITOR	4-1
4.1	Memory Introduction	4-1
4.2	Memory Map	4-5
4.3	Monitor	4-7
	4.3.1 System (Re-)start Routine	4-8
	4.3.2 Bell Routine	4-10
4.4	ROM Key	4-11
	4.4.1 ROM Key Structure	4-11
	4.4.2 ROM Key Identification Block	4-13
4.5	System and Extension RAM	4-15
	4.5.1 Monitor RAM Area	4-15
	4.5.2 Extension RAM	4-19
5	MINI-DIGITAL CASSETTE	5-1
5.1	Cassette Unit	5-1
5.2	Tape Format	5-3
	5.2.1 Blocks, Gaps, Marks and Records	5-3
	5.2.2 Header	5-7
	5.2.3 Data	5-12
5.3	Cassette Control	5-13
	5.3.1 Memory Areas	5-13
	5.3.2 Initialise	5-15
	5.3.3 Rewind	5-15
	5.3.4 Move Forward	5-15
	5.3.5 Move Backward	5-16
	5.3.6 Set End-of-File Record	5-16
	5.3.7 Write	5-17
	5.3.8 Read	5-17
	5.3.9 Status	5-18
5.4	Cassette Routine Error Codes	5-19

6	PRINTER	6-1
6.1	Printer Control	6-1
	6.1.1 Parameters	6-2
	6.1.2 Operation	6-3
6.2	Printer-select Jumper	6-4
6.3	Character-code Tables	6-5
	6.3.1 First Section (Non-backspace Printers)	6-6
	6.3.2 Second Section (Backspace Printers)	6-7
6.4	Baud Rate	6-8
6.5	'Transparent' Printing	6-9
7	VIDEO	7-1
7M	VIDEO ON THE M MODEL	7-2
7.1M	Monitor	7-2
7.2M	Video Memory	7-3
	7.2.1M Video Page 1	7-4
	7.2.2M Video Page 2	7-5
	7.2.3M Text and Graphics Characters	7-6
7.3M	Application Video Software	7-7
7.4M	Monitor Video Software	7-9
7.5M	Video Control and Function	7-10
7T	VIDEO ON THE T MODEL	7-12
7.1T	Television and RGB Monitor	7-12
7.2T	Video Memory	7-13
	7.2.1T Video Page 1	7-15
	7.2.2T Video Page 2	7-15
	7.2.3T Text and Graphics Characters	7-16
7.3T	Application Video Software	7-17
7.4T	Monitor Video Software	7-19
7.5T	Video Control and Function	7-20

APPENDIX A	Machine Architecture	App-A-1
APPENDIX B	Ports	App-B-1
APPENDIX C	I/O Slot Pinning	App-C-1
APPENDIX D	Character (Viewdata) Code	App-D-1
APPENDIX E	Key-codes	App-E-1
Index		
Manual Status Control Form		
Manual Comment Form		

1 THE STRUCTURE OF THE P2000

This chapter gives a broad outline of the units which make up the P2000 computer, and the way in which they relate to each other. Subsequent chapters of the manual consider each unit in detail.

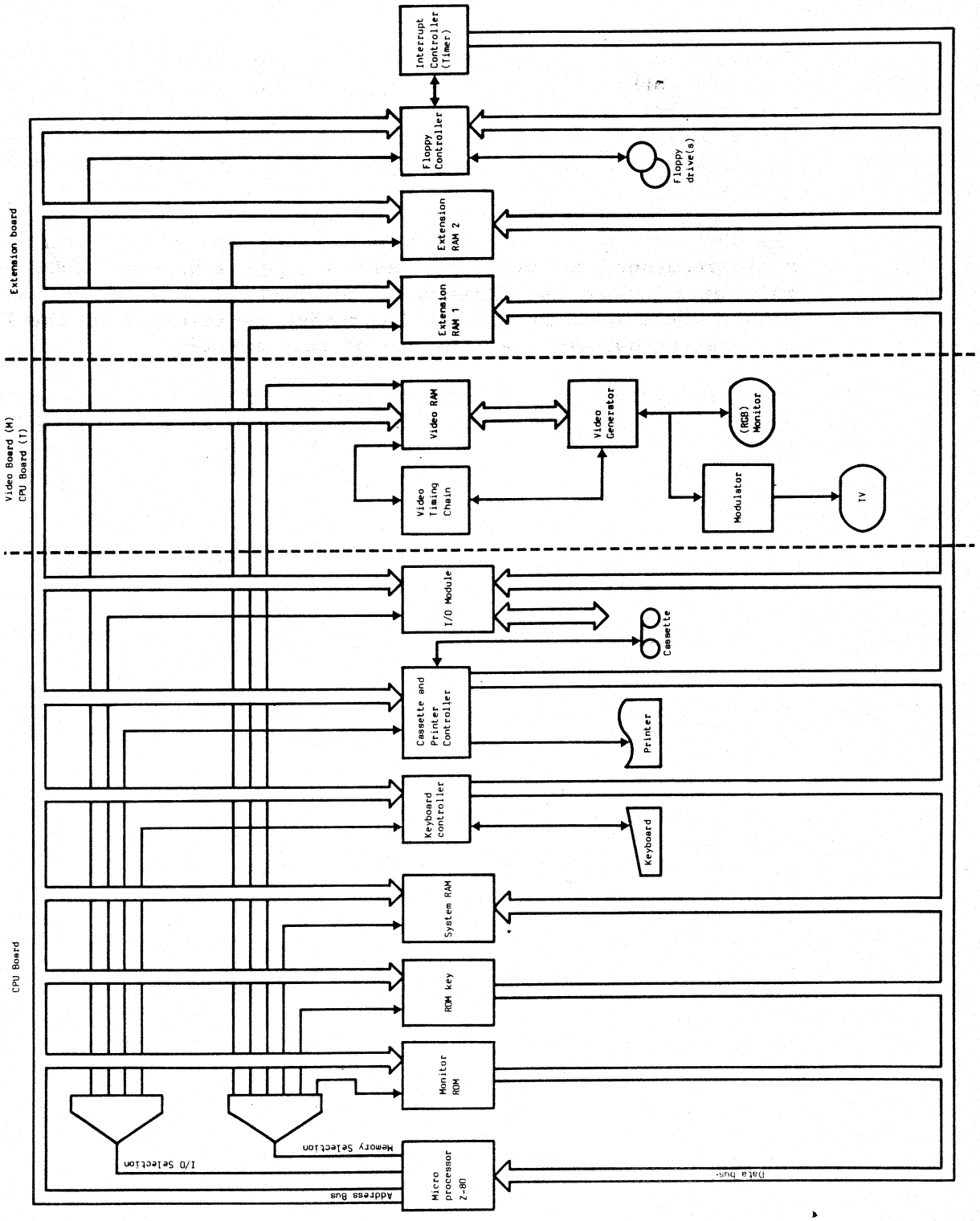
1.1 OVERVIEW

The P2000 is composed of several separate units linked together as shown in Figure 1. There are three basic types of unit:

- the microprocessor - performs the arithmetic and logical operations of the computer, executing a program, and controlling the operations of the other units.
- memory - holds the instructions which make up the program, and the data on which the microprocessor performs its operations.
- input/output devices - used for communication between the microprocessor and the outside world - for the input of instructions and data, and the output of the results of the microprocessor's operations.

The 'chips' and circuits which make up the units of the P2000 are housed on circuit boards inside the main unit (the unit with the keyboard). The number of circuit boards and the circuits on them differ between the different versions of the P2000. There are two types of P2000, known as the M model and the T model. The difference between the two is seen most clearly in the video - the M model has a built-in P2000 monitor, whereas the T model is connected to a television set or an RGB monitor - but there are other differences between the two, including the circuitry.

All P2000 models contain the CPU board, which holds the basic units and, on the T model, control circuits for the video. On the M model, a separate board - the video board - holds the video control circuits. The third board, the extension board, is present in some T and M models - this contains extra memory, and controls for the flexible disk units. This division of units between the three boards can be seen in Fig. 1.


Interrupt Controller (Timer)
Floppy Controller
Floppy drive(s)
Extension RAM 2
Extension RAM 1
Extension board
Video RAM
Video Generator
(RGB) Monitor
Video Timing Chain
Modulator
TV
Video Board (M)
CPU Board (T)
I/O Module
Cassette
Cassette and Printer Controller
Printer
Keyboard controller
Keyboard
System RAM
ROM key
Monitor ROM
Micro processor Z-80
I/O Selection
Memory Selection
Address Bus
Data bus
CPU Board

1.2 CENTRAL PROCESSING UNIT

The Central Processing Unit (CPU) is the centre of the P2000; all the other units are built around it, as a sort of life-support system. The CPU is the part of the computer which reads instructions from a program and carries out the processing required for each instruction (for example, arithmetic operations).

The CPU of the P2000 is the well-known and widely-used Z-80 microprocessor. So much has been written about the Z-80 that this manual does not attempt to describe it - to do so would take a whole book in itself! The reader is referred to the list of relevant documents at the end of this manual.

The Z-80 microprocessor chip is present on the CPU board of the P2000. The M and T models of the P2000 have different CPU boards, as can be seen in Figure 1; however, the connections between the Z-80 and other units in the P2000 are the same for both versions. There are three main groups of lines:

- address lines, with which the CPU addresses either a memory location or an input/output device.
- data lines, on which data is transferred between the CPU and the memory location or I/O device addressed.
- control lines, which determine whether a memory location or an I/O device is addressed, and control other functions.

Each location in the memory has a unique address and can be accessed by the CPU. When the control lines indicate 'memory access', the address decoder interprets the address lines as an address in memory. If the control lines indicate reading, the contents of the memory location are put on the data lines and input to the CPU; if writing, the CPU puts the information to be written in the location onto the data lines, and this is stored in the addressed location.

Just as each memory location has a unique address, each I/O device has a unique port, or group of ports. When the control lines indicate 'I/O', the address lines are interpreted by the I/O decoder as a port number, and the CPU accesses the corresponding device. If the control lines indicate reading, input is taken from the device and put onto the data lines; if writing, the CPU puts the data to be output on the data lines, and it is output by the device.

The P2000 does not use direct memory access; that is, data to be transferred between a device and memory must first be transferred to the CPU, and then to its final destination.

1.3 FLEXIBLE DISKS

The P2000 can support up to four drives for single-sided double-density flexible disks (also known as floppy disks), providing a mass-storage facility for data and programs. On the M model, two disk drives may be mounted in the same cabinet as the monitor unit, on the T model, the disk units are free standing; in both cases, there is no direct physical connection between the disk units and the monitor.

The control circuitry for the disk units is present on the extension board in the main unit. The disk units are connected to the main unit via a flat 34-pole cable which plugs into the connector socket at the rear; a separate earth wire is connected to an earth pin beside the socket.

Control of the flexible disk units is a complex function which is performed by the floppy controller, a separate Z-80 microprocessor with its own timimg circuitry, which is however under the control of the CPU. So the monitor does not control the disk units itself, but it will, if instructed, read some disk control software from disk into memory. Reading this software is the only disk control action the monitor performs for itself, but once the software has been read, the monitor can use it to handle all aspects of disk unit control.

1.4 KEYBOARD

The P2000 keyboard unit houses the keyboard itself, as well as the circuit boards which contain the CPU, memory and control circuitry for other units.

The keyboard itself comprises 74 keys, each of which generates a particular code when pressed. Under control of the monitor, the CPU addresses the keyboard at regular intervals; a signal is returned to the CPU on the data lines indicating whether a key has been pressed. If one has, the monitor will then check which key - the corresponding key-code will be found. Different application programs interpret the key-codes in different ways, which allows for different national keyboards, for example.

1.5 I/O SLOT

The second slot in the top of the main unit, behind the ROM key can hold one of several different modules to perform I/O. Although the function and the circuitry of each module differs their connection to the system is the same in all cases. When inserted, the module is connected directly to the address, data and control lines, in the same way as any other input/output device.

Each module contains its own control circuitry and memory if required. The monitor has no control over the functions of the module in the slot; application programs can only transfer data to and from the module, by means of the ports assigned to it.

1.6 MEMORY

The total amount of memory in the P2000 varies from version to version, and is divided into several distinct sections:

- monitor ROM - 4k of memory in which the monitor program is stored. ROM stands for read-only memory, indicating that the monitor can be read but not altered. The monitor ROM is located on the CPU board.

- ROM key - the plug-in ROM keys can each contain up to 16k of ROM, in which the different application programs are stored - for example, the word processing application.

- video memory - 4k of RAM in which the characters to be displayed on the video are stored. RAM stands for random access memory, and it can be read from or written to by the CPU. In the T model, the video RAM is located on the CPU board; in the M model, on the video board.

- system RAM - 16k of RAM for general use. The monitor uses a very small section of this memory, and the rest is freely usable by the application program. The system RAM is located on the CPU board.

- extension RAM - two extensions, each of 16k, for general use. If the system includes flexible disks, these extensions must both be present, since the disk controller software is stored in here. The extension RAMs are located on the extension board.

Each location in the memory has a unique address; the CPU can access any address in any section of memory, as described in section 1.2, but will do so only when instructed to by the monitor or application software. Each section of memory is laid out in a particular structure, which is described in detail in the appropriate chapter. Clearly, users writing their own application programs must know exactly how the memory is laid out if they intend to access memory directly.

When the P2000 is switched on, or the RESET button is pressed, the monitor program starts running immediately. The monitor will instruct the CPU to start executing the application program stored in the ROM key, if one is inserted. From that point on, the behaviour of the system depends on the application program, and the monitor is only working in the background, providing control functions for the application program to use.

The memory of the P2000 is described in detail in chapter 4, except the video memory, which is described in chapter 7.

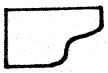
1.7 MINI-DIGITAL CASSETTE

Built into the main unit of the P2000 is the mini-digital cassette unit; this is the primary storage device for the system, particularly for versions without flexible disks.

The cassette unit circuitry is located on the CPU board, incorporated with the printer control circuitry; the monitor software controls the cassette unit via this circuitry. When the system is switched on, or the RESET button is pressed, if no ROM key is inserted, the monitor will start reading the cassette in the cassette unit, if one is present, will load the first program it finds on the tape into the system RAM and start executing it.

1.8 PRINTER

Although not all P2000 systems have a printer, the hardware and software needed to control one are present in all versions. The printer is connected to the main unit by the connector socket at the rear; this socket is a standard 25-pole connector according to the RS232-C standard, also known as V-24. The socket is marked:



The control circuitry for the printer is located on the CPU board, together with that for the mini-digital cassette unit. The monitor software controls the use of the printer by application programs. For example, if the printer signals that it is not ready for more than 10 seconds at any time during printing, the monitor will inform the application program that printing cannot be performed.

Different printers use different character code systems, which are not generally the same as the internal character code used in the P2000 itself; the monitor therefore translates the P2000 codes into the correct character codes for the printer, by looking up a conversion table which is stored in memory. Each application program must provide its own conversion table for the monitor, which allows different character-sets to be used for different national versions, for example.

1.9 VIDEO

The P2000 supports three different kinds of video - the monitor unit of the M model, a normal TV set or an RGB monitor on the T model. At the rear of the main unit are three connector sockets for the different video units. They are marked as follows:


connector to M model monitor


connector to TV


connector to RGB monitor.

Although all three sockets are marked on both models, on the M model, only the monitor socket is present; on the T model, only the TV and RGB monitor sockets are present.

Controlling the video is one of the most complex functions of the P2000, so the video control circuitry functions almost completely independantly of the monitor, although the CPU is still in control. The control circuitry required for the M and T models is slightly different; in the M model, the video controller is located on the video board, whereas in the T model, it is on the CPU board.

The video memory contains the codes representing the characters to be displayed on the screen, together with information about how they are to be displayed (inverted, for example). These codes are written into the memory by the CPU, the same way as storing data in any other part of memory. The memory holds 24 lines of 80 characters; on the M model, these are all displayed, on the T model, 24 lines of 40 characters can be displayed at one time. In addition, there is one extra line for vertical scrolling; when the text on the screen moves up one line, the new line which appears at the bottom is taken from the scrolling area.

Characters are represented on the monitor screen of the M model as patterns of green dots on the black background. The video memory is read, in strict synchronisation (provided by a special timing circuit), and for each character code, the corresponding pattern of dots is generated. This dot pattern is then sent to the monitor, where the electron beam scans the screen, producing the dots of light. Since the complete picture on the screen is made 50 times per second, it appears to be constant.

