MEGATEL COMPUTER CORPORATION INCORPORATED

THE MEGATEL QUARK® FAMILY OF COMPONENT COMPUTERS

MEGATEL QUARK® MANUAL FOR CP/M PLUS - 29 FEBRUARY 1984

edited by Graham Skells
Bill Misketis
Lisa Edwards

150 TURBINE DRIVE,
TORONTO, ONTARIO, CANADA M9L 2S2

How To Use This Manual

Part one, **the Installation Manual**, describes in detail how to set up a prototype system using any of the Megatel QUARK component computers, and how to install the CP/M Plus operating system and other software.

Part two, **the Technical Manual**, is the complete Technical Manual for the QUARK family of component computers, and describes in detail the operation of the hardware and software.

A special section dealing with the software interface to the Local-Area Network Interface used on the QUARK/200 is included with the QUARK/200 manual, which is otherwise identical to this manual.

Manuals published by Digital Reasearch for CP/M Plus are incuded with orders for that operating system, but not with "manual-only" orders.

The material on pages where the Digital Research Inc. trademark "CP/M Plus" appears is related to that operating system specifically. When no such mark appears, the material is of a more general nature, and refers to all models or software packages for the Megatel QUARK family.

Copyright Notice and Trademarks

"IBM" is a tradename of International Business Machines, Inc.

"Centronics" is a registered trademark of Centronics Data Computer Corporation.

Disclaimer

Megatel makes no representations nor warranties with respect to any circuitry not embodied within a Megatel product. Megatel specifically disclaims any implied warranties of merchantability or fitness for any particular purpose of any Megatel product. Megatel assumes no responsibility for errors which may appear in this document, and reserves the right to revise this document without notice.

Service Information

Should a QUARK board require service, please follow these steps to ensure prompt service.

1. Call Megatel and ask for the service department or, if you purchased your QUARK through a distributor, contact them. A service representative will try and diagnose the problem over the phone.

2. If the problem cannot be resolved over the phone, ask for a RMA number for returning the board. Please prepare a Purchase Order for the repair and provide us with the number.

3. Ship the board to the factory prepaid with the RMA number clearly marked on the outside of the package.

4. U.S. and overseas customers who purchased their boards directly from Megatel will receive a copy of their validated Canadian Custom B-13 form. When returning a board for service, a copy of the B-13 should be enclosed in an envelope marked to the attention of Thomas Meadows Company (our custom brokers). Including a copy of the validated B-13 avoids delays at customs. The box itself should be clearly marked as follows;

   Megatel Computer Corporation RMA#
   150 Turbine Drive
   Weston, Ontario Canada M9L 2S2
   c/o Thomas Meadows Company
   for custom clearance in Toronto

5. Depending on your credit terms with Megatel the board will be shipped back to you invoiced against the Purchase Order or upon receipt of payment for service. Boards still under warranty will be returned immediately upon completion of

repairs.

6. Included with the return board will be a service report describing the problem and an itemized bill of repair.

Service rates will be quoted on request.

Warranty

All products sold (including software) are under WARRANTY on a return to factory basis against defects in workmanship and material for a period of 90 days from the date of shipment.

WARRANTY is contingent upon proper use of the Product. WARRANTY will not apply if any repairs are necessary due to accident, unusual physical, electrical or electro-magnetic stress, neglect, misuse, or causes other than ordinary use. WARRANTY will also not apply if the product has been modified by BUYER, or if the product has been disassembled by BUYER. Disassembly includes the removal of the serial number label on the Product without prior written approval from MEGATEL.

Customer attempted repairs will void the WARRANTY. Any tips suggested in the manual which involve physical changes to the board or a reconfiguration of the software, if attempted will void the WARRANTY.

TABLE OF CONTENTS

INSTALLATION MANUAL

TECHNICAL MANUAL

HARDWARE

SOFTWARE

TECHNICAL MANUAL (continued)

APPENDIX

Electrical Specifications

Omninet Programmer's Guide (included for QUARK/200 only)

MEGATEL QUARK® INSTALLATION MANUAL
CP/M PLUS EDITION

1.0 Introduction to the Megatel QUARK single-board computer family

The Megatel QUARK is a family of high-performance CP/M-compatible microcomputers constructed on standard-sized Eurocard-format printed-circuit boards. Based on the six megahertz Zilog Z-80B 8-bit microprocessor, features of the QUARK family include dual-mode Alphanumeric/Graphic Video Display interfaces with a programmable character sets, floppy disc interfaces for up to four double-sided, single- or double-density 8- inch or 5-1/4- inch floppy disc drives, a full-duplex serial interface port as well as a simplex serial port, a parallel printer interface, and 22 general-purpose input/output lines for keyboard input and other I/O functions. Both serial ports are compatible with RS-232C signal levels. Additionally, access to the CPU's data bus and to a subset of its address and control lines is provided to allow the user to add special-purpose peripherals.

The Megatel QUARKs are available with 128 kbytes of main memory for the operating system, user programs, and the Video Display memory. As well, some models in the family have 256k or 64k of main memory. An additional 2k of static programmable memory is used to store the character set used in the Alphanumeric mode on those QUARK models equipped with a Monochrome Video Display Interface. On the Colour Video Display Interface of the QUARK/150, a separate 16kbyte programmable memory stores the Colour Translation Tables. A 512 byte ROM holds the system bootstrap loader on all models.

Programmable Array Logic (PAL) integrated circuits and a Synchronous Address Multiplexer include most of the random logic and timing functions used in the computer. The high level of integration on the QUARKs allows a minimum number of ICs to provide all of the functions required, thus enhancing reliability. The small size of the boards makes them suitable for inclusion into new or existing systems, and simplifies packaging and service.

Software packages based around Version 2.2 or 3.0 (PLUS) of Digital Research Corp.'s CP/M operating system is available for any of the QUARK microcomputers. Included with this package is the complete BIOS source needed to run CP/M 2.2 or CP/M PLUS on the QUARK, a menu-driven installation program which allows the user to customize the BIOS to suit the parameters of the target system, standard CP/M utilities, and a set of special utilities for the QUARK. These utilities include loaders and editors for the programmable video display interfaces, serial port configuration programs, and disc copying routines. Also available are MP/M II, a multi-tasking, multi-user operating system, and the CP/NET Network Operating System, both from the Digital Research .

All models of the QUARK family are availible in 60Hz or 50Hz versions. This allows the vertical refresh rate of the Video Display Interface to be matched exactly to the line frequency of the local power system.

All members of the QUARK family require only regulated 5 and 12 volt supplies. These are the same voltages as are required for industry-standard 5-1/4 floppy-diskette drives.

1.1 The QUARK/100

The QUARK/100 is the basic member of the QUARK family of single board computers. It includes the Z-80B processor, 128kbytes of main memory, a monochrome Video Display Interface with programmable Alphanumeric and Graphics modes, and the serial and parallel I/O ports mentioned in Sec 1.0. Memory options also include 256k or 64k of main memory.

The QUARK/100 is constructed on a 100-by-162mm printed-circuit board. All connections to the board are made through the 96-pin European Standard Interface Connector at one end of the board.

The QUARK/100 consumes appproximately 10 watts of power.

1.2 The QUARK/150

The Megatel QUARK/150 is designed to meet the requirements of users who need the capabilities of the QUARK/100 together with a full-colour alphanumeric or graphics display. A highly flexible Colour Video Display Interface is intended to be used with analog RGB colour monitors (although other types of monitors may be suitable) and can also be used with monochrome composite video monitors.

The Colour Video Display Interface of the QUARK/150 is fully-programmable in both the Alphanumeric and Graphics modes. In Alphanumeric Mode, as many as two independent colour character fonts of 256 characters each may be simultaneously resident in the Colour Translation Tables, a special 16kbyte memory in which the colour information is stored. Switching between the two character fonts may be accomplished on a row-by-row basis.

In Graphics mode, as many as sixteen variations on one of three resolution modes allows the user great flexibility of pictorial display. The user has the choice of trading spatial resolution for colour selection. The particular mode used may be changed on a raster scan-line-by-scan-line basis. The three modes are:

- 640 pixels horizontally, with a choice of one of two colours for each pixel. The two colours are pre-selected for each scan line from a 16-colour palette
- 320 pixels horizontally, with a choice of one of four colours for each pixel. The four colours are pre-selected for each scan line from a 16-colour palette
- 160 pixels horizontally, with a choice of any of 16 for each pixel.

The colour information is stored in a special 16kbyte memory which is external to and independent of the main memory of the QUARK/150. This memory, called the Colour Translation Table, replaces the Programmable Character Generator used on the QUARK/100. Two 16k-

by-four bit dynamic RAMs are used for the Colour Translation Table. The functions of the Video Character latch and the Video Shift Register of the QUARK/150 are replaced by two Programmable Array Logic (PAL) devices.

The colours generated by the RGB Colour Video Display Interface are compatible with the Apple III Designated Colour Set.

Except for the differences between the monochrome Video Display Interface of the QUARK/100 and the Colour Video Display Interface of the QUARK/150, the two machines are identical. The pinouts of their 96-pin connectors differ only on the four video-related lines. Both models include a floppy disc interface for up to four double-sided, single- or double-density 8- inch or 5-1/4- inch floppy disc drives, a full-duplex serial interface port and a simplex serial port, a parallel printer interface, and 22 general-purpose input/output lines for keyboard input and other I/O functions. The serial ports are compatible with RS-232C signal levels. Additionally, access to the CPU's data bus and to a subset of its address and control lines is provided to allow the user to add special-purpose peripherals.

The Megatel QUARK/150 includes 128kbytes of main memory for the operating system, user programs, and the Video Display memory. An additional 16k memory holds the colour information on the QUARK/150. A 512-byte ROM stores the system bootstrap loader.

In addition to the standard software available for the entire QUARK family, special utilities required to set up and use the Colour Translation Tables are provided with the QUARK/150.

The QUARK/150 consumes approximately 10 watts of power.

1.3 The QUARK/200

The QUARK/200 includes all of the features of the QUARK/100, and in addition provides a one-megabit-per-second local area network interface. The network protocol used is Corvus System's OMNINET local-area network system. It allows up to sixty-four computers to be connected over a single twisted-pair cable of up to 1300m (4000 ft) in overall length. The maximum node-to node distance is 650m (2000 ft). RS-422 differential signal levels are used on the cable.

The OMNINET interface on the QUARK/200 includes its own network processor. The interface implements the lower four layers of the ISO seven-layer network model. This relieves the host processor of many of the tasks required for the network. The interface handles all of the message acknowledgement and retransmission without disturbing the CPU of the QUARK/200. A carrier-sense multiple access distributed control mechanism is used to arbitrate the use of the single network channel.

The LAN interface communicates with the Z-80B processor by both Direct Memory Access (DMA) and by programmed I/O. Data received or

transmitted by the LAN interface is transferred in and out of a pre-programmed area in Main Memory. Sequences of commands to be executed by the LAN interface are entered as command vectors in memory and transferred by DMA to the network interface. Two status/command registers, which appear within the I/O space of the QUARK/200's Z-80B, are used to program the locations of data and command vectors within memory, to query the status of the interface, and to control the generation of interrupts by the LAN interface.

The software available from Megatel for the QUARK/200 includes the standard system and utility software for the QUARK family (mentioned above), and a demonstration network package using MP/M II and CP/NET. This demonstration package, available separately, is sufficient to permit several QUARK/200's to be connected together, with one acting as a file server and the rest as network slaves.

The QUARK/200 can be utilized as a network slave in conjunction with a file server. The QUARK/200 itself can perform the functions of a file server in limited applications.

The QUARK/200 is constructed on a standard 100 mm x 228 mm Eurocard-sized printed-circuit board. All connections to the computer, with the exception of the LAN interface, are made through a 96-pin connector at one end of the board. The pin-out of this connector is completely compatible with that of the QUARK/100. Connection to the LAN interface is through a 3-pin right-angle male header on the opposite end of the board from the ESIC connector.

I/2.0 Installation of the QUARK

In order to set up a prototype system using the QUARK, some peripheral devices are required. The minimum set of peripherals is as follows:

- at least one single- or double-sided, double-density floppy disc drive of the same size and track density (48TPI or 96TPI) as the distribution diskettes shipped with the QUARK

--AND--

- an ASCII-encoded 7- or 8-bit parallel-output keyboard with active-high DATA outputs and an active-low STROBE output
- for those QUARKs with a Monochrome Video Display Interface (such as the QUARK/100 or /200), a monochrome CRT monitor with either direct-drive video, horizontal-sync, and vertical-sync inputs, or with a composite video input that accepts composite video signals similar to RS-170
- for the QUARK/150, an analog RGB colour monitor with red, green, and blue video inputs and a composite sync input, or red and blue video inputs and a green sync input. The QUARK/150 may also be used with monochrome composite-video monitors.

--OR--

- a computer terminal (or its equivalent) with a full-duplex RS-232C asynchronous serial interface capable of operating at 1200 baud

--AND FINALLY--

- a regulated DC power supply capable of supplying +5.0V at between 2.0A and 2.5A, and +12V at 100mA. If this supply must also supply power to operate the floppy disc drives, a monitor, or a keyboard, then the power requirements of these devices must be allowed for.

2.10 ESIC Connector

All connections to any of the QUARK single-board computers (with the exception of those which include the Local-Area Network Interface) are made through the male 96-pin European Standard Interface Connector on the printed-circuit board. The name assigned to each pin and a brief description of its function can be found in Table VII of the Appendix. Table VIII lists the sets of pins on the ESIC connector required for commonly-used peripheral devices or interfaces. A separate three-pin header is used for the network connection on the QUARK/200.

A mating female 96-pin ESIC connector with wire-wrappable posts is included with the package to assist in assembling prototype systems using the QUARK. Extra connectors are available from Megatel, or from the sources suggested in Table X.

2.11 The QTB-2 Transition Board

The QTB-2 Transition Board is designed to provide users of the Megatel QUARK family of single-board computers a flexible and quick method of bringing up a prototype system. Additionally, some users may find that the interconnection capability provided by the QTB-2 renders it suitable for use in their intended application. This section describes some of the ways of using the QTB-2. As of the date of issue of this manual, the revision number of QTB-2 boards being shipped is REV 2.0. The previous revision, REV 1.0, is very similar to REV 2.0, and is no longer available.

The QTB-2 provides connection points for 8-inch or 5-1/4 inch floppy disc drives, a serial terminal and a serial printer or keyboard, a composite-video CRT monitor, and the power supply for the QUARK computer. Printed circuit patterns allow two 25-pin D-shell connectors, a 26-pin male header, a 34-pin male header or card-edge connector, a 4-pin DC power connector and a straight or right-angle 96-pin European Standard Interface Connector (ESIC) to be mounted directly on the board. REV 2.0 boards also provide a position for direct-drive (or "TTL") video monitors to be connected. Together with the composite-video output, these connectors form the connection points needed for the RGB monitor used with the QUARK/150. Finally, REV 2.0 boards provide pads for the installation of a reset switch.

The QTB-2 is shipped as a bare printed-circuit board. No peripheral connectors are included, as each user will require a specific set of connectors to meet his requirements. ESIC connectors are available from Megatel in small quantities.

2.12 Installing a mating ESIC connector for the QUARK

To connect the QUARK computer to the QTB-2, a female 96-pin connector must be soldered into the position marked on the QTB-2 board as "J1". The female connector used may be a straight wire-wrappable female connector or a right-angle PC mount connector. Either types of connectors **must** be inserted from the side of the QTB-2 board on which there are no white markings (i.e., the connector pins will be soldered from the side on which the white labelling appears). Furthermore, the orientation of the connector **must** match the pin numbers indicated on the labelled side of the board. (See fig I/2.1)

If a 96-pin connector with wire-wrappable pins is to be used it may be desirable to retain the long wrap-posts so that additional wires may be wrapped on these posts. If so, then it is recommended that one of the outside rows of pins (i.e. row "a" or "c") be soldered in place, then the other outside row be bent outward sufficiently to allow the inside row (row "b") to be soldered. When this row is soldered, the row previously bent outward may be straightened. Care should be taken when doing this that the pins are not broken nor that the connector is unseated from its mounting plane.

Two pairs of mounting holes are provided for securing the ESIC connector to the transition board. The pair nearest to the edge of the board should be used to mount right-angle connectors, and the inner pair for the straight connectors. The correct screw to be used is a 2.5mm machine screw, although a #4 screw will clear the hole in the PC board.

2.13 Connector for a serial terminal and serial printer or keyboard

Connector positions J4 and J5 are provided to allow a terminal with a standard RS-232C serial interface to be easily connected to the board. Position J5 is intended for a female (or, less likely, a male) 25-pin "D-shell" connector. Both straight and right-angle PC-mount or wrappable connectors may be used. If the right-angle variety is employed, then the connector may be fastened by two #4 machine screws and nuts through the holes provided adjacent to the PC pattern for J5.

In order to conform to the standard pinout for the RS-232C interface, a 25-pin female D-shell connector must be mounted from the labelled side of the board, whereas a 25-pin male D-shell connector should be mounted from the unlabelled side of the board. Figure I/2.2 shows a right-angle female DB-25S connector mounted this way. If this convention is used, then the pinout for the installed connector will be as shown in the table below.

PINOUT FOR D-SHELL CONNECTOR AT J5 (standard installation)

D-shell pin number	QUARK pin number	Function
1	N/C	Optional chassis ground
2	C-2	Full-duplex receive data
3	A-3	Full-duplex transmit data
4	C-4	Clear-to-send
5	B-3	Request-to-send
6	C-3	Data terminal ready
7	A-5	Signal ground
14	A-4	Simplex serial port output
16	B-2	Simplex serial port input
20	B-4	Data terminal ready

The Simplex Serial port on the QUARK board is normally strapped at the factory for operation as a serial output device. Thus pin 14 in the above pinout will be the Simplex Serial port Transmit Data line, and pin 16 will be the Simplex Serial port Protocol Input line. If the Simplex Serial port is re-strapped to act as an input, then the functions of these two pins will be reversed accordingly. Consult section 3.4 in the Technical Manual for further information regarding the use of this port.

Position J6 is wired to provided the Simplex Serial port output on pin 3, the Simplex Serial port input on pin 20, and a ground connection on pin 7. If the Simplex Serial port is strapped as an output (as is standard when shipped from the factory), then a 25-pin D-shell connector installed at J6 will provide a common connection to a serial printer. Note, however, that some serial printers will require their CTS/RTS, DSR/DTR, or other protocol lines to be tied to the active state in order for their serial interfaces to operate correctly. These interface requirements can usually be found in the manual for the printer.

Traces on the bottom of the QTB-2 between J5 and J6 allow the connections to pin 3 and 20 of J6 to be cut easily, and re-strapped or left open if the wiring of J6 is to be other than as described above. (See Fig. I/2.2)

If a CRT display and a keyboard, rather than a terminal, are used then the RTS and CTS lines on the Full-Duplex Serial Port MUST be tied together initially in order that the Asynchronous Communications Adapter (ACIA) transmitter will be enabled. If the CTS input line is allowed to float, then the transmitter in the ACIA will not clear, and the BIOS routine in the Distribution operating system which handles console output will get hung-up in a loop waiting for the transmitter to clear. Tying RTS to CTS will pull CTS high (+12V) and allow the transmitter to clear. (The operating system shipped on distribution diskettes is normally configured to send console output to both the Video Display Interface and the Full-duplex serial port, as well as to accept console input from the serial port or from a parallel keyboard. This gives users a choice of peripheral equipment needed to bring up a Quark system.) Later, when using QINSTALL to install the customized

operating system, the desired configuration for the console device may be specified. If a serial terminal will not be used as the console device in the final (target) system, then the strap between RTS and CTS may be removed after the installation process has been completed and the QUARK is running under the customized operating system. The RTS output line for the Full-Duplex Serial Port is pin B-3 on the ESIC connector, and the CTS input line is pin C-4.

If a serial terminal is used as the QUARK's console device, then the CTS input line to the QUARK should be connected to the RTS output from the the QUARK, or to the terminal's RTS output. Additionally, the QUARK's DSR input (pin B-4) must be at a high (+12V) level in order that the $\overline{DCD}$ input on the ACIA be at a low level. (A high level on the ACIA's $\overline{DCD}$ input disables its receiver.) Thus the DSR input should be connected to either the RTS output from the QUARK, or to the DTR output from the terminal. RS-232C inputs which must be connected to a high level may be connected to the +12V supply as well.

All of the QUARK's serial interface lines (inputs and outputs from the Full-duplex and Simplex ports) appear at the connector positions J4 and J5, at the pin numbers given in the table above. Additionally, all 25 pads forming connector position J5 are connected to the corresponding pads at position J4. The J4 pattern is provided to allow a 26-pin male header assembly to be used instead of (or in addition to) a D-shell connector for the Full-duplex and Simplex serial ports. Connections to the J4 pattern have been set up to correspond to the standard D-shell pin designations given in the table above. This permits a ribbon cable with a 26-pin female header at one end and a 25-pin D-shell connector (or other connector) at the other end to be used to extend the RS-232C connector from the QTB-2 board.

2.14 Connection to floppy-disc drives

The QTB-2 transition board may be used to connect to either 5-1/4-inch or 8-inch floppy disc drives.

Connection to a 5-1/4-inch drive may be may made through a ribbon cable from the board to the card-edge connector fingers on the drive, or by a card-edge connector mounted directly onto the QTB-2 board. In the former case, a 34-pin, 17-position male header should be soldered onto the QTB-2 board in the position indicated by the smaller and left-most rectangle associated with J3. This header connector may be mounted on either surface of the QTB-2 board, although the orientation of the ribbon cable will vary depending on the choice of mounting surface. It is recommended that the user review the topology of the connection between the QTB-2 and his disc drive carefully before proceeding with his installation. (See Fig. I/2.2)

In systems using a 5-1/4-inch drive, it is possible to solder a card-edge connector directly onto the QTB-2 board. In this configuration, the QTB-2 acts as a sort of motherboard, with both the floppy-disc drive and the QUARK single-board computer plugging into the QTB-2. To do this, a 17-position double-sided card-edge connector

the QTB-2. To do this, a 17-position double-sided card-edge connector with either PC-mount or wire-wrappable tails should be used. (Seventeen-position card-edge connectors may be difficult to obtain with PC or wire-wrappable tails. Eighteen-position double-sided card-edge connectors are usually more readily available and may be used if the last position is ignored.) The card-edge connector **must** be installed from the same surface of the transition board as the female ESIC connector for the QUARK, using the set of holes defined by the outer rectangle of J3.

The QTB-2 may be used in a motherboard configuration for a system incorporating more than one 5-1/4-inch floppy disc drive. To do this, install a 17-position double-sided edge connector with long wrappable posts from the unlabelled surface of the QTB-2 board. Then this connector should be installed so that the wrap posts extend approximately 8mm (5/16") or more through the circuit board. Solder the 34 pins in place, and then cut off all of the pins in the row nearest the 96-pin ESIC connector. This will leave one row of pins, all of which are connected to ground on the QTB-2 board. Then install a 17-position single-row male header on the opposite side of the board from the edge connector in the set of holes between the two rows of PC pads used for the edge connector pins. Trim the remaining row of edge connector pins to the same height as the male header pins. The resulting two rows of pins can then be used as a 34-pin double-row male header to which a 34-conductor ribbon cable can be connected. This ribbon cable would be constructed with a 34-contact female header connector on one end, and with one to three 34-pin ribbon cable edge connectors (one for each 5-1/4-inch floppy disc drive) along the rest of the length of the cable.

To connect one or more 8-inch floppy disc drives, three sets of traces on the QTB-2 transition board must be modified. Industry-standard 8-inch drives use a 25-position double-sided (50-contact) edge connector for their interfaces. Of these fifty pins, a subset of thirty-four is sufficient to connect the drive to the QUARK computer.

On the labelled side of the QTB-2 board, there are two sets of three small rectangles in which the words "SIDE", "SEL 3", and "INDEX" appear. Three of these rectangles enclose two plated-through holes which are connected by short traces, while the other three rectangles enclose only a single plated-through hole. In order to re-wire the QTB-2 board to configure the floppy-disc connector hole pattern (J3) for 8-inch drives, it is necessary to cut the short traces joining the pairs of holes within the rectangles marked "SEL 3" and "INDEX". (It is not necessary to cut the trace within the rectangle marked "SIDE" unless a male header with more than 34 pins is used on the QTB-2 board.) After making these cuts, install jumper wires between those holes which are now electrically connected only to the ESIC connector pattern and the corresponding single hole which bears the same name.

The following is a summary of the method of modifying the QTB-2 board for use with 8-inch floppy disc drives: (See Fig. I/2.1)

1. Cut the short trace joining the pair of holes enclosed by the rectangle marked "INDEX".

2. Cut the short trace joining the pair of holes enclosed by the rectangle marked "SEL 3".

3. Cut the short trace joining the pair of holes enclosed by the rectangle marked "SIDE" if a male header connector with more than 34 pins is to be used on the QTB-2 board.

4. Connect a wire from the hole nearest the letter "I" in the word "INDEX" of the rectangle in which the trace was cut to the hole within the other rectangle marked "INDEX". (dashed line in figure)

5. Connect a wire from the hole nearest the letter "S" in the word "SEL 3" of the rectangle in which the trace was cut to the hole within the other rectangle marked "SEL 3". (dashed line in figure)

6. Connect a wire from the hole nearest the letter "S" in the word "SIDE" of the rectangle in the centre of the board to the hole within the other rectangle marked "SIDE". (dashed line in figure)

This completes the modifications necessary to configure the QTB-2 board for use with 8-inch floppy disc drives.

After these modifications are carried out, install a 17-position double-row male header in the set of holes indicated by the outer and right-most rectangle for J3. This rectangle is marked by the number "8" at one end. The header may be installed from either surface of the board.

Up to four 8-inch drives may be connected by a 34-conductor ribbon cable to the header on the QTB-2 board. Since the drives use 25-position double readout edge connectors, the ribbon cable must be constructed with attention to the proper positioning of the 34-conductor cable on the 50-contact ribbon cable edge connectors. Pin 1 on the 34-pin header (which is connected to ground at the hole nearest the number "8" on the QTB-2 board) must be connected to pin 13 on the 8-inch floppy-disc drive. Using 34-conductor ribbon cable, pins 13 through 46 on the 50-contact edge connector will be used. This will leave pins 1 through 12 and 47 through 50 unconnected.

The connections required between standard 8- or 5-1/4-inch floppy disc drives and the QUARK's 96-pin ESIC connector are indicated in Tables VIIIc and VIIId of the Appendix. Note that the cables connecting the QUARK to the floppy disc drives, whether they be ribbon cables or twisted pairs, must be treated as transmission lines. This means that all output lines from the QUARK (e.g. the Write Data

output) must be terminated by 150ohm pullup resistors on the drive which is farthest from the QUARK. Pullup resistors are installed on the QUARK board for all lines which are outputs from the disc drives to the QUARK. No pullup resistors should be installed on any of the drives which are electrically "between" the QUARK and the last drive. The pullup resistors on input lines to the drives are usually installed as a resistor array in an IC socket, so that it is necessary only to remove the arrays from all but the "last" drive. The user should consult the manual for his disc drives if he is uncertain as to how to set up the terminating resistors. Finally, the configuration of the drives and the QUARK must be "straight-line", with the QUARK at one electrical "end" of the line and the floppy disc drives all on one electrical "side" of the board. "Stubs" on the cable should be kept to a minimun length.

The operating system on the Distribution Diskettes shipped with any of the "Complete QUARK packages" will initially recognize only one physical disc drive. In selecting this drive, it will assert (logic low output) the SEL 0 Drive Select Line . Therefore, one (and only one) drive must be set up to be selected by SEL 0. If the operating system is temporarily or permanently modified to permit the use of two drives (as explained in the section 4), then the system will attempt to select the second drive by asserting the SEL 1 Drive Select Line. Thus the second drive should be set up to be selected by the SEL 1 line.

With the Megatel Quark, it is possible to set up a a system which uses both 5-1/4- and 8-inch floppy disc drives. Such a system is referred to a a "mixed drive installation", and requires a few small modifications to the floppy disc drive connections.

For further information concerning the connections to either 5-1/4-inch or 8-inch drives, Megatel can provide an application note or "Q-tip" on setting up a mixed drive installation. Please consult the factory for any application note.

2.15 Parallel-output keyboard connection

The connections for an ASCII-encoded keyboard are listed in Table VIIIe of the Appendix. (See also Fig. 3.1) Although eight input lines are provided, the routine in the CP/M BIOS handling the parallel keyboard input (which is part of the Console Input routine) normally discards bit 7 (the most-significant), so that it is not absolutely necessary to connect any input to this pin. If a conventional computer terminal rather than an encoded-keyboard is to be used with the QUARK, then Port A of the PIA and its CA2 control line (designated Port 1 on the QUARK) may be used for other purposes.

Because there is no industry-standard connector for encoded parallel-output keyboards, no specific connector pattern has been provided for this purpose on the QTB-2 board. It is left to the user

provided for this purpose on the QTB-2 board. It is left to the user to choose a connector which meets his requirements, as well as to wire the connector to the appropriate pins on the female 96-pin connector for the QUARK. Section 3.2 in the QUARK Technical Manual describes the requirements for the parallel keyboard, and Table VIII(b) summarizes the pins of the QUARK's connector used for this type of keyboard.

One possibility for a keyboard connector is to use the spare "D-shell" hole pattern (J6) on the QTB-2 board. With a straight or right-angle D-shell connector mounted from either side of the board, the pins of the connector may be connected to the female 96-pin connector by short lengths of wire. Note that if the D-shell connector is mounted on the opposite side of the QTB-2 board from the 96-pin connector, it will be necessary to arrange for the wires to cross from one surface of the board to the other somehow, perhaps through unused holes in the "kluge" area of the board. Also note that 3, 7, and 20 of the J6 pattern are connected to the Simplex Serial port of the QUARK. See Sec. I/2.3 for more information on the use of the J6 connector position.

Although J6 is set up for a 25-pin D-shell connector, 15- and 9-pin D-shell connectors may also be soldered in the PC hole pattern provided. Note that only one of the mounting holes may be used for these smaller connectors.

Another method of connecting a parallel keyboard is to install an appropriate male header or other connector in the "kluge" area of the QTB-2 board. This connector would then be wired to those pins on the female 96-pin connector used for the parallel keyboard interface.

To use an encoded keyboard with active-low DATA outputs, the DATA outputs must initially be inverted with TTL-compatible inverters. After bringing up a prototype system, the user may modify the Console Input routine in the BIOS to allow the use of the keyboard without the need for inverters. If the STROBE output from the keyboard produces an active-high signal, then it may be necessary to invert this line initially. However, as long as the STROBE pulse includes both rising and falling edges, then the CA2 input on the PIA to which this line is connected will cause an interrupt to be generated. Some experimentation may be needed if this sort of keyboard is to be used.

2.16 CRT display monitor connection

Megatel QUARK single-board computers with the Monochrome Video Display Interfaces (such as the QUARK/100 and /200) provide both direct-drive (TTL) and composite video outputs. Connector position J7 is intended for the composite video output using a standard RCA-type phono jack. The large pad labelled "CV" around the .250-inch hole is connected to ground, while a trace from pin A-29 on the female 96-pin connector (which is the QUARK's composite video output) terminates at a plated-through hole adjacent to the grounded pad for J7. With the female phono jack installed from either side of the board, it is necessary only to connect a short wire from the centre conductor of the phono jack to the pad terminating the composite video line.