On the T model, a normal TV set or RGB monitor is used as the video, so the characters are represented in the same way as characters on a teletext system. As in the M model, the video memory is read in synchronisation, and the corresponding pattern produced for each character, together with signals for the red, green and blue colours. These signals are passed to the VHF modulator, which produces a composite signal suitable for input to a television; they are also passed to the RGB interface, for output to an RGB monitor. Again, the complete picture is made 50 times per second, giving a constant appearance.

2 FLEXIBLE DISKS

The flexible, or 'floppy' disks provide a large capacity mass storage facility for the P2000. The disks are 5¼" diameter, single-sided and double-density, and each disk can hold 140k bytes. The contents of a disk depends largely on the application software which recorded the information on the disk, but there are some features which are common to all disks.

2.1 DISK DRIVE UNITS

Two disk drive units can be supported by all P2000 models, and in some cases two more may be connected, allowing a total of four disks to be in use at one time, with a capacity of 560k bytes. On the M-model, two disk units may be installed in the same cabinet as the monitor, there is no physical connection between monitor and disks.

The flexible disk drive contains a motor with its own 12 V power supply, which rotates at 300 r.p.m. and drives a spindle via a belt-drive system; the motor speed is regulated by feedback from an internal tachometer. When the drive unit door is shut, a clamp holds the disk onto the hub of the spindle; registration marks on the face of the spindle hub ensure that the disk is centred correctly.

The read/write head of the disk drive is a single element glass bonded ferrite/ceramic head. There are erase elements on either side of the head which erase the areas between tracks whenever the head is reading - this improves the signal/noise ratio of recorded data.

The read/write head is mounted on rails and is positioned facing one of the 35 concentric tracks of the disk by a stepping motor and a face cam. The stepping motor receives pulses from the disk controller to indicate the number of tracks to be moved from the current position, and the direction (in or out), and rotates the cam two step increments per track. When the head is to be used, the disk is moved to directly touch the head by a solenoid.

Data is recorded on the disk in binary, using frequency modulation; that is, each bit is recorded together with an associated clock pulse. A bit cell is the period between (the leading edge of) one clock pulse and the next. Each bit cell may contain a pulse indicating a one, or no pulse, indicating a zero. A byte consists of eight consectutive bit cells; bit 0 is written first and bit 7 last. The format of data bytes on the disk is described in the next section.

The drive units also contains an LED and a photodetector positioned in line with the small hole in the disk itself. The photodetector generates a pulse once every rotation of the disk, when the hole passes it, allowing light from the LED to be received. Since the hole indicates the start of the first sector of the disk, this signal ensures the disk controller can activate the read/write head at the start of a sector.

A special signal is also sent to the controller when the head is positioned on the outermost track; in this way, a reference point is given to the controller. When a drive is selected, the red light on the front is lit, and the controller steps the read/write head out until the outermost track signal is received.The drive unit also senses whether the disk is write-protected, and signals this to the floppy controller. Data cannot be written on the disk when it is write protected.

2.2 DISK FORMAT

The layout of data on the disks - the disk format - is determined by the disk controlling software. The disks are pre-formatted by Philips, to mark the disks into tracks and sectors; this is required by the disk software, and for this reason only disks supplied by Philips can be used on the P2000.

Every disk contains 35 tracks, each divided into 16 sectors of 256 bytes. The physical layout of the sectors is consecutive, but the disk control software gives each a number, which is not consecutive; that is, sector 1 is not next to sector 2 on a track. This is known as the logical mapping or numbering of the sectors.

The methods of allocation of disk space to files, and the maintenance of a directory, varies from application to application, depending on the disk control software.

2.3 DISK CONTROL SOFTWARE

The disk units are handled by the floppy control circuits on the extension board, under software control. These select the drive to be used (only one drive can be active at a time), switch the motor on, control the head movement via the stepping motor as described in section 2.1, and control reading and writing from the disk.

There are several different versions of disk control software; different applications use different disk software. In general, the disk control software must perform a number of tasks. It is responsible for the allocation of disk space to files; a directory must be maintained on each disk, listing all the files and where on the disk they are recorded - the directory must be updated whenever a file is added, changed or deleted. The disk control software is also responsible for the code used to record data on the disk, and for the conversion of logical sector numbers to physical sector numbers.

The monitor disk bootstrap routine provides the monitor's only ability to directly control the disk units. The routine loads track 1 of the disk in the lefthand drive into RAM; track one should therefore contain the disk control software, which is therafter used to perform disk control.

The disk bootstrap routine is called from the system (re-)start routine, if this is specified by the ROM key, as described in section 4.3.2; it may also be called by an application program if wished. The routine takes no parameters, and starts at address 0E90 hex.

3 KEYBOARD

The keyboard is the same for the P2000 T and the P2000 M model. It comprises 74 keys; there is no distinction between the keys of the main keyboard and those of the keypad on the right. The RESET button is not a key.

Although the keys are marked with symbols, the meaning of each is competely dependant on the application; even the alphabetic keys can be assigned different meanings by an application if necessary. It is even possible for the same key to have different meanings under the same application, for example, if the application can be in several different 'modes'.

3.1 KEYBOARD UNIT

The keys are connected in a matrix form. Each key is a switch between an x-line and a y-line; there are ten x-lines and 8 y-lines. Normally all the lines are at logic one - when a key is pressed, one x-line and one y-line will be made zero. This x-y pair specifies which key has been pressed. The pressing of a key only registers in this way when the keyboard is enabled, and that is under software control, as described in the next section.

3.2 KEYBOARD CONTROL

The monitor performs all keyboard control functions, although the application determines the meaning given to each key. The application can prevent the monitor from controlling the keyboard if necessary.

There are two monitor keyboard handling routines:

- keyboard initialise routine - called by the system (re-)start routine to enable and initialize the keyboard.
- reading keyboard - the main monitor routine scans the keyboard regular intervals to see if a key is pressed.

These are described in the following sections.

3.2.1 Memory areas

A small part of the system RAM reserved for monitor variables is associated with the keyboard. There are three items:

- keyboard queue - 12 bytes used as a buffer to hold the codes of up to 12 keys which have been pressed; stored in locations 6000 to 600B hex.
- queue counter - one byte which counts the number of valid key-codes in the queue; stored in location 600C hex.
- current key - one byte which holds the key-code of the key currently pressed; stored in location 600D hex.

The way in which these areas are used by the monitor is explained in the descriptions of the routines which follow.

3.2.2 Keyboard initialise routine

This routine is called by the system (re-)start routine described in section 4.3.2. It is not usually desirable for this routine to be called by an application. The routine takes no parameters, and starts at address 1B hex, which jumps to address 0167 hex. The action of the routine is as follows.

If the extension board is fitted in the P2000, interrupt mode 2 is selected. The interrupt vectors which can be generated by the CTC are:

- on channel 0 - 'disk operation ready or timer CTC interrupt'; stored in system RAM location 6020 hex.
- on channel 1, 'disk not ready interrupt'; stored in system RAM location 6022 hex.
- on channel 2, 'communication interrupt'; stored in system RAM location 6024 hex.
- on channel 3, 'keyboard timer interrupt' (this is external to the CTC); stored in system RAM location 6026.

If no extension board is fitted, interrupt mode 1 is selected.

The CTC timer is initialised, and the keyboard queue counter is set to zero (indicating that the keyboard queue is empty). This ends the keyboard initialisation.

For more information on the CTC timer, refer to the list of reference material at the end of this manual.

3.2.3 Reading keyboard

Every 20 ms, an interrupt is generated by the interrupt control circuit on the CPU board. The monitor then scans the keyboard to see whether a key is being pressed. This is done as follows.

The nine x-lines of the keyboard matrix are made logic zero. An input is made from ports 0 to 9 (assigned to the keyboard), putting the signals from the eight y-lines onto the data bus as one byte. If one of these bits is zero, a key is being pressed.

If no key is being pressed, the monitor exits and control returns to the application - until the next interrupt 20 ms later. If however, a key is being pressed, the monitor continues, to establish which key. To do this, an input is made from port 0, making line x0 zero, and at the same time taking the signals from the y-lines onto the data bus. If one bit is zero, this indicates which key is being pressed. If not, an input is made from port 1, making line x1 zero, and so on - this process can be continued up to port 9. It is the combination of port (x-line) and bit (y-line) which determines the key being pressed.

The current key byte and the keyboard queue and counter are updated by the monitor. If a key was not pressed, the current key byte is set to zero, and the queue and counter remain unchanged. If a key was pressed, the equivalent key-code is stored in the current key byte; if the queue counter is less than 12, the new key-code is stored in the next free position in the queue, and the counter is increased by one. If the counter is already 12 (the queue is full), the new key-code overwrites the last one in the queue.

The key-codes used indicate only the position of the key on the keyboard, not its meaning. The meaning depends on the application, which also allows for the different keyboards used in different countries. The codes generated by each key are shown in Appendix E.

The keyboard reading routine will only register that a key is being pressed if it tests whilst the key is held down. The whole test is made every 20 ms, and takes considerably less than 20 ms to complete; an operator, however quickly he presses the keys holds each key down for longer than that, so it is not possible for the monitor to 'miss' a key being pressed. Indeed, the reverse is often the case - the key is held down for so long that the monitor registers the key several times over. To avoid repetitions, the monitor compares the key being pressed with the last key pressed, as stored in the current key byte; only if the same key is held down for a longer period does this register as a repeat, giving the useful 'repeat key' facility.

3.3 APPLICATION KEYBOARD ACCESS

There are five monitor routines which can be called by the application to examine the keyboard:

- keyboard status routine - called to see whether a key has been pressed.
- keyboard input routine - called when a key has been pressed, to determine which key it was.
- keyboard clear - called to empty the keyboard queue.
- keyboard disable routine - called to disable keyboard input.
- keyboard enable routine - called to enable keyboard input again.

These are described in the following sections. Note that even while these routines are being executed, the 20 ms interrupts are continuing, causing the monitor to examine the keyboard and update the keyboard queue.

3.3.1 Keyboard Status Routine

This monitor routine is called by an application to determine whether a key has been pressed; that is, whether the keyboard queue is empty. The routine starts at address 29 hex, which jumps to address 0489 hex, and takes no parameters.

The routine returns two indicators of keyboard status. If the keyboard queue is empty, the Z-flag of the CPU status register is set. If the key-code for the STOP key (SHIFT together with the bottom righthand key on the keypad) is one of the codes in the queue, the Carry flag is also set. This is useful for applications where operations must be able to be interrupted by pressing this key.

Once the keyboard queue is full, each new key-code overwrites the last in the queue. Therefore, to avoid loss of keyboard input, the status should be examined frequently, and the keyboard input routine called when the queue is not empty.

3.3.2 Keyboard Input Routine

The application calls this monitor routine to read the next key from the keyboard queue. The start address is 26 hex, which jumps to address 049B hex.

This routine puts the key-code of the next key in the queue into register A. If this is the code for the STOP key (SHIFT together with the bottom righthand key on the keypad), the Carry flag is also set. If the queue counter is zero (the queue is empty) the routine will wait until a key is pressed; to avoid this waiting, the application should call the keyboard status routine first, to check the queue is not empty.

The key-codes stored in the queue indicate only the position of the key on the keyboard; the code for each key is shown in Appendix E. When the keyboard input routine returns, the application must itself translate the key-code in register A into the P2000 internal code, by looking up the 'keyboard table'. Since each application has its own keyboard table, different applications can give the same key a different meaning.

The location of the keyboard table depends on the application. The table itself consists of 90 bytes, each containing the P2000 internal code for the key, in order; that is, the first byte contains the internal code to be used for the key with key-code 0. The application merely has to access the byte at the starting address plus offset of the value in register A, to find the appropriate code. The P2000 internal code is described in chapter 7.

3.3.3 Keyboard Clear Routine

The keyboard clear routine is called by an application to empty the keyboard queue; this is done by setting the queue counter to zero.

The start address is 2C hex, which jumps to address 04DC. There are no parameters.

3.3.4 Keyboard Disable Routine

By calling this routine, the application stops the monitor reading the keyboard every 20 ms. Keys pressed whilst the keyboard is disabled have no effect. This can be useful to increase application execution speed. The keyboard enable routine should be called afterwards to enable keyboard reading again.

The start address of the routine is 23 hex, which jumps to address BC hex. No parameters are taken.

3.3.5 Keyboard Enable Routine

This routine re-enables the monitor keyboard reading routine. The start address is 20 hex, which jumps to address B4. There are no parameters.

4 MEMORY AND MONITOR

The P2000 memory is the area used for storage of programs and data, which can be directly accessed by the CPU. The amount of memory varies depending on the model, but it all functions in the same way.

The monitor is the basic software built-in to the P2000 to provide primitive functions, such as system start-up, interrupt handling and some input/output control. There are two advantages in having a monitor. Firstly, because it is started as soon as the power is switched on, and instructs the CPU to start executing the application program is one is present - otherwise the CPU would have no instrcutions to execute. Secondly, because without the monitor, each application program would have to perform all the primitive tasks itself, which would make application programs longer and more complex. Instead, they can use the monitor to perform these functions.

4.1 MEMORY INTRODUCTION

Each element (or 'location') of memory is one byte, which is eight bits. Each byte has a different number, known as its 'address' - these range between 0 and 65535, that is 0 and FFFF hex, giving a maximum total memory size of 64k (some P2000 models have less). This range of numbers can be expressed in binary using 16 bits.

The CPU can access one element of memory at a time; it does this by putting the address of the element (in binary) onto the address lines A15 to A0. The control line MREQ (memory request) is made high, to indicate that the address lines are addressing a memory element (they can also be used to address an input/output port, as described in section 1.2).

Although the P2000 memory appears to be one complete unit, it comprises several different chips on the circuit boards; the complete structure of the memory is given in the next section. The address decoder circuit interprets the most significant five of the 16 address lines to determine which chip contains the element that the CPU is accessing; it gives an enable signal to the appropriate memory area. The remaining 11 address lines specify the element within the chip.

In the case of the system RAM, eight memory chips forming one unit of 16 k, address lines A13 to A0 specify the element within this area. The RAM structure requires that each memory location is addressed in two stages, a 'row address' and a 'column address' each of seven bits. The address multiplexer circuit derives the row address from address lines A6 to A0, followed by the column address derived from A7 to A13; together, these specify one of 16 k elements. The memory control circuit supplies signals to ensure that the row and column addresses are supplied with the correct time delay.

When addressing a memory location, the CPU must also make either the RD (read) or WR (write) control line true, to indicate whether it is going to read the value currently stored in the memory location, or store a new value in the location. Reading is possible from all memory elements, but some memory areas may not be written. When a location is read, the 8 bit value from the location is output to the bus driver, which passes the value along the data bus lines to the CPU. When writing, the CPU puts the 8 bit value to be written onto the data bus lines while it addresses the element, and the value is copied fom the lines into the addressed location by the bus driver.

The RAM memory can, due to its electronic construction, only retain its stored contents for a very short time without power (this is one of the reasons that all data and programs are lost when the P2000 is switched off). The memory control circuit therefore performs the additional task of 'refreshing' the memory. Following a clock signal from the system timing circuits, a pulse is sent to a complete row in RAM, to refresh it; this is continuously repeated for each row, with the result that the RAM memory's contents remain constant. The refresh action is performed during the second half of the CPU's instruction cycle, ensuring that the CPU never attempts to access the memory whilst it is being refreshed.

The refresh process takes place on all RAM memory, both on the CPU board,the video board of the M model, and the extension board if present. ROM memory has a different construction, and therefore does not require refreshing.

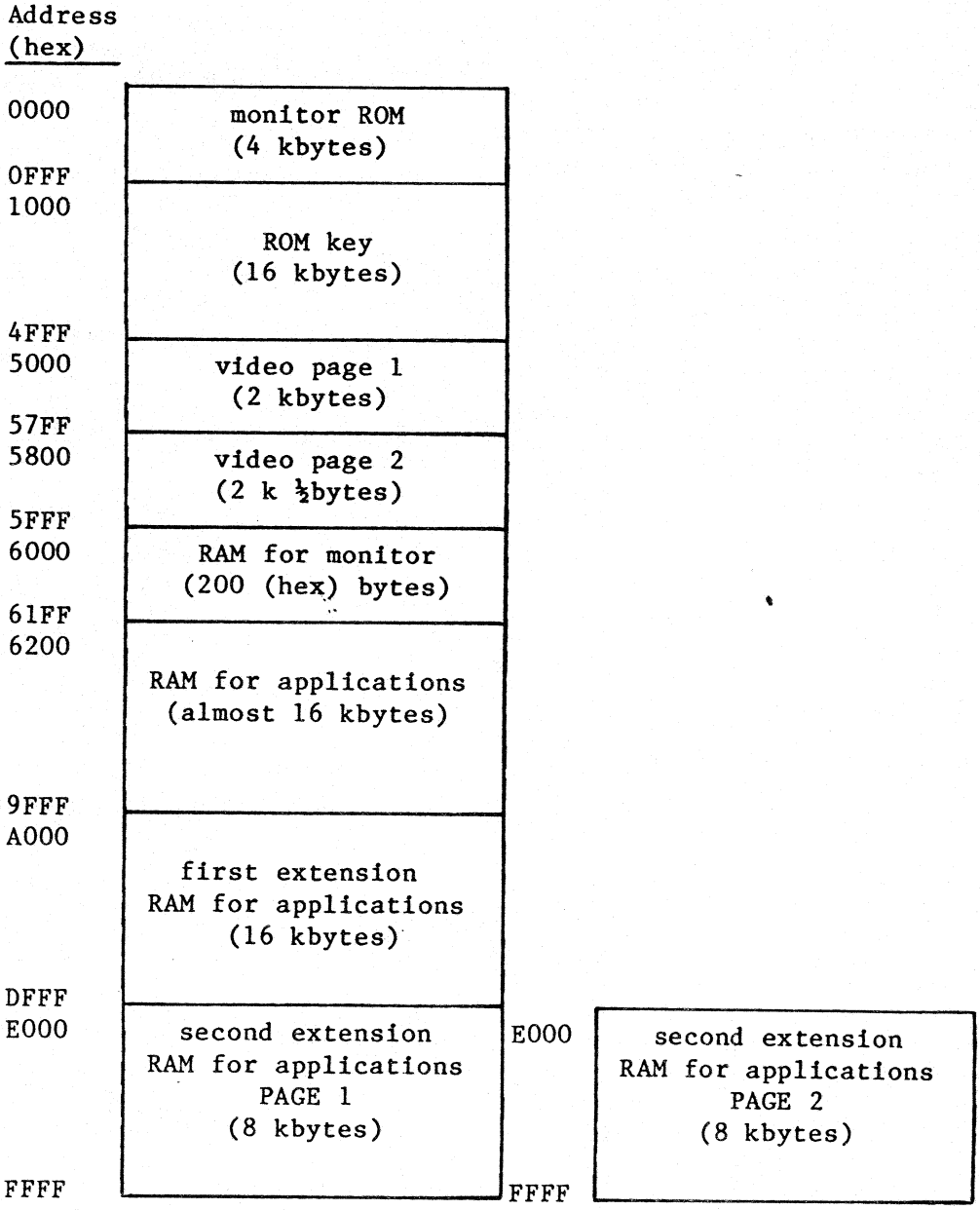


Figure 2: Memory map

4.2 MEMORY MAP

As can be seen from the memory map in figure 2, the P2000's total memory is divided into several distinct sections:

- monitor ROM - 4k of memory in which the monitor program is stored. ROM stands for read-only memory, indicating that the monitor can be read but not altered. The monitor ROM is located on the CPU board.

- ROM key - the plug-in ROM keys each contain 16k of ROM, in which the different application programs are stored - for example, the word processing application.

- video memory - 4k of RAM in which the characters to be displayed on the video are stored. RAM stands for random access memory, and it can be read from or written to by the CPU. In the T model, the video RAM is located on the CPU board; in the M model, on the video board.

- system RAM - 16k of RAM for general use. The monitor uses a very small section of this memory, and the rest is freely usable by the application program. The system RAM is located on the CPU board.

- extension RAM - two extensions, each of 16k, for general use. If the system includes flexible disks, these extensions must both be present, since the disk controller software is stored in here. The extension RAMs are located on the extension board.

Each section of the memory is discussed in the following sections, except the video memory, which is discussed in Chapter 7.

The distinctions between the different sections of memory is very carefully observed by the monitor and all application software supplied by Philips; user-written software should also observe these distinctions carefully, to avoid unexpected effects, which may cause problems.

4.3 MONITOR

The monitor is therefore a background support, or the bottom layer of software, on which application programs can rely. The monitor program itself is stored in 4k of read-only memory, composed of two ROM chips on the CPU board. A small section of the system RAM is also used by the monitor to store data values which it uses.

The monitor is a complex program. The major routines which it comprises are the following:

- system (re-)start routine
- disk bootstrap routine (chapter 2)
- printer routine (chapter 6)
- cassette routine (chapter 5)
- keyboard input routine (chapter 3)
- keyboard status routine (chapter 3)
- bell routine
- clear video screen routine (chapter 7).

The system (re-)start routine and the bell routine are described in the following sections; the other routines are described in the chapter given in parentheses.

4.3.1 System (re-)start routine

When the power is switched on, or the RESET button is pressed, the program counter of the CPU is cleared to zero, so the CPU starts executing instructions beginning at location 0000, which is where this routine begins. The following sequence of actions is performed:

1. Disable all interrupts.

2. Test whether the ROM key is a maintenance module. (The way in which this can be tested is described in section 4.4.2.) If it is, then start executing the maintenance program immediately; if it is not a maintenance module, or no ROM key is inserted, then continue.

3. Call the bell routine (see section 4.3.3) to sound the bell.

4. Test the amount of RAM available (that is, the amount present in this version), and store the result in the RAM area reserved for the monitor (see section 4.5.1.).

5. If the test for RAM results in an error, then display the message 'CALL SERVICE' on the video. (The way in which messages are displayed on the screen is described in chapter 7.)

6. If 48 k of RAM is available, and the ROM key specifies that the disk control software should be loaded, then call the disk bootstrap routine (this is described in chapter 2).

7. Set the printer transmission rate in the RAM area reserved for the monitor (see section 4.5.1 and 6.4).

8. Call the keyboard initialise routine to clear the keyboard queue. (Keyboard control is described in chapter 3.)

9. If a ROM key is inserted, start executing the application software in the ROM-key.

10. If no ROM-key is inserted, display the message 'P2000 PHILIPS MICROCOMPUTER', and continue.

11. Test if a cassette is inserted in the cassette unit. If a cassette is inserted, read the header from the first file found on the tape. If no cassette is inserted, repeat step 11; that is, keep testing until a cassette is inserted. (The way in which the testing and reading is done is described in chapter 5.)

12. If the header of the cassette file indicates that the file is a program file, read it into RAM, starting at the first location available for application software. If it is not a program file, repeat step 12; that is, read the header from the next file on the tape.

13. Start executing the application program.

It can be seen that this sequence of actions results in the system waiting until an application program, either in a ROM key or on cassette, can be executed.

The system (re-)start routine takes no parameters; it may be called by an application program if wished.

4.3.2 Bell routine

The bell routine sounds the bell by sending 80 hex alternating ones and zeros to the bell control circuit. On the M model, the bell control circuit is connected to a beeper, which produces a short tone. On the T model, the bell signal is sent to the TV or RGB monitor, and appears as flashing mark on the screen.

The bell routine is called by the monitor (for example, by the system (re-)start routine), and Application programs may call the bell routine if wished. The entry point is address 32 hex; no parameters are required. For shorter or longer sounds of the bell, applications may themselves send alternating ones and zeros to any of the bell control unit ports, numbers 50 to 5F hex.

4.4 ROM KEY

The P2000 ROM keys each contain up to 16k of read-only memory, in which is stored an application program. There are two advantages in the use of ROM keys; first, that changing application programs is very quick and simple; second, because it's read-only memory, application programs cannot become erased or overwritten, as could happen with a disk or cassette.

The application stored in the ROM key is a machine-code program which can be directly executed by the Z-80 microprocessor. When the power is switched on, or the RESET button is pressed, the microprocessor starts executing the monitor (re-)start routine, and this routine contains the instructions to start executing the program in the ROM key, if one is present; this is described in section 4.3.2.

4.4.1 ROM key structure

There are different sizes of ROM key. Most comprise one continuous block of 16k read-only memory, with addresses 1000 to 4FFF hex. Others comprise two sections of various sizes, with a gap in between. This is indicated in the ROM key identification byte, as explained in the next section. However, this is only an important factor if user programs are directly reading from the ROM key area - the monitor only refers to addresses 1000 to 1010 hex, and the structure of the rest of the key does not affect it.

The different ROM-key configurations are shown in Figure 3.

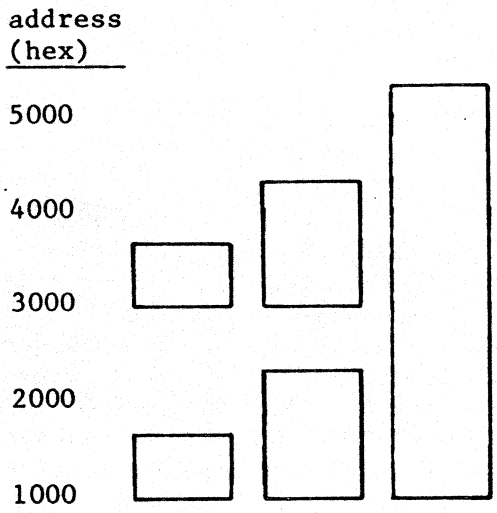


Figure 3: ROM key configurations

Those parts of the application software which differ in versions produced for use in different countries (for example, messages to be displayed on the video) are always contained in the ROM chip with the lowest addresses. Consequently, the contents of this chip may vary from version to version.

4.4.2 ROM key identification block

The first fourteen bytes of all ROM keys have the same layout, so that the monitor can check what sort of application is stored in the key. This 'ROM key identification block' starts at address 1000 hex; it is laid out as follows:

Address (hex)	Bit	Indication
1000	0	this bit is always zero.
	1	if this bit is one, the monitor (re-)start routine should load the disk control software from disk before starting to execute the application program itself, as explained in section 4.3.2.
	2	if this bit is zero, the ROM-key contains a maintenace program which the monitor will start executing immediately; if one, the ROM-key contains normal application software.
	3	if this bit is zero, the ROM-key consists of two address ranges with a gap in between; if one, it contains one continuous address range, as described in section 4.4.1.
	4	this bit is always one.
	5	this bit is always zero.
	6	this bit is always one.
	7	this bit is always zero.
		bits 4 to 7 allow the monitor (re)-start routine to check whether there is a ROM-key inserted or not.
1001		application program length in bytes (low byte).
1002		application program length in bytes (high byte); both monitor and user programs may need to know the length of the application.

(continued overleaf)

Address (hex)	Bit	Indication
1003		checksum (low byte).
1004		checksum (high byte); the monitor (re-)start routine performs a checksum calculation - the result should be the same as this value.
1005 - 100C		application program name; up to 8 characters in Viewdata code. This is the code used internally by the P2000, as described in Appendix D.
100D		release number in binary; different versions of an application may be released over a period of time.
100E		reserved.
100F		reserved; these bytes may be used in future applications.

If the ROM-key comprises two sections, the information in bytes 1000 to 1004 is repeated at the start of the second section, that is, at addresses 3000 to 3004 hex.

4.5 SYSTEM AND EXTENSION RAM

The system RAM is 16k of random-access memory present in all P2000 models, most of which is freely usable by application programs; a small part is reserved for the monitor's use. The extension RAM is one or two sections each of 16k, which are present only in some models; this RAM is entirely for the use of the application.

The extension RAM is described in section 4.5.2. Each application uses the system and extension RAM in a different way, although generally some part will be reserved for program and some for data; details of the structure used may be found in descriptions of the application concerned.

The monitor requires the use of 144 bytes of system RAM to store its data variables - these cannot be stored with the monitor program since this is in read-only memory. The structure of this area is detailed in the next section.

4.5.1 Monitor RAM Area

The monitor uses the first 144 bytes of RAM to store data for its own use; application programs may also need to refer to these variables. The memory map on the next page details the contents of the bytes.

Address (hex)	Indication	reference to other chapter (if applicable)
6000 - 600B	keyboard queue	3
600C	queue pointer	3
600D	current key	3
600E	CTC counter	
600F	shift/lock flag (for keyboard routine	
6010 - 6011	clock counter (incremented every 20 ms)	
6012	monitor flags - these have various meanings	
6013	application flags - this byte is for use by the application	
6014 - 6015	address of start of display area for different symbols on video	7
6016	printer baud rate	6
6017	cassette error byte	
6018 - 6019	cassette start address	
601A - 601B	cassette valid data section start address	
601C - 601F	temporary storage for cassette routine	
6020 - 602F	interrupt vector table (holds addresses to which monitor should jump when an interrupt is received)	

(continued overleaf)

Address (hex)	Indication	reference to other chapter (if applicable)
	locations 6030 to 604F comprise the cassette file descriptor:	5
6030 - 6031	data transfer address	
6032 - 6033	total file length	
6034 - 6035	valid data section length	
6036 - 603D	file name	
603E - 6040	file extension	
6041	file type	
6042	data code	
6043-6044	start address	
6045-6046	load address	
6047-604E	reserved	
604F	record number	
6050	cassette routine command flag	5
6051 - 6054	temporary storage area used by cassette routine	
6055 - 6056	used to save old value of stack pointer (SP)	
605C	memory size: value 1= 16 k value 2= 32 k value 3= 64 k	4
605D	disk inserted - if 1, a disk is currently inserted	2
605E - 605F	address of printer table	6
6060	cassette status flag	5
6061 - 6062	used by cassette routine to save old value of stack pointer (SP)	

(continued overleaf)

Address (hex)	Indication	reference to other chapter (if applicable)
6063	temporary storage for cassette routine	
6064 - 6065	start address cassette data section	
6066 - 6067	cassette data section length	
6068 - 6069	address of cassette header	
606A - 606B	length of cassette header	
606C - 606D	temporary storage for cassette routine	
606E - 606F	number of blocks making up cassette file	
6070 - 608F	this area is used by the disk control software.	2
6090	start of application RAM.	

Many applications, such as BASIC, allow the user to write programs. These are often confusingly called 'application programs' - a more suitable name would be 'user programs'. The area of RAM for use by applications is where the BASIC system program itself is stored; the BASIC system itself will set aside a part of this area to hold the user program. So it is the BASIC system itself which starts at address 6090 - the user program will be stored in another section of memory.

4.5.2 Extension RAM

There may be one or two memory extensions on the P2000 - with both extensions, the P2000 has a total of 72k memory. The first extension is 16k RAM, with addresses A000 to DFFF hex. This only leaves addresses E000 to FFFF for the second, which is 8k, not 16k. To double the available addresses on the second extension, the extension is divided into two banks each of 8k, with the same addresses. The first bank - bank 0 - is selected by setting a zero at port 94 hex; the other - bank 1 - is selected by sending a one to port 94. The signal on port 94 is used by the address decoding circuits to select the appropriate bank.

A bank remains selected until the other is selected; but when the monitor and the disk control software are called, they always select bank 0 before returning.

5 MINI-DIGITAL CASSETTE

5.1 CASSETTE UNIT

The motor in the cassette unit turns one of the two hubs of the cassette unit at 360 r.p.m., allowing forward and reverse movement of the tape at an average of 40 cm/s. When the cassette is inserted correctly and the drive door is shut, the tape is brought into contact with one of the two heads, read/write and erase. Only one head may be activated on one of the two tracks at a time.

Data is always recorded in the forward direction of the tape (that is the tape moves from right to left), and must be read in the same direction. Each track is therfore only accessible from one side; that is the cassette must be turned over to access the other track.

The cassette unit can detect a number of internal conditions:

- cassette inserted and unit door shut - a switch on the door and a mechanical feeler detect this condition.
- cassette wound to beginning or end of tape - this is detected by a check on the motor rotation.
- cassette is write enabled - detected by a mechanical feeler which is depressed if the write enable plug is present.

These conditions are represented by signals within the cassette unit which are passed to the output control circuit on the CPU board.

Data is recorded on the tape in phase-encoded form; that is, a one bit is represented by the transition between a low value and a high value on the read/write head line, and a zero bit by a high-low transition. The transitions occur at the start of each clock pulse, provided by the cassette timing circuit; the clock pulses are not themselves recorded. At the beginning of each block of data a 'preamble' is recorded, consisting of the bit pattern 10101010 - this is used in the read mode to synchronise the clock so that the transitions are read correctly. Data bytes are recorded in serial form, least significant bit first; both recording and reading are performed in the forward direction only.