Standard direct-drive CRT displays usually use a 10-pin 0.156" edge connector on the monitor's printed-circuit board. Five .025" diameter plated-through holes are provided for connection to a direct-drive monitor. This set of pads, labelled "TTL-V" provides (in order from left to right):

TTL Video pin	QUARK pin	FUNCTION Q/100, Q/200	FUNCTION Q/150
1	A-1	Ground	(Ground)*
2	A-2	Vertical Sync	Red
3	B-1	TTL-video	Green
4	power supply	+12V to monitor	
5	C-1	Horizontal sync	Blue

* On revision 01R00 Quark/150 boards, pin A-1 is open. Use A-5.

A set of five .025" square posts may be installed in this connector area to provide a convenient connector for TTL-video monitors, or a part of the connection to an RGB monitor.

Table VIII(g) in the Appendix lists the four pins on the QUARK's connector to be used for a direct-drive monitor.

If a direct-drive monochrome monitor is used, a 500-ohm contrast control potentiometer should be connected on the video input to the monitor. (Some monitors may include this control internally.) If a composite-video monitor is used, the composite video output should be terminated by a 75-ohm load to ground at the monitor. Some monitors have a "High Impedance/Terminating" load switch for this purpose. This switch, if present, should be at the "Terminating" position.

For the Colour Video Display Interface of the QUARK/150, four lines (plus ground) connect the analog RGB monitor to the board. The four pins used for the Red, Green, Blue, and Composite Sync outputs on the QUARK/150 are the pins which are used for Video, Horizontal Sync,

Vertical Sync, and Composite Video on the monochrome QUARK/100 and /200. If a connector providing all four (plus ground) of these lines is installed, then complete compatibility between monochrome and colour set-ups can be maintained when this is necessary. All that is required is that one cable harness be used for connection from the QTB-2 and a monochrome monitor (direct-drive or composite), and another, slightly different cable harness for the connection to a colour monitor. Thus, the composite-video connector position plus the five TTL-video pads provide all of the lines needed for an RGB monitor connections.

The Colour Video Display Interface may also be used to drive a monochrome composite-video monitor. To do this, the Red, Green, Blue, and Composite Sync outputs should be all tied together. The RGB resistor network on the QUARK/150 will then generate an analog grey-scale video output.

The composite sync polarity on the QUARK/150 is selectable by jumpers J14 and J15. (Negative polarity means that the horizontal sync pulses are low-going except during vertical retrace.) These jumpers consist of three male posts near the Z-80B processor on the board. The center of the three posts is connected to the composite sync output on the 96-pin connector. A jumper plug is provided to connect either J14 or J15 (but not both). J15 is the pair of posts nearest the 96-pin connector on the QUARK/150. With J15 connected, the composite sync signal will be of negative polarity. J14 is the other pair of posts, and gives positive sync polarity. The jumper plug should be installed so as to provide the polarity required by the monitor to be used.

2.17 Power supply connection

A set of four plated-through holes marked "J2" is intended for the power supply connection. The pads for this connection are laid out so as to allow the use of the same type of 4-pin connector as is used on standard 5-1/4-inch floppy disc drives. Such a printed-circuit-mount connector may be soldered on from either surface of the QTB-2 board, although the orientation of the connector will depend on which surface is used. Since these connectors are polarized, it is essential that the connector be installed correctly. The power supply voltage and polarity of each pin is clearly marked on the QTB-2 board.

Both straight-pin and right-angle varieties may be used for this power connector.

C A U T I O N

UNDER NO CIRCUMSTANCES SHOULD ANY OF THE POWER SUPPLY LINES BE CONNECTED INCORRECTLY! REVERSING ANY OF THESE LINES, OR INTERCHANGING +5 AND +12 MAY CAUSE GRIEVIOUS DAMAGE TO THE COMPONENTS OR TO THE PRINTED CIRCUIT BOARD OF THE QUARK! ANY SUCH DAMAGE IS NOT COVERED BY WARRANTY!

Before power is initially applied to the QUARK board, the power supply lines should be inspected carefully to insure their proper connection.

For convenience, it is recommended that a normally-open momentary-contact switch be connected between the RESET input line and any GROUND return line. The RESET input is pin C-31 on the QUARK's connector. Closing this switch causes a System Reset to occur. A System Reset is also generated when power is applied initially to the board. A set of three pads near the composite-video connector (labelled "RESET") is provided for mounting a reset switch. The middle of the three pads is connected to ground, while the pad nearest the composite video connector is connected to pin C-31, the RESET input to the QUARK. If it is not desired to mount the switch at this position, wires may be extended from this postion to the switch.

2.18 Kluge area

A "kluge" area of plated-through holes is provided to permit special connectors to be soldered to the QTB-2 board, and to allow additional circuitry to be assembled and connected to the QUARK computer. Between the kluge area and the 96-pin connector a set of holes provide connecting points to the +5V, +12V, and ground lines from the power supplies. Bypass capacitors may be installed here if they are necessary.

2.19 Connectors for the QTB-2 transition board

Twenty-five pin "D-shell" connectors for straight or right-angle mounting are available from many suppliers. Part numbers for female connnectors from AMP Inc., of Harrisburg, Pennsylvania, are as follows:

Straight: 207090-5
Right-angle: 206584-1

Male headers for the floppy disc connector and other connections are also widely available. T & B/Ansley Corp., of Los Angeles, manufacture a wide variety of headers for these purposes. Their part number for one style of 34-pin header is

609-3427

For female 96-pin ESIC connectors, see table X in the Appendix of the Technical Manual.

2.2 Local-Area Network Connections (QUARK/200 only)

The Local-Area Network used on the QUARK/200 communicates over a single twisted-pair cable. Connection to this cable is by a three-pin right-angle male header located at the opposite end of the QUARK/200 board from the 96-pin ESIC connector.

The two outside pins of this connector are the "+" and "-" network signal lines. Signal polarity conventions must be observed throughout the network for proper operation, although reversing the connection causes no damage.

The centre pin of the connector is connected to ground through a non-polarized capacitor of approximately 1.0uF.

Normally twisted-pair cable is sufficient for use on the network. The maximum overall length of the network cable is 1300m (4000 feet), although the maximum distance between two units which must communicate with each other is approximately half this distance. The cable should be terminated at both ends by a matching load impedance (150ohms) to cancel end reflections.

In electrically noisy environments it may be necessary to use shielded twisted-pair cable to achieve an adequate signal-to-noise ratio on the network. The shield of such a cable should be connected to the centre pin of the three-pin network connector. Because of the capacitance added by the shield, the maximum distances achievable on the network are usually reduced when using a shielded cable pair.

Leaving the network connections open or floating can occasionally cause glitches in the network interface. When not connected to the network, the network connections should be terminated by a 150-ohm load accross the "+" and "-" lines.

This page intentionally left blank.

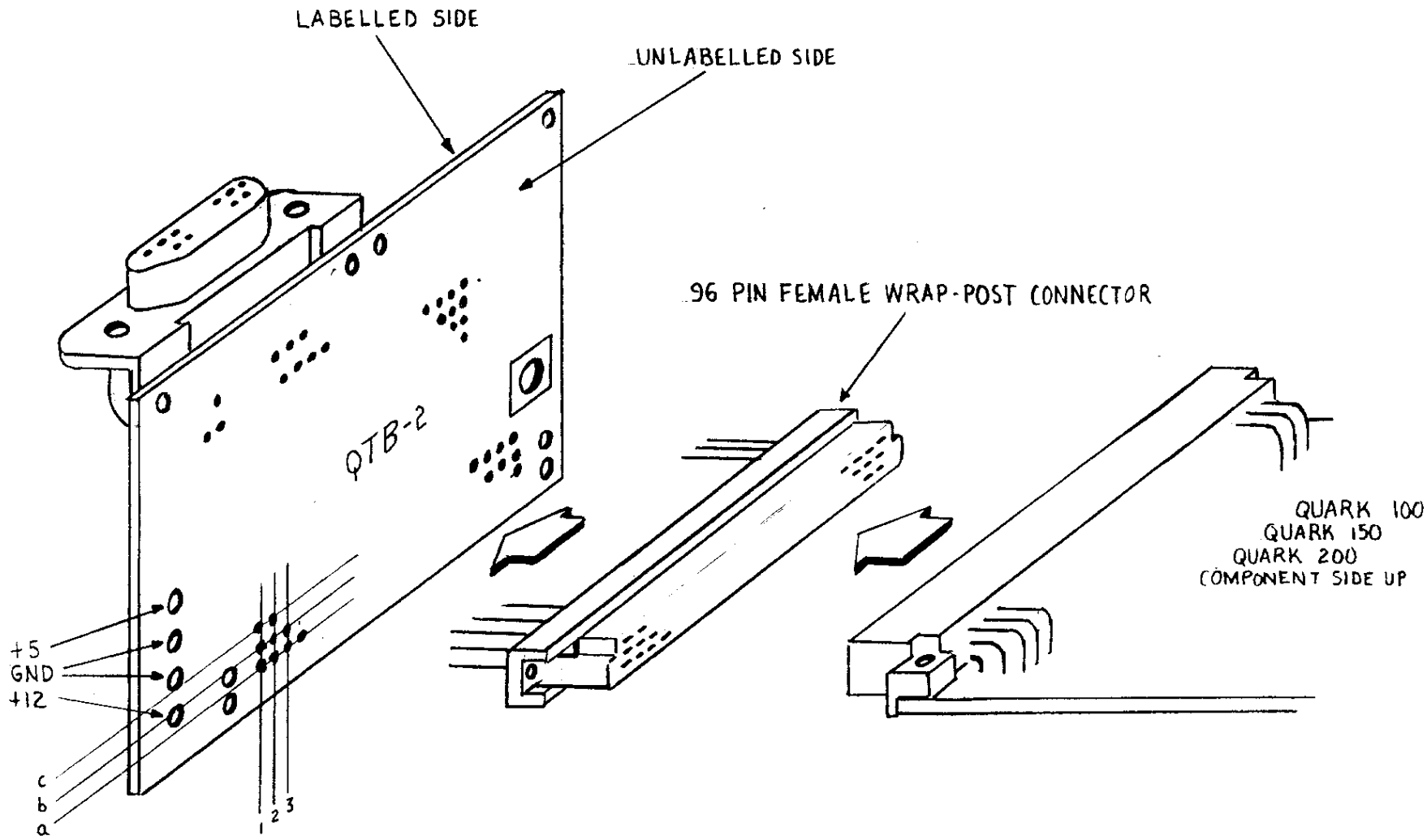


FIG. I/2.1 : QTB-2 CONNECTOR ORIENTATION

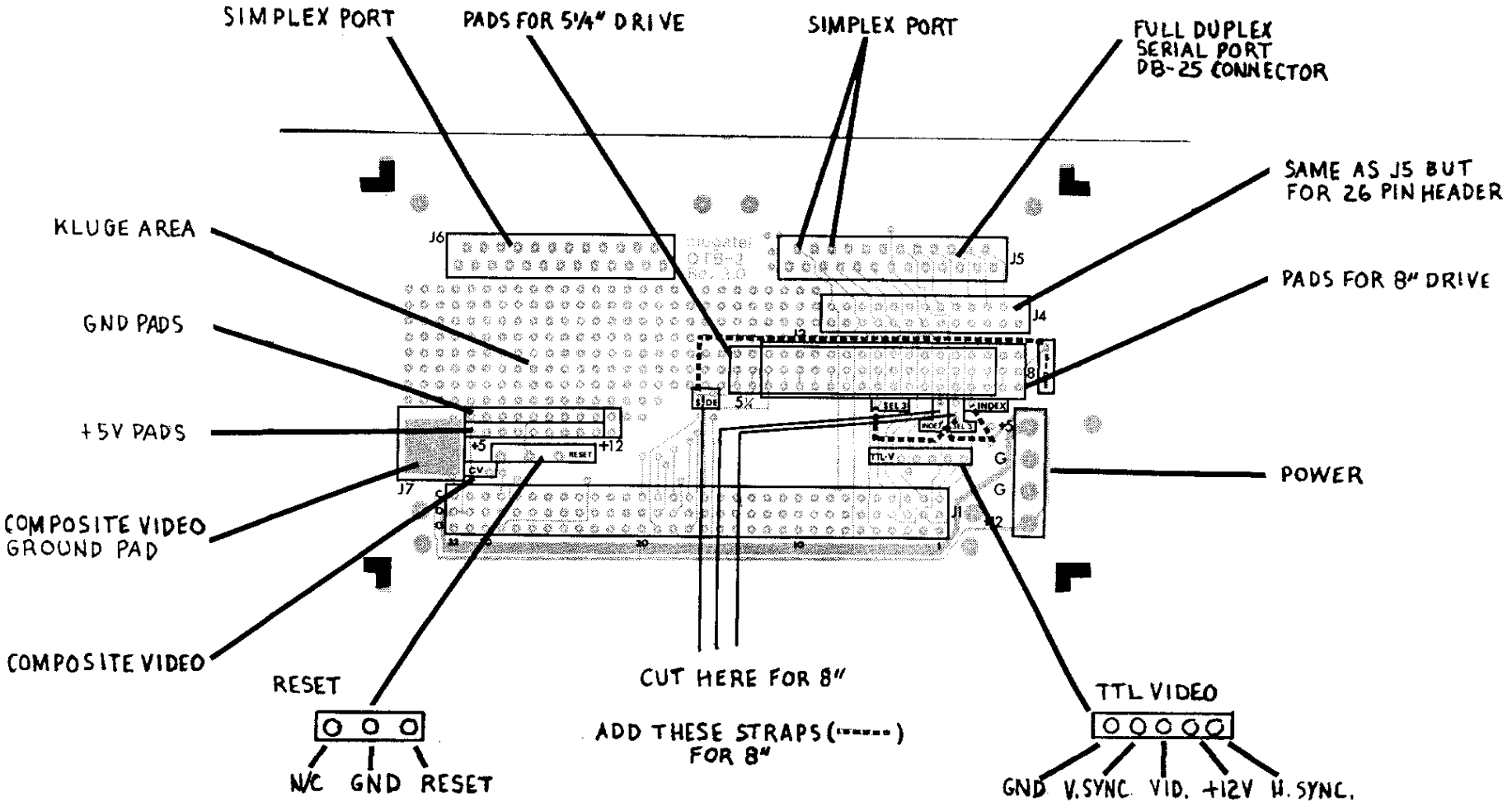


FIG. I/2.2 : QTB-2 LAYOUT (LABELLED SIDE)

3.0 The CP/M PLUS Software package for the QUARK

The Megatel QUARK is available with an extensive operating system and utility software package based on Digital Research's CP/M PLUS (CP/M V3.0) operating system. This package is available for use with most of the Megatel QUARK single-board computers, and may be ordered on either of the following sets of disc media:

-5-1/4 inch, double-density encoding (MFM), Megatel format, 96 tracks per inch, 80 tracks per side, single-sided Set of three (3) discs

-8 inch, double-density encoding (MFM), Megatel format 48 tracks per inch, 77 tracks per side, single-sided Set of two (2) discs

When shipped from the factory, all of the floppy diskettes are write-protected to guard against accidental erasure in an improperly-set-up system. It will not be possible for the QUARK to write on a write-protected diskette if the disc drive incorporates the appropriate write-protection mechanism, which is the case with almost all drives.

As shipped, the distributed operating system is configured with the following logical-to-physical I/O device assignments:

LOGICAL DEVICE	PHYSICAL DEVICE	
CONIN:	PKEY	DUPL
CONOUT:	CRT	DUPL
AUXIN:	DUPL	
AUXOUT:	DUPL	
LST:	CENT	

The peripheral interfaces on the QUARK computers and their identities under the CP/M PLUS BIOS are given in the table below:

PHYSICAL DEVICE ID	PHYSICAL DEVICE ASSIGNED
PKEY:	Input from the parallel keyboard port (PIA Port A + CA2)
DUPL:	Input from the full-duplex serial interface
CRT:	Output to the video display interface
DUPL:	Output to the full-duplex interface
SIMP:	Raw output to the simplex serial interface
LSTSIM:	Output to the simplex serial interface with protocol input line checking and automatic null insertion after CR/LF
LSTDUP:	Output to the full-duplex interface with automatic null insertion after CR/LF
CENT:	Output to the CENTRONICS-compatible parallel printer interface

3.1 The QUARK CP/M PLUS Operating System

After wiring up a prototype QUARK system (outlined in Section 2 in the Installation manual), an initial operating system will be ready to be loaded from the Distribution Diskettes. This operating system is configured by Megatel to operate under a "worst-case" system configuration - that is, a hardware configuration consisting of only one single-sided low-performance floppy disc drive, and either a keyboard/CRT console device or a RS-232C terminal.

When the initial operating system has been successfully loaded from one of the Distribution "SYSTEM" diskettes, the actual parameters of the "target" system may be entered using the submit procedure QINSTALL. The QINSTALL submit file contains menu-driven CP/M customization programs which have been written by Megatel specifically for use with the QUARK. QINSTALL allows the QUARK user to change the basic I/O configuration of the CP/M PLUS operating system without requiring a thorough understanding of the operation of the BIOS or of the programming of the Z-80B.

With QINSTALL the user may modify the parameters in the operating system for optimum performance with his final system hardware. As well, various I/O ports not needed for functions assumed by the Distribution operating system may be freed for other uses. Finally, other system parameters, such as the baud rates to be initialized when the operating system is loaded, can be specified through QINSTALL.

The CP/M PLUS operating system supplied on the Distribution Diskettes is configured to support one physical drive (i.e. a single floppy disc drive unit) and eight logical drives, named A:, B:, C:, D:, E:, N:, O:, and P:. Before installation of the system, the disc formats of logical drives A: to E: and N: will all be compatible with the "Megatel format" used on the Distribution Diskettes. Drive P: will be compatible with the "Interchange format" for the particular disc size (8-inch or 5-1/4-inch).

After the operating system is properly installed, the eight logical drives will support the following formats:

A: to E:	Format will correspond to the definition provided by the user-specified parameters entered through QINSTALL.
N:	Format will be compatible with the supplied CP/M Plus Distribution Diskettes
O:	Format will be compatible with Megatel Quark CP/M version 2.2 Distribution Diskettes
P:	For 8-inch diskettes, this format will be compatible with the single-density single-sided IBM 3740 format. For 5-1/4-inch diskettes, this format will be compatible with the Osborne single-density single-sided format

One to four physical drives can be supported in the installed CP/M PLUS system. If only one drive is used, then all logical drives will access the single physical drive. If more than one drive is used, then each logical drive will access the physical drive to which it was assigned during the Installation.

The N: logical drive will always be compatible with the format used for the QUARK CP/M Plus Distribution Diskettes. For users who previously purchased CP/M 2.2 Distribution software, logical drive O: will always be compatible with the version 2.2 Distribution format. These logical drives will be used after the operating system is installed to transfer files from the pertinent distribution diskettes, as well as for transferring files from diskettes obtained from Megatel in the future. On systems using 8-inch floppy disc drives, logical drive P: will be compatible with the IBM 3740 single-sided single-density 8-inch format. For 5-1/4-inch drives, the P: drive will be compatible with the Osborne single-density single-sided format. Thus drive P: is the so-called "Interchange format", and allows program exchange and the purchasing of software from commercial software vendors.

QINSTALL proceeds step-by-step through the CP/M customization process. It queries the user for parameters relating to the target system, and endeavors to ensure that mistakes due to erroneous entries are minimized. The end products of the installation procedure are as follows:

-a customized version of the QUARK CPM3EQU.LIB equate library
-a modified version of the system parameter file QSYS.DAT
-a modified version of the system file CPM3.SYS

At the completion of the QINSTALL procedure, a temporary CP/M system file will be written to the disc. A re-boot will load this temporary system. (A re-boot, or cold start, occurs whenever the QUARK is reset by power-on or externally, such as would be caused by closing the reset switch.) Depending on whether an alphanumeric-mode or graphics-mode terminal driver is being installed, one of the two submit procedures QASETUP.SUB or QGSETUP.SUB is run to assemble and link all the files required for the final system. At the completion of these setup procedures, the new CP/M system file will be complete.

The QINSTALL and QASETUP (or QGSETUP) installation procedures simplify the task of customizing the CP/M PLUS system. In addition to running these procedures, some other house-keeping tasks must be performed during the installation, such as formatting and copying diskettes.

For the complete Installation procedure, refer to Section 4 of the Installation manual.

This page intentionally left blank.

4.0 Installing the Customized CP/M PLUS System

To generate a customized CP/M Plus system for the target hardware configuration the following steps, which described in detail in the following sections, should be performed:

1. Make two sets of copies of the CP/M Distribution Diskettes. One of these sets of copies will be used in the course of the Installation. The other is a back-up copy for security purposes. (Sec. 4.3 and 4.4)

2. Start the first step of the automatic installation procedure, QINSTALL. This procedure is in the form of a CP/M SUBMIT file. The procedure will query the user regarding the parameters of the target system, and use the responses to the queries to create modified versions of QSYS.DAT and CPM3EQU.LIB, which are later used when creating the temporary system.

3. Re-boot the system and start the second installation procedure. QASETUP.SUB is used when an alphanumeric-mode terminal driver is being installed, while QGSETUP.SUB is used for the graphics-mode driver. These procedures will assemble and link all files required to create the target CP/M PLUS operating system.

4. Format a diskette with QCERT. Using COPYSYS write the new CP/M loader onto this diskette, and then PIP the new CP/M system files to this disc.

5. Finally, re-boot with this new CP/M System disc and transfer all of the files on the Distribution Diskettes to this diskette.

This completes the list of steps needed to install a CP/M PLUS system for the target QUARK system. Each of the steps is described in greater detail in the next several sections.

Throughout the software examples in this manual, all **boldface** text indicates a user response to a prompt or query from the computer. All prompts or communication which will be displayed on the screen will be indented. The examples will assume two physical drives. This convention is also followed in the CP/M Operating System Manual.

The convention followed here is that **logical** disc drives, which are the storage devices "seen" by the operating system, are identified by a letter followed by a colon (A:, E:, etc.) while physical drives, the real drive mechanisms themselves, are identified by the SELect line used when accessing that drive (SEL 0, SEL 2, etc.).

4.1 Operation of a one-drive system

This section is for users intending to set up a system with only one floppy disc drive. If more than one drive is to be used, this section may be ignored.

Although it is possible to use the QUARK with only one floppy disc drive, most users will find this configuration inconvenient because of the extra effort involved when making copies of files or entire diskettes. On such a one-drive system, a substantial amount of time will be spent removing and inserting diskettes in the single drive. However, if only one drive is available, or if the target system requires only one drive, it is possible to bring up and operate a one-drive system with any of the QUARK computers.

With a one-drive system, all of the logical drives access the same physical drive unit. With drive A: in operation (which is always the case when the CP/M system is first booted up), to access drive B: requires the removal of the diskette in the drive and the insertion of the diskette which is to be accessed on logical drive B:. Whenever logical drive B: is to be accessed after having previously accesed drive A:, the following prompt will be displayed on the screen:

Please mount disc B in Drive Ø, press <RET> when ready.

This prompt indicates that the diskette then in the drive should be removed, and that the "B:" diskette should be inserted. (RET indicates that the carriage return key should be typed.)

When logical drive A: is again accessed, the following prompt will be displayed:

Please mount disc A in Drive Ø, press <RET> when ready.

At this point, the "B:" diskette should be removed from the drive, and the "A:" diskette (the original diskette) be re-inserted in the drive. This disc-swapping procedure will take place any time a logical drive other than the presently logged-in drive is accessed. The same holds true for all logical drives. It is essential that when diskettes are swapped that the logical drive diskettes are not confused. Failure to keep the distinction between the different diskettes clear may result in data being written on the wrong diskette, or even the data being lost if the wrong format is used when writing.

4.2 Temporary enabling of the second drive on a two-drive system

As stated in Sec. 3.1, the CP/M operating system on the Distribution Diskettes is configured to support only one physical disc drive. Such a system can be tedious to use because of the necessity of removing and inserting diskettes into the single drive when copies of diskettes are being made. However, the CP/M system on the Distribution Diskettes can be patched directly to allow the use of a second drive.

To patch the system for two physical disc drives, the following program should be run:

```
A>QDSKTWO
```

QDSKTWO modifies the operating system in memory. Please note that the patch performed by QDSKTWO is temporary, if the system is re-booted, it will be necessary to run QDSKTWO again.

4.3 Formatting diskettes to make copies of the Distribution Diskettes

Before the QINSTALL program is run, a secure backup of the QUARK Distribution Diskettes should be made in the same format as is used on the Distribution Diskette. It is advised that two copies of each diskette received be made.

To make the copies, it is necessary first to format the appropriate number of diskettes on which the copies will be made. The utility routine QCERT.COM, which will write the same format as is used on the Distribution Diskettes, is used for this purpose.

The following examples will assume that there are two physical drives. Logical drive A being selected by drive 0 (SEL 0) and logical drive B selected by drive 1 (SEL 1).

To run QCERT, the example below should be followed:

```
A>QCERT

MEGATEL DISC FORMATTER ROUTINE VERSION 3.01

Which drive, (A,B...P then RETURN)? B

Formatting disc B with ..............

Put disc in B: in drive 1 and press return to continue  <ret>

        FORMATTING BEGUN ON DRIVE B
```

QCERT will indicate the number of tracks and sectors used in the format in the space indicated by the ".....", shown above.

When QCERT is complete, the following prompt will appear, to which the user should reply by typing the return key to re-boot the operating system, or by typing any other key to format another diskette.

```
****** HIT RETURN to Reboot, any other key to restart
```

The user should format four 8 inch or six 5-1/4 diskettes initially using QCERT. These diskettes will be used to make backup copies of the Distribution Diskette. It is essential that the diskettes be inserted in the "B:" drive and not in the "A:" drive. To ensure that the Distribution Diskette will not be accidentally erased, the write-protection feature on the diskette should be enabled if the floppy disc drives in use recognize a write-protected diskette. (The user is reminded that covering the notch will write-protect 5-1/4-inch diskettes, while exposing the notch will write-protect 8-inch diskettes.)

4.4 Making backup copies of the Distribution Diskettes

The utility routine QCOPY.COM is included on the distribution diskettes. This utility is used primarily to make backup copies of diskettes. Two sets of copies of the Distribution Diskettes should be made on diskettes formatted using the QCERT.COM utility. (See Sec. 4.3) To make the copies of the Distribution Diskettes, insert each diskette in drive A, and follow the example shown below.

```
A>QCOPY

MEGATEL COMPUTER Quark Disk copy Routine version 3.01

Please enter the source disk
      (Disk to be copied) A

Please enter the destination disk
      (Disk to be copied to) B

Please confirm copy of drive A to drive B (Y/N)? Y

Copy completed.

Hit RETURN to reboot, any other key to restart
```

When the copy has completed, hitting any key but the RETURN key will cause the copy utility to be run again. The two diskettes can be removed and another of the Distribution Diskettes can be inserted into drive "A:" to be copied to a blank diskette in drive "B:". When complete, two exact copies of the Distribution Diskettes including the CP/M Operating System will have been made. The Master Distribution Diskettes should now be stored in a safe place. The discs remaining should be labelled as follows:

8-inch	"Work"	System disc Source	(one set of copies)
	"Back-up"	System disc Source	(the other set)
5-1/4-inch	"Work"	System disc Source disc Utility disc	(one set of copies)
	"Back-up"	System disc Source disc Utility disc	(the other set)

The "Work" copies of the diskettes will be used during the installation of the operating system. Some of the files on these discs will be modified by the Installation procedure. The "Back-up" copies are intended as additional security against accidental loss of data.

4.5 Running the CP/M PLUS QINSTALL procedure

This section should not be attempted without having made backup copies of the Distribution Discs (Sec 4.4).

To run the QINSTALL procedure, specifications for the floppy disc drives to be used should be at hand. The QINSTALL program will require some information regarding the characteristics of the disc drives in order to provide a nearly-optimal implementation of CP/M Plus.

Because of the numerous system configurations possible with the QUARK, only a limited set of combinations can be created using QINSTALL. Most users should find the configurations provided by QINSTALL adequate. Advanced users may wish to modify the source files themselves to tailor their operating system more closely to their needs.

To run the installation procedure, insert diskettes into the drives as follows. (It is assumed the two physical drives are being used throughout the installation process; for one-drive systems prompts will be displayed when diskettes must be swapped.)

```
8 inch      - "Work" System disc in drive 0
              "Work" Source disc in drive 1

5-1/4 inch  - "Work" System disc in drive 0
              "Work" Source disc in drive 1
```

Type the following:

A>SUBMIT QINSTALL

This "SUBMIT" command causes a number of CP/M Plus commands and programs to be executed in a sequential fashion. The file QINSTALL.SUB contains the list of commands to be automatically executed, the first of which is QINSTALL.COM. It will clear the screen and display its version number, and then produce the queries described below.

The default values for each query will be displayed after the "ENTER - " prompt. The value used as the default is taken from the file QSYS.DAT, which is setup on the Distribution Diskettes with values suited for the disc size distributed. Entering a "carriage return" as the response to a query will cause the displayed default value to be entered.

To install the operating system properly, one must select and answer all queries from options A, B, and C. Any queries for which a response is not entered will assume the default values. Option D must then be selected to update QSYS.DAT with the target system parameters and to continue with the alteration and assembling of the source files.

```
SYSTEM INSTALLATION PROGRAM FOR CP/M 3.0 Version 3.01

Please select option A, B, or C to enter the CP/M
system's configuration for your MEGATEL QUARK.
When you have completed the configurations, select
option D to continue with the installation
procedure or select option X to abort.  If option
D is selected, all user input will be saved in
QSYS.DAT, and will be used as defaults for the
next installation run.

A - DISC DRIVE HARDWARE SPECIFICATIONS
B - DISKETTE FORMAT SPECIFICATIONS
C - OTHER PERIPHERALS CONFIGURATION
D - CONTINUE THE INSTALLATION PROCEDURE
X - ABORT PROCEDURE

Enter -
```

Each of the above menu selections is described in the sections which follow.