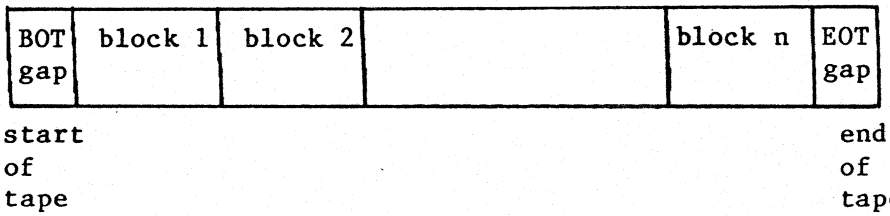
The layout of data on the tape is described in the next section.

5.2 TAPE FORMAT

Each track (or side) of a cassette is divided into 40 blocks of information. A file may comprise between 1 and 40 blocks, depending on its length.

5.2.1 Blocks, Gaps, Marks and Records

At the start of the tape is an area of clear (erased) tape, the BOT (beginning of tape) gap; to read past this gap takes approximately 1 second. After this, the first block starts, followed directly by the second, third and so on. After the last block on the track comes the EOT (end of tape) gap; if all 40 blocks on the track are used, this area of erased tape has a length equivalent to 1.8 seconds reading time.



BLOCK

Each tape block is made up of five sections of tape:

start gap	mark	mark gap	data record	end gap

- start gap - a section of erased tape separating the start of one block from the end of the previous block. This takes approximately 515 ms to read over.
- mark - four bytes of recorded information (described below).
- mark gap - a section of erased tape separating the mark from the data record; length about 85 ms equivalent.
- data record - 1056 bytes of recorded data (described below).
- end gap - a section of erased tape of length approximately 155 ms equivalent.

MARK

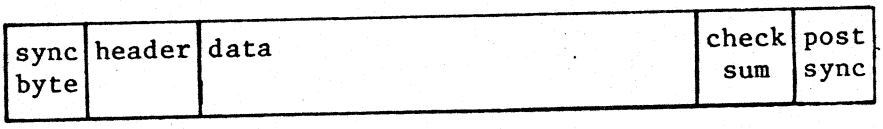
The mark is made up of four bytes with the following bit patterns:

byte 1: preamble synchronisation pattern 10101010 (AA hex)
byte 2: 00000000 (0 hex)
byte 3: 00000000 (0 hex)
byte 4: postamble synchronisation pattern 10101010 (AA hex)

The function of the synchronisation bytes is described in section 5.1. Bytes 2 and 3 are reserved for a check-sum or other data control information.

DATA RECORD

The data record contains the information which has been written onto the tape. It comprises five sections:



- sync byte - one byte with the preamble synchronisation bit pattern 10101010 (AA hex).

- header - 32 bytes which specify the contents and type of the data section. This is described in section 5.2.2.

- data section - 1024 bytes of information; described in section 5.2.3.

- check-sum - 2 bytes holding a value which can be used to check for reading errors.

- post-sync byte - the postamble synchronisation byte.

5.2.2 Header

The 32 bytes which make up the header section of the data record describe the contents of the data section which follows. This information is supplied by the monitor cassette routine when writing a data record; when the cassette routine reads a file from tape, the header is read first so that the cassette routine knows what to do with the data from the tape. The header bytes are always written from and read into the same area of system RAM, the 32 bytes starting at location 6030 hex, known as the 'cassette file descriptor'. Thus this area has the same structure as the header, as shown on the next page.

byte

	Data transfer address:
0	high byte
1	low byte.
	Total file length, in bytes:
2	high byte
3	low byte.
	Valid data section length, in bytes:
4	high byte
5	low byte.
	File name:
6-13	eight bytes in the P2000 internal character code.
	File extension:
14-16	three bytes in the P2000 internal character code.
	File type:
17	one byte in the P2000 internal character code.
	Data code:
18	one byte in the P2000 internal character code.
	Start address:
19	high byte
20	low byte.
	Load address:
21	high byte
22	low byte
23-30	Reserved for future use.
	Record number:
31	one byte, a number in binary.

Each item in the header is described below.

DATA TRANSFER ADDRESS

This is the starting address of the area in system RAM to which the data is to be transferred when read from cassette. This address is supplied by the application program when calling the monitor cassette routine to write the data on cassette in the first place.

TOTAL FILE LENGTH

This specifies the total number of bytes which make up the file. Files may comprise between 1 and 40 blocks on cassette; the monitor cassette routine uses this value to calculate the number of blocks which must be read to complete the file.

VALID DATA SECTION LENGTH

The length of the valid data section indicates how many of the 1024 data bytes of this record are actually used; for example, if this is 256, then only the first 256 bytes of the data section will be loaded into RAM. To avoid wasting space on the cassette, all records except the last in a file will usually use all 1024 bytes of data.

FILE NAME

The eight character file name identifies the file to which the record belongs; it will be the same in all records making up the file. Each record except the first is considered to be an extension, and has a further three character name to distinguish it. The file and extension names are in the P2000 internal code, listed in Appendix D.

FILE TYPE

The file type byte specifies the type of information represented by the bytes in the data part of the record. It is a single character in the P2000 internal code, with the following indication:

Character	File type
B	(Microsoft) BASIC program file
P	program file (other than Microsoft BASIC)
V	file of viewdata characters (same as P2000 internal character code)
W	Word processing text file
O	other

This indicates to the application the code in which the data is recorded.

DATA CODE

The data code identifier indicates the type of P2000 internal code which is used for the data - this is because different codes are used in different countries. The identifier is itself one character in the internal code, with the following indication:

Character	Data Code
D	German
S	Swedish
U	English, Dutch

Additional national versions are in preparation.

This byte only has relevance if the File Type byte has the value 'P'.

START ADDRESS

The start address specifies the address in system RAM of the first executable instruction of the program which the file contains. This is only valid when the file type= P.

LOAD ADDRESS

The load address specifies the address in system RAM at which loading of the program which the file contains is to begin; that is, the first byte of the program (which may not be the first executable instruction). This is only valid when the file type= P.

RESERVED BYTES

These eight bytes in the record header are not currently used, but may be required in future version of the monitor. They are therefore reserved, and applications should avoid writing information in these bytes.

RECORD NUMBER

This byte indicates the number of the record within the file (the extension number).

5.2.3 Data

The data section of the cassette records consists of 1024 bytes written in serial form, least significant byte first. When called by the application, the monitor cassette routine writes these bytes by copying the values from the specified area of RAM; thus the coding system used depends on the application which wrote the data into RAM, as described in section 5.2.2 under FILE TYPE.

5.3 CASSETTE CONTROL

The monitor provides basic cassette control software for use by application programs.

5.3.1 Memory areas

The monitor cassette routine uses two areas of system RAM to store data values used in cassette handling. These areas start at address 6017 hex and at address 6030 hex. The structure of the areas is as follows:

Address (hex)	Indication	
6017	cassette error byte	
6018 - 6019	cassette start address	
601A - 601B	cassette valid data section. start address	
601C - 601F	temporary storage	
6030 - 6031	data transfer address	
6032 - 6033	total file length	
6034 - 6035	valid data section length	
6036 - 603D	file name	
603E - 6040	file extension	together, these items correspond to a cassette record header
6041	file type	
6042	data code	
6043 - 6044	start address	
6045 - 6046	load address	
6047 - 604E	reserved	
604F	record number	

(continued overleaf)

Address (hex)	Indication
6050	command flag tape
6051 - 6054	temporary storage
6060	cassette status flag
6061 - 6062	used by cassette routine to save old value of stack pointer (SP)
6063	temporary storage
6064 - 6065	start address cassette data section
6066 - 6067	cassette data section length
6068 - 6069	address of cassette header
606A - 606B	length of cassette header
606C - 606D	temporary storage
606E - 606F	number of blocks making up cassette file

The purpose of these items of data storage will be described as they occur in the following sections. The most important area is the 'cassette file descriptor', locations 6030 to 604F hex, which holds a copy of the header from a cassette file. When a new file is to be written, the application must supply some of the information in the cassette file descriptor, the monitor cassette routine supplies the rest. When reading a file, the monitor reads the header into the cassette file descriptor, where the information can be accessed both by the monitor routine and by the application.

Several of the data items in the locations beginning at 6017 hex and at 6061 hex are used in the cassette routine to duplicate some of the items from the cassette file descriptor; arithmetic and logical operations can be performed on these duplicates, leaving the original data intact.

5.3.2 Initialise

The initialise function is called by loading value 0 into register A and calling the cassette routine at address 18 hex. The application should examine the Zero flag and/or register A when the routine returns, to ensure no error has occurred. Error codes and their meanings are listed in section 5.4.

5.3.3 Rewind

The rewind function of the monitor cassette control routine causes the cassette in the drive unit to be rewound to the start (that is, onto the lefthand side of the cassette).

This function is called by loading value 1 into register A and calling the routine at address 18 hex. The application should examine the Zero flag and/or register A when the routine returns, to ensure no error has occurred. Error codes and their meanings are listed in section 5.4.

5.3.4 Move forward

This function of the monitor cassette control routine will move the tape in the cassette forward one or more blocks, that is until the next header is found.

This function is called by loading value 2 into register A and calling the routine at address 18 hex. The number of blocks to be moved forward should be loaded as the value in the cassette file descriptor 'record number', address 604F hex. The application should examine the Zero flag and/or register A when the routine returns, to ensure no error has occurred. Error codes and their meanings are listed in section 5.4.

5.3.5 Move Backward

This function of the monitor cassette control routine will move the tape in the cassette backward one or more blocks.

This function is called by loading value 3 into register A and calling the routine at address 18 hex. The number of blocks to be moved backward should be loaded as the value in the cassette file descriptor 'record number', address 604F hex. The application should examine the Zero flag and/or register A when the routine returns, to ensure no error has occurred. Error codes and their meanings are listed in section 5.4.

5.3.6 Set End-of-File Record

This function writes an end-of-file record on the tape after the last block. This will indicate that there is no information stored after this point on the tape, when the tape is read.

This function is called by loading value 4 into register A and calling the routine at address 18 hex. The application should examine the Zero flag and/or register A when the routine returns, to ensure no error has occurred. Error codes and their meanings are listed in section 5.4.

5.3.7 Write

This function writes information from system RAM onto the cassette. Any area of system RAM may be written; the writing does not destroy the information in RAM.

This function is called by loading value 5 into register A and calling the routine at address 18 hex. The calling program must also fill in the details of the Cassette file descriptor (address 6030 to 604F hex) before calling the routine, since these details tell the routine which area of RAM is to be written to disk. The details from the Cassette file descriptor are copied onto the cassette, to form the cassette record header which precedes the data in the record.

The application should examine the Zero flag and/or register A when the routine returns, to ensure no error has occurred. Error codes and their meanings are listed in section 5.4.

5.3.8 Read

This function reads information from the cassette into system RAM. Any area of system RAM may be written; the writing does not destroy the information on the cassette.

This function is called by loading value 6 into register A and calling the routine at address 18 hex. The routine reads the next record from the cassette. First, the contents of the cassette record header are copied into the cassette file descriptor (address 6030 to 604F hex). Then the data section of the record is read into the area of system RAM specified in the header. When the routine returns, the application can examine the cassette file descriptor to see what action to take.

The application should examine the Zero flag and/or register A when the routine returns, to ensure no error has occurred. Error codes and their meanings are listed in section 5.4.

5.3.9 Status

This function does not affect the cassette, but only returns a value indicating the status of the cassette unit.

This function is called by loading value 7 into register A and calling the routine at address 18 hex. If no tape is present, then when the routine returns, the Zero flag will be set (one); if the tape is write protected, the Carry flag will be set.

The application should also examine the Zero flag and/or register A when the routine returns, to ensure no error has occurred. Error codes and their meanings are listed in section 5.4.

5 4 CASSETTE ROUTINE ERROR CODES

If the monitor cassette routine returns with an error, the Zero flag will be set (one). In this case, the value of register A indicates the specific error. These are as follows:

Value (hex)	Indication
0	No error
41	No tape in cassette unit OR Cassette unit door not closed
42	Beginning of tape (BOT)
43	Checksum error
44	Mark checksum error
45	End of tape (EOT) during write function
46	Tape full; EOT reached during set end-of-file function
47	Cassette is write protected
49	Time-out during rewind
4A	Tape torn
4B	Invalid function code
4C	End of tape (EOT) during read function
4D	No mark found
4E	No record found.

6 PRINTER

6.1 PRINTER CONTROL

The monitor provides the software needed to control the printer. The print routine converts the P2000 internal codes for the characters to be printed into the codes used by the printer and sends these codes to the printer control circuit, where they are converted to serial form and sent to the printer itself.

The print routine starts at address 8, which is a jump to location D00 hex. To use the routine, the application should store the characters to be printed in a contiguous area of system RAM, load the three parameters required, and call the routine. When the routine returns, the application must check the return status to be sure that the printing has been successful.

Note that a second, 'transparent' print routine is described in section 6.5. This may, however, be too simple for the requirements of many applications and printers..

6.1.1 Parameters

The input parameters are as follows:

in register A, the number of lines to be printed
in register C, the maximum line length
in register HL, the address of the first character of the first line.

Each line is assumed to be 80 characters long; that is, the starting address of the second line is 80 bytes after the start of the first line.

The return parameter is the Carry bit; if unset (zero), the printing operation has been completed normally. If at any time during the printing operation, the printer signals 'not ready' for more than 14 seconds, the Carry bit is set (one) and the routine returns.

6.1.2 Operation

When the print routine is called, characters in the P2000 internal code (the Viewdata code) are read consecutively from memory, starting at the address defined by register HL.

For normal character printing, each character is translated from the P2000 internal code into the printer's character code by looking it up in a table; this table must be supplied by the application, and is described in detail in section 6.3. The codes are sent to the printer control circuit and from there to the printer. In the translation process, Viewdata control characters are replaced by spaces (except for those Viewdata codes which appear in the translation table); at the end of each line, a CR code followed by a LF code is given. Control characters and spaces after the last printable character in a line are ignored.

If bit 7 of the P2000 internal code is one, the character is to be underlined. Most printers respond to this indication.

If the line length, specified by register C, is zero, only a CR code followed by an LF code are sent.

If the number of lines to be printed, specified by register A, is zero, all character codes up to the first null character (code 0 hex) are sent to the printer without translation. In this case registers C and HL and the translation table are ignored, and no CR/LF codes are inserted. This feature is used to send strings of control characters to the printer.

6.2 PRINTER-SELECT JUMPER

The table supplied by the application for use by the monitor can be composed of two parts, to allow for the fact that different printers can be connected to the P2000; one part of the table is used for one type of printer, the other part for another type of printer. Generally, the first part of the table is used for a printer which cannot backspace (that is, go back one character and print another character on top); and the second for a printer which can.

The monitor has no way of testing the printer itself to see which type it is, so an indicator is built-in to the hardware - the printer-select jumper. The jumper consists of two pins and a small connector; the connector can be pushed over both pins, to connect them, or pushed onto just one pin, in which case the pins are not connected. The monitor can check whether the pins are connected by reading from any of ports 20 to 2F hex - if bit 2 is set, the pins are connected. When the pins are connected, the first part of the printer translation table is used by the print routine; when they are not connected, the second part is used.

The P2000 is delivered with the jumper connecting both pins.

The printer-select jumper is found on the reverse of the CPU board. Since this board is the lowest in the main body of the P2000, the jumper is accessible from the bottom side of the body. In most P2000 models, there is a small circular cap covering a hole leading to the jumper; in this case, the cap can be removed to expose the jumper. You can then position the jumper over one or both pins as required. In some older models, the cap is not provided; in these cases, the cover of the main body must be opened, for which you are advised to consult a service engineer.

6.3 CHARACTER CODE TABLE

The printer character code table is to be provided by the application for the monitor print routine to use. The application stores the table at any position in system RAM, and gives the start address of the table to the monitor, by storing it in locations 605E and 605F hex (605E is the lower byte). The monitor can then access the table via this address.

The table is divided into two sections; the first section is accessed by the print routine if the print-select jumper is on only one pin, the second if the jumper is connecting both pins (see section 6.2). The two sections are consecutive.

6.3.1 First Section (Non-backspace Printers)

All sections of the printer character code table can be of variable length. The first byte of the first section indicates the number of entries in that section. (If this value is zero, there is no first section, so the next byte is the first byte of the second section.) Each entry in the first section of the table comprises two bytes: the first byte is a P2000 internal character code, the second the byte which is to be sent to the printer in place of that code. The entries are given in any order.

Assuming the printer-select jumper is set for the first part of the table, when the monitor print routine is called, it reads characters one by one from the user area. Each of these characters (in P2000 internal code) is compared with the first byte of all the entries; if the same character code is found, the second byte of that entry is sent to the printer, if not, the character code itself is sent to the printer.

Both sections of the table only contain entries for those characters which must be replaced. This minimises memory space, and allows different character sets to be used by different applications, for different languages for example.

6.3.2 Second Section (Backspace Printers)

The second section of the printer character table starts directly after the first section. Its starting address can therefore be found by reading the value of the first byte of the first section, multiplying by two, and adding this to the start address of the table + 1.

The second section is itself divided into two sub-sections, 'double' and 'single'; the 'double' sub-section comes before the single. The first byte of the 'double' sub-section indicates the number of entries in the sub-section. Each entry comprises three bytes: the first is a byte is a P2000 internal character code, the second and third are two bytes which are to be sent to the printer in place of that code. The entries are given in any order.

The structure of the 'single' sub-section is exactly the same as that of the first section of the table. Either or both of the sub-sections may contain zero entries, in which case the sub-section consists of a single byte of value zero.

When using the second section of the table, the monitor print routine compares the P2000 internal character code with the first byte of each element, first in the 'double' sub-section. If the same character code is found, the second byte of that entry is sent to the printer, followed by a backspace character code, followed by the third byte of that entry. If not, the monitor searches the 'single' sub-section in the same way as in the first section. If the matching code is not found in either sub-section, the character code itself is sent to the printer.

The 'double' sub-section of the table is used to build up special characters by printing two characters on top of each other.

If bit 7 of the second or third byte of an entry is 1, then an escape character (code 1B hex) is sent to the printer before the character itself. This facility is provided to allow the printing of 'escape - 20 hex' and 'escape - 7F' constructions, which have special significance for some printers.

6.4 BAUD RATE

The monitor printer routine can send characters to the printer at several different speeds. This feature is useful because different printers accept characters at different rates. The speed at which characters are sent is known as the baud rate (1 baud = 1 bit/second).

The baud rate of the monitor print routine is controlled by the value of location 6016 hex, as shown in the table below. The application should set the value of this location to specify the baud rate required by the printer in use.

value of location 6016 (hex)	baud rate
0	2400
1	1200
3	600
7	300
F	150
1F	75

The default value assigned at system start-up is 1; that is, 1200 baud.

Normally, at the end of each line, a carriage return and line feed are automatically sent to the printer. However, if bit 7 of location 6016 hex is set (one), this is not done; the application must send carriage return and line feed when required. This feature is useful when printing graphics characters.

6.5 'TRANSPARENT' PRINTING

A second print routine is available within the monitor. This so-called 'transparent' print routine sends one character only to the printer. It performs no translation, that is, it sends the byte value of the character exactly as given, with no carriage return, line feed or any other codes. This basic print facility may be useful in those applications which wish to handle the printer at a very low level; it should however be borne in mind that calling this (or any other) routine very frequently will inevitably slow down execution of the application.

The routine starts at address E5D hex. The character to be sent to the printer should be in register C. On return, the Carry bit is set (one) if an error has occurred.

7 VIDEO

The video unit of the P2000 M model is known as a 'monitor'; one of the two video units of the T model is an 'RGB monitor'. These units have no connection with the basic operating software of the P2000, also known as the 'monitor'. This chapter has been written to avoid as much as possible any confusion between these three.

The video is one of the most complex units of the P2000, and one of the most important from the user's point of view, since through the video the user is informed of the system's status and functioning.

There are a number of major differences between the structure and function of the video on the M model and the T model. For this reason, chapter 7 has been split into two parallel chapters - the first deals with the M model, the second with the T model. Sections with the same number describe the equivalent aspect of the video; those concerning the M model end in the suffix 'M', those describing the T model end with 'T'.

For example, section 7.2M describes the video memory of the M model, and section 7.2T the video memory of the T model.

7.1M MONITOR

The M model video monitor is housed in the same unit as the one or two disk drive units, although there is no connection between them. The monitor is connected to the P2000 main unit through the lead at the rear; the control circuitry on the video board sends the video signal to the video monitor down this lead.

The 12" (30 cm) 90° screen of the video monitor displays 24 lines of 80 characters on green phosphor. Each character is represented by a pattern of green dots on the black background. The character is the smallest unit of the screen which is addressable by the CPU; the dots and dot patterns are controlled by the character generator circuit, which is completely independant.

The monitor unit has its own 12 V d.c. 1 A power supply from a transformer drawing current from the mains supply, not via the main (keyboard and CPU) unit of the P2000. The connector at the rear of the main unit takes the 5 pin plug from the monitor to allow the monitor to receive the video signal from the video control circuits on the video board.

Note that under some application software - the Word Processing package, for example - line lengths of more than 80 characters may be used. However, no more than 80 characters are ever present on the screen at one moment; the horizontal movement of longer lines across the screen is performed by the application.

7.2M VIDEO MEMORY

The section of RAM reserved for the video comprises 4 k bytes, extending from address 5000 to 5FFF hex. This video memory is divided into the following structure:

address (hex)	
5000 - 577F	video page 1 characters, 1920 bytes
5780 - 57FF	scrolling area for page 1, 128 bytes
5800 - 5F7F	video page 2 attributes, 1920 half-bytes
5F80 - 5FFF	scrolling area for page 2, 128 half-bytes

The video control circuitry operates on the basis of a display made up of 24 lines each of 80 characters, a total of 1920 characters. The characters to be displayed are stored in the bytes of video page 1, starting from the leftmost character of the first line. Video page 2 consists of 1920 half-bytes (bits 0 to 3), each of which corresponds to a byte in page 1. The values of these bits specify the manner in which the corresponding character is to be displayed. Each half-byte has a separate address as if it were a complete byte, but bits 4 to 7 do not exist.

The scrolling areas of pages 1 and 2 correspond to one another in the same way. Scrolling (the vertical movement of text up the screen as new lines are added at the bottom) takes place under the control of the application; the scrolling areas can be used by the application to hold the characters and attributes for the new line which is to appear in the scrolling process.

The horizontal scrolling port (port numbers 30 to 3F hex), used on the T model has no function on the M model. Any value output to this port is ignored.

7.2.1M Video Page 1

Video page 1 contains the codes representing the characters to be displayed; these are in the P2000 internal code, also known as the Viewdata code, since it is the same as the internationally-agreed code used for viewdata (teletext) transmission. This code is listed in Appendix D.

This coding uses only the seven least significant bits (0 to 6) of each byte to represent a character; the most significant bit (bit 7) of each byte in video page 1 indicates how the character should be displayed. If bit 7 is zero, the character is displayed normally; if it is one, the character is displayed with a flashing underline.

Since the P2000 internal code is used in all data in RAM, as well as for characters in the video page, monitor and application software can treat all RAM data, including characters to be displayed, in the same way.

7.2.2M Video Page 2

Video page 2 contains the attributes specifying the way in which the characters are to be displayed. Each element of video page 2 comprises 4 bits, with the following indications:

bit

0 Text/graphics mode
If this bit is zero, the corresponding code in video page 1 is interpreted and displayed as a text character.
If one, the code is interpreted and displayed as a graphics character (these are described in section 7.2.3M).

1 Underline
If this bit is zero, the character is displayed alone.
If one, the character is displayed underlined.

2 Flash
If this bit is zero, the character is displayed steady.
If one, the character is displayed flashing.

3 Invert
If this bit is zero, the character is displayed as normal - green on black background.
If one, the character is displayed inverted - black on green background.

Any combination of these bit values is possible for each character.

7.2.3M Text and Graphics Characters

The P2000 internal character coding consists, in fact, of two character sets, known as 'text' and 'graphics' characters. For each character code stored in video page 1, either the equivalent text character or the equivalent graphics character will be displayed, depending on the value of bit 0 of the corresponding half-byte in video page 2.

The text characters comprise the alphabet, digits and punctuation symbols in a coding system very similar to the well-known ASCII. The first 32 characters are non-printable control characters; these have no meaning for the M model, are ignored and not displayed.

Each graphics character has the form of a rectangle made up of six units, arranged two across by three down. The units may be 'on' or 'off', each combination of on and off units being a different graphics character. An on unit is displayed green, an off unit is displayed black. Using groups of graphics characters, pictures can be drawn on the screen. Graphics characters can also be underlined, flashing, or inverted according to bits 2, 3 and 4 of the attributes.

The graphics and text characters sets overlap to some extent - the graphics character set includes the full upper-case (capital) alphabet from the text set, and the first 32 graphics characters are the same control characters as are used in the text characters. The M model has seven additional symbols not present in the T model - these are part of the graphics character set:

symbol	code value (hex)
	A
	B
	43
	44
	4C
	50
	54

The complete text and graphics character sets are listed in Appendix D.

7.3M APPLICATION VIDEO SOFTWARE

The information to be displayed on the video units is totally determined by the application software. The application stores the codes for the characters to be displayed in the appropriate locations in RAM video page 1, and the attributes required in the corresponding half-bytes in video page 2. Since the P2000 internal character code is used for the screen, as far as the application is concerned, accessing the video memory in order to store a character is the same as accessing any other memory location.

Typically, an application will spend a certain amount of time reading a key from the keyboard queue, displaying it on the screen, moving the cursor one character right and scrolling vertically if necessary. Pressing a key on the keyboard does not automatically (that is, through hardware action) result in that character appearing on the screen - this operation must be performed by the application, it is not done by the monitor. Similarly, for example, if the cursor is to be displayed inverted, the application must specify this by setting bit 3 of the appropriate byte in video page 2 to one.

Similarly, scrolling is also the responsibility of the application. The new characters to be displayed when scrolling occurs may be built up in the scrolling area of video page 1, and their attributes in the equivalent part of video page 2; or they may be stored in any other area of RAM as wished. When scrolling is to take place, these characters must be moved into the correct area of video pages 1 and 2, and the existing contents of these areas may either be moved, or simply overwritten by the new values. The Z80 instruction LDI is normally used to move areas of data values in this way.

Since the video control circuitry is constantly reading the video memory in order to generate the image to be displayed on the video, changes made to the video memory by the application are shown immediately on the screen. Thus applications may, for example, modify the values in video page 2 only, with the effect that the same character codes (in page 1) are used, but with different attributes.

Although the application treats the video memory like any other area of RAM, access by the CPU to the video memory is, in fact, restricted. This is because the video control circuitry is constantly reading the video memory to be able to generate the picture on the screen, and the CPU must be prevented from accessing the memory at the same time. Thus when the video controller starts access to the memory, it disables the CPU's ability to access the video RAM; once the video controller is finished with the current memory access, the CPU is allowed to proceed. The software is not aware of this temporary waiting period.

During access to cassette or disk, this ability to prevent CPU access is disabled by setting port number 7x (that is, any of ports 70 to 7F hex). This is necessary to preserve the timing constraints on the disk and tape access, which would otherwise be altered by the generation of the CPU-disable interrupt.

7.4M MONITOR VIDEO SOFTWARE

There is only one monitor routine dealing with the video, that for clearing an area of the screen; all other video functions - vertical scrolling, moving the cursor and so on - are performed directly by the application itself.

This monitor routine is called to clear a number of lines of the video screen; this is done by writing spaces into the appropriate section of RAM video page 1. The routine starts at address 35 hex, and takes two parameters:

in register A, the number of lines to be cleared
in register HL, the address within RAM video page 1 of the first character of the first line.

The address given in HL need not be the start of a line on the screen, it may be midway along a line; the routine assumes a line length of 80 characters, and will therefore clear 80 times register A consecutive characters starting from the point given.

There are no return parameters.

7.5M VIDEO CONTROL AND FUNCTION

The process by which the character codes in the video RAM are transferred to character patterns on the screen is performed by the video control circuitry. This circuitry operates independantly of the rest of the system, so that interrupts or monitor program activity do not affect the appearance of the display on the screen. A block diagram of the control circuits is shown in Appendix A.

Each character on the screen is made up of a matrix of dots - this matrix is 8 dots wide and 12 dots high. The pattern of dots which makes each text character is predetermined and stored in the character generator circuit, which can be varied from model to model to allow for different national alphabets. For the graphics characters, each of the six units in the character is a 4 by 4 square of dots.

The counting and timing circuits perform the important synchronisation function for the video circuitry. The dot counter counts from 0 to 7, representing the horizontal dots of each character, whilst the line counter counts from 0 to 11, for the vertical dots. The character counter circuit counts up to 96, representing the number of characters in a row on the screen, allowing for the time delay between generation of a character pattern and its display on the screen. The character decoder interprets signals from the character counter which are used to reset the line counter and by the counter control. The row counter circuit counts up to 26, representing the number of rows of characters on the screen, again allowing for delay; its signals are similarly interpreted for use by the counter control.

Under control from these counting circuits, the counter control generates an address within the video page memory; with this address, the refresh control multiplexer accesses the specified byte of video page 1 and, simultaneously, the corresponding half-byte of page 2. (The CPU may only access page 1 or page 2, never both.) The bit-values from page 1, seven of which indicate the character code and one of which indicates whether a cursor is present or not, are sent to the row memory; the 4 bit values from page 2 are sent to the attribute row memory. The row memory and attribute memory each hold the information for 80 characters, that is, one screen row at a time.

A signal from the dot counter circuit indicates the start of a new character (that is, at its leftmost dot), which causes the character code for the next character to be given to the character generator. This interprets the code into the required dot pattern, and passes that to the video shift register, where the dot pattern is serialised, to be mixed with the attribute information before being sent to the monitor for display. This process is repeated for each of the 12 lines of dots in each character; then a new row of 80 characters is read from video memory. RAM access is thus performed only once at the start of each row, when initiated by the signal from the row decoder, which reduces the number of occasions on which the CPU will have to be disabled from video memory access.

The process repeates for each row of characters on the screen, until a signal from the row decoder indicates that the last row of the screen is being written, when the whole process starts again with the top row.

7.1T TELEVISION AND RGB MONITOR

One of two different video units may be connected to the P2000 T model - a normal domestic television (colour or black-and-white), or an RGB monitor (as used in closed circuit systems). This manual does not discuss either the televison or RGB monitor in detail since they do not form part of the P2000 range of equipment.

At the rear of the P2000 main unit are the two sockets for connection of the TV or RGB monitor. These allow the connection between the video unit and the control circuits on the CPU board. Only one video unit may be connected at a time.

The same video signals are sent from the two sockets, except that those from the TV socket have been VHF modulated. This means that the signal is given to the television as if it came from an aerial - the television demodulates the signal itself to form the picture. Signals for the RGB monitor are not modulated because RGB monitors do not have demodulators, since they are not designed to receive signals from an aerial.

The characters displayed on the TV or RGB monitor are from the standard character set used internationally for viewdata (teletext) transmission. Each character is made up of a number of dots on the screen, under the control of a viewdata character generator as used in televisions fitted for viewdata reception. A number of possibilities are available under the viewdata standard, including different colours of letters and background, flashing letters and letters of double height.

Although the characters produced on the TV are the viewdata characters as used in viewdata TVs, your TV does not have to be one of those capable of receiving viewdata in order to be able to be connected to the P2000. Any ordinary TV will do.

7.2T VIDEO MEMORY

The section of RAM reserved for the video comprises 4 k that extends from address 5000 to 5FFF hex. The video memory is divided into the following structure:

address (hex)	
5000 - 577F	video page 1 characters, 1920 bytes
5780 - 57FF	scrolling area for page 1, 128 bytes
5800 - 5F7F	video page 2 attributes, 1920 half-bytes
5F80 - 5FFF	scrolling area for page 2, 128 half-bytes

All the video circuitry functions on the basis of a display made up of 24 lines each of 80 characters, a total of 1920 characters. The characters to be displayed are stored in the bytes of video page 1, starting from the leftmost character of the first line. Video page 2 is not used by the T model and does not in fact exist; however, the address space is reserved as if page 2 does exist to ensure software compatability between T and M models.