4.5.1 Selection 'A' - Disc Drive Hardware Specifications

When menu selection **A** is chosen, the following queries relating to the physical disc drive hardware specifications will be given. The number of drives to be connected to the system, the masks and ports for the side-select, the low-write-current-select, and the auto 5/8-select lines are entered only once. All other drive hardware queries will be asked once for each physical drive in the system depending on the number of drives having been entered in the first query. If the hardware specifications to be entered for a drive are identical to those entered for the previous drive, the specifications may be copied to save effort.

```
Number of physical drives
 1.  One drive unit
 2.  Two drive units
 3.  Three drive units
 4.  Four drive units
 Enter -
```

The QUARK provides buffered drive select lines for up to four floppy disc drives. Each floppy disc drive connected should be set up to be selected by one of the four lines, which are called SEL 0, SEL 1, SEL 2, and SEL 3.

```
Enter the port address to be used
 for the low write-current line
 Minimum of 95
 Maximum of 255
Enter --
```

Except for those lines dedicated for specific functions, any of the Quark's parallel I/O lines may be used for the floppy disc low write-current control line. Bits 0-4 on port B of the PIA (68A21, I/O address 76h) are normally intended for use as drive select, side select, or low write-current outputs, and are buffered by medium-current non-inverting TTL drivers, making them suitable for these purposes. Other parallel I/O lines on the Quark are MOS outputs, and must be buffered in order to drive standard disc drive inputs. Also note that due to the fact that the contents of the input buffers on most of the parallel I/O devices reflect the state of the output pins rather than the contents of the output registers, it is possible for other lines which are configured as outputs on the same port which is being used for a floppy-disc control function to be inadvertently changed when the control lines are toggled. If the low write-current is not invoked a value of 127 may be entered.

```
Give the mask for the
 low write-current line
 Minimum of 1
 Maximum of 255
 Enter 'X' if the above is not required
Enter --
```

The mask referred to is the 8-bit binary pattern that will be used to determine which of the eight lines of the port chosen in the previous query is to be used as the "low write-current" line. The value of mask will be the decimal equivalent of the 8-bit binary number used as the pattern. The pattern itself consists of eight bits, seven of which are '0', with the solitary '1' in the bit position corresponding to the I/O line to be used for the low write-current output. For convenience, the table below gives the decimal values to be entered to use any one of the eight lines of a port to be used as the low write-current line.

MASK (DECIMAL)	LINE USED FOR LOW WRITE-CURRENT	CORRESPONDING PIN ON QUARK
01	SEL 0	B-19
02	SEL 1	C-19
04	SEL 2	C-18
08	SEL 3	B-18
16	SIDE	A-19

MASK (DECIMAL)	LINE USED FOR FUNCTION
1	BIT 0
2	BIT 1
4	BIT 2
8	BIT 3
16	BIT 4
32	BIT 5
64	BIT 6
128	BIT 7

The mask will be "ANDed" to the value at the given port to release the line, and its complement will be "ORed" to set the line.

```
Give the mask for the
 side-select line
 Minimum of 1
 Maximum of 128
Enter --


Automatic 8/5-1/4 inch selection
 by parallel I/O line?
 1. Enable auto-select line
 2. Disable auto-select line
Enter --
```

Under Megatel's CP/M 3.0 implementation, mixed drive configurations (in which both 8-inch and 5-1/4-inch floppy disc drives are connected) are supported through this installation process. This query determines whether a parallel I/O line will be used to indicate that an 8-inch or 5-1/4-inch drive is currently selected. If this feature is enabled, the line chosen for this purpose can be connected to the Quark's Drive Size Select line to automatically switch operating modes for the internal Floppy Disc Interface.

```
Enter the address of port
 to use for auto-select
 Minimum of 1
 Maximum of 155
Enter --
```

As Write-current line, any parallel I/O line may be used for the Auto-select function. If the auto select is not invoked a value of 127 should be used.

```
Enter mask value for
 auto-select line
 Minimum of 1
 Maximum of 255
 Enter 'X' if the above is not required
Enter --
```

Again, a mask must be entered to determine which I/O line of the chosen port will be used for the Auto-select line.

This completes the set of disc drive hardware specifications which are common to all drives connected to the Quark. The Install program will now proceed with queries more directly related to each floppy disc drive. This set of queries will be given once for each physical drive in the system. (Again, "physical drive" refers to a single drive mechanism. For this installation, these are numbered from 0 up to 3, depending on the number given in the earlier query.)

```
DRIVE UNIT 0 SPECIFICATIONS

Size of drive
 1.  5-1/4 inch
 2.  8 inch
Enter --
```

The response to this query determines what type of drive the system expects for the current physical drive.

```
Number of Tracks Per Inch
 1.  96 TPI
 2.  48 TPI
Enter --
```

While none of the Megatel Quark software actually requires this information, it is saved in the QSYS.DAT file for completeness.

```
Stepping rate mode
 1.  Hardware stepping rate
 2.  Software stepping rate
Enter -
```

Step pulses for the positioning of the read/write head on the floppy disc drive may be generated by the 1793 floppy disc controller (selection 1), or by a software timing loop in the drive-handling routines (selection 2). If one of the available hardware stepping rates (shown in the table below) is appropriate for the current drive, then hardware stepping rates can be selected. Otherwise, the software rate should be selected.

```
If hardware stepping rate
    5-1/4 inch          8 inch
 1.  6  ms               3  ms
 2.  12 ms               6  ms
 3.  20 ms               10 ms
 4.  30 ms               15 ms
Enter --

For software stepping rate
 Minimum of 3   msec
 Maximum of 300 msec
Enter --
```

The stepping rate should be equal to or greater than the rate specified for the floppy disk drives. If the drives require an extra-long interval between pulses in some circumstances, such as when changing directions, the maximum required interval (or greater) should be installed.

```
Motor start time
 Minimum of 0.1 msec
 Maximum of 650 msec
 Enter 'X' if the above is not required
Enter --
```

The "motor start time" is the delay inserted between the end of the head load operation and the beginning of any write operation. It must be less than five disc revolutions or the 1793 may release the drive.

```
Wait time for the head load operation
 Minimum of 0.5  msec
 Maximum of 100  msec
Enter --
```

The "wait time for the head load operation" is the delay inserted between the selecting of a drive and the beginning of most operations. Note that if the operation is a write operation, the motor start delay will be added to the head load delay.

```
Head settling time after positioning
 Minimum of 0.1  msec
 Maximum of 100  msec
Enter --
```

The "head settling time after positioning" is the delay inserted after any operation in which the heads are moved, except certain operations used when booting up, or when certain errors occur.

```
Delay between drive selects in msec
 Minimum of 0.1  msec
 Maximum of 1000 msec
 Enter 'X' if the above is not required
Enter --
```

The "delay between drive selects" is inserted after one physical drive is de-selected and another physical drive is selected. This allows time for the power supply to recover from the surge which can occur when a drive is first selected. This surge can be aggravated when one drive is selected very soon after another. This delay is not used when switching between different logical drives running on the same physical drive.

```
Tunnel erase delay
 Minimum of 0.5  msec
 Maximum of 10   msec
Enter --
```

After a write operation, the write head must not be allowed to move from one track to another until the tunnel erase has been turned off. The "tunnel erase delay" is the time required by the drive to turn off the tunnel erase following a write operation.

```
If low-write current is required, at
 which physical track is it invoked
 Maximum of 512  trks
 Enter 'X' if the above is not required
Enter --
```

Enter the number of the track at or above which the low write-current line will be active. The "physical track number" is the track number for the drive mechanism itself, and may be specified by the drive manufacturer. Track 0 may not be chosen as the track at which low write-current is enabled. Most newer disc drives do not require an external low write-current control line.

```
Number of retries for read
 and write operation
 Minimum of 1
 Maximum of 255
Enter --
```

When the 1793 floppy disc controller detects an error during a disc read or write operation it will retry the operation 5 times, if there is still an error the operating system will normally instruct the controller to repeat the operation. The number of retries is the number of such attempts the system will permit before reporting the problem to the operator and asking if further attempts are to be made. Fifteen to thirty retries is a recommended range of values; values much greater will cause excessive delays when genuine read or write errors do occur, while substantially smaller values will allow uneccessary error reports for soft disc errors.

```
Number of seek retries for
 verification operation
 Minimum of 1
 Maximum of 255
Enter --
```

In order to verify that the correct track and sector have been found, a 1793 "SEEK" operation is performed. If not found the entire procedure to find the desired track and sector is performed until either the track and sector is found or the number of seek retries is reached. This "Number of seek retries" is the number of times this seek procedure will be performed before an error is reported.

```
PHYSICAL DRIVE UNIT 1 SPECIFICATIONS

Same as drive unit 0 (Y/N)
```

After the specifications for one physical drive unit have been entered, the installation program will repeat the process until specifications have been entered for as many physical drives as were specified earlier. If the specifications for the physical drive just given are the same as those for the next drive to be entered, this query may be answered "Y" to duplicate the specifications. If "N" is entered, then the specifications for the next unit may be entered by the same process.

4.5.2 Selection 'B'- Diskette format specifications

The queries of this section are used to set up the format used on each "logical disc drive".

```
Drive for Megatel distribution
 format (logical drive N:, O:, P:)
 will be selected by
 1.   SEL 0
 2.   SEL 1
 3.   SEL 2
 4.   SEL 3
Enter --
```

The QUARK has four buffered drive select lines, SEL 0, 1, 2, and 3. The response to this query determines which of these four outputs will go active (logic low) when the operating system wishes to access the given logical drive. The format used by each of the logical drives assigned to a particular physical drive must be compatible with that drive (i.e., a logical drive using a double-sided format cannot be assigned to a single-sided disc drive.)

```
LOGICAL DRIVE A FORMAT
Single or double density
 1.  Single density
 2.  Double density
Enter --
```

The response to this query determines the recording format used on the diskette. Single density is FM, and double density is MFM.

```
Single or double sided
 1.  Single sided
 2.  Double sided
Enter --
```

The response to this query determines whether one or both sides of the diskette will be used for recording. In the double-sided mode, the Side Select line will be used to select the surface of the diskette used in any read or write operation. The choice of an I/O line to use as the Side Select line is left to the user through the responses to queries at the beginning of menu selection 'A'. In the double-sided mode, consecutive logical track numbers refer to tracks on opposite sides of the diskette. With standard floppy-disc drives (and assuming that the Side Select line represents the true state of the disc handler's output), Track 0 will be the non-labelled side of the diskette.

```
Number of tracks per side
 Minimum of 10  trks
 Maximum of 255  trks
Enter --
```

The response to this query determines the number of tracks used on each side of the diskette.

```
Double tracking
 1.   Use double tracking
 2.   Do not use double tracking
Enter --
```

If double tracking is invoked, two step pulses will be issued each time the heads moved in or out one track, so that alternate physical tracks are skipped. This feature allows 48TPI diskettes to be read or written on a 96TPI drive. However, it is often not possible to read a diskette on a 48TPI drive if the diskette was ever written using 96 tracks per inch, since the data between the 48TPI tracks will interfere with the data on the proper tracks. Thus, when writing in a 48TPI format using a 96TPI drive, it is not recommended to use a diskette that has ever been written using 96TPI.

```
Physical sector size
 1.  128 bytes
 2.  256 bytes
 3.  512 bytes
 4.  1024 bytes
Enter --
```

CP/M 3.0 supports variable sector sizes. The physical sector size is the number of bytes per sector on the diskette. The Megatel Quark CP/M 3.0 software is distributed in a format using nine 512-byte sectors per track. Choosing 1024 bytes/sector will generally yield the greatest capacity, but some empirical results suggest that small variations between drives can make interchange in this format less reliable.

```
Number of physical sectors per track
 Minimum of 1
 Maximum of 48
Enter --
```

A physical sector is the unit of data read or written by the floppy disc controller. The following table shows the estimated bytes per track available:

Recording method	8 inch	5 1/4 inch
Single Density (FM)	5208	3125
Double density (MFM)	10416	6250

```
Skew factor
 Minimum of 1
 Maximum of 8
Enter --
```

The skew factor is the number of physical sectors skipped between consecutive logical sectors. The format used on Megatel Quark CP/M 3.0 Distribution Diskettes has a skew factor of 2.

The motivation for skewing (or interlacing) physical sectors is to allow the system sufficient time after reading or writing one logical sector to process its buffers and then be ready to read or write the next logical sector. If that sector were the very next physical sector on the track, the system would likely miss the beginning of the sector, and thus have to wait for an entire revolution of the disc to get the sector again. By skipping several physical sectors between sequential logical sectors, the response time of the system can be greatly improved. See the CP/M manuals for a full discussion of this topic.

```
Gap size
 Minimum of 4  bytes
 Maximum of 31 bytes
Enter --
```

The gap size is the the number of bytes between the end of the data field and the ID mark. Its purpose is to act as a "safe zone" to allow for uncertainty in the position of the end of the data field. This uncertainty is caused by variations in the disc rotation speed and other factors.

```
Block size
 1.  1024 bytes
 2.  2048 bytes
 3.  4096 bytes
 4.  8192 bytes
 5.  16384 bytes
Enter --
```

A block is the smallest unit of file storage recognized by CP/M. A small block size improves disc utilization efficiency, while a large block size increases disc access speed. A popular compromise is 2048 bytes. See Sec 3.3 in the CP/M Plus System Guide.

```
Number of reserved tracks
 Minimum of 1
 Maximum of 4
 Enter 'X' if the above is not required
Enter --
```

The reserved tracks are those that are set aside for the operating system. Megatel CP/M 3.0 Distribution Diskettes use three reserved tracks for the 8-inch format, and four reserved tracks for the 96 TPI 5-1/4-inch format. The reserved tracks are only required on logical drive "A:". The other logical drives do not require any reserved tracks unless compatability with drive "A:" is required.

```
Number of directory entries
 Minimum of 64
 Maximum of 1024
Enter --
```

The number of directory entries is dependent on the aplication of the particular logical drive. If many small files are going to be stored on the drive then a greater number of directory entries will ensure against using up all the directory space while still having plenty of storage on the diskette.

```
This logical drive to be selected by
 1.  SEL 0
 2.  SEL 1
 3.  SEL 2
 4.  SEL 3
Enter --
```

The QUARK has four buffered drive select lines, SEL 0, 1, 2, and 3. The response to this query determines which of these four outputs will go active (logic low) when the operating system wishes to access the given logical drive. More than one logical drive can be assigned this way to a single physical drive, although the format used by each of the logical drives assigned to a particular physical drive must be compatible with that drive (i.e., a logical drive using a double-sided format cannot be assigned to a single-sided disc drive.) The only restriction to this question being that logical drive A: must always be selected by SEL 0. This is to enable the "Boot PROM" to access the proper drive in order to boot up.

```
FORMAT FOR LOGICAL DRIVE B
Same as drive A
(Y/N)
```

```
This logical drive to be selected by
 1.   SEL 0
 2.   SEL 1
 3.   SEL 2
 4.   SEL 3
Enter --
```

After the format for one logical drive has been entered, the installation program will proceed to the next drive. If the new drive has the same format as the previous drive, this query may be answered 'Y' to copy the previous specifications. If the next drive differs, then a negative response will cause the install program to repeat the queries for the new drive. This process will continue until formats for all of the logical drives have been entered.

4.5.3 Selection 'C' - Other Peripherals Configuration

The queries of this section are used to set up the configuration for the various serial and parallel I/O interfaces on the Quark which are under the control of CP/M 3.0. The name of each logical device appears in parentheses next to the name of the hardware interface.

```
Console input (CONIN:)
 1. Input from full-duplex serial port (DUPL:)
 2. Input from parallel keyboard port  (PKEY:)
 3. Input from full-duplex serial port (DUPL:)
    and parallel keyboard port (PKEY:)
Enter --
```

The response to this query determines which of the Quark's peripheral interfaces will be used for CP/M console input. With selection 3 characters arriving from either port will be interpreted as console input.

```
Console output (CONOUT:)
 1. Output to full-duplex serial port (DUPL:)
 2. Output to video display interface (CRT:)
 3. Output to full-duplex serial port (DUPL:)
    and memory-mapped CRT monitor (CRT:)
Enter --
```

The response to this query determines which of the Quark's peripheral interfaces will be used for CP/M console output. With selection 3 characters will be sent to both the full-duplex port and the video display interface.

```
Auxiliary input (AUXIN:)
 1. Input from full-duplex serial port (DUPL:)
 2. Input from parallel keyboard port  (PKEY:)
Enter --
```

This determines the function of the CP/M Auxiliary Input logical device.

```
Auxiliary output (AUXOUT:)
 1.  Output to the full-duplex serial port (DUPL:)
 2.  Output to the simplex serial port     (SIMP:)
 3.  Output to the parallel printer port   (CENT:)
 4.  Output to the video display interface (CRT:)
Enter --
```

This determines the function of the CP/M Auxiliary Output logical device.

```
List device (LST:)
 1.  Output to the full-duplex serial port (LSTDUP:)
 2.  Output to the simplex serial port     (LSTSIM:)
 3.  Output to the parallel printer port   (CENT:)
 4.  Output to the video display interface (CRT:)
Enter --
```

The response to this query determines the peripheral interface on the Quark to which CP/M LST: data will be sent.

```
List device end-of-line to next-line delay.
 Maximum of 255 nulls
 Enter 'X' if the above is not required
Enter --
```

Some printers require that there be a delay between sending a carriage return character to the printer and the transmission of the next printable character. The number of "null" characters (=00hex) sent after a carriage return can be specified here.

```
Simplex Serial Port baud rate
 1.  9600 baud
 2.  7200 baud
 3.  4800 baud
 4.  3600 baud
 5.  2400 baud
 6.  1800 baud
 7.  1200 baud
Enter --
```

The baud rate for the Quark's Simplex Serial port, which uses the Shift Register of the VIA, will be initialized when the CP/M system is first booted up. The response to the query determines what baud rate will be used. See Table II in the Appendix for more information on these baud rate values.

```
Full duplex serial port baud rate
 1.  9600   baud
 2.  7200   baud
 3.  4800   baud
 4.  3600   baud
 5.  2400   baud
 6.  1800   baud
 7.  1200   baud
 8.   600   baud
 9.   300   baud
10.  150    baud
11.  134.5  baud
12.  110    baud
13.   75    baud
14.   50    baud
Enter --
```

Similarly, the baud rate for the Quark's Full-duplex Serial port will be initialized when the CP/M system is first booted up. The response to the query determines what baud rate will be used. See Table II in the Appendix for more information on these baud rate values.

```
Display mode for the video display interface
 1.  Alphanumeric mode
 2.  Graphic mode
Enter --
```

Two terminal drivers are provided for use with the Quark operating system. Both display alphanumeric data, using a user-definable character set. Choosing "Alphanumeric mode" will cause the terminal driver which runs in the Quark's alphanumeric display mode to be linked in when assembling the operating system. Choosing the "Graphic mode" will link in the the terminal driver which runs in the Quark's graphics display mode. Both terminal drivers have more or less the same features and accept the same terminal control codes, which appear in Table XI of the Appendix. The alphanumeric-mode driver uses less memory for video display and operates slightly faster. Options in the graphics-mode driver permit longer horizontal retrace periods and can be used as the starting point for users wishing to implement additional graphics features.

The number of rows on the video display is determined by the next three queries. In the alphanumeric-mode terminal driver, three distinct screen areas are available. Each screen is independent of the other, and can be installed with any number of rows, as long as the total does not exceed the capacity of the video display memory. (The maximum is 30 rows for 60Hz units, 38 for 50Hz, and possibly fewer depending on the vertical retrace interval required by the monitor. The total number of lines used is the sum of the installed DISPLAY, STATUS and MAIN screens.) The upper and lower screens (DISPLAY and STATUS) may be individually included or ommitted from the terminal driver. In the graphics-mode terminal driver, only the MAIN screen is available.

```
If the alphanumeric mode is chosen, number
 of rows for the DISPLAY screen
 Enter 'X' if the above is not required
Enter --
```

The DISPLAY screen is the uppermost set of rows on the video display. It ends at the start of the MAIN screen, and includes the number of rows given as the response to this query. The top row of the DISPLAY screen will be the upper-most row of the video display. If no DISPLAY screen is required, enter an 'X'. The display screen feature is not implemented in the graphics-mode terminal driver.

```
For graphic mode  or alphanumeric mode, number
 of rows for the MAIN screen
 Minimum of 1
Enter --
```

Using the alphanumeric-mode terminal driver, the MAIN screen ends at the start of the STATUS screen, or, if the STATUS screen is not included (by entering 'X' to the next query), at the bottom-most row on the video display. With either the alphanumeric- or graphics-mode terminal drivers, the size of the MAIN screen will be determined by the response to this query. The MAIN screen is the display which is used when the operating system boots up.

```
If the alphanumeric mode is chosen, number
 of rows for the STATUS screen
 Enter 'X' if the above is not required
Enter --
```

The STATUS screen is the bottom-most set of rows on the video display. The number of rows making up the STATUS screen will be determined by the response to this query. The last row of this screen corresponds to the last 96 bytes in the video display memory. If no status screen is to be included in the terminal driver, an 'X' should be entered for this query. The STATUS screen feature can be implemented only in the alphanumeric-mode terminal driver.

```
If the alphanumeric mode is chosen
 1.  Enable the clock display
 2.  Disable the clock display
Enter --
```

The clock may be displayed only in the alphanumeric-mode terminal driver.

```
If the clock display is enabled
 1.  Display at top left corner
 2.  Display at bottom right corner
Enter --
```

The clock may be displayed at either of two positions on the screen. The "top left corner" position is in the third character row from the top of the CRT display (i.e., 17 scan line periods after the beginning of the vertical retrace). The clock display occupies 10 character positions, and starts in the 13th column from the left. The "bottom right corner" is in the bottom-most row of the screen, regardless of whether there is a STATUS screen in use. The last chacater of the clock display appears in the 5th column from the right side of the screen. The positioning of the clock display is independent of the size of the DISPLAY, MAIN, and STATUS screens.

```
If the Graphics display mode is chosen,
 select column offset from the left side
 of the screen
 Minimum of 1
 Maximum of 7
 Enter 'X' if the above is not required
Enter --
```

The offset referred to is the number of 1-byte wide character columns offset from the left edge of the screen. Offsetting the start of the data field allows more time for the horizontal retrace on those monitors requiring this.

```
Colour selection for graphics-mode
 video display:
 Foreground colour:
 1.  Black           9.  Brown
 2.  Burgundy        10. Orange
 3.  Dark Blue       11. Grey
 4.  Mauve           12. Pink
 5.  Dark Green      13. Green
 6.  Dark Grey       14. Beige
 7.  Cyam            15. Turquoise
 8.  Blue            16. White
 Minimum of 1
 Maximum of 16
 Enter 'X' if the above is not required
Enter --
```

```
Background colour:
 Minimum of 1
 Maximum of 16
 Enter 'X' if the above is not required
Enter --
```

In the graphics-mode terminal driver for the Quark/150, the foreground and background colours may be specified.

When menu options **A**, **B**, and **C** have been completed to the user's satisfaction, the **D** option should be chosen.

Function **D** will continue with the QINSTALL submit procedure. The first file to be altered is the system parameter file QSYS.DAT on the work diskette. The default values obtained from the copy of QSYS.DAT on the Distribution Diskette will be replaced by the parameters entered during QINSTALL which were until this point stored in memory.

The QINSTALL submit procedure will run approximately 10 minutes before it reaches the point of continuing with the temporary system (Section 4.5.4).

4.5.4 Final steps in the Installation procedure

At the end of this phase of the Installation, a temporary CP/M 3.0 system file will be written to the diskette in drive 0. A prompt will appear on the screen requesting the user to reset (caused by closing the reset switch) to load the temporary system.

The formats of the logical drives used in this system are as follows:

- -The temporary system's drive 'A:' format is the same as the Distribution Disc format
- -The temporary system's drive 'B:' format is the same as the Distribution Disc format
- -The temporary system's drive 'N:' format is the same as the target system's drive 'A:' (system drive)

Therefore, the user can read or write the target system format through drive N: while operating under the temporary system. Additionally, he can read and write to the Distribution Diskettes through drive A: or B:.

4.6 The New CP/M Plus system and disc format

The second part of the Installation procedure requires another submit to be executed. The submit to be executed is dependent on the mode chosen for the video output. If the alphanumeric mode is to be used then **QASETUP.SUB** is executed. If the graphics mode is to be used **QGSETUP.SUB** is executed. To continue with the proper procedure enter one of the following:

```
A>SUBMIT QASETUP
```

if the alphanumeric-mode terminal driver is being installed,

-or-

```
A>SUBMIT QGSETUP
```

if the graphics-mode terminal driver is being installed.

The second part of the installation procedure will involve approximately 20 minutes and will produce a system file called CPM3.SYS and a loader file called CPM.LDR. The CPM3.SYS file is configured to the target systems specifications and the CPM.LDR file is the file used by COPYSYS to write to the reserved track of a diskette in order to load the target system. The CPM3.SYS file is renamed to CPM3.NEW and the original CPM3.SYS file (the one which is on the Distribution disk) is restored.

4.6.1 Formatting a diskette under the *new format*

To format a new diskette with the user's customized format, under the temporary system, simply run QCERT and format the logical **N:** drive.

```
A>QCERT

MEGATEL DISC FORMATTER ROUTINE VERSION 3.01

Which drive, (A,B...P then RETURN)? N

Formatting disc N with ..............

Put disc in N: in drive 1 and press return to continue  <ret>

        FORMATTING BEGUN ON DRIVE N
```

4.6.2 Writing the new system on a diskette

After formatting one or several diskettes with the user's customized format, the customized operating system may be written onto one of these diskettes. To do this, the programs COPYSYS.COM and PIP.COM are used. These files are included on the Distribution Diskette, and hence should also appear on the "Work" disc copy of the Distribution Diskette.

To use the COPYSYS utility at this point, follow the example presented below.

```
A>COPYSYS CPM.LDR
CP/M 3 COPYSYS - Version 3.0
Destination drive name (or return to reboot) N
Destination on N then type return <ret>
Destination drive name (or return to reboot) <ret>

A>
```

By specifying CPM.LDR, COPYSYS will read the file CPM.LDR from drive A: and write the file to the reserved tracks of the diskette in drive N:. The file CPM.LDR was created in the second half of the Installation procedure. Copysys waits for the proper disc to be inserted, after which the user types a carriage return to start the copy process.

The PIP.COM program is used to copy the files needed to load the new system. To copy the files needed the following should be entered.

```
A>PIP N:CPM3.SYS=A:CPM3.NEW [VO]

A>PIP N:=A:CCP.COM [VO]
```

After generating and writing the customized CP/M system, this diskette can now be used as the "system diskette". What remains to be done is transferring of the CP/M and Megatel utility files from the Distribution Diskette (or the "Work" copy) to the user's system diskette and testing the customized system.

4.6.3 Booting the new system

To verify that the new CP/M system can be properly loaded, insert the new system diskette into the "A:" drive, and reset the QUARK system. If the system operates properly, the new system will be loaded and the "A> " prompt will be displayed on the screen. At this point, utilities and other files on one of the copies of the Distribution diskette may be copied onto the new system diskette. For details on drive assignments of the new system, please refer to Section I/3.1.

If the boot fails, improper data may have been entered in the QINSTALL procedure. It is recommended that the user restore all the original files from the Distribution Diskettes to their "Work" discs, and then return to the beginning of section 4.0 to repeat the Installation procedure.

4.7 Transferring files from the Work Diskette

Having successfully booted the customized CP/M system from the new System Diskette, the various files outlined in section 3.0 may be transferred from one of the copies of the Distribution Diskette to the new System Diskette. This will be done using the CP/M utility PIP (for Peripheral Interchange Program). The file PIP.COM should be on the "Work" diskette. All of the files on this diskette can be copied **EXCEPT CPM3.SYS**.

As PIP copies each file from drive N: to drive A:, the names of the files will be displayed. Having completed the transferring of all of the files, additional copies of the new system diskette may be made on other diskettes formatted using the QCERT utility.

At this point, the following diskettes should be on hand:

1. The Distribution Diskettes received with the QUARK. These diskettes should be stored in a safe place.

2. An exact set of backup copies of the Distribution Diskettes, labelled "Back-up".

3. A diskette with the CP/M system used on the Distribution Diskette and new utilities, labelled "Work". This diskette, used in running the QINSTALL procedure, can be re-formatted.

4. The new master Target System Diskette together with all backup copies made.

Logical drive N: will always remain compatible with the Megatel format used on the Distribution Diskette, so programs, data, or

utilities on the Distribution Diskette, as well as any future software releases from Megatel, may be copied to the user's customized format simply by copying from drive N: to one of the user's customized logical drives.

4.8 Mixed Drive Installations

When the user installs a system that employs different size disc drives the installation is called a "mixed drive" installation. In such systems the user should make sure that the formats used for logical drives N:, O:, and P: are compatible with the physical drive on which that format is to be used. For instance, a logical drive using the I.B.M. 3740 eight inch format cannot be assigned to a 5-1/4 inch disc drive.

4.9 Optimizing the System using the GENCPM Utility

After a working system is installed it can be optimized to make better use of the memory available. The Digital Research utility GENCPM.COM is used to relocate system modules, allocate physical record buffers, allocation vectors, checksum vectors, and hash tables. For a complete description of the GENCPM.COM utility see section 5 of the CP/M Plus System Guide.

The memory maps in Fig. 4.1 can be used as a guide in optimizing the new system. Some of the memory locations shown in the figure are approximate, and depend on the logical drive formats installed.

The total amount of memory available will depend on the target system installed. (The memory locations affected are recognized by a "~" preceding the memory address.) Therefore, the memory available for buffer space between the bottom of the video memory and the BNKBDOS3 will vary while the memory available between the top of the video memory and location 0100H will remain constant.

When allocating buffer space, DATA BUFFERS can not be allocated to bank 0. To compensate for this a memory segment can be assigned to logical bank 2. Logical bank 2 will then be mapped directly onto physical bank 0.

To get a better understanding of the GENCPM utility it can be run after a working system has been installed. The user can run GENCPM using the default values without worry of any modifications to the installed system.

This page intentionally left blank.

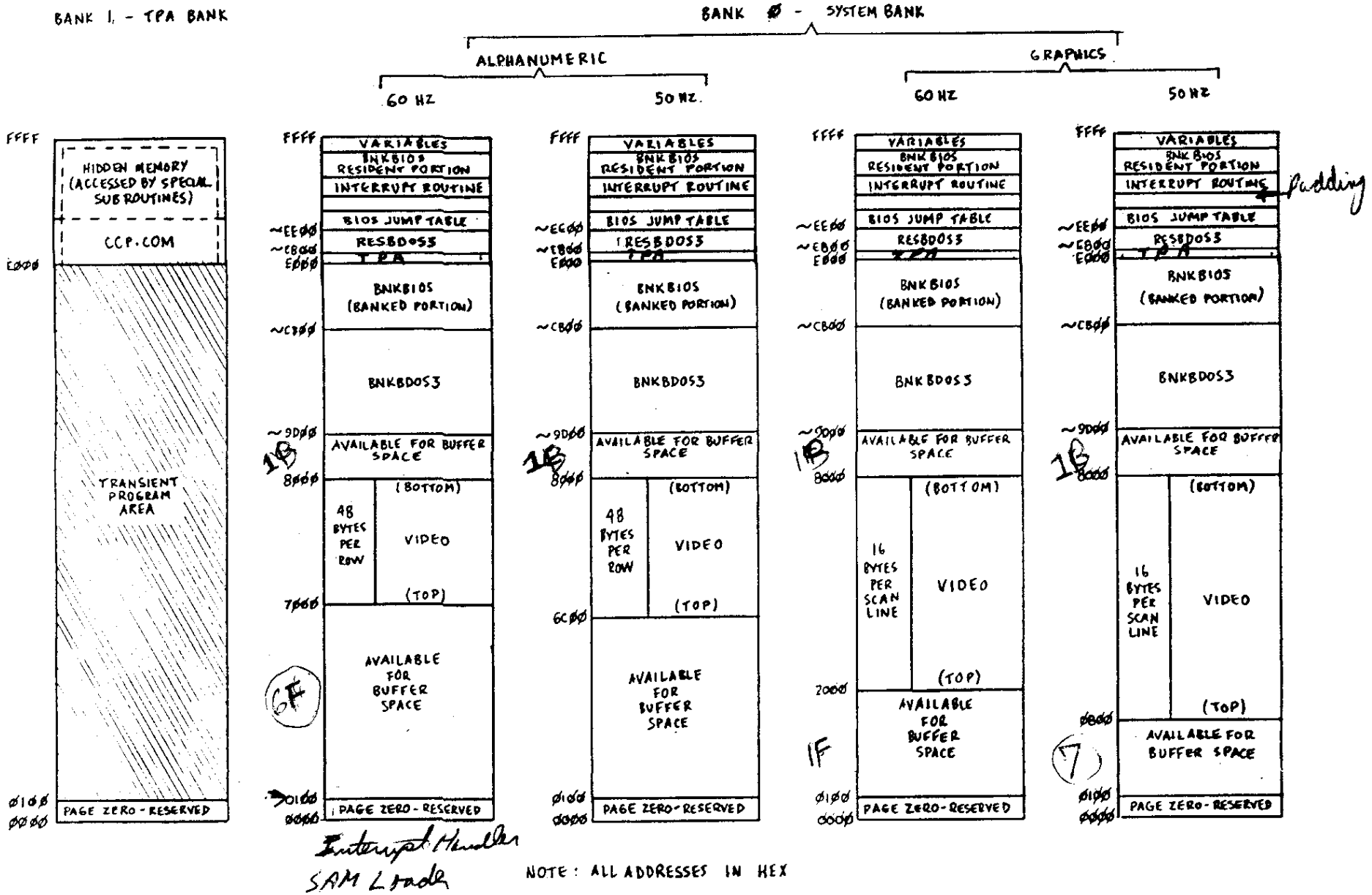


Fig 4.1 Memory organization for 128k banked version of CP/M 3.
Logical Bank 1 is in physical Bank B.
All other logical Banks are in Bank A.