Scrolling (the vertical movement of text up the screen as new lines are added the bottom) takes place under the control of the application; the scrolling area of page 1 can be used by the application to hold the characters for the new line which is to appear in the scrolling process. The scrolling area of page 2 is not present on the T model.

Although each screen line is considered as 80 characters wide, a line on the TV screen is only 40 characters wide, so that half the total display is shown at any one time. The video horizontal scroll port (port numbers 30 to 3F hex) specifies which

character is displayed at the leftmost end of each line; an application may output any value between 1 and 80 to this port to control the horizontal scrolling of text across the screen. These values between 1 and 80 use only the 7 least significant bits of the port.

The most significant bit of the horizontal scroll port has another function. When this bit is one, the video display is disabled. For normal operation, this bit should be zero.

7.2.1T Video Page 1

Video page 1 contains the codes representing the characters to be displayed; these are in the P2000 internal code, also known as the Viewdata code, since it is the same as the internationally-agreed code used for viewdata (teletext) transmission. This code is listed in Appendix D.

Since the P2000 internal code is used for all data in RAM, as well as for characters in the video page, monitor and application software can treat all RAM data, including characters to be displayed, in the same way.

7.2.2T Video Page 2

In the M model, video page 2 contains information specifying the way in which the characters in page 1 are to be displayed. This display information is an integral part of the T model character set, so page 2 is not required. Page 2 is not in fact present, although the address space is reserved to ensure software compatability between T and M models. Addresses 5800 to 5FFF hex are therefore permitted addresses but do not access any memory locations.

7.2.3T Text and Graphics Characters

The P2000 internal character coding consists, in fact, of two character sets, known as 'text' and 'graphics' characters. For each character code stored in video page 1, the equivalent text or graphics character will be displayed depending on the value of bit 0 of the corresponding half-byte in video page 2.

Video page 2 does not exist on the T model, the address space is reserved as in the M model to ensure software compatibility between the M and T models.

The text characters comprise the alphabet, digits and punctuation symbols in a coding system very similar to the well-known ASCII. The first 32 characters (with code values between 0 and 1F hex) are not displayed as characters on the screen, but serve to specify the size and colour of characters, and the colour of the background on which they are to be displayed.

Each graphics character has the form of a rectangle made up of six units, arranged two by three. The units may be 'on' or 'off', each combination of on and off units being a different graphics character. The colour of an on unit is controlled by the preceding control characters, and off units are displayed the colour of the background. Using groups of graphics characters, pictures can be drawn on the screen.

The first 32 graphics characters are the same control characters as are used in the text characters. The complete characters sets are listed in Appendix D.

7.3T APPLICATION VIDEO SOFTWARE

The information to be displayed on the video units is totally determined by the application software. The application stores the codes for the characters to be displayed in the appropriate locations in RAM video page 1, and the attributes required in the corresponding half-bytes in video page 2. Since the P2000 internal character code is used for the screen, as far as the application is concerned, accessing the video memory in order to store a character is the same as accessing any other memory location.

Typically, an application will spend a certain amount of time reading a key from the keyboard queue, displaying it on the screen, moving the cursor one character right and scrolling vertically if necessary. Pressing a key on the keyboard does not automatically (that is, through hardware action) result in that character appearing on the screen - this operation must be performed by the application, it is not done by the monitor. Similarly, for example, if the cursor is to be displayed inverted, the application must specify this by setting bit 7 of the appropriate byte in video page 1 to one.

Since the video control circuitry is constantly reading the video memory in order to generate the image to be displayed on the video, changes made to the video memory by the application are shown immediately on the screen. Thus applications may, for example, modify the values in video page 2 only, with the effect that the same character codes (in page 1) are used, but with different attributes.

Similarly, scrolling is also the responsibility of the application. The new characters to be displayed when scrolling occurs may be built up in the scrolling area of video page 1, or they may be stored in any other area of RAM as wished. When scrolling is to take place, these characters must be moved into the correct area of the video memory and the existing contents of this area may either be moved, or simply overwritten by the new values. The Z80 instruction LDI is normally used to move areas of data values in this way.

Under some application software - such as the Word Processing package - line lengths of more than 80 characters may be used. The physical size of the screen prevents more than 40 characters ever being present on the screen at one moment, and the video memory can only hold 80 by 24 characters. In such cases, therefore, the application must use another area of RAM to hold the excess characters, and must transfer these into video memory at the correct moment (when the cursor moves to the right) to provide the horizontal movement of longer lines across the screen. The horizontal shift register (described in section 7.2.1) provides horizontal movement only within the 80 by 24 characters of the video memory, and there are no built-in facilities within the video controller for larger movements.

Although the application treats the video memory like any other area of RAM, access by the CPU to the video memory is, in fact, restricted. This is because the video control circuitry is constantly reading the video memory to be able to generate the picture on the screen, and the CPU must be prevented from accessing the memory at the same time. Thus when the video controller starts access to the memory, it disables the CPU's ability to access the video RAM; once the video controller is finished with the current memory access, the CPU is allowed to proceed. The software is not aware of this temporary waiting period.

During access to cassette or disk, this ability to prevent CPU access is disabled by setting port number 7x (that is, any of ports 70 to 7F hex). This is necessary to preserve the timing constraints on the disk and tape access, which would otherwise be altered by the generation of the CPU-disable interrupt.

7.4T MONITOR VIDEO SOFTWARE

There is only one monitor routine dealing with the video, that for clearing an area of the screen; all other video functions - vertical scrolling, moving the cursor and so on - are performed directly by the application itself.

This monitor routine is called to clear a number of lines of the video screen; this is done by writing spaces into the appropriate section of RAM video page 1. The routine starts at address 35 hex, and takes two parameters:

in register A, the number of lines to be cleared
in register HL, the address within RAM video page 1 where the first line starts.

The address given in HL need not be the start of a line on the screen, it may be halfway along a line; the routine assumes a line length of 80 characters, and will therefore clear 80 times register A consecutive characters starting from the point given.

There are no return parameters.

7.5T VIDEO CONTROL AND FUNCTION

The timing chain performs the synchronisation function for the video circuitry. The 6 MHz signal is used by the character generator to time the generation of the dots which make up the characters on the screen. The refresh control circuit uses the timing signal to count from 1 to 80, representing the character within the row on the screen; and a further counter provides an indication of the row, counting to 24.

The value from the horizontal scrolling port is stored in the scroll register and indicates the address of the memory location which is to be the top lefthand character on the screen. This value is read by the refresh control at the start of each row, and its counting from 1 to 80 modifies this address to provide the 80 consecutive addresses for each row. The addressed character is read from the video RAM and the 7 least significant bit-values passed to the character generator, which interprets them into the appropriate signals to generate a character, be it printable or a control character. The most significant bit (bit 7) is passed to the inverse video control which will invert the signals to generate an inverted image when this bit is one.

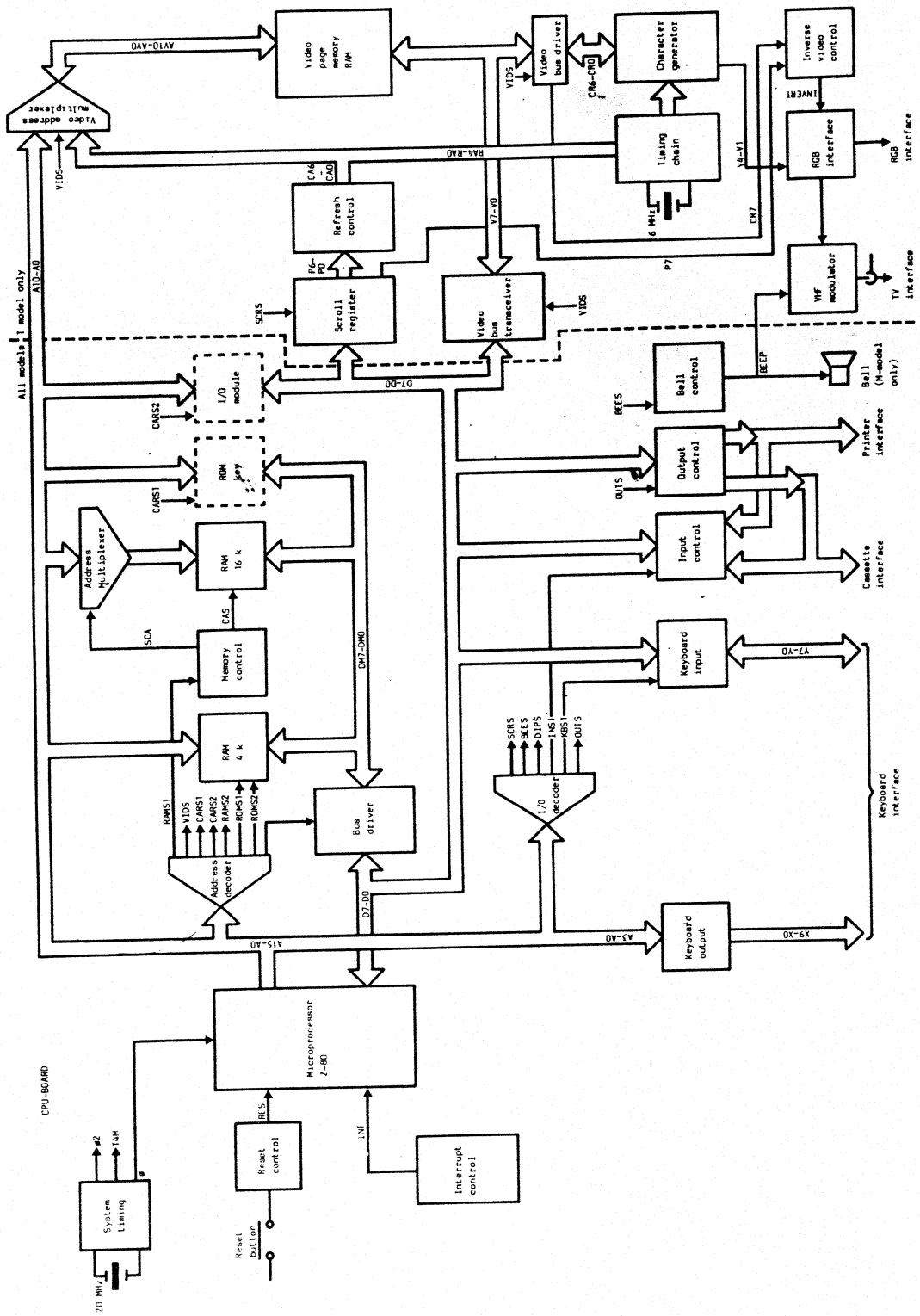
Four signals come from the character generator - dot pulses, red, green and blue - these and the inverse video control are passed to the RGB interface and can be sent directly out to an RGB monitor. The signals from the character generator are also passed to the VHF modulator, where they are modulated so that, to a television, they appear to be signals from an aerial; the TV will demodulate the signals in the normal way to form the picture.

APPENDIX A MACHINE ARCHITECTURE

CPU BOARD SIGNAL LISTING

ϕ	2.5 MHz CPU clock
ϕ2	5 MHz general clock
A15-A0	Address bus; 16-bit, used for memory and I/O port addressing.
AV10-AV0	Video address bus; offers either A10-A0 CPU address or the address formed by the binary value of CA6-CA0 character address and RA4-RA0 row address.
BEEP	Sound bell; activates bell (M model) or audio signal (T model).
BEES	Bell select; selects bell.
CA6-CA0	T Character address; offers the character address during screen refresh. Receives 40 clock pulses every scan line from the read address clock; preset before every line with the start character from the scroll register.
CARS1	ROM key (cartridge) select; active for ROM key access.
CAS	Column address strobe; active one clock pulse after SCA select column address, used to strobe column address into system RAM.
CR6-CR0	Character; offers video memory signals to character generator.
CR7	Character; offers most significant bit of video memory signals to video inverse control.
D7-D0	Data bus; 8-bit, bi-directional.
DIPS	Disable selection; active when CPU attempts to access video memory at the same time as video controller (M model).
DM7-DM0	Memory data bus; offered to D7-D0 data bus when MBEN2-N is active.

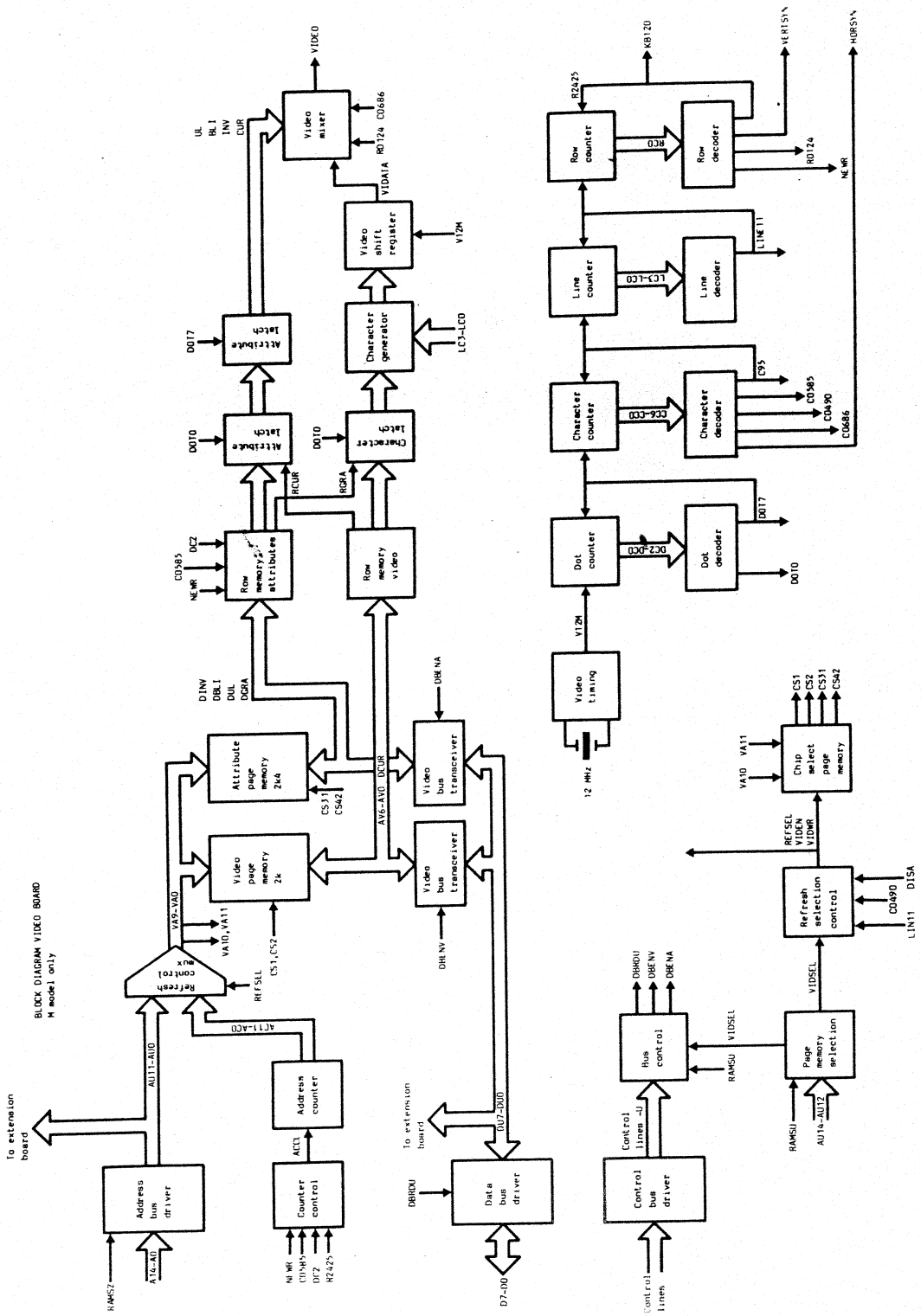
INS1	Input select; together with RD, strobes input from port onto D7-D0 data bus.
INT	Interrupt; from interrupt control or from CTC if extension board present.
INVERT	Invert; exclusive-or input to video data, used for cursor control.
KBS1	Keyboard select; together with RD read, strobes output from keyboard onto D7-D0 data bus.
P6-P0 T	Preset character count value; indicates the first character of the row in video RAM to be displayed.
P7 T	Preset count 7; enables display of data and cursor.
RA4-RA0	RAM address; provides RAM row address, incremented every 10 scan lines (every 20 when displaying double-size characters).
RAMS1	RAM select 1; active for system RAM access.
RAMS2	RAM select 2; active for extension RAM access.
RES	Reset signal to all boards
ROMS1	ROM select 1; active for accesses to the first 2 k of monitor ROM.
ROMS2	ROM select 2; active for accesses to the second 2 k of monitor ROM.
SCA	Select column address; active one clock pulse after MREQ memory request, used to select the high order part of the address to be offered to the memory.
SCRS	Scroll register selection; selects scroll register for video (T model).
T4M	4 MHz for disk control on extension board
V1	Video output 1; monochrome.
V2	Video output 2; blue.
V3	Video output 3; green.
V4	Video output 4; red.
V7-V0	Video data bus; coupled to D7-D0 data bus for accesses to video RAM. High impedance during refreshing.
VIDS	Video RAM select; active for video RAM access (T model).
X9-X0	X-matrix lines; output of ten scan lines to keyboard matrix. One out of the ten is selected by decoding on A3-A0; if keyboard enable is active, all ten are selected together.
Y7-Y0	Y-matrix lines; input from keyboard to D7-D0 data bus.


Video page memory RAM
Video bus driver
Character generator
Inverse video control
Video address multiplexer
A10-A0
VIDS
CR6-CR0
INVERT
Timing chain
6 MHz
RGB interface
RGB interface
RA4-RA0
CA6
CA0
Refresh control
V7-V0
V6-V1
CR7
P7
P6-P0
VHF modulator
TV interface
SCRS
Scroll register
Video bus transceiver
VIDS
All models 1 model only
A10-A0
D7-D0
Bell control
BEEP
Bell (M-model only)
I/O module
CARS2
BEES
ROM
CARS1
Output control
Printer interface
OUTS
Address Multiplexer
RAM 16 k
Input control
Cassette interface
CAS
SCA
Memory control
DM7-DM0
Keyboard input
X7-X0
RAM 4 k
SCRS
BEES
DIPS
INKS
KBS1
OUTS
I/O decoder
Keyboard interface
RAMS1
VIDS
CARS1
CARS2
RAMS2
ROMS1
ROMS2
Address decoder
Bus driver
D7-D0
A15-A0
A3-A0
X9-X0
Keyboard output
Microprocessor Z-80
CPU-BOARD
Reset control
RES
INT
Interrupt control
System timing
Ø2
14M
20 MHz
Reset button

VIDEO BOARD SIGNAL LISTING (M MODEL ONLY)

A15-A0	Address bus; 16-bit, used for memory and I/O port addressing.
AC11-AC0	Screen refresh counter (0 to 1919, total number of characters).
ACCL	Address counter clock; 80 times every row.
AU14-AU0	Address bus; to upper boards.
AV6-AV0	Data character to Row memory.
BLI	Blinking (flashing) attribute
CC6-CC0	Binary counter (divided by 96).
CS1,CS2	Chip select 1 and 2; selects the first and second 1 k of video RAM (video page 1, characters - addresses 5000 to 57FF hex).
CS31,CS42	Chip select 3 and 4; selects the third and fourth 1 k of video RAM (video page 2, attributes - addresses 5800 to 5FFF hex).
CUR	Cursor attribute
C0490	Character 4 to 90.
C0585	Character 5 to 85 (enable address counter clock).
C0686	Character 6 to 86 (enable video output).
C95	Character 95, last character.
D7-D0	Data bus; 8-bit, bi-directional.
DBENA	Data bus enable attributes (page 2 video memory).
DBENV	Data bus enable video characters (page 1 video memory).
DCUR	Cursor attribute; from bit 7 of character code.
DBLI	Blinking (flashing) attribute.
DBRDU	Data bus read enable upper boards; active during memory read from video RAM or extension RAM, I/O read on extension board and interrupt handling on extension board.
DCUR	Cursor attribute; from bit 7 of character code.
DC2-DC0	Binary counter (divided by 8).
DGRA	Graphic attribute.
DINV	Inverted attribute.
DISA	Disable access.

DOT0	Dot 0; first dot of character.
DOT7	Dot 7; last dot of character.
DUL	Underline attribute.
DU7-DU0	Data bus to upper boards.
HORSYN	Horizontal synchronisation; signal to monitor unit.
INV	Inverted attribute.
KBI20	Keyboard interrupt; generated every 20 ms by DEW data entry window (T model) or the signal R2425 Row 24/25 (M model video board), initiates monitor keyboard scan.
LC3-LC0	Binary counter (divided by 12); part of the addressing for Character generator ROM.
LINE11	Line 11; last line in a character.
L3-L0	Binary counter (divided by 12); part of the addressing for Character.
NEWR	New row; enables loading of new information into Row memory for screen refresh.
RAMS2	RAM select 2; active for extension RAM access.
RAMSU	RAM select upper boards.
RCUR	Cursor attribute
RC4-RC0	Binary counter (divided by 26).
RGRA	Graphic attribute
REFSEL	Refresh selection; active for video memory access by video controller, that is during scan line 11 of each character row.
R0124	Row 1 to 24; active during these displayable rows.
R2425	Row 24 and 25; active during these last two rows.
UL	Underline attribute
VA11-VA0	Address bus to page memories.
VERTSYN	Vertical synchronisation; signal to monitor.
VESY	Vertical synchronisation; signal to monitor.
VIDATA	Video data; serial dot information.
VIDEO	Composite video signal to monitor unit.
VIDEN	Video memory enable.
VIDSEL	Video memory selection; active for video memory access by CPU.
VIWR	Video memory write enable.
V12M	12 MHz video timing dot clock.
WE4-WE1	Write enable for chips 1 to 4.


BLOCK DIAGRAM VIDEO BOARD
M model only
To extension board
Address bus driver
Address counter
Counter control
Refresh mux control
Attribute page memory 2k
Video page memory 2k
Video bus transceiver
Video bus transceiver
Data bus driver
Row memory attributes
Row memory video
Attribute latch
Character latch
Character generator
Video shift register
Video mixer
Row counter
Row decoder
Line counter
Line decoder
Character counter
Character decoder
Dot counter
Dot decoder
Video timing
12 MHz
Chip select page memory
Refresh selection control
Page memory selection
Bus control
Control bus driver
VIDEO
KB120
VERSYN
HORSYN
RAMSU
AU16-AU17
Control lines

EXTENSION BOARD SIGNAL LISTING

AE7-AE0 Address bus; from CPU board (T model) or video board (M model).

AM6-AM0 Memory address bus; multiplexing of either row address (AU6-AU0) or column address (A12-A7, SEL8K).

BS0 Bank select 0; active during selection of bank 1 (addresses A000 to DFFF hex).

BS1 Bank select 1; active during selection of bank 1 (addresses E000 to FFFF hex).

BS10 Bank select 0 and 1; active during selection of extension memory bank 0 or 1 (addresses A000 to FFFF hex).

CAS1,CAS0 Columns address select bank 1; clocks information from extension memory bus into column address register.

CTCSEL Select Z80 CTC timer (ports 88 to 8B hex).

D7-D0 Data bus; 8-bit, bi-directional.

DE7-DE0 data bus on extension board; controlled by RDE.

DIRECT Direction for track movement; in or out with regard to centre.

DISA Disable access.

DM7-DM0 Memory data bus; bidirectional.

DRISEL3-DRISEL1 Select drive 3, 2 or 1.

FCDK Floppy controller data acknowledge; active during data read/write.

FCDR Floppy controller data request; active when controller requests data transfer.

FCINT Floppy controller interrupt; raised by the controller when it requires a non-service interrupt.

FCS Floppy controller select.A15-A0 Address bus; 16-bit, used for memory and I/O port addressing.

FDCS Floppy disk chip select (output command).

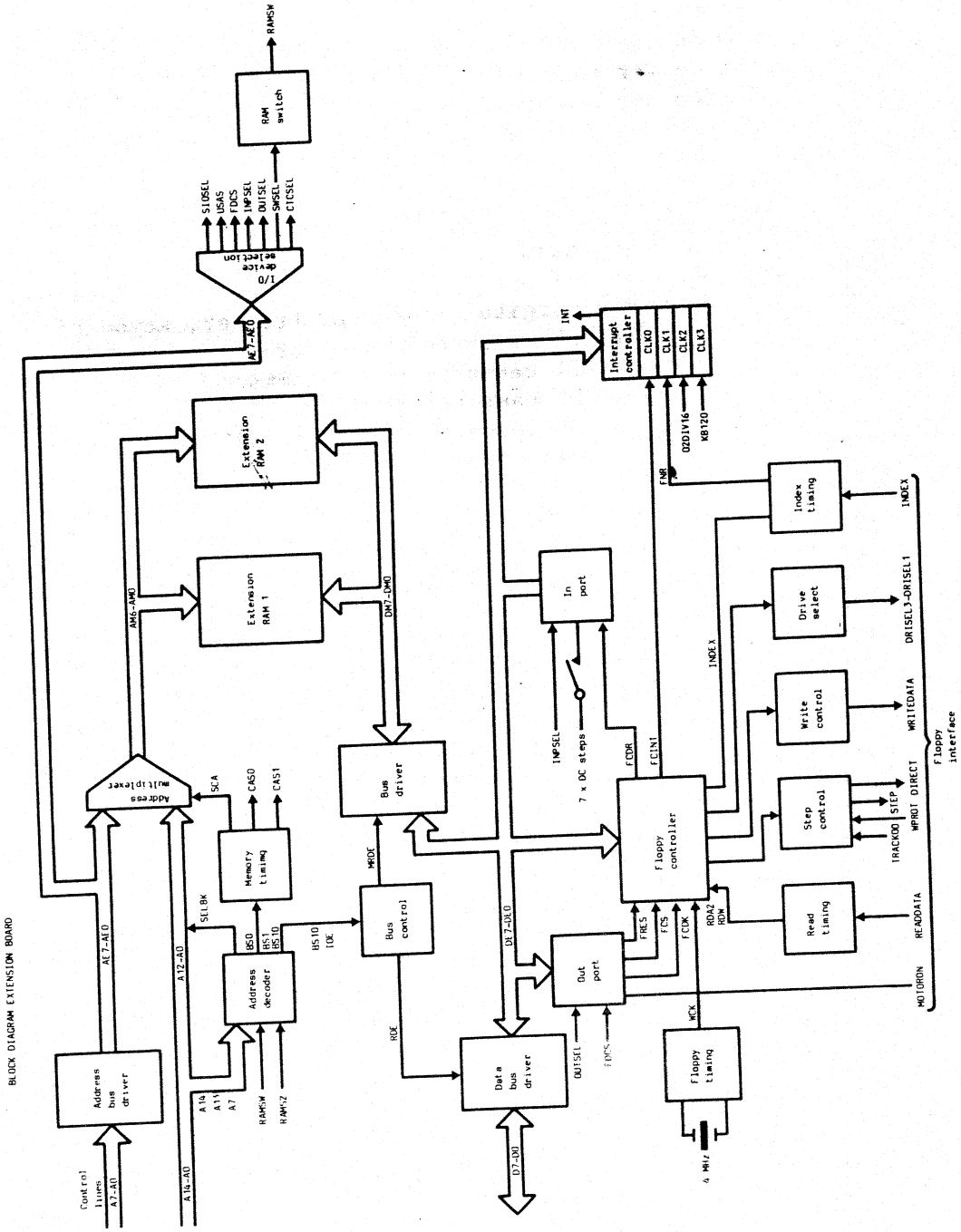
FDCSEL Select UPD 765 FDC floppy controller (ports 8C to 8F hex).

FNR Floppy not ready; active if, after 'head load', no index pulses are received from disk.

FRES Floppy controller reset.

INDEX Index signal; received from disk drive each time sector hole passes detector.

INPSEL	Input selection; active to strobe contents of input port onto data bus.
IOE	Input/output extension board; active when A7=1, that is ports 80 to FF hex.
INT	Interrupt to CPU; active when the floppy controller receives a KBIZO, FNR or FCINT signal.
KBI20	Keyboard interrupt; generated every 20 ms by data entry window (T model) or the signal R2425 Row 24/25 (M model video board); initiates monitor keyboard scan.
MOTORON	Motor on to disk drive.
MRDE	Memory bus read enable; active during read from extension memory, determines direction of data flow.
OUTSEL	Select output port; data acknowledge, motor on, etc. (port 90).
RAMS2	RAM select 2; active for extension RAM access.
RAMSW	Ram switch; for bank 1, 2 and 8 k selection.
RDA2	Read data; synchronised with WCK.
RDE	Data bus read enable; active during read from extension memory or extension input ports, determines direction of data flow.
RDW	Read data window.
READATA	Data read from disk (serial).
SCA	Select column address; active one clock pulse after MREQ.
SEL8K	Select 8 k; active during selection of bank 1 when signal RAMSWITCH is active. Selects upper 8 k of bank 1.
SIOSEL	Select Z80-S10 communications control (ports 80 to 83).
STEP	Step one track; causes disk drive head to move one track.
SWSEL	Select RAM switch.
TRACK00	Track 0; indicates that head of disk drive is positioned on track 0.
USAS	Select 8215 USART communications control (ports 84 to 87).
WCK	Timing signal; 4 MHz.
WPROT	Write protect; indicates that the disk is write protected.
WRITEDATA	Data written to disk (serial).


BLOCK DIAGRAM EXTENSION BOARD
RAM switch
RAMSW
S1DSEL
USAS
FDCS
INPSEL
DUTSEL
SMSEL
CTCSEL
I/O device selection
A7-A10
Extension RAM 2
Extension RAM 1
AM6-AM0
DM7-DM0
Address multiplexer
SCA
CAS0
CAS1
Memory timing
SELBK
BS0
BS1
BS10
IOE
Address decoder
A12-A0
A7-A10
Address bus driver
Control lines
A7-A0
A14
A11
A7
RAMSW
RAMA2
Bus driver
MODE
Bus control
RDE
Interrupt controller
INT
CLK0
CLK1
CLK2
CLK3
Q2D1Y16
KB120
FNR
In port
INPSEL
7 x DC steps
FCDR
FCIN1
Floppy controller
FRES
FCS
FCDK
RDA2
RDA1
Out port
DUTSEL
FDRS
DM7-DM0
Data bus driver
D7-D0
WCK
Floppy timing
4 MHz
Index timing
INDEX
Drive select
DRISEL3-DRISEL1
Write control
WRITEDATA
Step control
STEP
WPROT DIRECT
TRACK00
Read timing
READDATA
MOTORON
Floppy interface

APPENDIX B PORTS

The following are all the ports used in the P2000. Where 'x' is shown in the port number, all ports in that range have the same effect - for example, ports 40 to 4F hex are assigned to the input/output module.

Port	I/O	Device
0x	I	Keyboard
1x	O	Mini-digital cassette, Printer, Keyboard bit D0 cassette write data D1 cassette write command D2 cassette rewind D3 cassette forward wind D4 D5 D6 keyboard reset interrupt D7 printer output to printer connector pin 3
2x	I	Mini-digital cassette, Printer bit D0 printer input from printer connector pin 2 D1 printer ready from printer connector pin 20 D2 printer-select jumper on CPU board D3 cassette write enable D4 cassette in position D5 cassette begin/end of tape D6 cassette read clock D7 cassette read data
3x	O	Video bit D0-D6 horizontal scroll for T model; number of leftmost column on screen D7 video disable

(continued overleaf)

Port	I/O	Device
4x	I/O	I/O module even port numbers (that is, 40, 42, 44 etc.) - data in/out odd port numbers (41, 43, 45 etc.) - control/status
5x	O	Bell
6x	I	I/O module even port numbers - dip-switch 1; transmission rate odd port numbers - dip-switch 2
7x	O	Video video wait lock; disables interrupt from video to CPU while disk/cassette transfer in process
88	I/O	Counter timer circuit channel 0
89	I/O	Counter timer circuit channel 1
8A	I/O	Counter timer circuit channel 2
8B	I/O	Counter timer circuit channel 3
8C-8D	I/O	Flexible disk controller
90-93	I/O	Flexible disk controller

APPENDIX C I/O SLOT PINNING

The I/O slot is pinned according to the convention below; however, not all modules make use of all the pins listed.