TECHNICAL MANUAL

1.0 Megatel QUARK CPU Overview

The Megatel QUARK uses the Z-80B microprocessor, manufactured by Zilog Inc. The Z-80B clock frequency is 5.97MHz (6.2MHz on 50Hz models), leading to an execution time of 667ns (645ns on 50Hz models) for a typical 4-cycle instruction, such as a register-to-register ADD.

The main memory of the QUARK is 128kbytes in size. A memory-management scheme employing the Z-80B's I-register is used to provide simple yet flexible bank-switching to allow the full use of the extended physical memory. The Synchronous Address Multiplexer (SAM) also participates in the memory-mapping process, and permits the use of some special mappping modes.

To accomodate the Video Display Interface, the main memory of the QUARK operates as a dual-ported RAM. One of the ports is a bidirectional input/output port to the CPU, while the other port is essentially an unidirectional output-only port for the Video Display Interface. Through the programmable registers of the SAM, the size and location of the area of main memory used as the Video Display Memory can be set under software control.

As a consequence of the operation of this dual-ported memory, all Z-80B memory cycles have wait states added to extend them to a multiple of 4 T-states. Thus an instruction which would execute in, say, 7 T-states if no wait states were added will have one wait state added, to extend the instruction to 8 cycles. The insertion of these wait states is fundamental to the synchronization of memory accesses by the Video Display Interface and by the CPU.

Input/output instructions are also extended to modulo-4 cycles, but then have four additional wait states added. For instance, an 11-state IN A,r type of instruction will have a total of five wait states added, extending it to 16 T-states. The addition of the extra four cycles relaxes the speed requirements for the peripheral controller devices, both those on the board as well as user-added external controllers.

1.1 CPU Memory Management

The Z-80B processor used on the QUARK has a sixteen-bit address bus, and is therefore able to directly address 65536 (64k) locations in memory. However, the QUARK has 128k of memory, and therefore a process is required whereby the 64k possible addresses generated by the CPU are "mapped" into the 128k of main memory provided. In this discussion, the addresses generated by the CPU on its Address Bus will be called "logical addresses", whereas the actual locations in memory where data is stored will be called "physical addresses". Logical addresses are therefore the hexidecimal values normally used in programming.

The Z-80's I-register and the Synchronous Address Multiplexer (SAM) are used in the mapping process between the logical and physical address spaces. The I-register is used to control bank-switching between bank A and bank B of the main memory, and to determine in which bank the Video Display Memory is located. The SAM determines the location of the Video Display Memory within a 64k address space, and controls the mapping of the upper and lower halves of the logical address space.

The most-significant three bits (bits 5, 6, and 7) of the I register are used to define an 8k address boundary within the CPU's 64k logical address space. Any logical address appearing on the Address Bus is compared to the address boundary as determined by these bits. The result of this comparison determines in which memory bank the physical address is to be found.

If the value of the logical address is greater than or equal to the specified address boundary, the logical address will be mapped onto a physical address (that is, a memory location) in memory bank A. Conversely, if the value of the logical address is less than this boundary, then the logical address will be mapped onto a physical address in memory bank B. For example, if bits 5, 6, and 7 of the I-register were all zero, then the address boundary specified would be ØØØØ. Since all logical addresses are greater than or equal to ØØØØ, then the CPU's entire logical address space would be mapped into the physical address space of Bank A. If bits 6 and 7 were set and bit 5 were clear (I register = CØ), then logical addresses between ØØØØ and BFFF would be mapped onto physical addresses in bank B, while logical addresses equal to or above CØØØ would be mapped onto physical addresses in bank A. Figure 1.1 shows the operation of this mapping process for the eight possible address boundaries.

To read or write the contents of the I register the Z-80's LD A,I and LD I,A instructions are used. After changing the value in the I register, the new boundary will take effect after the next opcode fetch. Consider the following code sequence:

```
LD    A,BDYVAL        ;LOAD A WITH NEW BOUNDARY VALUE
LD    I,A             ;LOAD I WITH VALUE
RET                   ;RETURN
```

The return address used for the RET instruction will be popped off the stack in the memory bank determined by the new boundary value in the I register, whereas the opcode for the RET instruction will be fetched from the memory bank in effect prior to loading the new value into the I register.

Note that the use of the Z-80 I-register for selecting bank-switching boundaries precludes its usual use in the Z-80 Mode 2 (vectored) interrupt. See section 3.9 for further information regarding the interrupt system of the QUARK family of computers.

It might appear that a difficulty with this method of bank-switching is that there is no logical address which maps onto any physical address above EØØØ in Bank B, that is, that the top 8k of bank B cannot be directly accessed by the CPU. This would seem to be the case because the highest 8k address boundary which can be specified by bits 5, 6, and 7 of the I-register is EØØØ, so that logical addresses above EØØØ will always be mapped onto physical addresses in Bank A, and never in Bank B. (See fig 1.2) This problem can be overcome through the use of the MAP TYPE and PAGE MODE bits of the SAM, as explained below.

The MAP TYPE bit in the SAM (designated "TY") allows addresses in either the upper or the lower halves of the 64k logical address space to be effectively translated into the other half of the logical address space. In the "normal" mode of operation (the mode which is initialized at power-up) the TY bit is set. In this mode, each logical address is mapped onto a unique physical address in Bank A or Bank B by the mapping process described previously and shown in fig 1.2. If, however, the TY bit is cleared, then the upper and lower halves of the 64k logical address space map onto a physical address space whose size is between 32k and 64k. The actual size and location of the physical address space within the two Main Memory Banks is determined by both the bank-switching boundary set up in the I-register and by the PAGE MODE bit in the SAM (designated "P1"), as described below.

With both the MAP TYPE and the PAGE MODE bits cleared, the position of the bank-switch boundary will determine both the bank into which the logical addresses are mapped as well as whether the mapping is one-to-one or many-to-one. Consider first the simplest case, where the bank-switch boundary is 8ØØØhex. In this special case, logical addresses less than the bank-switch boundary will be mapped into the physical address space of Bank B, just as they would be if the MAP TYPE bit were set ("Normal" mode). However, logical addresses equal to or greater than the bank-switch boundary are mapped onto physical addresses in the "bottom" half of Bank A. This part of the physical address space of Bank A is the space into which logical address from Ø to 7FFFhex are mapped when the bank-switch boundary is set to ØØØØ and the MAP TYPE bit is set (normal mode). Thus logical address from 8ØØØhex to FFFFhex are essentially translated to ØØØØ to 7FFFhex. Note, however that this is still a one-to-one mapping, since each half of the logical address space is mapped into the bottom half of a distinct Main Memory Bank.

Continuing with the case where the bank-switch boundary is at 8ØØØhex and the MAP TYPE bit cleared, the effect of the PAGE MODE bit is to determine whether it is the upper or lower half of the logical address space which is mapped into a "foreign" part of the physical address space. As explained above, if the PAGE MODE bit is cleared, the upper half of the logical address space is mapped into the bottom half of Bank A, while the lower half is mapped into the lower half of Bank B. If, instead, the PAGE MODE bit is set, then the upper half of the logical space will be mapped into the upper half of Bank A. This is the physical space into which the upper half of the logical space would be mapped if the MAP TYPE bit were set. The lower half of the logical address space is now mapped into the upper half of Bank B.

With the MAP TYPE bit cleared and the PAGE MODE bit set, it becomes possible to access the otherwise-hidden upper 8k of Bank B. Setting the bank-switch boundary to 8ØØØH, as in the description above, forces logical addresses from ØØØØ to 7FFFhex to be mapped onto physical addresses in Bank B from 8ØØØhex to FFFFhex, a range which includes the hidden 8k. Thus the logical address range 6ØØØhex to 7FFFhex will be mappped onto the upper 8k of Bank B.

By appropriately setting the F-bits in the SAM control register and bit Ø of the I-register, it is possible to position the Video Display Memory so that this hidden 8k region is used for all or part of the Video Display Memory. Table V in the Appendix gives suggested values for the F-bits and the resultant Video Memory address boundaries.

1.2 QUARK Boot Mode Memory Mapping

The QUARK has a special memory-mapping mode which is used after the system has been reset by a low-level signal at the RESET input, or when the internal BOOT MODE line has been set. This memory mapping mode is called the Boot Mode. This mode is used when the system is loaded from the floppy disc drive (in a so-called "bootstrap" manner), and whenever the SAM control registers are to be written.

In this mode, any memory read operation by the CPU (including instruction fetching) will read from the contents of the 512-byte Bootstrap PROM, rather than from Main Memory. Memory write operations will write to Main Memory, using the memory mapping process described in Sec. 1.1. I/O read or write operations will reference I/O ports in the usual way, except that the Character Generator (I/O addresses ØØ-3F) cannot be written when in the Boot Mode.

When the QUARK is reset, the registers of the Z-80B, including the Program Counter, are cleared. Also, various I/O lines on the VIA and PIA are cleared and set to act as inputs. In particular, the BOOT MODE control line, which is the CB2 control line on the PIA, will be reset to act as an input, allowing a pullup resistor on this line to assert a "BOOT MODE" condition.

The Z-80B will now fetch the instruction at location ØØØØ. Since the BOOT MODE control line is high, this will cause the instruction at location ØØØØ of the Bootstrap PROM to be fetched. Sucessive instructions will be fetched from the PROM until the BOOT MODE control line is cleared to a low-level output condition. This puts the QUARK into the Normal Mode of operation, where all memory read or write operations will reference the Main Memory. In the Normal Mode, the PROM cannot be accessed.

The logical address space occupied by the Bootstrap PROM consists of the first 64 addresses (i.e. ØØ-3Fhex) of each 256-byte page in memory. Thus the entire PROM can be accessed in eight 64-byte "chunks" in the bottom 2k of memory. This logical address space is aliased within the thirty-two 2k blocks in memory. Normally, only addresses between ØØØØ and Ø73F(hex) are used in accessing the Bootstrap PROM.

Addresses between the 64-byte chunks cause I/O read or write operations. It is not recommended that these addresses be used. I/O operations may be performed in the usual manner using I/O instructions while in the Boot Mode.

It should be pointed out that it is unlikely that the user will ever need to directly use the Boot Mode of operation, nor is it ever necessary to directly access any part of the Bootstrap PROM. After the operating system has been booted from the floppy disc, the only operation requiring a return to the Boot Mode is when the SAM control registers are to be re-written. In this case, a special routine called the SAM Loader Access Routine, resident on the bottom page of Main Memory, will automatically handle the entry into and return from the Boot Mode. This use of this routine is discussed in Sec. 1.4.

1.3 QUARK Video Display Memory Mapping

The size of the Video Display Memory and its location within the 128k Main Memory are under the control of registers in the SAM and by bit Ø of the I-register.

Bit Ø of the I register determines in which memory bank the Video Display Memory is located. When this bit is clear, the Video Display Memory is located entirely within Bank A, and when it is set, the Video Display Memory is located entirely within Bank B. The bank-switching apparatus for the Main Memory using bits 5-7 of the I-register does not apply to addresses generated by the Video Display Controller, but only to addresses generated by the CPU, regardless of whether these CPU addresses fall within the Video Display Memory or not.

In either Bank A or Bank B, the physical address boundaries of the Video Display memory within the selected bank are determined by

the settings of bits FØ, F1, F2, F3, F5, and F6 in the SAM control Register, and by the Video Display Mode (ALPHA or GRAPHICS). The starting address (or the lower bound) of the Video Display Memory is the binary address

(F6)(F5)(F3)(F2) (F1)(FØ)ØØ ØØØØ ØØØØ,

where (Fn) represents the contents on the Fn bit in the SAM Control Register. The final address (or the upper bound) of the Video Display Memory depends on the Video Display Mode. In Alphanumeric Mode, the final address is the first 16k address boundary followimg the starting address, whereas in Graphics Mode, the final address is the second 16k address boundary following the starting address.

From the above, it can be seen that the size of the Video Display Memory may be set anywhere from zero to 32k in 1024-byte increments. However, only a certain set of sizes are likely to be of use in most applications. First, when operating in Graphics mode, the size of the Video Display memory must be an integer multiple of 3k (3072) bytes in order for the horizontal sync signal to be generated correctly. Second, if it is desired to have the Vertical Sync frequency match the frequency of the local AC power system (to avoid moving "hum-bars" on the CRT and related phenomena), the size of the Video Display Memory must be adjusted so that its entire contents will be read and displayed once during one cycle of the AC power line. In practice, this means that the most useful sizes are likely to be:

LINE FREQ.	MODE	SIZE
50Hz	ALPHA	5k
50Hz	GRAPHIC	30k
60Hz	ALPHA	4k
60Hz	GRAPHIC	24k

In order to achieve a Vertical Sync frequency of exactly 50Hz or 60Hz, it is necessary that the appropriate frequency crystal be used in the Master Clock generator. Thus it is not posible to generate a 50Hz vertical sync frequency on a board equipped with a crystal intended to permit operation at 60Hz, and vice versa. However, a "60Hz" QUARK can be programmed to operate with a Vertical Sync frequency of 48.1Hz, and a "50Hz" QUARK can be operated at 62.5Hz. See Table VI in the Appendix for suggested settings and the resultant address ranges.

Note that the F4 bit in the SAM in not used in determining the starting address of the Video Display Memory, and must always be set. Bits 1 through 4 of the I-register are "don't care" bits, and have no effect on the operation of the QUARK.

1.4 Programming the SAM Registers

The Synchronous Address Multiplexer (SAM) is an essential part of the CPU, Video Display Interface, and Memory Management of the QUARK.

The Control Registers of the SAM are initialized by the Bootstrap PROM after a system reset. It may be necessary at some time to change some of the values in the Registers of the SAM. To allow the user to do this, a special routine is automatically loaded into Page Zero of Bank A of the Main Memory. This routine is called the SAM Loader Access Routine, since it provides access to the SAM Register Loader routine in the Bootstrap PROM. This routine is not loaded into Bank B automatically, so any CALLs to location ØØØBh in Bank B should allow for this fact.

To set or clear any of the SAM registers, the address corresponding to the register is loaded into the HL register, and then the SAM Register Loader Access routine at location ØØØBhex is called. For example, the following sequence will clear register F1:

```
LXI     H,ØFF88H        ;LOAD ADDRESS FOR F1-CLEAR
CALL    ØØØBH           ;CALL SAM LOADER ACCESS ROUTINE
```

Table V in the Appendix lists the addresses to be used for clearing or setting each of the registers in the SAM, as well as indicating the functions of some of the register for the QUARK. Note that only some of the registers are ever likely to be changed. Even though there are no actual restrictions on the settings for any of the registers, only a particular set of combinations is useful on the QUARK.

Table VI indicates the useful set of values for the F-registers. Registers FØ, F1, F2, F3, F5, and F6 are used in determining the starting address of the Video Display Memory. (See Sec. 1.3.) The location of the Video Display Memory which results from each of the combinations is also shown in Table V.

If it is necessary to use the area in either bank of Main Memory occupied by the SAM Loader Access Routine, then the routine can be copied to the corresponding location in any 2k block of memory. Only the section of the Access Routine between locations ØØØBH and ØØ17H inclusive need be copied.

For instance, if the area from ØØØBH to ØØ17H is copied into the area from FØØBH to FØ17H, then the following example will have the same effect as the previous example:

```
LXI     H,ØFF88H        ;LOAD ADDRESS FOR F1-CLEAR
CALL    FØØBH           ;CALL SAM LOADER ACCESS ROUT
```

1.5 QUARK Interrupt system

On the QUARK, both of the two Z-80B interrupt inputs NMI and INT are used. The maskable interrupt input INT is wired-ORed to the interrupt request outputs from the VIA, the PIA, and the ACIA. On the QUARK/200, the Local-area Network Interface is also capable of generating an interrupt request to the INT input. The non-maskable interrupt input NMI is connected to the 1793 Floppy-disc Controller's Data Request output (DRQ) through an inverter.

Interrupt requests from each of the VIA, PIA, and ACIA can be individually enabled or disabled by writing to the appropriate control registers in these devices. Consult the data sheets at the back of this manual for further information on enabling interrupts from these devices.

The Vertical Sync line is connected to the CA1 input of the PIA. Since the frequency on this line is normally 60Hz (50Hz on 50Hz versions of the QUARK), the CA1 input can be used to generate a real-time clock interrupt to the CPU. Note that if the size of the Video Display Memory is changed from the "standard" sizes for Alphanumeric and Graphics Modes, the Vertical Sync frequency will change proportionately. Thus, if the Vertical Sync Frequency is changed, any software which relies on this interrupt as a time base should take this into account.

The Interrupt Request line INTRQ from the 1793 Floppy-disc Controller is connected to the CB1 input on the VIA. This interrupt output is set at the completion of any command to the 1793. In order for the 1793 to interrupt the CPU, the CB1 interrupt on the PIA must be enabled for low-to-high transitions on the CB1 pin.

The Floppy-disc Controller's Data Request output is set each time a byte is ready for the CPU or each time the controller is ready to receive a byte from the CPU. The DRQ output is inverted to drive the NMI input. When transferring data, the DRQ signal will go active at a rate proportional to the data transfer rate for the floppy disc drives in use. The QUARK is sufficiently fast so that even with the worst-case data transfer rate (500kbits/second for 8-inch double-density discs), only 70% of the CPU's time is used in moving data between the controller and main memory. This leaves 30% of the processor's capacity available to perform other operations. The left-over capacity is essential in certain applications, such as when an interrupt-driven communications routine must continue despite data transfers to or from the disc. Lesser transfer rates will of course leave correspondingly more of the CPU's time available.

Version 2.22 operating systems place a jump instruction at location 38hex in both Bank A and Bank B. In Bank A, the jump address points to the interrupt handler for the INT interrupt in the BIOS. In Bank B, the jump address points to the entry point of a special routine in the BIOS. This routine saves the current value in the I-register (which determines the bank-switching boundary when the interrupt occured), sets the bank-switching boundary to location ØØØØ

(so that all CPU memory is in Bank A), and then pushes a special return address onto the stack. The routine then jumps to location 38hex in Bank A. This causes the interrupt to proceed as it would have if the bank-switching boundary had been at location ØØØØ when the interrupt occured.

When the routine handling the interrupt returns, it will pop the return address off the stack. If this return address is the special address pushed earlier by the Bank B interrupt handler, then control will be transferred back to this routine. It will then restore the former bank-switching boundary, and return from the original interrupt.

Versions of the operating system earlier than V2.22 provide no support for the Bank B interrupt.

Version 2.21 and 2.22 operating systems distributed with the QUARK enable only two interrupt sources, the real-time clock interrupt from the CA1 input on the PIA, and the NMI interrupt, which is connected to the 1793 floppy-disc controller's DRQ output. When an INT interrupt occurs (which vectors to location 38hex), the interrupt handler in the distributed version of the BIOS does not verify that the source of the interrupt is the real-time clock. Instead, it immediately updates the displayed clock on the screen, if the system was installed with the clock option.

Version 3.01 (CP/M Plus) operating systems require that interrupt-handling routines not use memory between 8ØØØ and 8Ø7F inclusive if they expect to re-boot the CCP.

In order to use other interrupt sources on the QUARK, the user may either patch the source file for the BIOS himself, or include in the application program sufficient code to handle the additional interrupts.

For the latter approach, the general idea would be for the application program to save the jump address at locations 39hex and 3Ahex, and replace them with a pointer to the interrupt entry point in the program. When an interrupt occurs, the application program would check for the source of the interrupt, and take the appropriate action if the interrupt was intended for use by the application. If the interrupt was not for the application program, then the program should jump to the regular interrupt entry point in the BIOS, the address of which was saved by the application program initially. Thus the BIOS routine will have its opportunity to respond to the interrupt.

Notice that the mechanism handling interrupts which occur while operating in Bank B will mesh perfectly with the above approach. Of course, the user is free to handle Bank B interrupts in his own way.

The NMI interrupt is largely a system rather than a user feature on the QUARK. The operating system always handles the NMI interrupt from the floppy disc controller in an appropriate fashion. No mechanism is provided by the operating system for trapping Bank B NMI interrupts, as none are expected.

This page intentionally left blank.

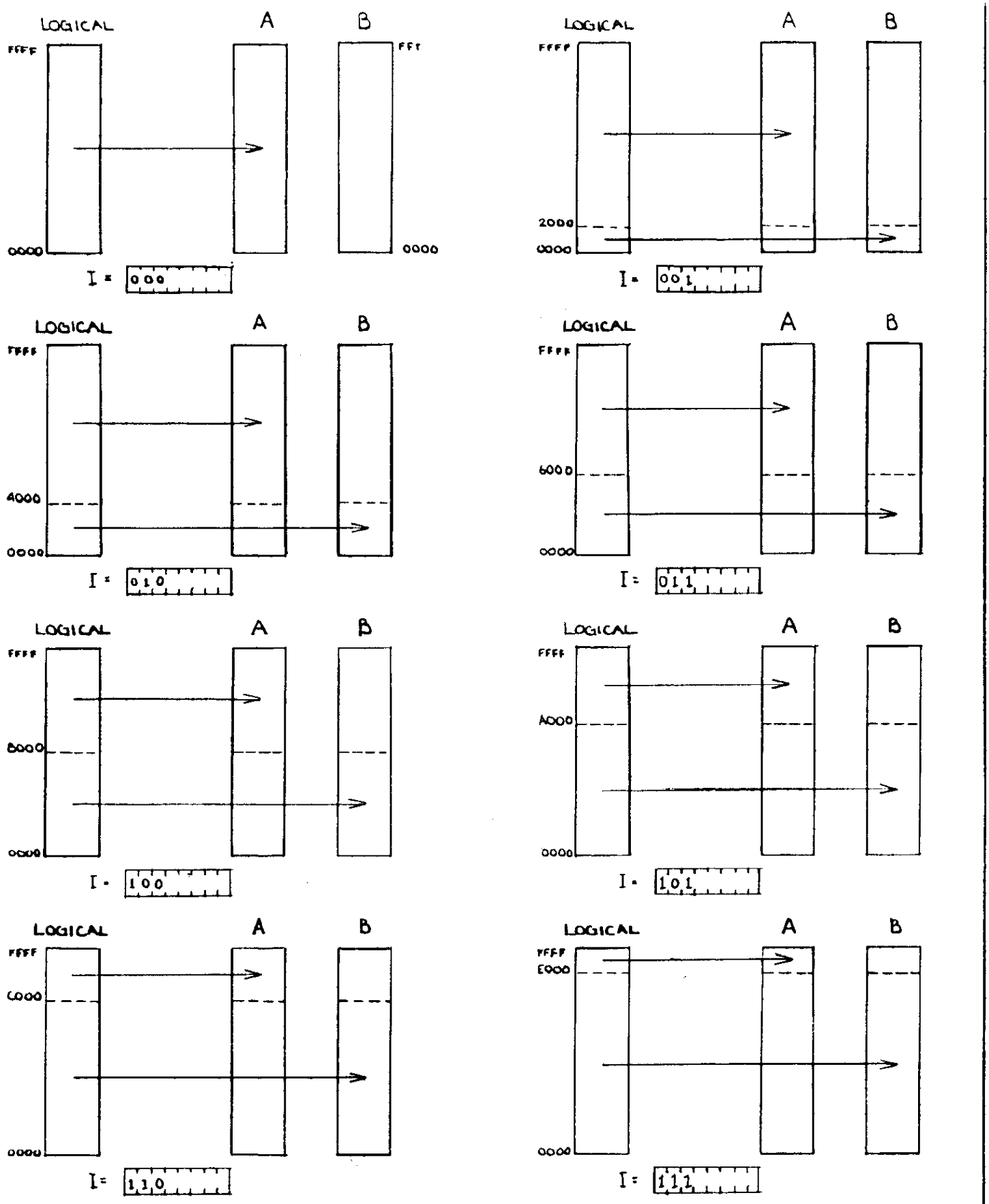


Figure 1.1 QUARK Memory bank-switching

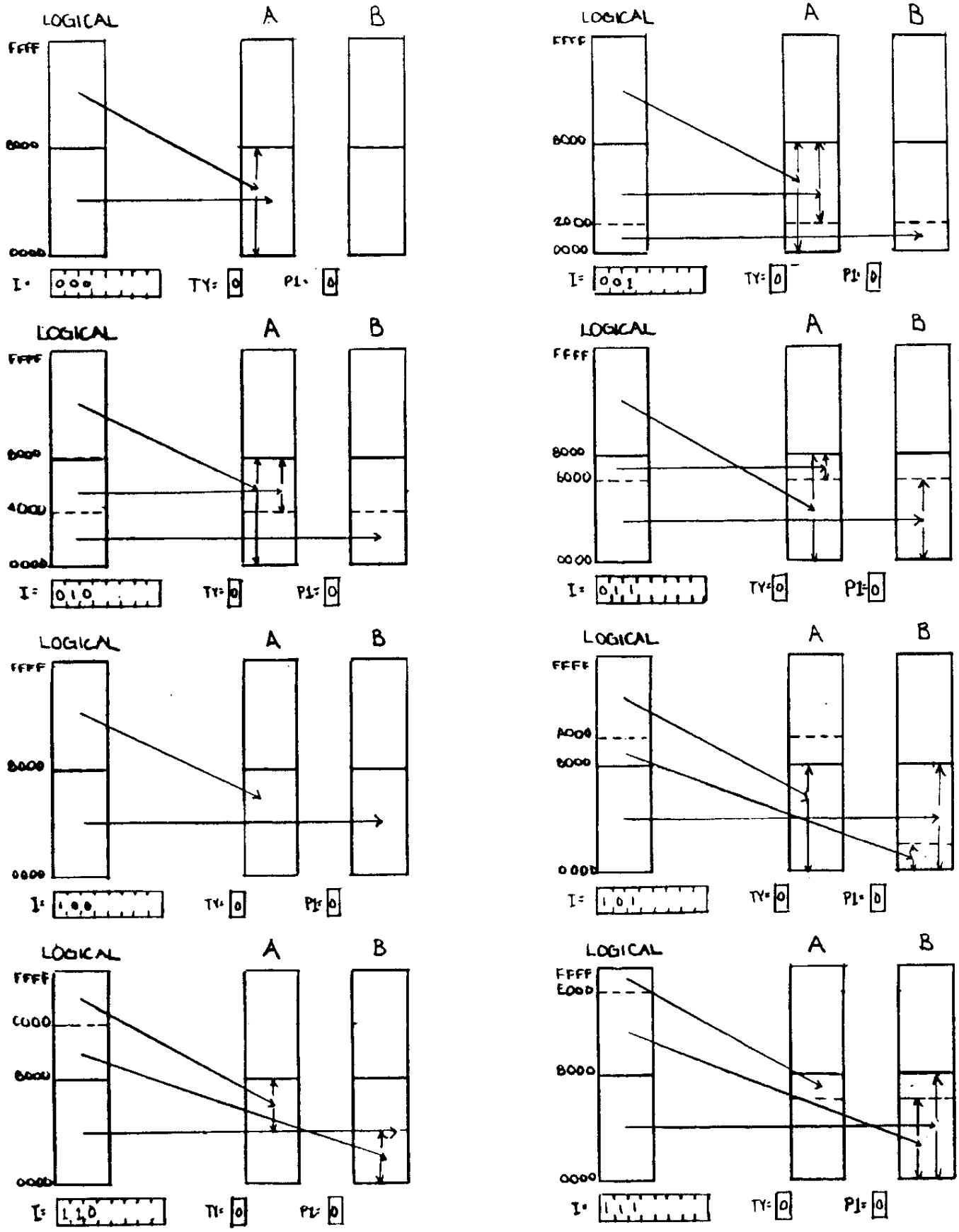


Figure 1.2 Use of SAM Map Type bit in memory-mapping Page mode P1=0

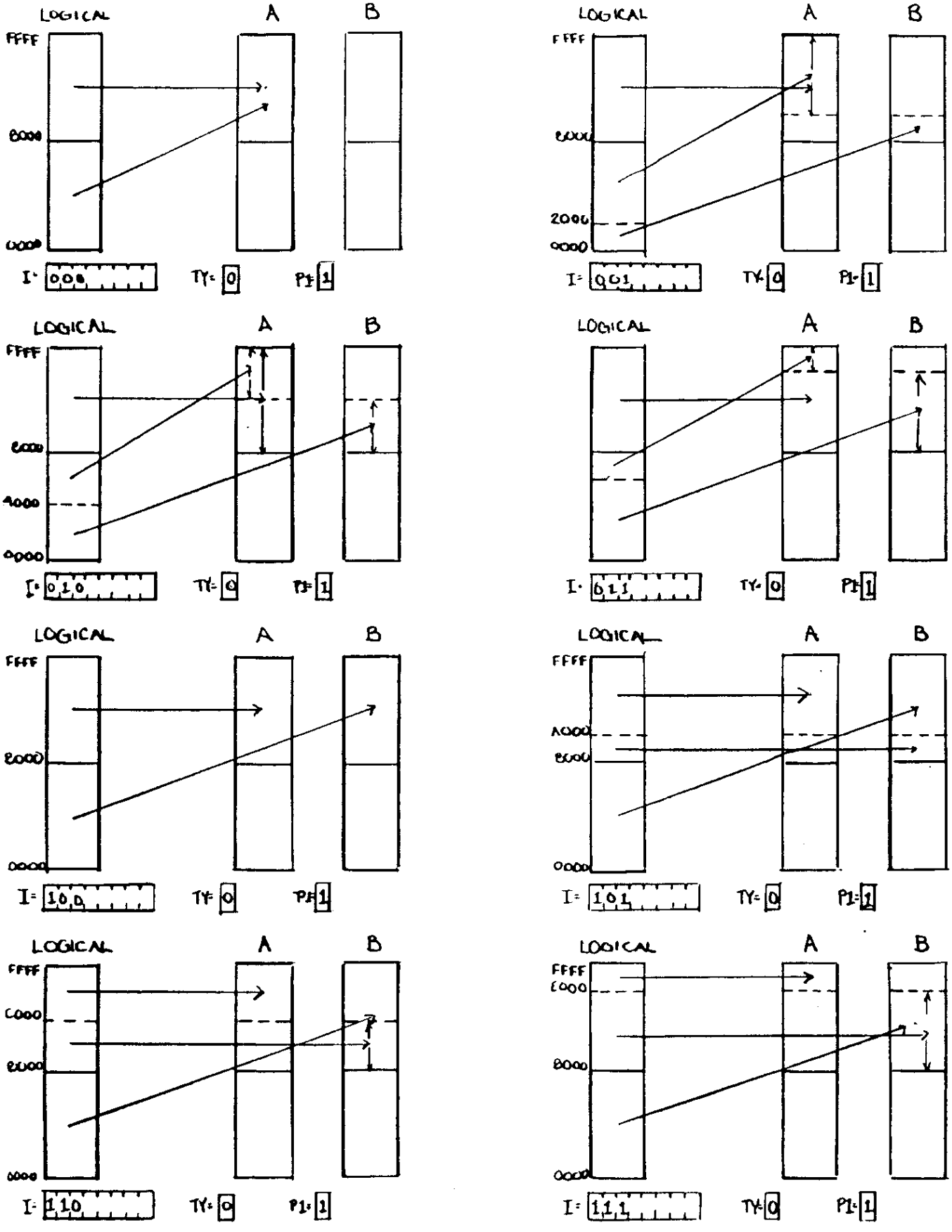


Figure 1.3 Use of SAM Map Type bit in memory-mapping Page mode P1=1

2.0 The QUARK Video Display Interfaces

The on-card Monochrome or Colour Video Display Interfaces are integral parts of all of the members of the Megatel QUARK family of single-board computers. Both types of Interfaces are capable of operating in either Alphanumeric or Graphics modes. On monochrome members of the family (such as the QUARK/100 and /200), TTL-driven video, horizontal sync, and vertical sync signals are provided for connection to direct-drive CRT data displays. Additionally, a composite video output at approximately $1V_{p-p}$ is provided for use with monochrome displays with composite video inputs. On the Colour Video Display member of the family (the QUARK/150), analog RGB and composite sync outputs are provided. Polarity of the composite sync is strap-selectable, and the four outputs can be wire-ORed for a grey-scale monochrome composite video output.

The data displayed on the CRT are stored in a segment of the Main Memory of the QUARK. The memory bank in which the Video Display Memory is located is determined by bit 0 of the CPU's I register. See sections T/1.1 through T/1.3 for more information about memory management on the QUARK computers.

In Alphanumeric mode, a total of 32 character rows are scanned for each video frame on the 60Hz version (assuming a 60Hz frame rate), and 40 rows on the 50Hz version (assuming a 50Hz frame rate). However, not all of the scanned rows may be displayed. The first row, which represents data within the first half-page of the Video Display Memory, is displayed during the Vertical Retrace Period. There being no hardware mechanism to blank the video output during the Vertical Retrace period, this part of the Video Display Memory must be loaded with data that will generate a null video output. The simplest manner in which to do this is to load ØØ bytes into that part of the Video Display Memory which is scanned during the retrace period. The Bootstrap PROM routine loads a "blank" character into the Character Generator for the ØØ character code, so that when a row of ØØ bytes is displayed, no video output will result.

The second line of the display can be used for display purposes if the CRT monitor employed terminates the Vertical Retrace Interval sufficiently quickly. On many monitors, however, characters displayed in this row will appear slanted because of the monitor's inability to recover from the Vertical Retrace Interval in time to properly display the first several scan lines. Thus the video driver routines included with standard QUARK operating systems do not use the second displayable row on the video display.

The video driver for CP/M 3.01 systems can be configured during installation to set up three separate areas within the full display. Each of these areas, called the DISPLAY, MAIN, and STATUS screens, use a user-specified number of character rows starting from the top of the screen. The alphanumeric-mode driver also permits an optional clock to be displayed near the upper-left or lower-right corners of the CRT.

CP/M 2.22 video drivers are set up for 1 row in the DISPLAY screen and 2 rows in the STATUS screen. The intervening 27 lines (on the 60Hz version) or 35 lines (on 50Hz versions) are used as the general video display area (the MAIN area). No clock is displayed.

Standard versions of the QUARK with monochrome Video Display Interfaces are equipped with a Programmable Character Generator. This allows the eight-bit by eight-bit patterns for the characters displayed on the CRT to be loaded or altered under software control. Custom character sets may be designed, saved on floppy disc, and loaded when needed. This simplifies the task of accomodating application programs requiring languages other than English, of running programs using special symbols, or of using programming languages which employ special character sets (e.g. the APL programming language). While normally an entire character set (256 characters) would be loaded as one step (as is done by the utility program CHRLD.COM -see sec. 9.1 in the Technical manual), character patterns can be loaded or modified on a byte-by-byte basis, so it is not necessary to load an entire set of characters. This may permit special video effects in some applications.

The Programmable Character Generator used on the QUARK monochrome Video Display Interface is a 2048-byte static memory which is independent of the Main Memory. A utility routine to load the Programmable Character Generator with a standard character set as the operating system is booted is included with the QUARK operating system software. Also included is a character set editor utility, which can be used to customize the standard character set or to create new character sets. These user-defined character sets can be saved on a floppy diskette. A full discussion of the software included for the Programmable Character Generator can be found in section 9 of the Technical manual.

The Bit-mapped Graphics Mode of the Monochrome Video Display Interface allows graphic data to be displayed on the CRT. In this mode, individual bits in the Video Display Memory are mapped onto single dots (pixels) on the CRT. For 60Hz models, 24k of Main Memory is used to display (typically) 143,360 pixels, organized as 640 horizontally by 224 vertically. On 50Hz models, 30k is used to display (typically) 179,200 pixels, organized as 640 by 280 pixels.

With three exceptions, the Colour Video Display Interface of the QUARK/150 operates in the same fashion as the monochrome version. The first difference is that instead of a 2kbyte Character Generator, a 16kbyte Colour Translation Table is used to map the data in the Video Display Memory onto the CRT. Second, this Translation Table is used in both Alphanumeric and Graphics mode, whereas the Character Generator on monochrome versions is used only in the Alphanumeric mode. Finally, the 16-byte "holes" in the Video Display Memory preceding the 80-byte displayed field are used by the Colour Video Display Interface for mode selection and memory refreshing. More information on the Colour Video Display Interface can be found in section 2.5.

2.1 Video Display Memory

The size and location of the Video Display Memory within the Main Memory of the QUARK is under software control through the values stored in the control register of the SAM, as well as by the value in the Z-80B's I register. The size of the Video Display memory may be set to values between 1k and 32k, although only certain memory sizes are appropriate, as will be explained below. The starting address for the Video Display memory is determined by the values loaded into the F6, F5, and F3 to FØ bits of the SAM Control Register. (Bit F4 is not used in the determination of the starting address.) The top address of the Video Display Memory is determined by the mode (Alphanumeric or Graphics) and by the values of the SAM Video Display Counter bits corresponding to F3 and F4.

The Video Display Interface reads 96 consecutive bytes from the Video Display Memory for each raster scan line displayed on the CRT. In Alphanumeric mode, these 96 bytes are the last 96 of each half-page (128 bytes) within the Video Display Memory. The first 32 bytes of each half-page are not read by the Video Display Interface in Alphanumeric mode.

In Graphics mode, the 96 bytes scanned for each line displayed are contiguous within the Video Display Memory. Thus there are no unscanned memory areas within the Video Display Memory when operating in Graphics mode.

In both display modes, the second through sixteenth of the 96 bytes, plus one more from the displayed 80 bytes, are read during the horizontal retrace interval between succesive scan lines on the CRT monitor. The Video output is blanked during this interval, so the contents of these bytes will not directly cause any visible output on the CRT. The last eighty bytes contain either the ASCII codes to be translated into character data on the CRT, or the graphic information to be displayed as pixels.

In the Alphanumeric mode, the same 96-byte block is read eight times for each character row displayed. It is necessary to do this because each character row is built up from eight raster scan lines, each line adding one horizontal "slice" of the character patterns. The ASCII code for each of the eighty characters in the row must be read eight times while the Scan Line Counter counts from 0 to 7.

The Graphics mode operates in a similar fashion to the Alphanumeric mode, except that each of the sets of 96 bytes is read only once for each raster scan line displayed, rather than eight times as in the Alphanumeric Mode. In the Monochrome Video Display Interface, the data in the last 80 of the 96 bytes is sent directly to the Video Shift Register, bypassing the Character Generator. Thus the pattern of bits in each of the 80 bytes determines the pattern of pixels displayed on each scan line of the CRT. Since 80 bytes of eight bits each are read for each line, a total of 640 pixels can be displayed horizontally. The most-significant bit of each byte (bit 7)

is the first bit shifted out by the Video Shift Register, and thus appears on the CRT as the left-most pixel of each group of eight.

The time required to read the 96 bytes from the Video Display Memory determines the Horizontal Sync pulse period and frequency. Four Z-80B T-states are required for each byte read. Thus on the 60Hz version, the Horizontal Sync frequency is 15.540kHz (64.368 us), and on the 50Hz version it is 16.145kHz (61.939 us).

In the Graphics mode, the top address of the Video Display Memory is equal to the output from the Video Counter at the time when the Video Counter bits corresponding to F3 and F4 reach Ø and 1, respectively. In Alphanumeric mode, the top address is such that the Video Counter bit corresponding to F4 reaches zero. Thus in Alphanumeric mode the top address of the Video Display memory is the address of the first 16k boundary following the starting address minus one. In Graphics mode the top address is the address of the second 16k boundary following the starting address minus one.

The Vertical Sync frequency is determined by the display mode, the size of the Video Display Memory area, and the master clock frequency. The exact Vertical Sync period is given by

$$t_V = (96(n+2) + 88) * t_E,$$

where n is the number of scan lines being displayed, and t_E is the period of the E-clock. On the 60Hz version of the QUARK, this is 670ns, while on the 50Hz version it is 645ns. In Alphanumeric mode, the number of displayed lines is the Video Display Memory size (in bytes) divided by 16, while in Graphics mode, it is the Video Display Memory size divided by 96.

For a 60Hz Vertical Sync frequency, the Video RAM area should be 4k in Alphanumeric Mode, and 24k in Graphics Mode (assuming that the master clock frequency is 23.86176Mhz, as is the case for "60Hz" versions of the QUARK). For a 50Hz Vertical Sync frequency, the memory sizes should be 5k and 30k, respectively (assuming the 24.8MHz master clock frequency used on "50Hz" versions of the QUARK).

If it is not necessary for the Vertical Sync frequency to be exactly 60Hz (or 50Hz on the 50Hz model), then one is free to pick the starting address of one's choice. The starting address of the Video Display Memory may be placed on any 1k address boundary, although in Graphics mode only 3k boundaries will result in a prpoer Horizontal Sync signal. For example, a 60Hz board could be operated in the Graphics mode with a Video Display Memory size of 27k instead of 24k. This larger size would result in a vertical sync frequency of 53.4Hz, with a total of 288 lines per frame, instead of 256. The horizontal sync frequency would remain unchanged at 15540Hz.

See Table VI in the Appendix for SAM Control Register values needed to select various Video Display Memory sizes and locations.

2.2 Monochrome Alphanumeric Mode

In Alphanumeric Mode on versions of the QUARK with the Monochrome Video Display Interface, data stored in the Video Display Memory are interpreted as eight-bit character codes. These codes are fetched from the 96 scanned locations in each half-page of the Video Display Memory and presented to the input of the Character Generator. The Character Generator contains the patterns which represent the characters to be displayed on the CRT.

The standard character set for the QUARK uses characters formed from a 5-by-7 matrix of dots. This matrix is imbedded in a larger 8-by-8 background field. The background field is effectively part of the character; in the standard QUARK character set the top row of dots and the first two and the last columns of dots for each character are blank to provide the necessary space between adjacent characters. The standard character set includes 128 "normal video" characters and 128 "reverse video" characters, and is designed to use seven-bit ASCII codes. The reverse video characters are essentially a second set of 128 characters where the dots forming both the character and the background are inverted from the corresponding normal video characters. In the standard character set, a reverse video character is displayed whenever bit 7 (the most significant bit) of any 8-bit value stored in the Video Display Memory is set. The remaining seven bits form the ASCII code for the character that is to be displayed.

There are no extra dot columns or dot rows between the 8-by-8 background fields of each character, that is, that the 8-by-8 fields completely fill the displayable area of the screen. Thus it is also possible to create graphic characters (as distinguished from Bit-mapped Graphics, below) which allow continuous lines, bars, and other figures to be displayed on the screen in Alphanumeric Mode.

2.3 Monochrome Bit-mapped Graphics Mode

In the Bit-mapped Graphics Mode, the Monochrome Video Display Interface fetches data from the Video Display Memory in exactly the same manner as is used in the Alphanumeric mode. However, instead of this data going to the character generator (together with the three Scan Line Counter bits), the data goes directly to the Video Shift Register. Thus the patterns of "ones" and "zeroes" stored in each byte of the displayed portion of the Video Display Memory will directly determine which dots, or pixels, are illuminated on the CRT.

Because the Character Generator is not used to map eight-bit bytes onto eight-by-eight dot patterns in the Graphics Mode, eight times as much memory must be allocated for the Video Display Memory when operating in Graphics Mode as when in Alphanumeric mode. Note, however that in Graphics mode the 96-byte blocks scanned for each

raster line displayed are contiguous in the QUARK's Main Memory, whereas in Alphanumeric Mode, these blocks have 32-byte gaps between each block. Thus the actual memory area used in Graphics mode need be only six times that used for Alphanumeric mode. The size of the Video Display Memory is increased by moving its starting address downward within the Main Memory. See Sec. 1.3 on the Video Display Memory for more information on this topic.

To switch between the Bit-mapped Graphics Mode and Alphanumeric Mode, the Graphics Mode Bit of the QUARK and the V2 Mode bit of the SAM must be changed. The Graphics Mode Bit is bit 6 of the PIA Port B output (I/O address 76hex). The Graphics Mode bit is at a logic low level for Graphics Mode and a logic high level for Alphanumeric Mode. This bit is cleared by writing to Port B with accumulator bit 6 set to zero, and is set by writing to the same port with the same accumulator bit set to one. Note that bit 6 of the PIA Data Direction Register (I/O address 74hex) must be high in order that PB6 be enabled as an output line. The V2 Mode bit of the SAM must be set to 1 for the Graphics Mode, and cleared for the Alphanumeric Mode. The VØ and V1 mode bits are left cleared in both display modes.

The Graphics Mode bit and the V2 mode bit must be changed in synchrony with the Vertical Sync (VS) signal. A suggested approach using the Real-time Clock interrupt (which is generated by the positive-going edge of the VS signal) to initiate the sequence of mode bit changes is described below.

When the Real-time Clock interrupt occurs, create a delay of at least five microseconds. This delay ensures that the seven pulses (at the E-clock frequency) immediately following the first rising edge of the VS signal will be bypassed. At the end of this delay, load the new starting address of the Video Display Memory (defined by bits FØ-F3, F5, and F6) into the SAM. Then change the polarity of the active transition on the CA1 input of the PIA from positive-transition active to negative-transition active. (This requires that bit 1 of the PIA Control Register A be set low.) Now wait for the negative transition of the VS signal by polling the PIA IRQA1 flag (bit 7 of Control Register A, I/O address 75hex). Immediately upon detecting this negative transition change the Graphics mode bit and the V2 bit of the SAM to the values required for the mode to be selected. After changing these bits restore the original value of bit 1 of the PIA Control Register A by loading a one into this bit. This completes the sequence of operations required to change from Alphanumeric mode to Graphics mode.

The following routine can be called to enter into the Graphic Mode. It is assumed here that the Real-time Clock interrupt is disabled before entering the routine.

```
;Graphics-entry example routine
;revised sept 15, 1983 to restore piacra and eliminate unnecessary
;sync loops
;
SAMV2S  EQU     ØFF85H  ;ADDRESS TO SET V2 BIT IN SAM
SAMFØC  EQU     ØFF86H  ;ADDRESS TO CLEAR FØ BIT IN SAM
SAMF1C  EQU     ØFF88H  ;ADDRESS TO CLEAR F1 BIT IN SAM
SAMF2C  EQU     ØFF8AH  ;ADDRESS TO CLEAR F2 BIT IN SAM
SAMF3S  EQU     ØFF8DH  ;ADDRESS TO SET F3 BIT IN SAM
SAMF5C  EQU     ØFF9ØH  ;ADDRESS TO CLEAR F5 BIT IN SAM
SAMF6C  EQU     ØFF92H  ;ADDRESS TO CLEAR F6 BIT IN SAM
SAMSET  EQU     ØØØØBH  ;ADDRESS OF ROUTINE TO SET/CLEAR SAM BITS
PIAPA   EQU     Ø74H
PIACRA  EQU     Ø75H
PIAPB   EQU     Ø76H
        ;
GRAPHICS:
        DI
        IN      PIAPB   ;READ THE CURRENT STATE OF PIA PORT B
        ANI     ØBFH    ;CLEAR BIT 6 (GRAHICS/ALPHA BIT)
        MOV     B,A     ;SAVE THIS IN REGISTER B
        CALL    SYNC    ;CALL ROUTINE TO SYNCHRONIZE WITH
                        ; VERTICAL SYNC LINE
        MOV     A,B     ;RESTORE VALUE TO SEND TO PIA PORT B
        OUT     PIAPB   ;RESTORE PIA PORT B WITH GRAPHICS BIT CLEARED
        LXI     H,SAMV2S        ;GET SAM V2-SET ADDRESS
        CALL    SAMSET  ;SET V2 FOR GRAPHICS MODE
        LDA     PIASAVE ;GET FORMER VALUE FOR PIA CRA
        OUT     PIACRA  ;RESTORE PIA CRA
        RET             ;RETURN FROM GRAPHIC-ENTRY ROUTINE
        ;
SYNC:
        IN      PIACRA  ;GET CURRENT PIA CRA
        STA     PIASAVE ;SAVE IT
        MVI     A,ØC4H  ;VALUE FOR PIA CONTROL REGISTER A
        OUT     PIACRA  ;DISABLE INTERRUPTS FROM CA2, SET IRQA2 ON
                        ; HIGH-TO-LOW TRANSITION OF CA1 (VERT. SYNC)
        IN      PIAPA   ;READ PIA PORT A TO CLEAR IRQA-1 AND -2 FLAGS
LOOP1:
        IN      PIACRA  ;READ PIA CRA
        ANI     Ø8ØH    ;EXAMINE BIT 7 (=IRQA1 FLAG)
        JZ      LOOP1   ;LOOP UNTIL IRQA1 IS SET BY HIGH-TO-LOW
                        ; TRANSITION OF VERTICAL SYNC LINE
;
;THE FOLLOWING WILL CONFIGURE THE VIDEO DISPLAY MEMORY BETWEEN
;ADDRESSES 2ØØØH AND 7FFFH
;SEE TABLE VI IN THE APPENDIX FOR OTHER LOCATIONS FOR THE VIDEO
;DISPLAY MEMORY
;
        LXI     H,SAMFØC  ;CLEAR FØ BIT
        CALL    SAMSET
```

```
        LXI     H,SAMF1C  ;CLEAR F1 BIT
        CALL    SAMSET
        LXI     H,SAMF2C  ;CLEAR F2 BIT
        CALL    SAMSET
        LXI     H,SAMF3S  ;SET F3 BIT
        CALL    SAMSET
        LXI     H,SAMF5C  ;CLEAR F5 BIT
        CALL    SAMSET
        LXI     H,SAMF6C  ;CLEAR F6 BIT
        CALL    SAMSET
        ;
        RET
PIASAVE:
        DS      1         ;ONE BYTE FOR SAVING PIA CRA
```

The example and description above assumed that the interrupts were disabled throughout the procedure. It would also be possible to re-enable interrupts after loading the Video Memory starting address and changing the polarity of the CA1 active transition. The next negative transition of the VS signal would generate an interrupt, immediately following which the Graphics mode bit and the V2 bit would be changed, just as in the non-interrupt procedure above.

To return to Alphnumeric mode, it is necessary only to set the Graphics/Alphanumeric mode bit, and then clear the V2 bit in the SAM, and set up the F-bits for the desired Video Display Memory location. The routine below will set up the Alphanumeric mode with the Video Display Memory located between FØØØ and FFFF, as is set up by the Bootstrap PROM.

```
;alpha-entry example routine
SAMV2C  EQU     ØFF84H  ;ADDRESS TO CLEAR V2 BIT IN SAM
SAMFØC  EQU     ØFF86H  ;ADDRESS TO CLEAR FØ BIT IN SAM
SAMF1C  EQU     ØFF88H  ;ADDRESS TO CLEAR F1 BIT IN SAM
SAMF2S  EQU     ØFF8BH  ;ADDRESS TO SET F2 BIT IN SAM
SAMF3S  EQU     ØFF8DH  ;ADDRESS TO SET F3 BIT IN SAM
SAMF5S  EQU     ØFF91H  ;ADDRESS TO SET F5 BIT IN SAM
SAMF6S  EQU     ØFF93H  ;ADDRESS TO SET F6 BIT IN SAM
PIAPA   EQU     Ø74H
PIACRA  EQU     Ø75H
PIAPB   EQU     Ø76H
SAMSET  EQU     ØØØØBH
        ;
        ;
ALPHA:
        ;TO AVOID GLITCHING SCREEN, THIS POINT SHOULD SYNCHRONIZED WITH
        ;VERTICAL SYNC SIGNAL (SEE GRAPHICS-ENTRY EXAMPLE)
        DI              ;DISABLE INTERRUPTS
        IN      PIAPB   ;READ CONTENTS OF PIA PORT B OUTPUT
        ORI     4ØH     ;SET BIT 6 FOR ALPHANUMERIC MODE
        OUT     PIAPB
        LXI     H,SAMV2C
        CALL    SAMSET
        LXI     H,SAMFØC
```

```
CALL    SAMSET
LXI     H,SAMF1C
CALL    SAMSET
LXI     H,SAMF2S
CALL    SAMSET
LXI     H,SAMF3S
CALL    SAMSET
LXI     H,SAMF5S
CALL    SAMSET
LXI     H,SAMF6S
CALL    SAMSET
EI
RET
```

This completes the routine to switch from graphics mode to alphanumeric mode.

2.4 Loading the Monochrome Programmable Character Generator

On monochrome models of the QUARK with the Programmable Character Generator, the patterns for the characters displayed in the Alphanumeric Mode are loaded into the Character Generator under software control. Once loaded, the contents of the Character Generator remain until another character set is loaded or until power to the board is removed. The standard set of utility routines included with the Distribution Software include programs for loading the Programmable Character Generator and for designing character patterns to be loaded. It is therefore not necessary to understand how the Programmable Character Generator operates in order to be able to make use of this feature. However, the information in this section is provided for the user who wants to fully understand this aspect of the operation of the QUARK.

In normal Alphanumeric Mode operation (that is, when the Programmmable Character Generator is not being loaded) the Character Generator is accessed only to be read by the Video Display Interface. This system makes use of the Synchronous Address Multiplexer (SAM), some of the Programmable Array Logic (PAL) devices, the Video Data Latch, the Character Generator, and the Video Shift Register. The basic function of the Video Display Interface in Alphanumeric Mode is to fetch data from that part of the Main Memory which has been set up as the Video Display Memory, present this data to the Character Generator, and then convert the output of the Character Generator to a form suitable for display on a CRT monitor.

The code representing the character to be displayed is fetched from the Video Display Memory. This code (usually ASCII) is eight bits long, and is used to provide eight bits of the address to the Character Generator. The remaining three address bits determine which of the eight lines of the character pattern is to be displayed on the

current raster scan line. These three Character Generator address lines are generated by the least-significant three bits of the Scan Line Counter, which counts scan lines from the top of the video frame to the bottom. As these three bits count from zero to seven, the "zeroeth" through seventh lines of the character pattern selected by the character code are fetched from the Character Generator and sent to the Video Shift Register. The VSR shifts these eight-bit patterns one bit at a time out to the CRT on the Video output line.

To load, or write, the Programmable Character Generator, it is necessary to set up an eleven-bit address on the input of the Character Generator, to set up the eight-bit pattern to be loaded on its data I/O lines, and then to strobe the Write-enable line. Eight of the address bits represent the character code for the pattern to be loaded into the Character Generator. The remaining three bits determine which scan line of the character is to be loaded. The addresses and data must simultaneously be present at the inputs of the Programmable Character Generator at the time that the Write-enable line is strobed. The operations required to do all of this are described below.

There is no way to directly set up an address on the input of the Character Generator. As explained above, the fetching of data from the Video Display Memory together with the three bits from the Scan Line Counter is the only method by which the address for the Character Generator is formed. Thus, to set up a particular address to the Character Generator it is necessary to "fake" this process.

In order to set up the eight bits of the Character Generator address that represent the character code for the character pattern to be loaded, that character code is loaded into some location(s) of the Video Display Memory. The normal operation of the Video Display Interface is allowed to fetch this character code and present the code to the address inputs of the Character Generator. This provides a means to get eight of the eleven address lines set up under software control.

There is no way to force the Scan Line Counter to produce any particular combination of the three remaining bits needed to address the Character Generator. Furthermore, there is no way to directly read the output of the Scan Line Counter. However, the frequency at which the Counter is clocked is known, so if the value of the output from the Counter were known at a particular time, the value at a later time could be inferred by the amount of time which had passed since the reference time. Such a reference time is provided by the Vertical Sync line.

A transition on the Vertical Sync line indicates the end of one video frame and the start of another. The frequency of such transitions is determined by the size of the Video Display Memory and by the Master Clock Frequency of the QUARK board. Assuming the QUARK is being operated normally, the frequency of this signal will be 60Hz on 60Hz versions, and 50Hz on 50Hz versions. What is significant about this signal for the purpose of loading the Character Generator is that it can be used to indicate the end of the Vertical Retrace

period (the period during which the CRT's electron beam moves from the bottom to the top of the CRT). At the end of the Vertical Retrace period, the value of the output of the Scan Line Counter is known to be exactly zero, the uppermost line of the CRT being line zero.

The Vertical Sync Line is connected internally to the CA1 interrupt input on the PIA (68A21). This connection allows the state of the VS line to be sampled so that the software can determine exactly when the Vertical Retrace period ends.

The signal on the Vertical Sync line consists of a series of pulses. The start of the Vertical Retrace period is indicated by the first falling edge of the VS line falling a long period during which the line is at a logical high level. After this falling edge the line remains low for a short period, followed by a series of seven pulses at the E-clock frequency. When the last of these pulses finishes, the VS line is left at a high level, a state which persists until the next Vertical Retrace period.

This last rising edge before the longer period at a high level indicates the end of the Vertical Retrace period. At the occurence of this edge, the output from the Scan Line Counter has a value of zero. From this point the Counter will start counting, incrementing by one for each new scan line displayed.

Now, in order to produce a certain value on the output of the Scan Line Counter (a value which forms the remaining three bits of the address for the Programmable Character Generator), it is necessary to wait for the desired number of scan lines to go by from the end of the Vertical Retrace period. Although the pulses on the Horizontal Sync line occur once every scan line, it is not possible to read this line. However, the Horizontal Sync period - the length of time to display one scan line - is known, so a software timing loop, started at the end of the Vertical Retrace period and designed to terminate "n" times the Horizontal Sync period later will terminate when the output from Scan Line Counter has reached "n". At the termination of the timing loop, the three address bits for the Character Generator will form a known combination. This method provides a way to set up the three remaining address bits on the Character Generator.

The third and final requirement for loading the Programmable Character Generator is that the byte representing the eight-bit pattern for the "nth" line of the character being loaded must be present on the data I/O lines of the Character Generator at the time that the write-enable line is strobed. This is accomplished in the following way.

The write-enable input of the Programmable Character Generator will be strobed when an I/O read operation is performed at any I/O address between ØØ and 3Fhex. To load a byte representing a line of a character pattern, this sort of I/O read operation must be performed when the eleven-bit Character Generator address is properly set up on the input of the Character Generator. If a "register-indirect input" instruction is used, then a side effect of its execution will be that the 16-bit value in the register pair referenced by the I/O

instruction will be used as an address to fetch a byte from Main Memory. The byte so obtained will appear on the Memory Data Bus and the data I/O lines of the Character Generator at the precise time that the Write-enable line will be strobed, causing that memory byte to be loaded into the Character Generator.

Thus, using an IN r,(C) register-indirect input instruction will cause the byte stored at the location referenced by the contents of the BC register to be loaded into the Character Generator. The character code (part of the Character Generator address) under which the byte will be stored is determined by the byte currently being fetched from the Video Display Memory. There is, however, a restriction on possible values of the contents of register C. The contents of this register is used as the I/O address, and this address must be between ØØ and 3Fhex in order to cause the write-enable line on the Programmable Character Generator to be strobed, as stated previously. Therefore, the address stored in BC must fall within the first 64 bytes of any page in Main Memory. (There is no restriction on the value used in register B for this operation.)

The following steps summarize the procedure for loading the Programmable Character Generator:

1. Store the 8-bit ASCII code for the character to be programmed into a byte or block of bytes in the scanned portion of the Video Display Memory. The scanned portion of the Video Display Memory is the last 96 bytes of each of the 128 byte segments which correspond to a row of characters. Note that of these 96 bytes, only the last 80 bytes are actually displayed. The first 16 bytes are fetched by the VDI during the horizontal retrace period but are not displayed.

2. Load the 8-bit dot pattern of the character scan line to be loaded (line 'n') into any byte in the first 64 bytes of any 256-byte page of the main memory.

3. Load the address of this location into the BC register. Since this location must be one of the first 64 locations on any page, bits 6 and 7 of C must always be zero.

4. At the end of the Vertical Retrace period, start a software timing loop. This loop must create a delay from the end of the Vertical Sync pulse to such time as the VDI is fetching characters from the byte or bytes in which the ASCII code to be programmed was previously stored. Furthermore, the VDI must at this time be displaying the same scan line of the characters being displayed as the scan line being programmed. The effect of this procedure is to provide to the character generator the proper address for the character line pattern being programmed. The ASCII code stored in the Video Display Memory provides the eight most-significant bits of the character generator address, and the vertical scan line counter of the VDI provides the remaining three least-significant address bits.

5. On the completion of this loop, execute a Register Indirect Input instruction, such as IN r,(C). The execution of this instruction will cause the 8-bit pattern previously stored at the address specified by BC to be loaded into the character generator at the eleven bit address specified by the ASCII code stored in the Video Display Memory (8 bits) and by the value of the vertical scan line counter at the time that the Input instruction is executed (8 bits).