Pin	
1	Protective ground
2	Transmit data (103)
3	Receive data (104)
4	Request to send (105)
5	Clear to send (106)
6	Data set ready (107)
7	Signal ground (102)
8	Carrier detect (109)
9	Select standby (116)
20	Data terminal ready (108/1, 108/2)
22	Calling indicator (125)
23	data signalling rate selector (111)
24	Terminal equipment transmitter signal element timing (113)

APPENDIX D CHARACTER (VIEWDATA) CODE

binary	dec-imal	hexa-deci-mal	text character	graphics character
0000000	0	0	nul	nul
0000001	1	1	alpha red	alpha red
0000010	2	2	alpha green	alpha green
0000011	3	3	alpha yellow	alpha yellow
0000100	4	4	alpha blue	alpha blue
0000101	5	5	alpha magenta	alpha magenta
0000110	6	6	alpha cyan	alpha cyan
0000111	7	7	alpha white*	alpha white*
0001000	8	8	flash	flash
0001001	9	9	steady*	steady*
0001010	10	A	end box* (`)	end box* [ç]
0001011	11	B	start box (´)	start box ["]
0001100	12	C	normal height*	normal height*
0001101	13	D	double height	double height
0001110	14	E	so (^)	so
0001111	15	F	si ([)	si

(continued overleaf)

binary	dec-imal	hexa-deci-mal	text character	graphics character
0010000	16	10	dle (])	dle
0010001	17	11	graphics red	graphics red
0010010	18	12	graphics green	graphics green
0010011	19	13	graphics yellow	graphics yellow
0010100	20	14	graphics blue	graphics blue
0010101	21	15	graphics magenta	graphics magenta
0010110	22	16	graphics cyan	graphics cyan
0010111	23	17	graphics white	graphics white
0011000	24	18	conceal display	conceal display
0011001	25	19	contiguous graphics	contiguous graphics
0011010	26	1A	separated graphics	separated graphics
0011011	27	1B	esc	esc [°]
0011100	28	1C	black background	black background
0011101	29	1D	new background	new background
0011110	30	1E	hold graphics	hold graphics

(continued overleaf)

binary	dec-imal	hexa-deci-mal	text character	graphics character
0011111	31	1F	release graphics	release graphics
0100000	32	20	space	space
0100001	33	21	!	
0100010	34	22	"	
0100011	35	23	s	
0100100	36	24	$	
0100101	37	25	%	
0100110	38	26	&	
0100111	39	27	'	
0101000	40	28	(	
0101001	41	29	)	
0101010	42	2A	*	
0101011	43	2B	+	
0101100	44	2C	,	
0101101	45	2D	-	

(continued overleaf)

binary	dec-imal	hexa-deci-mal	text character	graphics character
0101110	46	2E	.	
0101111	47	2F	/	
0110000	48	30	0	
0110001	49	31	1	
0110010	50	32	2	
0110011	51	33	3	
0110100	52	34	4	
0110101	53	35	5	
0110110	54	36	6	
0110111	55	37	7	
0111000	56	38	8	
0111001	57	39	9	
0111010	58	3A	:	
0111011	59	3B	;	
0111100	60	3C	<	

(continued overleaf)

binary	dec-imal	hexa-deci-mal	text character	graphics character
0111101	61	3D	=	
0111110	62	3E	>	
0111111	63	3F	?	
1000000	64	40	@	@
1000001	65	41	A	A
1000010	66	42	B	B
1000011	67	43	C	C [↯]
1000100	68	44	D	D [⊙]
1000101	69	45	E	E
1000110	70	46	F	F
1000111	71	47	G	G
1001000	72	48	H	H
1001001	73	49	I	I
1001010	74	4A	J	J
1001011	75	4B	K	K

(continued overleaf)

binary	decimal	hexadecimal	text character	graphics character
1001100	76	4C	L	L [↑]
1001101	77	4D	M	M
1001110	78	4E	N	N
1001111	79	4F	O	O
1010000	80	50	P	P [▱]
1010001	81	51	Q	Q
1010010	82	52	R	R
1010011	83	53	S	S
1010100	84	54	T	T [▭]
1010101	85	55	U	U
1010110	86	56	V	V
1010111	87	57	W	W
1011000	88	58	X	X
1011001	89	59	Y	Y
1011010	90	5A	Z	Z

(continued overleaf)

binary	dec-imal	hexa-deci-mal	text character	graphics character
1011011	91	5B	←	←
1011100	92	5C	½	½
1011101	93	5D	→	→
1011110	94	5E	↑	↑
1011111	95	5F	#	#
1100000	96	60	-	
1100001	97	61	a	
1100010	98	62	b	
1100011	99	63	c	
1100100	100	64	d	
1100101	101	65	e	
1100110	102	66	f	
1100111	103	67	g	
1101000	104	68	h	
1101001	105	69	i	

(continued overleaf)

binary	decimal	hexadecimal	text character	graphics character
1101010	106	6A	j	
1101011	107	6B	k	
1101100	108	6C	l	
1101101	109	6D	m	
1101110	110	6E	n	
1101111	111	6F	o	
1110000	112	70	p	
1110001	113	71	q	
1110010	114	72	r	
1110011	115	73	s	
1110100	116	74	t	
1110101	117	75	u	
1110110	118	76	v	
1110111	119	77	w	
1111000	120	78	x	

(continued overleaf)

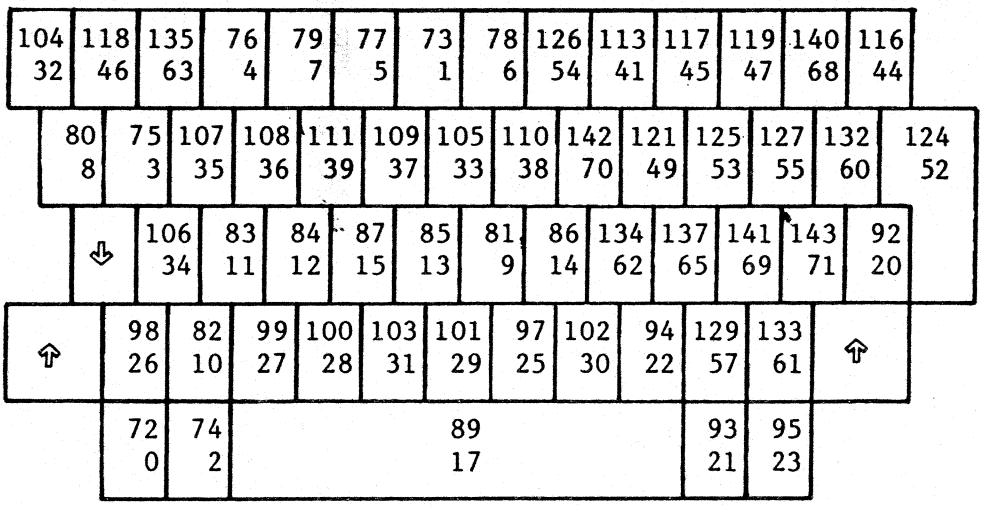
binary	dec-imal	hexa-deci-mal	text character	graphics character
1111001	121	79	y	
1111010	122	7A	z	
1111011	123	7B	½	
1111100	124	7C	|	
1111101	125	7D	¾	
1111110	126	7E	÷	
1111111	127	7F	↵	

Note:
Codes marked with an asterisk * are presumed before the start of each row.
Codes shown in parenthesis () are present in text character set of M model only.
Codes shown in brackets [] are present in graphics character set of M model only.

APPENDIX E KEY-CODES

The codes listed below are the decimal values of the key codes. The monitor reads and stores the hex values in the keyboard buffer when a key is pressed.

The lower number is the value of the key alone, the upper number, the value when the key is pressed together with the SHIFT, or during shift lock.



115 43	114 42	112 40
123 51	122 50	120 48
139 67	138 66	136 64
131 59	130 58	128 56
91 19	90 18	88 16

INDEX

Address 1-3, 4-1, 7-11, 7-20
Address decoder 4-1
Address lines 1-3, 4-1
Application flags 4-16
Application program 1-9
Application RAM 4-4, 4-5
Application video software 7-7, 7-17
ASCII 7-16
Attributes, video 7-3, 7-5, 7-11, 7-13, 7-17

Backspace printers 6-4, 6-5
Banks of extension RAM 4-19
Baud rate, printer 6-8
Bell, on T model 4-10
Bell, ports for AppB-2
Bell routine 4-10
Block, on cassette 5-3
BOT, beginning of tape 5-3

Cassette 1-10
Cassette control hardware 5-1
Cassette control software 5-13
Cassette file descriptor 4-17, 5-7, 5-13, 5-17
Cassette, memory areas for 5-13
Cassette, ports for AppB-1
Cassette routine 4-16
Cassette unit 5-1
Central processing unit 1-3
CFD, cassette file descriptor 4-17, 5-7, 5-13, 5-17
Character code AppD-1
Character code table 6-5
Character codes, for printers 1-11
Character decoding, video 7-10
Character generator 7-20

Character translation,
for printer 6-3, 6-5, 6-6, 6-7
Characters, on video 7-10, 7-20, 1-13
Check-sum, on cassette 5-6
Clear screen routine 7-9, 7-19
Clock counter 4-16
Colours, video 7-12, 7-17, 7-20
Column address 4-2
Control lines 1-3
CPU central processor unit 1-3
CPU, access to memory by 4-1
CPU board 1-2, 6-4, AppA-1
CTC, ports for AppB-2
CTC timer 3-3
Current key, on keyboard 3-2, 3-4, 3-8, 4-16

Data bus 4-2
Data code, on cassette 5-2, 5-8, 5-10, 5-12
Data code, on disk 2-2
Data format, on cassette 5-3
Data lines 1-3
Data record, on cassette 5-3, 5-4, 5-5
Data section, on cassette 5-6
Data transfer address,
on cassette 5-8, 5-9
Directory, on disk 2-4
Disabling of access to
video memory 7-8, 7-18
Disabling of video 7-13
Disk bootstrap routine 2-4
Disk control hardware 2-2
Disk control software 1-5, 2-4
Disk drive units 2-1
Disk format 2-3
Disk ports AppB-2
Dot matrix, of video 7-10, 7-12, 7-20
Double subsection, of
printer table 6-7

EOT, end of tape 5-3
Error codes, of cassette routine 5-19
Extension board 1-2, AppA-5
Extension RAM 1-8, 4-4, 4-5, 4-19

File extension, on cassette 5-8
File name, on cassette 5-8, 5-9
File type, on cassette 5-8, 5-10
First section, of printer table 6-6
Flash, on video 7-5
Flexible disks 1-5, 2-1
Flexible disks, ports for AppB-2
Floppy controller 1-5
Floppy disks, ports for AppB-2

Gap, on cassette 5-3, 5-4
Graphics characters 7-5, 7-6, 7-10, 7-16
AppD-1

Header, on cassette 5-5, 5-7
Horizontal scrolling 7-3, 7-13, 7-18
Horizontal scrolling, ports for AppB-1

I/O 1-3
I/O module, ports for AppB-2
I/O slot 1-7, AppC-1
Identification block, ROM key 4-11, 4-13
Initialise cassette routine 5-15
Input/output devices 1-1
Internal character code 6-1, 6-3, 6-5, 6-6, 6-7,
7-4, 7-14, 7-16, AppD-1
Interrupt 3-4
Interrupt vector 3-3
Interrupt vector table 4-16
Invert, on video 7-5

Key-codes 3-4, 3-5, 3-8, AppE-1
Keyboard 1-6, AppE-1
Keyboard, access from application 3-6
Keyboard clear routine 3-6, 3-9
Keyboard control software 3-2
Keyboard counter 4-16
Keyboard disable routine 3-6, 3-9
Keyboard enable routine 3-6, 3-9
Keyboard initialise routine 3-3
Keyboard input routine 3-6, 3-8
Keyboard, ports for AppB-1
Keyboard queue 3-2, 3-4, 3-7, 4-16
Keyboard, reading 3-4
Keyboard status routine 3-6
Keyboard table 3-8
Keyboard unit 3-1

Load address, on cassette 5-8, 5-11
Location, memory 4-1
Logical sectors, on disk 2-3

Mark, on cassette 5-3, 5-4, 5-5
Memory 1-1, 1-8, 4-1
Memory, access to 1-3
Memory, map of 4-5
Microprocessor 1-1
Mini-digital cassette 1-10, 5-1
Mini-digital cassette, ports for AppB-1
Modulation, of video signals 7-12, 7-20
Monitor 4-7, 6-4
Monitor flags 4-16
Monitor, M model (video) 1-12, 7-1, 7-2
Monitor, M model (video) hardware 7-2, 7-12
Monitor RAM 4-4, 4-5, 4-15
Monitor ROM 1-8, 4-4, 4-5, 4-7
Monitor video software 7-9, 7-19
Move backward cassette routine 5-16
Move forward cassette routine 5-15
MREQ, memory request line 4-1

Non-backspace printers 6-4, 6-5, 6-6

Page 1 of video memory 7-3, 7-4, 7-11, 7-13, 7-14, 7-17

Page 2 of video memory 7-3, 7-4, 7-5, 7-11, 7-13, 7-14, 7-17

Ports 6-4, 7-8, 7-18, AppB-1
Post-sync byte, on cassette 5-6
Preamble, on cassette 5-2, 5-5
Print routine 6-1, 6-3, 6-5
Printer 1-11, 6-1
Printer baud rate 4-16
Printer connector 1-12
Printer control hardware 6-1
Printer control software 6-1
Printer, ports for AppB-1
Printer table 4-17, 6-3, 6-4, 6-5
Printer-select jumper 6-4, 6-5, 6-6, 6-7

Queue counter, keyboard 3-2, 3-4, 3-8

RAM 4-2
RD, read line 4-2
Read cassette routine 5-17
Reading keyboard 3-4
Read/write, on disk 2-1
Record number, on cassette 5-8, 5-11, 5-15, 5-16
Refreshing memory 4-2
Repeat key facility 3-5
RESET button 1-9, 4-8
Rewind cassette routine 5-15
RGB monitor 1-13, 7-1, 7-12, 7-20
ROM key 1-8, 4-4, 4-5, 4-11
ROM key identification block 4-11, 4-13
ROM key structure 4-11
Row address 4-2

Scrolling, on video 7-3, 7-13, 7-17, 7-18
Second section, of printer table 6-7
Sectors, on disk 2-3
Set end-of-file record cassette routine 5-16
Single subsection, of printer table 6-7
Start address, on cassette 5-8, 5-11
Start gap, on cassette 5-4
Status cassette routine 5-18
STOP key 3-7, 3-8
Sync byte, on cassette 5-5
System (re-)start routine 4-8
System RAM 1-8, 4-15, 4-2, 5-17

Tape format 5-3
Text characters, on video 7-5, 7-6, 7-10, 7-16, AppD-1
Timing circuits, video 7-10, 7-20
Total file length, on cassette 5-8, 5-9
Tracks, on disk 2-3
Transparent print routine 6-1, 6-9
TV 1-13, 7-12, 7-20

Underline, of video 7-5

Valid data section length, on cassette 5-8, 5-9
VHF modulator 7-20
Video 1-12, 7-1
Video board 1-2, AppA-3
Video connectors 1-12
Video control hardware 1-12, 7-3, 7-11, 7-12
Video controller 7-8, 7-18
Video memory 1-12, 1-8, 4-4, 4-5, 7-3, 7-13
Video page 1 7-4, 7-11, 7-13, 7-14, 7-17
Video page 2 7-5, 7-11, 7-13, 7-14, 7-17
Video, ports for AppB-1
Video software, application 7-7, 7-17
Video software, monitor 7-9, 7-19
Video symbols 4-16
Viewdata code 6-3, 7-4, 7-14, 7-16, AppD-1

WR, write line 4-2
Write cassette routine 5-17
Write enable, of cassette 5-1
Write-protect, on disk 2-2

X-line, of keyboard 3-1, 3-4

Y-line, of keyboard 3-1, 3-4

Z-80 1-3, 7-7, 7-18

P2000 SYSTEM T&M REFERENCE MANUAL

12NC: 5103 991 30421

This issue comprises the following updates:

No updates

P2000 SYSTEM T&M REFERENCE MANUAL

12NC: 5103 991 30421

Including updates: .....

Originator:
Name .....
Address .....
.....
.....
.....
.....

Comment (if possible, add a copy of the page(s) affected by the comment, marked with the proposed changes):

Please return this form to: Osterr. Philips Ind. G.m.b.H.,
MAG-PERCO / MARKETING SUPPORT
Triesterstr. 64,
A-1100 WIEN,
A U S T R I A