6. Repeat steps (1) through (5) for each line of each character to be programmed.

The value of the delay in the software timing loop can be calculated from the timing of the CRT display. Characters are fetched from the CRT memory every four CPU cycles. Since 96 characters are scanned in each row, the length of the delay required when loading the 'nth' line of a character is between 4x96xn and 4x96x(n+1) CPU cycles, if the length of time required to respond to the interrupt is ignored. In practice, if the entire 96 bytes of the row are filled with the ASCII code then picking a delay approximately half-way between 384n and 480n would provide adequate timing margins. Using more sophisticated approaches, it is possible to avoid displaying "garbage" on the CRT when loading the character generator. With very precise timing loops the ASCII code may be stored in the undisplayed 16 byte block of each 128-byte row segment. Since the last fifteen of these sixteen bytes are fetched only during the horizontal retrace period, the ASCII codes stored therein never cause characters to be displayed on the CRT. Furthermore, more than one character may be loaded during each Vertical Sync period. There are 32x8 = 256 (40x8 = 320 on 50Hz version) 16-byte blocks available during each Vertical Sync period, so in theory the entire 256-character generator can be loaded in no more than two Vertical Sync periods. To achieve this level of performance precision in the software timing loops is essential.

It should be noted that the Video Display Interface must be operating in Alphanumeric mode while attempting to load the Character Generator. Also recall that wait states are added on the QUARK to extend every Z-80B instruction to mod 4 cycles (i.e. 1-, 2-, 3-, or 4-cycle instructions all take 4 cycles, 5-, 6-, 7-, and 8-cycle instructions take 8 cycles, etc.).

2.5 Colour Video Display Interface

One of the principal features of the hardware for the Colour Video Display Interface of the QUARK/150 is a 16kbyte Colour Translation Table RAM. This RAM is independent of the QUARK's Main Memory, and is used to translate data fetched from the Video Display Memory into patterns of coloured pixels (picture elements) for display on an analog RGB monitor.

Although from a hardware point of view the Translation Table consists of two 16k-by-4 dynamic RAMs, it can be best thought of as a set of sixteen individual symbol tables. Each such table has 256 entries. Each entry in the tables consists of a symbol, which is a sequence of eight "coloured dots". The colours for each of the dots of the symbol may be chosen from a sixteen-colour palette.

Consider first just one of these sixteen symbol tables.

When a data byte is fetched from the Video Display Memory, it is used to select one of the 256 entries in the table. The symbol which was stored for that entry of the table is sent to the RGB Video Shift Register, which will generate the necessary signals on the RGB video outputs to display eight coloured pixels on the monitor corresponding to the selected symbol.

Sucessive data bytes fetched from the Video Display Memory will select particular entries in the table, causing the corresponding symbols to appear on the monitor. In all, 80 bytes will be fetched for each scan line displayed, so 80 sets of eight pixels will be displayed on each line - a total of 640 pixels horizontally.

It should be clear now that the image formed on any line of the display will depend on two things: the symbols loaded into the symbol table, and the data fetched from the Video Display Memory.

Until this point, only one of the sixteen tables has been considered. Selection of a symbol table is controlled by a special byte, called the Table Control Byte. As the Colour Video Display Interface fetches data from memory, the byte which immediately precedes the 80 displayed bytes for each scan line is captured. The function performed by this byte depends on the operating mode (Alphanumeric or Graphics) of the Video Display Interface.

If the interface is operating in the Graphics Mode, then the least-significant four bits of the Table Control Byte determines which of the sixteen symbol tables will be used for the current scan line. The selection of a table is done for each scan line as the Table Control Byte is fetched, so it is possible to select any symbol table for each line randomly.

The situation for Alphanumeric mode is somewhat more complicated, since a single symbol table is insufficient to store the colour data for a complete character set. In the Alphanumeric Mode, a set of

eight symbol tables are used to store what is in essence two complete character sets, each of 128 characters. One of the character sets will be selected when bit 7 of the video data bytes fetched from memory is clear, while the other set will be selected when bit 7 is set. Thus the two character sets are roughly analogous to the "normal" and "reverse" character sets on the monochrome Video Display Interface used on the QUARK/100, etc.

In Alphanumeric Mode, only bit 3 of the Table Control Byte is used in determining the symbol tables which are to be used. If bit 3 is clear, then tables 0 through 7 are used as a character generator for the current character row. If bit 3 is set, then tables 8 through 15 are used as a character generator.

Any of the 16 available colours may be used for each of the 64 pixels forming one alphanumeric character. There is complete freedom of colour choice when designing colour character sets. While a conventional character set might use only two colours (one for the character, one for the background), it is entirely possible to create special symbols with many colours.

The two character sets can also use completely different colours. For instance, the first set might use pink characters against a blue background, while the other set could use orange characters against a black background. Of course for an actual reverse-video effect, one would simply reverse the foreground and background colour combinations for the two sets.

Since only eight of the symbol tables are used to store one pair of character sets, eight tables remain for other uses. One way to use the remaining table space is to store another pair of character sets in the other eight tables. In order to switch usage between the two pairs of character sets, it is necessary only to invert bit 3 of all of the Table Control Bytes in the memory. There are 32 such bytes on the 60Hz version, and 40 on the 50Hz version. Some interesting applications might be possible by using one of the pairs of sets for some rows, and the other pair for the remaining rows.

The other way of using the remaining eight symbol tables is by loading graphics-mode colour data into them. With eight tables available, one could be used for the 160-pixel resolution mode, perhaps two for two sets of 320-mode data, and five for different 640-mode data.

In either the colour Alphanumeric or Graphics modes the 15 bytes in the Video Display Memory which precede the 80 displayed bytes for each line or row are used to refresh the symbol table memories. These tables are located in two 64k dynamic RAMs which must be refreshed periodically to retain their contents. The colour terminal emulators which are included with the operting systems for the Quark/150 automatically stuffs these refresh bytes with values from 0 to 255 so that during one video frame the tables are refreshed sufficiently. The data in the refresh bytes should not be changed, or the tables may lose the data stored in them.

2.6 Colour Alphanumeric Mode

In Alphanumeric Mode, eight of the sixteen symbol tables are used as a character generator. Bit 3 of the Table Control Byte determines whether tables 0 through 7 or 8 through 15 are used as the character generator for the current character row. Each of the eight tables is used to store the pattern for one horizontal line of each of the 256 character patterns. Table 7 (or 15 if the second set of eight tables is used as a character generator) is used to store the pattern for the top line of each of the 256 characters, table 6 (or 14) is used to store the pattern for the second-from-top line of each of the 256 characters, and so on to table 0 (or 8). In Alphanumeric Mode, the eight tables are automatically sequenced from 7 to 0 (or 15 to 8) for the eight scan lines required to display one row of characters.

When they are used for the Alphanumeric Mode, symbol tables 0 through 7 are referred to as character generator 0, and tables 8 through 15 as character generator 1.

2.7 Colour Graphics Mode

In Graphics Mode, the least-significant four bits of the Table Control Byte selects a Symbol Table for use on the current scan line. The selected table is used to translate the graphics data bytes fetched from the Video Display Memory into pixels to be displayed for the length of that line. The selection of any of the 16 Symbol Tables is completely arbitrary, and table selection may be changed on every scan line if desired.

It is convenient to consider the QUARK/150's Colour Video Display Interface as being able to operate in any of three graphics resolution modes. The different modes, which allow spatial resolution of the displayed pixels to be traded for the variety of their colour selection, are defined solely by the data stored in the symbol tables. No specific hardware is used in determining the various resolution modes.

The three principal resolution modes are:

Mode A: 640 pixels horizontally, with a choice of one of two colours for each pixel. One colour is displayed for each bit which is set in the bytes fetched from memory, while the other colour is displayed for bits which are zero. The two colours are pre-selected for each scan line from the 16-colour palette.

Mode B: 320 pixels horizontally, with a choice of one of four colours for each pixel. The four colours are pre-selected for each scan line from a 16-colour palette

Mode C: 160 pixels horizontally, with a choice of any of 16 colours for each pixel.

In Mode A, one of the sixteen possible colours would be selected for the foreground and another colour for the background. The foreground colour will be displayed for any bit which is set in a byte fetched from the Video Display Memory, while the background colour will be displayed for any bit which is clear. To accomplish this particular translation rule, the contents of each of the 256 symbols in the symbol table would be set up in such a manner that the sequence of the two colours of the eight pixels would reflect the sequence of ones and zeroes in the byte which selects the symbol. Under this rule, the binary value ØØØØ ØØØØ should cause eight consecutive pixels in the background colour to be displayed. Therefore, the symbol selected by ØØØØ ØØØØ would be programmed with eight pixels all in the background colour. The "next" symbol, which is selected by the binary value ØØØØ ØØØ1, would consist of seven pixels in the background colour followed by one pixel in the foreground colour. Likewise, the next symbol would consist of six background-coloured pixels, one foreground pixel, and a final background pixel.

The patterns of foreground- and background-coloured pixels thus proceed in a binary fashion for the remainder of the 256 symbols, as illustrated in the table below.

TABLE ADDRESS (HEX)	SYMBOL (PATTERN GENERATED)
ØØ	11111111
Ø1	11111112
Ø2	11111121
Ø3	11111122
Ø4	11111211
..	....
..	....
FE	22222221
FF	22222222

The numbers 1 and 2 represent pixels programmed to display the background or foreground colours, respectively.

Thus with this set of patterns in a symbol table, the user can

position pixels at any of 640 locations (80 bytes/scan line times 8 pixels/byte) across the width of the display. One of sixteen colours (including black) may be chosen as the colour of the foreground pixels, with a second colour for the background.

In Mode B, the resolution is halved in order to double the number of colours available in the symbol table for each pixel. This is done by doubling the width of each pixel. Each pixel will now be displayed as a pair of dots on the RGB monitor.

The contents of the symbol table in this resolution mode would look like the table below, where the numbers 1, 2, 3, and 4 represent the four colours selected for use from the sixteen possible colours:

TABLE ADDRESS (HEX)	SYMBOL (PATTERN GENERATED)
ØØ	11111111
Ø1	11111122
Ø2	11111133
Ø3	11111144
Ø4	11112211
Ø5	11112222
Ø6	11112233
Ø7	11112244
Ø8	11113311
Ø9	11113322
ØA	11113333
ØB	11113344
ØC	11114411
ØD	11114422
ØE	11114433
ØF	11114444
1Ø	11221111
..	....
..	....
FE	44444433
FF	44444444

Studying the example above, it can be seen that although all eight dots of each symbol are used, there are only four distinct pairs of colours represented. Thus each byte in memory can cause only four distinct pixels to be displayed, each twice the width of those in Mode A. However, there is a choice of one of four colours for each pixel, rather than one of two colours as in Mode A

In Mode C, the resolution is halved once again in order to provide the maximum possible choice of colours for each pixel. This is done by defining each of the two halves of each symbol to be the same colour.

The contents of the symbol table in this mode would look like the table below, where the numbers 0, 1, 2,...F represent the sixteen colours.

TABLE ADDRESS (HEX)	SYMBOL (PATTERN GENERATED)
ØØ	ØØØØØØØØ
Ø1	ØØØØ1111
Ø2	ØØØØ2222
Ø3	ØØØØ3333
..	.....
..	.....
ØF	ØØØØFFFF
1Ø	1111ØØØØ
1A	11111111
1B	11112222
1C	11113333
..	.....
..	.....
FE	FFFFEEEE
FF	FFFFFFFF

In this mode, there are only two distinct pixels displayed for each byte in the Video Display Memory, but any of the sixteen colours may be used for each pixel.

The high, medium, and low resolution modes described above represent only some of the possibilities for programming the symbol tables. For instance, in the Mode B medium-resolution mode, it is not necessary that the pair of colour values making up each pixel be identical. Instead, some of the pixels can be programmed as two-colour pairs. This can be useful in "dithering", and in creating "tone" areas on the display. The concept of dithering can be applied to the low-resolution Mode C, as well.

Going beyond the idea of loading regular patterns of data into the symbol tables, it becomes clear that the Translation Table can permit substantial data compression for the display of colour graphic information. By the judicious choice of symbols for the tables, and by judicious selection of the table into which any particular symbol is loaded, it could be possible to achieve an apparent resolution of, say, 640 by 240 pixels, with a choice of any of the 16 colours for any pixel.

This sort of data compression might be done by predetermining the set of symbols which would be required to represent a given image. The set of symbols so obtained would be sorted out on a scan-line-by-scan-line basis to find "maximally redundant sets" of symbols (i.e., finding sets of scan lines which share the largest number of symbols). The symbols in these redundant sets would then be assigned to the smallest number of symbol tables sufficient to hold all of the symbols. If the number of symbols exceeded the available table space,

then further optimizing procedures would be used to reduce the number of symbols, or by assigning symbols to the table in different ways.

Finally, the appropriate data would be loaded into the Video Display Memory. The Table Control Bytes would be set up so that the symbol table containing the compressed colour information required for each scan line would be correctly selected.

Needless to say, such a program would be highly non-trivial. The purpose of discussing it here is to illustrate some of the flexibility inherent in the design of the Colour Video Display Interface.

Another capability inherent in the design of the Colour Video Display Interface is that of animation using the symbol tables. Consider the interface operating in the Graphics mode, with several tables, say numbers 0, 1, and 2, set up for the low-resolution 160-pixel mode. Suppose in table 0 the values 0, 1, and 2 have been set up to produce the colours red, green, and blue, respectively. In table 1, these same three values will produce the colours green, blue, and red, respectively. Finally table 2 will give blue, red, and green, respectively for the same values.

Now, if a "picture" consisting of three vertical bars is "drawn" in the video memory using the values 0, 1, and 2, then depending on which of tables 0, 1, or 2 is used throughout the video memory, the display will show red, green, and blue bars in a sequence determined by the values used in the video memory.

Suppose table 0 was used for every display line in the video memory, and this resulted in a red, then a green, and then a blue bar. If all of the Table Control Bytes are changed so that table 1 is now used on every line (without changing the data being displayed), the bars will assume their new colours as dictated by the contents of table 1. Thus the previous red-green-blue sequence appearing on the display will change to a green-blue-red sequence. If the Table Control Bytes are changed again so that table 2 is used, then the display changes to blue-red-green. If the Table Control Bytes are changed at a suitable rate, the bars can be made to appear to move across the screen.

Note that this sort of animation effect involves very modest amounts of data movement. To change all of the Table Control Bytes in the video memory is only 256 (or whatever number of scan lines are in use) write operations. This is contrasted with animation in which the displayed data in the video memory itself is moved. In the Graphics mode, this amounts to at least 20kbytes to be moved.

The animation effects are not limited to the low resolution mode, nor are they useful only with a colour monitor. Up to sixteen tables could be employed for more sophisticated animation schemes. It is also not necessary to change the Table Control bytes on every line in memory, but rather only on those scan lines on which the animation effect is to occur.

One conventional way to configure the sixteen symbol tables

forming the Translation Tables is to split the symbol table usage between the Alphanumerics Mode and the Graphics Modes. Eight of the tables can be loaded with the patterns necessary to implement a pair of colour character sets for the Alphanumeric mode. Of the remaining eight tables, one can be used to provide the symbols necessary for low-resolution Mode C graphics, the 160-pixel, 16-colour mode. Several tables could be used for medium-resolution Mode B graphics, which would provide the 320-pixel 4-colour mode. Each of the tables employed for Mode B would be loaded with different sets of the four colour choices. The remaining symbol tables would be available for the high-resolution Mode A graphics. As with the tables being used for the medium-resolution mode, each of the tables supporting the high-resolution mode would normally be loaded with a different pair of colours.

Other ways of configuring the symbol tables are possible, of course. If two character sets are required, tables 0 through 7 can be used for one set, and tables 8 through 15 another set. If more colour choices are required in the 640-pixel mode, then eight or even sixteen character generators could be used for this mode, although using more than eight would preclude the use of a set of eight tables as a colour character generator for the Alphanumeric mode.

2.8 Loading the Colour Translation Table

The 16kbyte Colour Translation Table used in the Colour Video Display Interface of the QUARK/150 is loaded by a procedure similar to that used to load the Programmable Character Generator on the QUARK monochrome Video Display Interface (see sec. 2.3). However, the procedure is more complicated.

A program called CLRLD.COM is provided with the software package for the QUARK/150 to load character sets into the Translation Table (see sec. 8.3). Additional routines are used for loading individual or sets of symbol tables for the Graphics Mode. Thus, the information presented in this section is largely for information only; the user is unlikely to ever need to write software to load a symbol table.

The following is a brief outline of the method by which a symbol table is loaded.

An 8-bit value is placed into the scanned portion of the Video Display Memory. This value is used to select one of the 256 entries in the selected symbol table.

A special byte is loaded into the Main Memory of the QUARK/150. Bits 0-3 of the byte will select one of the 16 symbol tables to be loaded. Bits 4 and 5 will determine which of four adjacent pixel pairs in the selected entry will be loaded.

The location of this byte in memory is loaded into the BC

register. However, several restrictions on the value in the BC register (and thus on the address of the byte in memory) apply.

First, the value in the C register must be between ØØ and 3Fhex, as this value is used as the I/O address to select and write the symbol table. Therefore the byte must be on the first quarter of a page in memory.

Second, bit 0 in the C register determines which one of the selected pair of pixels in the selected symbol table entry is to be loaded. Therefore the byte must by on an even address if the "0" pixel of the pair is to be loaded, or on an odd address if the "1" pixel is to be loaded.

Third, bits 1 through 4 of the B register determine which of the 16 colour values will be loaded into the selected pixel. Thus the byte must be positioned so that bits 9-12 of its 16-bit address match the colour value to be loaded.

Bits 2-5 of the C register and bits 0 and 5-7 of the B register are don't-care bits for this procedure.

Now, a software timing loop is used to determine when the byte previously stored in the Video Display Memory is being fetched by the Colour Video Display Interface. When it is being fetched, a Register-Indirect Input instruction, such as IN r,(C) is executed. The execution of this instruction causes the colour value in bits 1-4 of the B register to be loaded into the pixel selected by bit 0 of the C register and bits 4-5 of the byte stored at the memory address given by the contents of BC. The symbol table used for this load is determined by bits 0-3 of the byte stored at the address given by the contents of BC, and the entry in the symbol table used is determined by the byte being fetched from memory when the register-indirect input instruction is executed.

For additional information on this sort of loading procedure, compare with sec. 2.4 for the monochrome version of the QUARK.

2.9 RGB and monochrome monitors

The Colour Video Display of the QUARK/150 is intended for use with analog-input RGB video monitors. Such monitors usually have either separate Red, Green, Blue and composite sync inputs, or Red, Blue, and Green + Sync inputs.

For the four-input types, the QUARK provides the RED, GREEN, BLUE, and COMPOSITE SYNC outputs on its 96-pin connector. The RGB monitor may be connected directly to these outputs. The polarity (positive-going or negative-going) of the composite sync line is determined by jumpers J14 and J15. Installing J14 makes the sync signal positive-going, while installing J15 makes the signal negative-

going. See Table V in the Appendix for the pins of the QUARK's 96-pin connector used for the colour video outputs.

Some RGB monitors combine the Green and negative-going Sync lines. This can be done with the QUARK/150 by tying the GREEN video output to the COMPOSITE SYNC output, with J15 installed for negative sync. Since the SYNC output is pulled high on the QUARK/150, the RED and BLUE video outputs should be pulled high externally if the GREEN and SYNC lines are connected. All three of these lines should be terminated to ground at the monitor by 75ohm loads.

A monochrome composite video monitor may also be used with the QUARK/150. Tying together all of the colour video outputs - RED, GREEN, BLUE, and SYNC, with J15 installed for negative-sync will produce a standard composite video output. The "colour" outputs will now generate a multiple-level grey-scale video signal.

This page intentionally left blank.

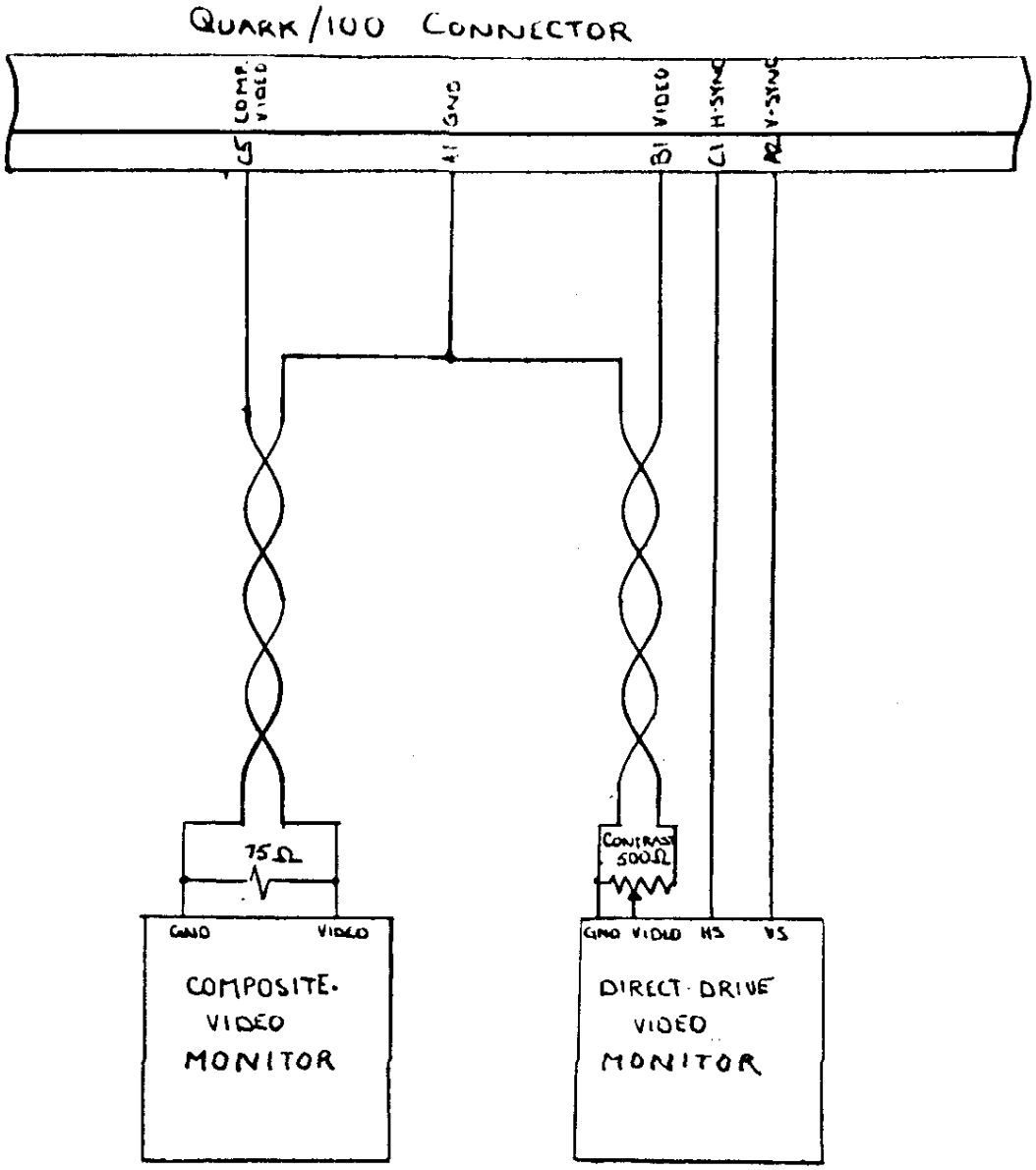


Figure 2.1 Connections for direct-drive and composite video monitors. If the composite video output is used, it should be terminated by a 75-ohm resistor at the monitor. If the direct-drive video output is used, the video input of the monitor should be driven through a 500-ohm potentiometer (unless the monitor provides a contrast control internally).

3.0 Peripheral Interfaces

The Megatel QUARK provides a number of parallel and serial I/O lines. While some of these lines are intended for use with specific peripherals, such as parallel- or serial-interface printers, many of these I/O lines may be used for more general purposes if the intended devices are not being used in a particular application. A discussion of these I/O ports follows. In addition, specialized peripheral subsystems, such as the Floppy-disc Controller and the QUARK/200's Local-Area Network Interface provide a high level of I/O support.

3.1 Parallel Printer Interface

The QUARK includes a port intended for the connection of an eight-bit parallel-interface printer. This port consists of an eight-bit latch for the output data, a Data Strobe output line, and an Acknowledge input line. All of the input and output lines for this port are TTL-compatible.

Eight-bit parallel data is written to the port by an output instruction to I/O address 5Fhex. The data appearing on the output pins of the QUARK (see Table I for the pinout of the connector) represent the true state of the data written to the port.

The Data Strobe line for the port is controlled by the CA2 control line of the VIA. The CA2 control line must be configured as an output in the Peripheral Control Register (PCR) of the VIA, by setting bits 2 and 3 of the PCR. The CA2 output drives a TTL inverter, the output of which is connected to the Data Strobe output pin (pin C-13). Thus the logic state of the actual Data Strobe output is the inverse of the state of the CA2 line, as determined by bit 1 of the PCR. This TTL inverter is capable of sinking 24mA.

The Acknowledge input for the Parallel Printer Port is directly connected to the CA1 input line of the VIA. The active transition of the CA1 input of this line will set a flag in the Interrupt Flag Register (IFR) of the VIA. The setting of this flag may also generate an interrupt if the appropriate bit in the Interrupt Enable Register (IER) of the VIA is set. This interrupt can be used to interrupt the CPU when the printer is ready to accept another character, depending on the operation of the printer.

Handshaking using the Data Strobe and Acknowledge lines is not automatic, that is that the routine handling the Parallel Printer Interface must write the output latch, toggle the Data Strobe line in the manner required for the interface of the printer, and then act accordingly for the printer's response on the Acknowledge input.

If it is not desired to use this port with a parallel printer,

then the eight Data lines, the Acknowledge Input line, and the inverted output line from CA2 may be used for other purposes such as might be required for a particular application of the QUARK.

Consult the Data Sheet for the VIA for more detailed information regarding the use of the CA1 and CA2 control lines.

3.2 Parallel Keyboard Interface

The QUARK provides an interface for an ASCII-encoded parallel-output keyboard. This interface uses the Port A I/O lines and the CA2 control line of the PIA. As initialized by standard CP/M operating systems provided on the Distribution Diskettes, active-high ASCII data present on the PA0-7 lines of the PIA will be read after a negative-going stobe pulse on CA2. Note that the data present on the input lines is not actually latched into the Input Data register when the strobe occurs, so the keyboard data must remain valid between the strobe and the read. (This is not usually a problem.)

If it is not intended to use an encoded keyboard for a particular application of the QUARK, then these eight I/O lines and the CA2 control line may be used for any other I/O functions which might be required. It is possible, for instance, to connect an un-encoded keyboard to the QUARK using the I/O lines from Port A of the PIA and Ports A and B of the VIA. For such a keyboard the user would include his own keyboard-scanning routine as part of the CP/M BIOS. (This software is not provided by Megatel.)

The I/O address for reading or writing either Port A or Data Direction Register A of the PIA is 74hex. The addresses of the PIA Control Register A is 75hex. For information on programming the PIA, consult the data sheet for the PIA (part number 68A21). Table VIId in the Appendix gives the pin connections required for connecting a keyboard to this interface.

3.3 Full-Duplex Serial Interface

The QUARK provides a full-duplex asynchronous serial data port. This port uses an Asynchronous Communications Interface Adapter (part number 68A50) and includes line drivers and receivers for compatibilty with RS-232C signal levels. As implied by the term full-duplex, this port can perform bidirectional simultaneous communication.

This port allows the QUARK to be connected to standard terminals and communications peripherals, such as telephone-line modems. The port also allows a terminal to be used as the console I/O device.

Parallel data written to the Transmit Data Register of the ACIA will be transmitted serially on the TxDATA pin of the QUARK connector (pin A-3). Serial data received on the RxDATA pin (pin C-2) is read from the Receive Data Register. The I/O address of both the Transmit and the Receive Registers is 79hex.

A total of four protocol lines are provided for the Port. The Clear-to-Send input line (pin C-4) provides direct control of the transmitter of the ACIA. The Data Set Ready input (pin B-4) drives the ACIA's Data Carrier Detect input, which provides direct control of the receiver of the ACIA. The Request-to-Send output (pin B-3) is controlled by the RTS bit in the Control register of the ACIA. The operation of these three lines is discussed in the data sheet for the ACIA. The Data Terminal Ready output (pin C-3) is controlled by the PB7 I/O line of the PIA. The PB7 line should be configured as an output by setting bit 7 of Data Direction Register B in the PIA (I/O address 76hex). All of the four protocol lines are RS-232C compatible.

The serial communications speed (or baud rate) for the serial input and output data is controlled by one or both of the programmable timers in the VIA. On the QUARK the PB7 line from the VIA is connected directly to the Transmit Clock input of the ACIA. When Timer 1 is operated in the free-run mode (see the data sheet on the VIA), a square wave is generated on the PB7 output line. This square wave provides the basic Transmit Clock frequency, which is then divided by 1, 16, or 64, according to the settings of bit 0 and bit 1 of the ACIA Control Register.

The Receive Clock input of the ACIA can be connected to the Transmit Clock by jumper J3. Unless ordered otherwise, this jumper is installed at the factory. With J3 installed, the Transmit and Receive baud rates will be identical, both being generated by Timer 1 of the VIA.

Timer 1 uses a 16-bit counter which, in the free-run mode, is automatically re-loaded from the 16-bit Timer 1 latch each time the counter reaches zero. The value in the latch determines the period of the square wave appearing on PB7. The period of this square wave is given by

$$(2N+3.5) * t_E,$$

where t_E is the period of the system E-clock and N is the value in the Timer 1 latch. The count-down clock for Timer 1 is the system E-clock. The period of the E-clock is 670.5 ns on 60Hz models, and 645.2 ns on 50Hz models. Table II in the Appendix gives the values for the Timer 1 latch required to generate commonly-used baud rates on this serial port. Because Timer 1 is a true 16-bit timer, it is able to produce the widest range of baud rates.

If "split" baud rates on the Full-duplex Serial Port - where the Transmit and Receive baud rates are independently generated - are required, then J3 should be removed and J4 installed. The installation of J4 connects the CB1 line from the VIA to the Receiver Clock input on the ACIA. Pulses may be generated on the CB1 line when the Shift Register (SR) of the VIA is operated in the "Shift out free-running at Timer 2 rate" mode. Values for the Timer 2 low-order latch (I/O address 68hex) to produce commonly-used baud rates can be found in Table III(a) and III(b) in the Appendix. Jumpers J5 and J6 should not both be installed when attempting to use split baud rates, or the Shift Register output on CB2 will short out the T2-generated clock output on CB1.

To enable this Shift Register Mode, the Auxiliary Control Register (ACR, I/O address 6Bhex) of the VIA must be written with bits 2 and 3 low and bit 4 high. According to the manufacturer's data sheet for the VIA, it is also necessary to perform an I/O read or write operation to the SR after setting up the Shift Mode to start the clocking of the SR (and the clock pulse output on CB1). Note that using Timer 2 and the Shift Register for this purpose precludes the use of the Simplex Serial Port, which uses the VIA's Shift Register, as a serial data channel. See Sec 3.4 for details on the Simplex Serial Port.

Although Timer 2 uses a 16-bit counter, only the least-significant eight bits are automatically re-loaded in its free-run mode. Thus, the range of baud rates directly available from this timer is more limited than that of Timer 1. However, this range can be extended by using the divide-by-16 and divide-by-64 modes of the ACIA. Note that the divide ratio selected applies to both the Transmit and Receive clocks. Table IV gives the values for the Timer 2 latch required for common baud rates. The period of the basic clock signal generated by Timer 2 is given by

$$(2N+4) * t_E,$$

where t_E is the period of the system E-clock (given previously) and N is the value in the Timer 2 latch.

For more information on programming the Full-duplex Serial Port, consult the data sheet for the ACIA (part number 68A50), as well as those sections of the data sheet for the VIA (part number 6522A) dealing with Timers 1, 2, and the Shift Register, and those sections of the data sheet for the PIA (part number 68A21) dealing with the programming of the Port B I/O lines.

Note that on the QUARK the negative supply voltage used for the RS-232C drivers is developed on-card by a charge-pump circuit. As a result, the voltage swing on the RS-232C outputs is usually in the range of +11V to -8V.

3.4 Simplex Serial Interface

In addition to the Full-Duplex Serial Port, the QUARK provides a simplex (unidirectional) asynchronous serial data port. This port uses the Shift Register of the VIA (part number 6522A) and includes line drivers and receivers for compatibilty with RS-232C signal levels.

The Simplex Port allows the QUARK to be connected to receive-only or transmit-only peripherals with serial interfaces, such as printers or serial-output encoded keyboards. The port can also be used for other purposes, such as generating tones. By means of jumpers J5-8, the port can be configured for serial output or input (but not both). One protocol line is also provided. Depending on which jumpers are installed, this protocol line may be used as an input or an output.

For serial output, the Shift Register (SR), the CB2 control line, and the PB6 I/O line of the VIA are used. In this mode, data in the shift register is shifted out on the CB2 pin of the VIA. With J6 installed (done at the factory unless ordered otherwise), the serial data is shifted to RS232C output voltage levels and is available on pin A-4 of the QUARK's connector. The protocol line, an RS-232C compatible input for this mode, is pin B-2. Jumper J8 (also installed at the factory) allows the state of the protocol line to be read on the PB6 I/O line of the VIA. Bit 6 in Data Direction Register B of the VIA must be zero to allow the use of PB6 as an input.

In order for the Shift Register to be used in the output mode, bit 4 in the Auxilliary Control Register (ACR) of the VIA must be set. With this bit set, bits 2 and 3 of the ACR will determine the rate at which the SR is shifted, as well as its operational mode.

Summary of Shift Register output modes (ACR-4 = 1)

ACR-3	ACR-2	Remarks
0	0	Continuous shifting at T2 rate. Useful for waveform-generation applications.
0	1	8 bits only shifted at T2 rate after each SR load. SR Interrupt Flag set after 8 bits shifted.
1	0	8 bits only shifted at E-clock rate after each SR load. SR Interrupt Flag set after 8 bits shifted.
1	1	8 bits or more shifted at CB1 input rate. SR Interrupt Flag set after 8 bits shifted. Install J3 & J4 for Timer 1 clock to CB1 input.

With ACR bit 2 and 3 cleared, Timer 2 determines the rate at which data is shifted out of the SR. In this mode, shifting is continuous, and does not stop automatically after eight bits have been shifted. This mode can be used for generating retangular waveforms (repeating patterns of eight bits) on pin A-4. This might be useful in some applications for generating audio signals.

In the next mode (ACR-2=1, ACR-3=0), shifting stops automatically after eight bits are shifted, and the SR Interrupt Flag in the Interrupt Flag Register (IFR) of the VIA is set. If the SR Interrupt Enable bit in the Interrupt Enable Register (bit 2 of the IER) is set, the VIA will assert its Interrupt Request output, which will cause a Z-80 interrupt if the Z-80's interrupt system is enabled. The shifting rate is determined by Timer 2, as in the previous mode. This is the mode used by the Serial Printer Handler included in the Operating System.

The next mode (ACR-2=0, ACR-3=1) operates in the same manner as the previous mode, except that the System E-clock is used as the shift clock, rather than Timer 2. The frequency of the E-clock is 1.49MHz for 60Hz models, and 1.55MHz for 50Hz models.

In the final mode (ACR-2=1, ACR-3=1), the shift rate is controlled by pulses applied to the CB1 input on the VIA. If jumpers J3 and J4 are installed, then the PB7 I/O line will be connected to the CB1 line (as well as to the Transmit and Receive clocks on the ACIA - see Sec 3.3). This allows Timer 1, normally used to generate the baud rates for the ACIA, to also generate the shift clock for the Shift Register. Note that J5 must not be installed in this mode, or the CB1 clock input will be shorted to the CB2 SR output.

For serial input, the Shift Register, and the CB1 and CB2 control lines are used. In this mode, data is shifted into the SR on the CB2 line. Pin B-2 on the QUARK's connector provides an RS-232C compatible input line for the Simplex Port input mode. Jumper J7 should be installed to allow the signal from pin B-2 to reach the CB2 input on the VIA. (Note that J8, installed at the factory unless ordered otherwise, will connect CB2 and PB6 together when J7 is installed. If both J7 and J8 are installed, PB6 must be configured as an input or it will contend with the signal at CB2.)

In order for the Shift Register to be used in the input mode, bit 4 of the ACR must be cleared. As in the output mode, bits 2 and 3 of the ACR will determine the rate at which the SR is shifted, as well as its operational mode.

Summary of Shift Register input modes (ACR-4 = 0)

ACR-3	ACR-2	Remarks
0	0	Shift register disabled.
0	1	8 bits only shifted in at T2 rate. SR Interrupt Flag set after 8 bits shifted. Shift pulses generated on CB1 during shifting.
1	0	8 bits only shifted in at E-clock rate. SR Interrupt Flag set after 8 bits shifted. Shift pulses generated on CB1 during shifting.
1	1	8 bits or more shifted at CB1 input rate. SR Interrupt Flag set each time 8 bits shifted in. Install J3 & J4 for Timer 1 clock to CB1 input.

In the first mode (ACR-2=0, ACR-3=0) the SR is disabled. The SR can be read or written, but no shifting occurs, and the CB1 and CB2 lines are under the control of the appropriate bits in the Peripheral Control Register.

In the next mode (ACR-2=1, ACR-3=0), data is shifted into the SR at a rate controlled by Timer 2. Shift pulses are generated on the CB1 line. If J5 is installed, then these pulses will appear (at RS-232C signal levels) on pin A-4 of the connector. (Note that if J4 is installed, these pulses, at TTL signal levels, will also appear on the Receive Clock input of the ACIA. This may interfere with the operation of the ACIA.)

The third mode is similar to the previous, except that the E-clock is used to control the shift rate.

In the final mode, pulses on the CB1 line control the shift rate. The only way to provide a signal input to the CB1 line while CB2 is being used for the SR input is to install J3 and J4. This connects the PB7 line to the Transmit and Receive clocks of the ACIA and to the CB1 input. This allows Timer 1 to generate a clock signal for the ACIA and for the Shift Register.

Values for Timer 1 to generate commonly-used baud rates for the Simplex Port are given in Table IV.

By various combinations of straps, the lines and RS-232C drivers and receivers associated with this port may be used for a variety of purposes. It is recommended that the schematic for the I/O section and the data sheet for the VIA (part number 6522A) be studied carefully to determine if a desired application can be accomodated with this port.

3.5 Parallel I/O lines

The QUARK provides fourteen general-purpose I/O lines (not including the nine lines on the PIA used for the parallel keyboard interface). These lines are connected to Ports A and B of the VIA. Each of these lines may be programmed to act as an input or as an output by setting or clearing the corresponding bit in the Data Direction Register.

Parallel Port 2 of the QUARK provides eight I/O lines. These are connected to the PA0 to PA7 lines on the VIA. These lines are not dedicated to any particular purpose in the system software for the QUARK.

Parallel Port 3 of the QUARK provides six I/O lines, which are connected to the PB0 to PB5 lines. Under the standard distributed operating system, the PB0 line is configured as the "bell" output from the QUARK. When an ASCII control-G character (code 07) is encountered by the terminal driver, a square wave will be produced on this output. The PB0 output line may be capable of driving some piezo-electric acoustic transducers directly, or an external buffer amplifier using a transistor or gate can be used to drive a small speaker.

The two remaining I/O lines of Port B on the VIA are intended for some specific uses. PB7 is normally used as the output line for Timer 1-generated baud rates for the Full-Duplex Serial Port (ACIA) or for the Simplex Serial Port. PB6 is used in the output mode of the Simplex Serial Port as a protocol input line. Neither PB6 nor PB7 are available directly on the QUARK's connector. (See Fig. 3.5.)

Consult the data sheet for the VIA for additional information on using Port A or B of the VIA. Note that when the VIA's Port B lines are configured as outputs, the value read in Input Register B (I/O address 60hex) is the value programmed for the corresponding bit in Output Register B, not the logic level actually present on the output pin of the VIA.

3.6 Serial Keyboard Interface

The QUARK can be used with serial-output keyboards. Many applications may require the use of a "detachable", or stand-alone keyboard. These keyboards generally use asynchronous serial communication over a single pair of conductors to reduce the size of the cable connecting the keyboard to the main housing of the system.

The Simplex Serial Interface can be used for this application. When strapped in the input mode (explained below), the CB2 control line of the VIA as well as that device's Shift Register can be used as a receiver for the asynchronous keyboard output. Timer 2 can be used

to generate the clock for the Shift Register.

The basic method of operation is to enable the CB2 control line to act as an edge-triggered independent-interrupt input. This mode is enabled by setting bit 5 and clearing bit 7 of the VIA's Peripheral Control Register (PCR, I/O address 6Chex). Also, the state of bit 6 in the PCR determines the active edge on CB2 which will set the CB2 Interrupt flag. The active transition will be positive-to-negative when bit 6 is cleared, and negative-to-positive when bit 6 is set. Bit 6 should be configured so that the transmission of the start bit from the keyboard will be recognized as the active transition. Note that the RS-232C line receiver driving the CB2 control line will invert the signal from the keyboard.

Jumper J7 must be installed to allow the output of the RS-232C receiver whose input is from pin B-2 to be connected to the CB2 line on the VIA. Jumper J8, which connects the output of the receiver to the PB6 I/O should either be removed, or PB6 must be configured as an input. If this is not done, then the output on PB6 will short out the input on CB2. Although pin B-2 would ordinarily be driven by an RS-232C driver, most TTL serial keyboard outputs should be able to drive this input properly.

When the keyboard transmits the start bit, the active transition of the CB2 line will set the CB2 Interrupt flag in the Interrupt Flag Register. If the CB2 Interrupt Enable bit in the Interrupt Enable Register is set, the VIA will generate an interrupt to the Z-80B.

Once in the interrupt service routine for the serial keyboard, the VIA shift register must be enabled in one of its input modes. The most useful mode is likely to be the "Shift in Under Control of T2", since in this mode Timer 2 is used to generate the Shift Register clock. If the Shift Register Flag in the IFR is not set, then shifting will begin as soon as this mode is enabled. At the completion of eight shifts, the SR flag in the IFR will be set, and, if the SR Interrupt in the IER is enabled, the VIA will generate an interrupt to the Z-80B. The contents of the SR can then be read, and the SR disabled. This allows the CB2 input to return to the edge-triggered independent-interrupt mode, so that the next start bit from the serial keyboard will be recognized.

Baud rates for the Shift Register can be found in Table III. It should be noted that depending on the interrupt service routine for the serial keyboard as well as the timing of the serial data stream it is possible that the first bit (or the start bit) will not be recognized as a data bit. It is recommended that the data sheet for the VIA and Sec. 3.4 on the Simplex Serial Interface be studied closely when designing a serial keyboard interface for the QUARK. As of the date of issue of this document, Megatel does not offer software for a serial keyboard interface using the simplex port.

The Full-duplex port may also be used for a serial keyboard input. This port would be particularly simple to implement if the keyboard's output is standard 7- or 8-bit RS-232C protocol. If this is the case, then it is necessary only to enable the ACIA properly for

the receiving mode used, and then poll the Received Data Register Full lag for an incoming character. The CP/M operating systems distributed with the QUARK generally allow the "TTY:" or "AUX:" input (the full-duplex port) to be installed as the console input device. See the Installation maunual for more information.

3.7 Special I/O functions

Some of the control lines on the PIA are connected internally on the QUARK for special functions. Care should be taken when programming the PIA that these functions are not disturbed by inadvertently modifying the Control Registers.

The CA1 control line on the PIA is connected to the Vertical Sync output for the Video Display. The CA1 line can be programmed to generate an interrupt to the Z-80B on either the rising or falling edge of the Vertical Sync line. This allows the use of a real-time clock function, a feature which is incorporated into some of the QUARK operating systems.

The CB1 control line on the PIA is connected to the Interrupt Request (INTRQ) output from the 1793 Floppy Disc Controller. The 1793 produces an interrupt request signal at the completion of any command. The CB1 input is initialized as a positive-transition active input. When a Floppy Disc Controller interrupt occurs, the IRQB flag in the PIA will be set. This configuration of the CB1 line must not be altered if the Floppy Disc Interface is to operate properly.

The IRQ bit can be programmmed to generate an interrupt to the Z-80B when this occurs, although this interrupt is not used under the standard operating system for the QUARK.

The CB2 control line is the internal Boot Mode control line. This line will be set to act as an input after a system reset has occured. When it is configured as an input, or as an active-high output, a pull-up resistor pulls the Boot Mode line high, putting the QUARK into Boot Mode operation. (See Sec 1.2.) At the end of the Bootstrap routine, this line will be cleared and configured to act as an output, returning the system to Normal mode operation. There is no need for the user to change the state or configuration of this line during normal operation of the QUARK.

All of the PIA Port B I/O lines are used for specific output functions. PB0-3 are the Floppy Disc SEL0 to SEL3 outputs. PB4 is the Floppy Disc SIDE select output. If floppy disc drives requiring a "Low Write-current" input are used, QUARK operating systems can be installed to allow any of the SELect lines or the SIDE line to be used for this purpose. (See the Installation manual.)

PB5 is the Floppy Disc Single/Double-density control line. Single-density (FM) operation is selected when this line is at a high

level. PB6 is the Alphanumeric/Graphics Mode line. A high level on this output selects Alphanumeric Mode. PB7 is the Full-duplex Serial Interface DTR output. A high level on PB7 causes a negative-voltage output on pin C-3.

3.8 Floppy Disc Interface

The on-card Floppy Disc Interface is capable of controlling up to four double-sided floppy disc drives. Both single-density (FM) and double-density (MFM) recording formats are supported, and either 8-inch or 5-1/4-inch drives may be used. The selection of single- or double-density operation is under software control, through bit 5 of PIA port B (PIA PB5). This output is set (logical 1) for single-density operation, and is cleared (logical 0) for double-density. Jumper 2 (designated J2), located adjacent to the crystal, is open (not installed) when 5-1/4 inch drives are to be used, and is closed (installed) when 8-inch drives are to be used. The QUARK board is shipped with J2 installed if the system was ordered with the distribution software on an 8-inch diskette, while J2 is not installed on boards ordered with 5-1/4-inch diskettes.

Four Drive Select lines (SEL 0 to SEL 3) are provided to select one of four floppy disc drives. Additionally, a Side Select output (SIDE) is provided for use with double-sided drives. The Select and SIDE outputs are controlled by PB0, 1, 2, 3, and 4 of Port B of the PIA. These lines are initialized to act as outputs in the Bootstrap PROM routine.

The Side Select line or one of the Drive Select lines may be used as a Low Write-current line. Some floppy disc drives require an external control line to reduce the write current to the recording head when writing the inner tracks on the diskette. If the drives to be connected to the QUARK require such a control line, then the system installation program allows any of the Drive Select lines or the Side Select line to be used for this purpose. Note that using a Drive Select line for this purpose would reduce the maximum number of drives that could be connected to the QUARK to three. If only single-sided drives are used, then the Side Select line may be used as the Low Write-current line with no effect on the maximum number of drives permitted.

Note that normally all of the drives connected to the QUARK must be of the same size, that is, that a combination of 8-inch and 5-1/4-inch drives cannot be directly supported without a modification to the board. However, since the installation of J2 merely ties the internal Disc Size Select line low (permitting 8-inch operation), a special control line connected to the Disc Size Select line can be used to change the selected Disc Size under software control. If a particular application of the QUARK requires between one and three 5-1/4-inch floppy disc drives and only one 8-inch drive, then the simplest solution is to install a strap from the Disc Select line (SEL0, 1, 2,

or 3) which is to be used to select the 8-inch drive to the Disc Size Select line. Since the Drive Select lines are active low, then whenever the 8-inch drive is selected, the Disc Size Select line will be pulled low, enabling 8-inch drive operation. When any of the 5-inch drives are selected, the Drive Select line for the 8-inch drive, and thus the Disc Size Select line, will be at a logical high level, putting the Floppy Disc Interface into 5-1/4-inch operation, just as is required.

If more then one 8-inch drive is to be used with some number of 5-1/4-inch drives, then the Disc Size Select line must either be connected to an unused parallel output line from the PIA or VIA on the QUARK (which can be then controlled in a software patch to the BIOS), or the Disc Select lines used to select the 8-inch drives must be AND-ed together and the result of this combination used to control the Disc Size Select line. The latter approach will, of course, require an external AND gate as well as the strap connecting the gate's output to the Disc Size Select line.

The outputs from the QUARK to the floppy disc drives are driven by medium-current low-power Schottky TTL drivers. Inputs from the floppy disc drives are terminated by 150ohm pullup resistors to the +5V supply on the QUARK. The floppy disc drive most distant from the computer should have 150ohm terminating pullups on the output lines for proper transmission line characteristics. None of the intermediate drives should have passive loads on either input or output lines.

Most of the remaining hardware for the Floppy Disc Interface is incorporated within the Western Digital 1793-02 Floppy Disc Controller (or equivalent). Consult the data sheet for this device for more information on the Floppy Disc Interface.

3.9 Local-Area Network Interface (QUARK/200 only)

The Local Area Network Interface used on the QUARK/200 is compatible with the Corvus OMNINET network system. The hardware of the interface includes a slave processor to handle message generation, transmission, reception, acknowledgement, and errors. Network control is distributed using a carrier-sense multiple-access scheme with collision avoidance (CSMA/CA). The hardware performs the lower four layers of the ISO seven-layer network model.

The network interface adds two I/O ports in the Z-80B's I/O space. These ports are used for passing addresses to areas in main memory of command vectors, for interrogating the ready status of the LAN interface, and for LAN interrupt control.

I/O port address 90hex is used for passing the three-byte main memory address of the command vector to the LAN interface. Consult the "Omninet Programmer's Guide", found at the back of this manual,

for a complete description of the format of the command vector and its address.

The same port address (90hex), when read, returns the LAN READY status in bit 7. If bit 7 is set, the the LAN interface is ready to accept a byte of the command vector address. If this bit is clear, then the LAN is not ready to accept a command vector address byte. Bits 0 to 6 are "don't care" bits.

Interrupts from the LAN interface to the Z-80B are controlled through port 98h. If a LAN interrupt is pending, then bit 7 of this port will be set when the port is read. If no interrupt is pending, then this bit will be clear. Writing to this port clears any pending interrupts, and, depending on the state of bit 7 when the port is written, optionally enables or disables the generation of LAN interrupts. If bit 7 is set when this port is written, then the LAN interface will be enabled to generate interrupts. If bit 7 is clear, then interrupts from the LAN interface will be disabled. Bits 0 to 6 of this port are "don't care" bits for both I/O read or write operations.

A copy of the "Omninet Programmer's Guide" by Corvus Systems is included with the QUARK/200 manual. Further information regarding the operation of the Omninet interface is proprietary to Corvus. The "Omninet Technical Reference Manual" is available from Corvus after a non-disclosure agreement has been signed with them. Their address is

Corvus Systems Inc.,
2029 O'Toole Ave.,
SAN JOSE, California,
U.S.A. 95131

3.10 Expansion of the Megatel QUARK

Although the various members of the Megatel QUARK family of single-board computers provide all of the functions necessary to integrate standard peripheral devices into a complete, stand-alone computer system, there may be special or additional I/O functions needed in some applications. The QUARK can accomodate these special functions through its Peripheral Expansion Bus. This bus provides external access to the eight data lines, the six least-significant address lines, and to appropriate timing and select lines.

The Read ($\overline{RD}$) and Write ($\overline{WR}$) lines generated by the Z-80B are brought out on the Peripheral Expansion Bus. These two lines would be used with 8080-compatible peripheral devices, such as the 8251 UART, or the 8255 PPI. To connect 6500- and 6800-compatible devices the E-clock output and the $\overline{WR}$ line are used. For these devices, the $\overline{WR}$ line functions as the 6500/6800 RD/$\overline{WR}$ line.

The Z-80B's Interrupt input line ($\overline{INT}$) is available on the

Peripheral Expansion Bus to allow external devices to generate interrupts to the Z-80B. A power-on reset output (active low) is also provided to reset external peripheral devices. Finally, a decoded active-low chip-select line responding to I/O addresses between C0 and FF(hex) is created on-board and can be used to select a single external peripheral device, or to qualify the decoding of some of the address lines for several external chip-select lines.

The Power-on-reset ($\overline{POR}$) output on the Peripheral Expansion Bus is an active-low buffered reset line which should be used in resetting external peripheral devices. Note that 8080-type devices require an active-high reset signal, so the POR line would have to be inverted to service these devices.

To ease timing requirements for interfacing external peripheral devices to the six megahertz CPU, four wait states (T-states) are added to all Z-80 I/O machine cycles. This is over and above the standard extension to mod-4 cycles for any memory cycle. Thus an I/O instruction which might require, for example, 10 cycles would be first extended to 12 cycles, and then further extend to 16 cycles. This allows "A"-version (1.5MHz) peripheral devices to be used. Because of the exact rule used for wait-state insertion, Table XII should be consulted for precise instruction timing information.

If several external peripheral devices are to be connected to the QUARK's Peripheral Expansion Bus, or if the Bus is to be extended any significant distance, it is recommended that the address, data, and control lines being used be buffered by TTL drivers. Two TTL packages are sufficient for this purpose if only 8080- or only 6500/6800-type peripheral chips are used; if both types are employed then an additional buffer is necessary, unless five or fewer address lines are required. An application note on use if this bus is available.

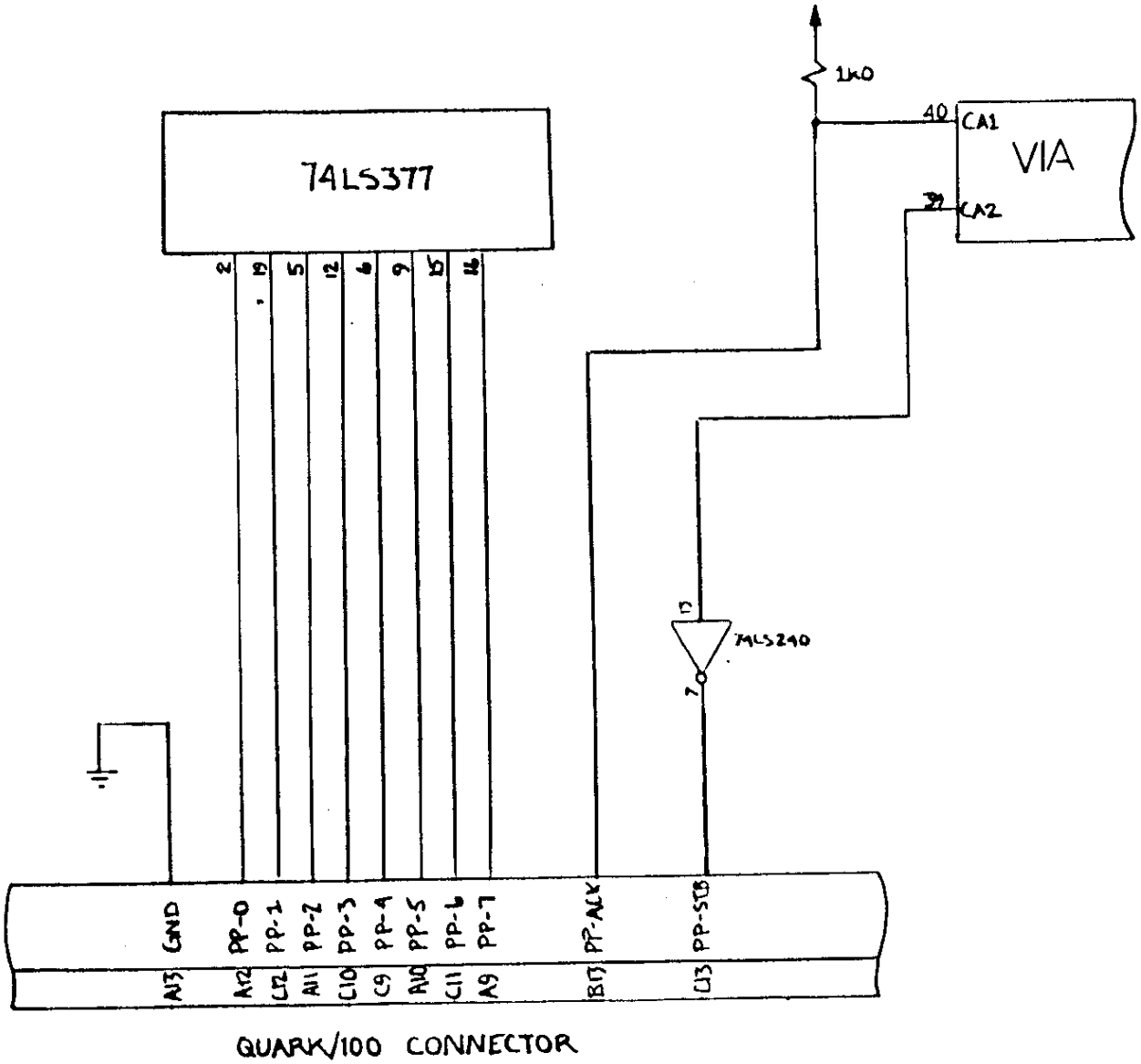


Figure 3.1 Parallel Printer Interface

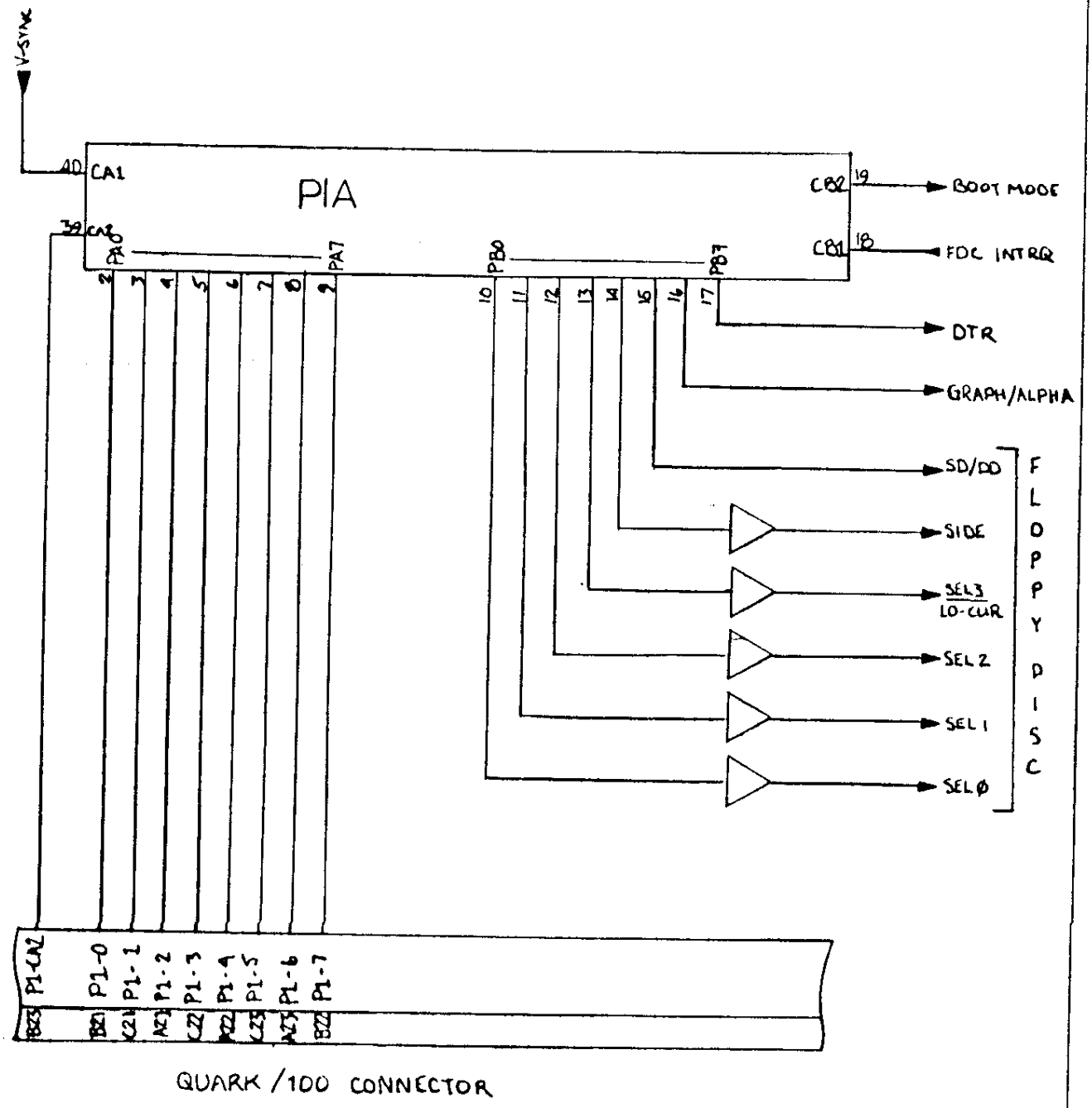


Figure 3.2 Port 1 (Encoded Keyboard Interface), PIA internal control lines, and Floppy Disc control lines

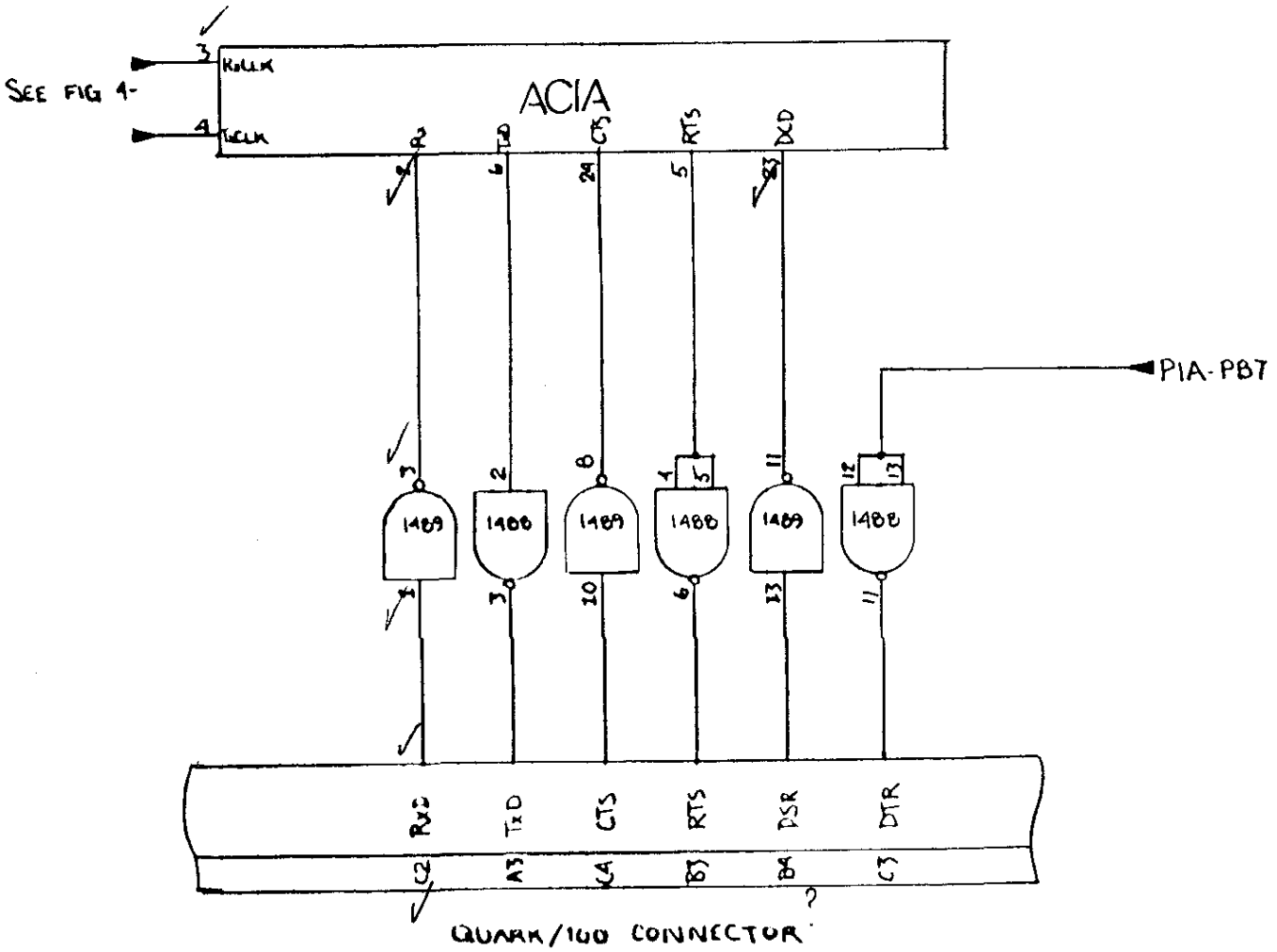


Figure 3.3 Full-duplex Serial Interface

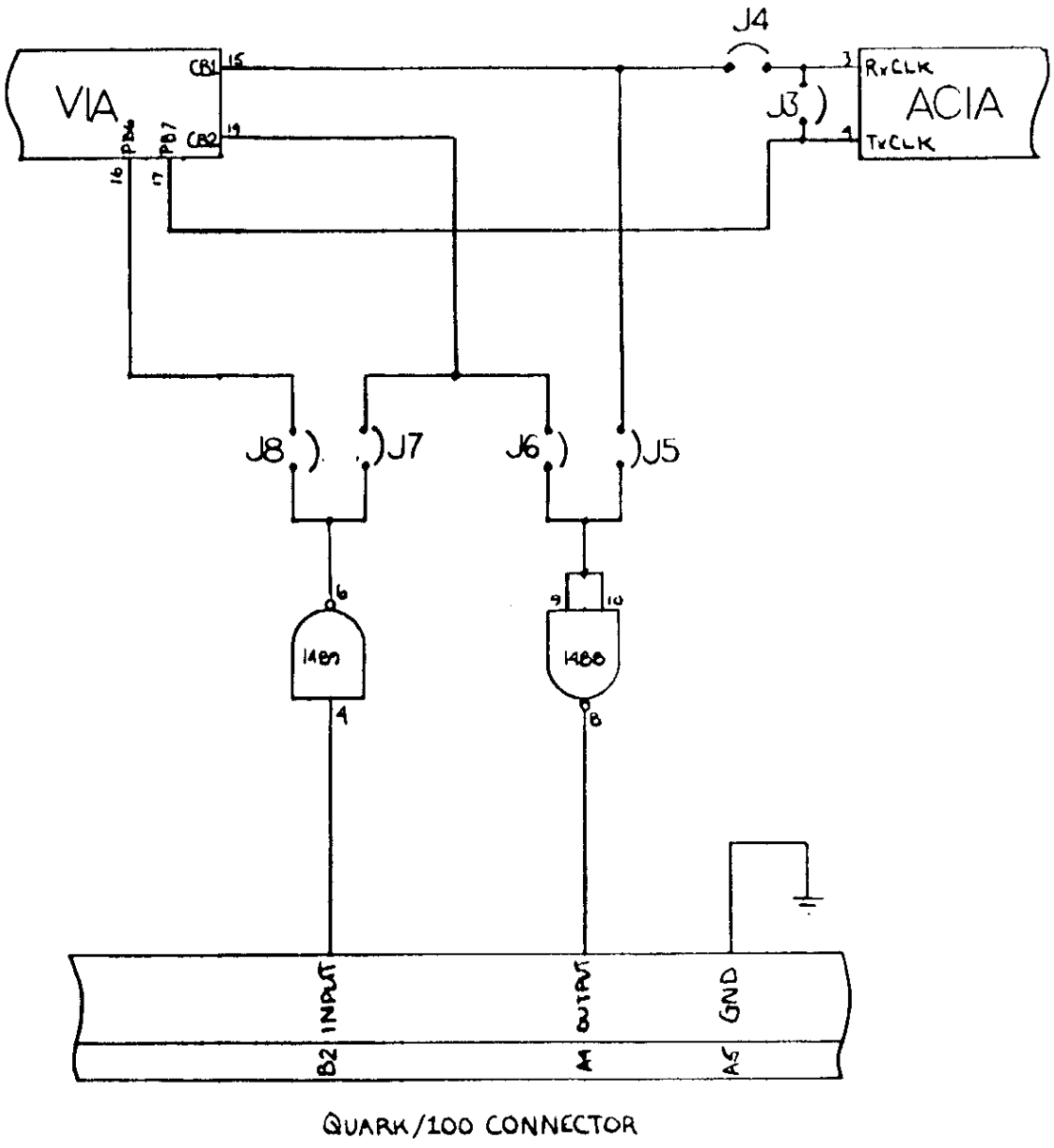


Figure 3.4 Simplex Serial Interface and jumpers for Serial Interface clocks
Standard configuration is J3, J6 and J8 installed

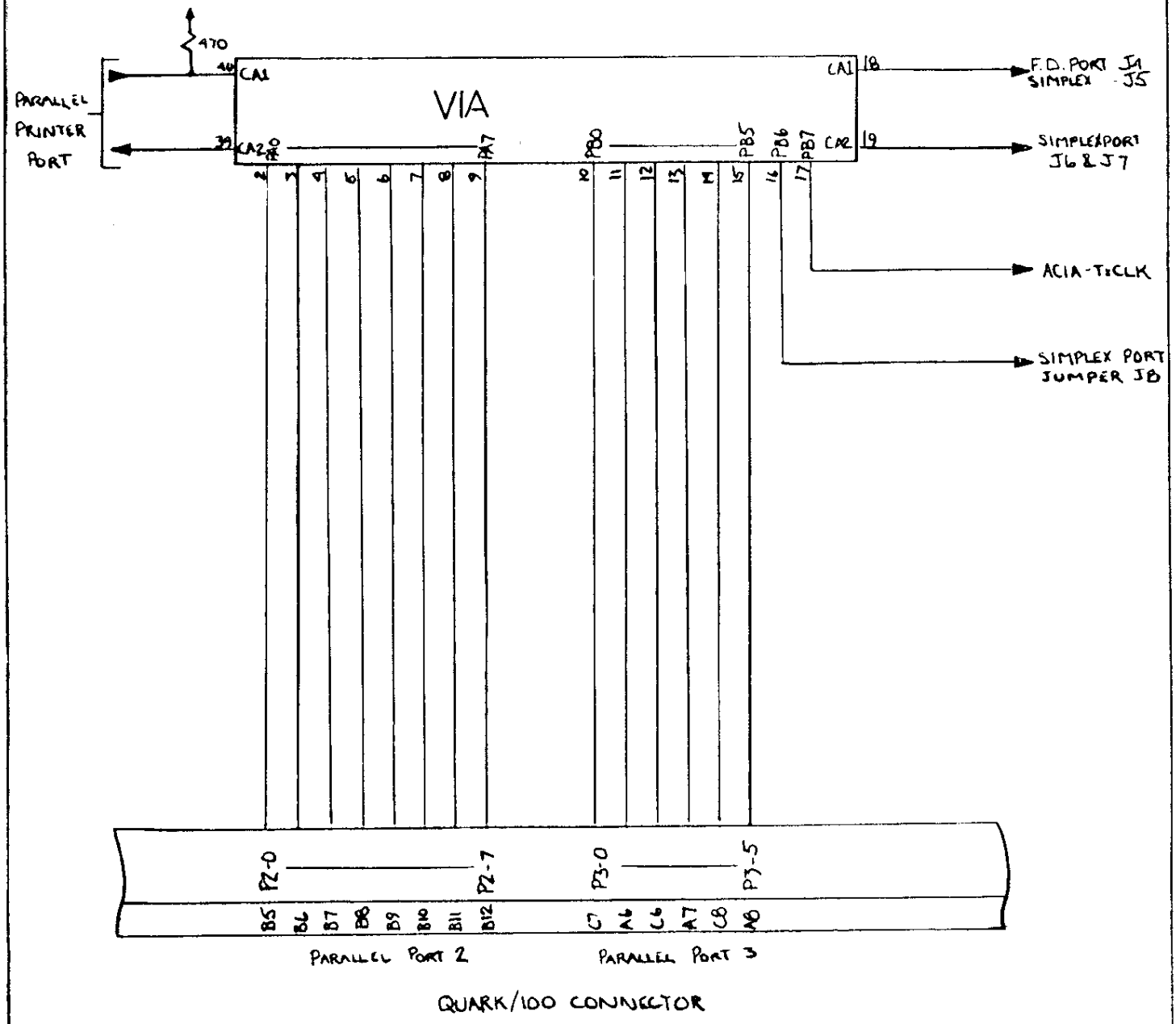


Figure 3.5 Parallel I/O Ports 2 and 3, and VIA control lines

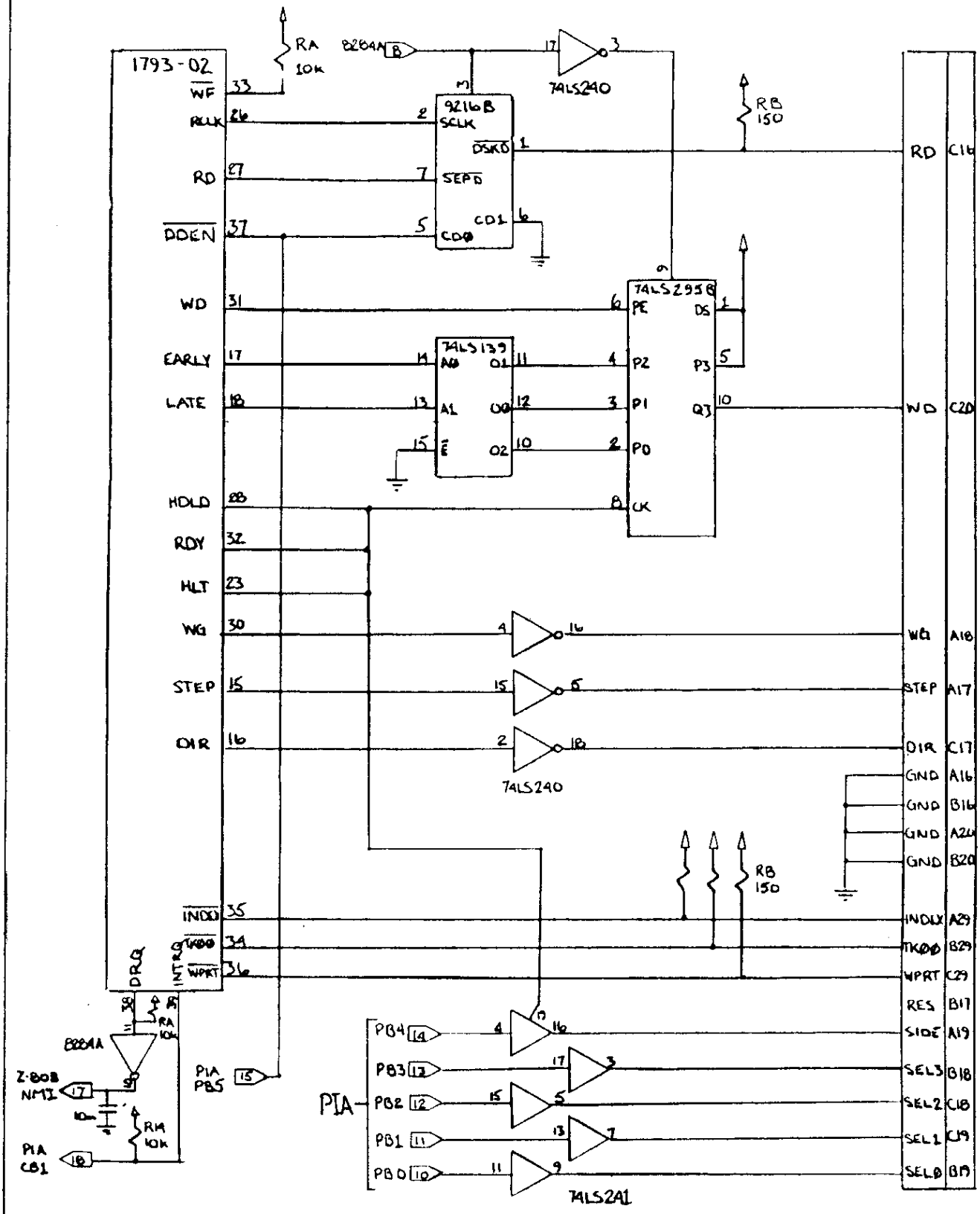


Figure 3.6 Floppy Disc Interface

4.0 Miscellaneous hardware notes for the QUARK

1. The QUARK component computers use some high-speed bipolar integrated circuits in order to achieve their high-performance specifications. Some of these parts become very warm in the course of operation. This is perfectly normal, and is no cause for alarm. However, adequate ventilation for the board should be provided, as it should for any piece of electronic equipment. The QUARK boards are tested at the factory for proper operation at an ambient temperature of over 50 degrees Celsius, a temperature that exceeds the specifications of standard floppy-disc drives. The total power dissipation of any of the QUARK boards is in the range of 10 to 14 watts.

2. Further technical information regarding the expansion of the QUARK hardware and utilization of the Peripheral Expansion Bus is available in the form of "Quark Application Notes - Q-Tips". A list of existing Q-Tips will be included in the sleeve of the binder for the QUARK manual and is available on request from the factory or our distributors.

3. The red light-emitting diode (LED) on any of the QUARKs is connected to the Z-80B's HLT output. This LED will be on when the Z-80 is running. Executing a HLT instruction, for instance, will extinguish the LED.

The green LED on any of the QUARK/200s indicates activity on the Local-Area Network Interface. This LED should normally be off, and will flash rapidly when network operations occur.

4. If the Local-Area Network Interface of the QUARK/200 is not connected, a 150-ohm resistor should be installed accross the "+" and "-" network connections on the QUARK/200. This prevents noise from causing false signals interfering with the Interface.

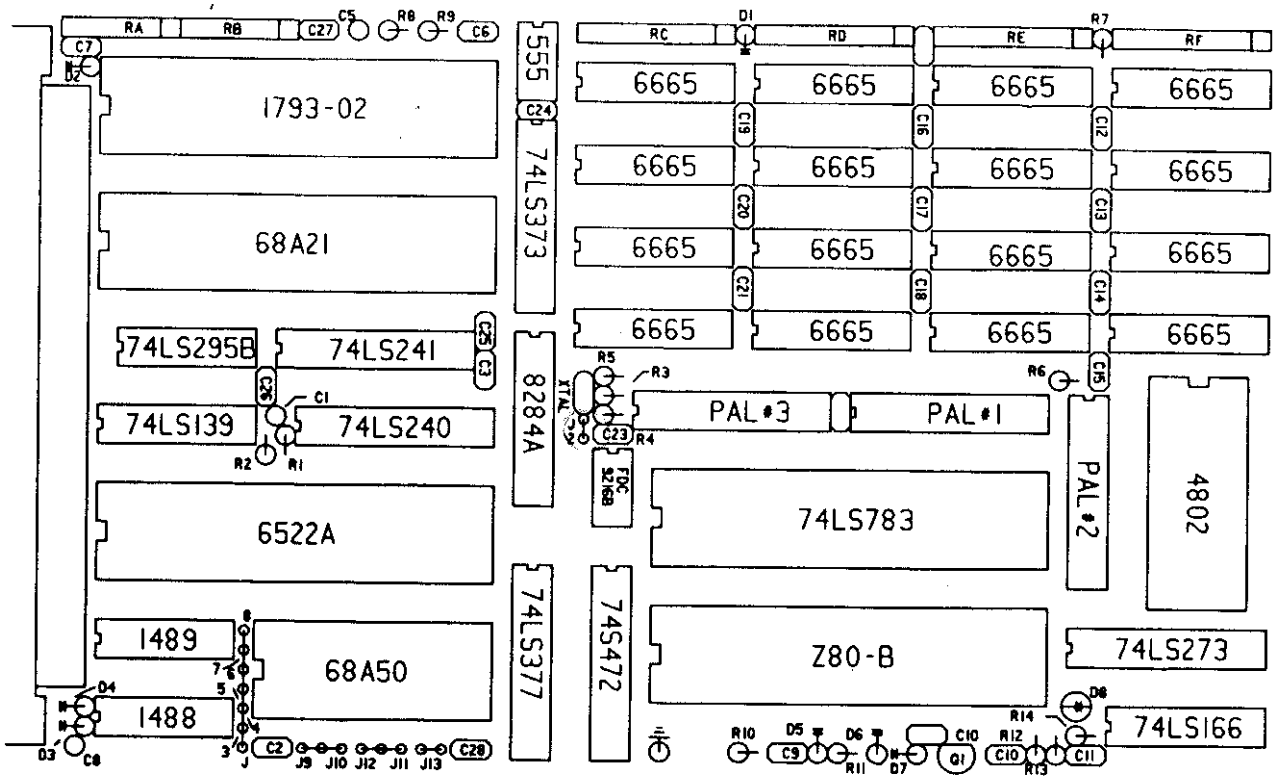
5. If a QUARK computer is unsucessful in locating track 0, sector 1 of the system disc drive when cold-booting, it will, after 10 tries, display a diagonal "staircase" pattern on the Video Display interface, and then halt (red LED off). No other indication of the error will be given.

6. The Parallel Printer STROBE output (from the CA2 line on the 6522A to pin C-13 on the connnector) is buffered by a medium-current TTL driver. On all QUARK/100s bearing revision 05R00 or greater, this driver is a non-inverting buffer, one of the eight such buffers in the 74LS241. On all earlier QUARK/100s, as well as all models of QUARK/150 and QUARK/200 in production at the time of writing, this driver is an inverting buffer, part of the 74LS240.

As a consequence of this change, the sense of the Parallel Printer STROBE output will be inverted with respect to its former sense. This means, among other things, that a software driver intended for use with the inverter-driven STROBE line will not,

strictly speaking, be compatible with the non-inverting buffer. However, this may not be as much of a problen as it seems.

Most printers latch the input data on the rising edge of the STROBE line. As long as the data is valid on its data inputs for at least the minimum set-up period before the edge, the data will be accepted by the printer. Since the parallel printer driver in the BIOS of the distributed CP/M 2.21 or 2.22 systems generates a short low-to-high-to-low pulse (as seen on the CA2 pin) after it sends the parallel data to the output port, the printer will see either a high-to-low-to-high pulse (if an inverter drives the STROBE line) or a low-to-high-to-low pulse (if a non-inverting buffer drives the STROBE line). If the inverter is driving the line, then the printer will latch the data on the second transition of the pulse. If the non-inverting buffer drives the line, it will latch the data on the first transition. Since the parallel data is valid on the output port well before either transition occurs, and because the both transitions of the pulse occur while the data is still valid, the printer will have no difficuly acquiring the data, regardless of the polarity of the signal.

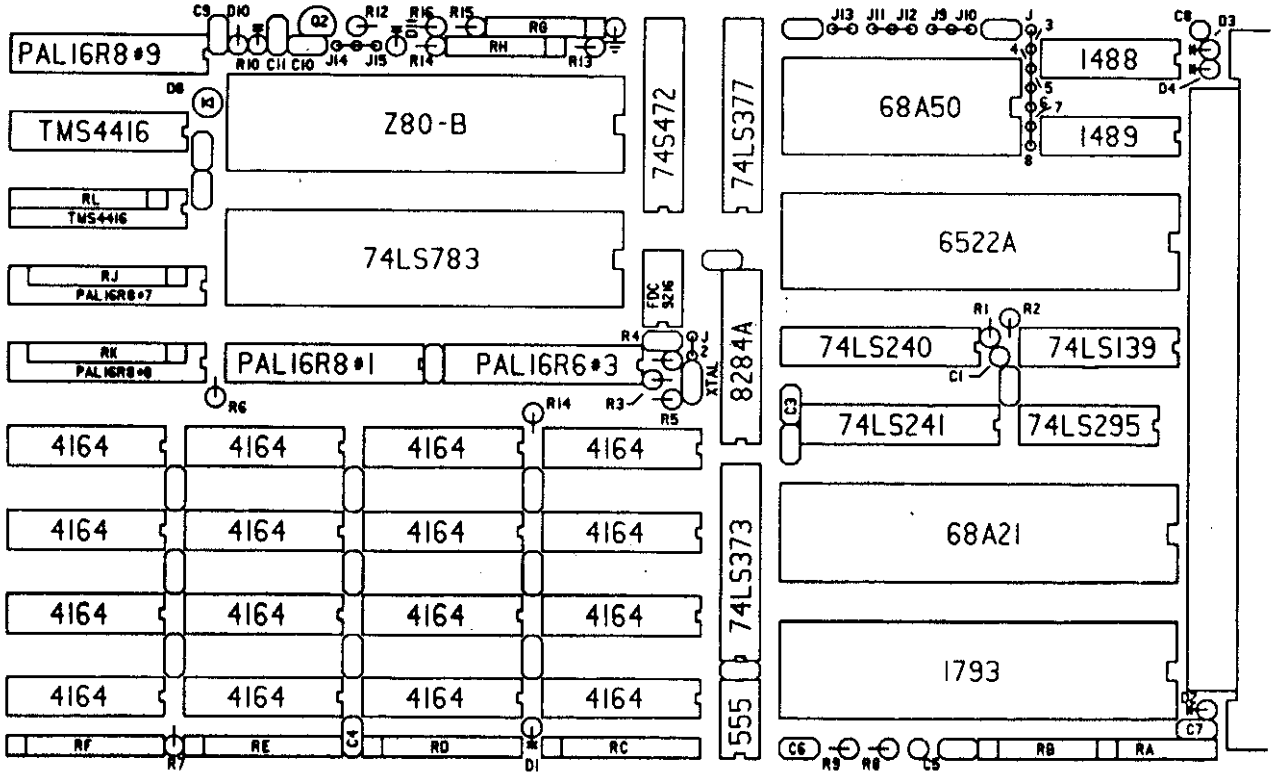


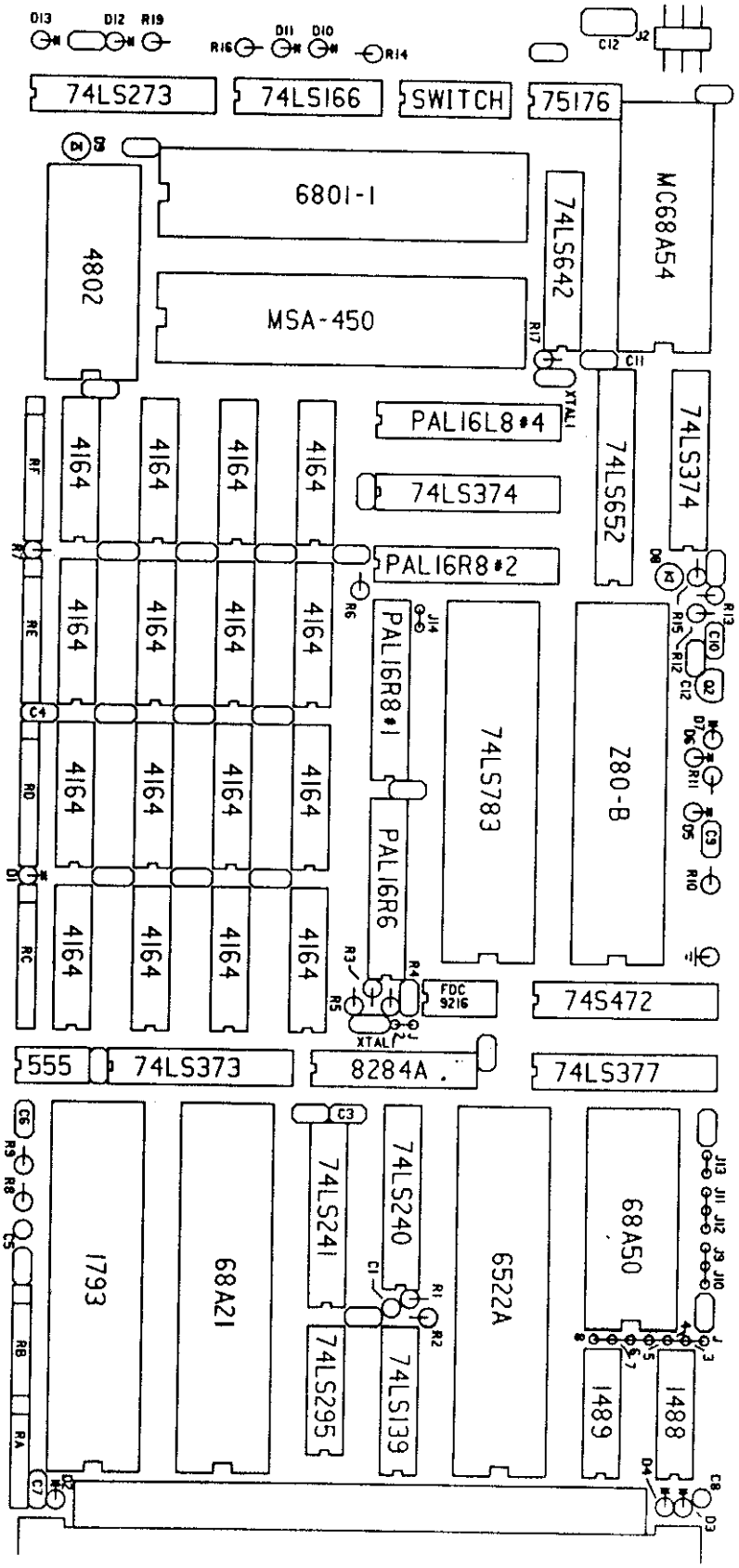
QUARK COMPONENT AND JUMPER LAYOUT

a) QUARK/100 (upper)

b) QUARK/150 (lower)

Component part numbers shown are typical.





QUARK/200 COMPONENT AND JUMPER LAYOUT

5.0 The Megatel QUARK CP/M Plus Software Package

Three classes of utility routines and other files are included in the CP/M Plus (CP/M 3.0) software package for the QUARK. In section 5.1, those commands and routines which are part of the CP/M Plus operating system package are listed. Section 5.2 describes briefly the utilities written by Megatel specifically for the QUARK. Section 5.3 describes CP/M Users' Group files which are included with the QUARK software package.

5.1 CP/M PLUS Commands, Utilities, and Files

The following utilities and commands are part of the CP/M 3.0 operating system from Digital Research and are included on the QUARK CP/M Plus Distribution Diskettes. A full discussion of the use of these utilities can be found in the CP/M Plus Manual.

DIR
DIRS

The DIR and DIRS commands display the names of files and the characteristics associated with the files. DIR and DIRS are built-in utilities. The DIR command with Options is a transient utility.

ERASE

The ERASE command removes one or more files from the directory of a disc, allowing the file's directory and data space to be reclaimed for later use by another file. The ERASE command can be abbreviated to ERA.

TYPE

The TYPE command displays the contents of an ASCII character file on the CONSOLE device.

USER

The USER command sets the current user number. The disc directory can be divided into distinct groups with "User Numbers" 0 through 15.

RENAME

RENAME changes the name of one or more files in the directory of a disc. The RENAME command, which can be abbreviated REN, accepts command line arguments or will prompt for console input.

SAVE

SAVE copies the contents of memory to a file.

DATE

The DATE command lets you display and set the date and time of day.

DEVICE

DEVICE displays and assigns current logical device assignments and physical device names. DEVICE also sets communications protocol and speed of a peripheral device, and displays or sets the current console screen size. To ensure that DEVICE displays the most recent assignments it should be run twice.

DUMP

DUMP displays the contents of a file in hexadecimal and ASCII format.

ED

ED is a line-oriented context editor. It can be used to create and change character files line-by-line, or by referencing individual characters within a line.

GENCOM

The GENCOM command creates a special COM file with attached RSX files. It can also restore a previously GENCOMed file to the original COM file without the header and RSX's or attach header records to COM files.

GET

GET directs the system to take console input from a file for the duration of the next system command or user program entered at the console.

HELP

HELP displays a list of topics and provides summarized information for CP/M 3.

HEXCOM

The HEXCOM command generates a command file (filetype .COM) from a HEX input file.

INITDIR

The INITDIR Command initializes a disc directory to allow date and time stamping of files on that disc. INITDIR can also recover time/date directory space.

LIB

A library is a file that contains a collection of object modules. The LIB utility can be used to create libraries, to append, replace, select or delete modules from an existing library, and to obtain information about the contents of library files.

LIB creates and maintains library files that contain object modules in Microsoft REL file format. These modules are produced by Digital Research's relocatable macro-assembler program, RMAC, or any other language translator that produces modules in Microsoft REL file format.

RMAC

RMAC, a relocatable macro assembler, assembles ASM files into REL files that can be linked to create COM files. See also MAC

XREF

XREF generates a cross-reference summary of variable usage in a program from the PRN and SYM files produced by MAC or RMAC. The file generated by XREF is of type XRF.

LINK

LINK is used to combine relocatable object modules (such as those produced by RMAC and PL/I-80) into a COM file ready for execution. Relocatable files can contain external references and publics, and can reference modules in library files. LINK will search library files and include any referenced modules in its output file. See the CP/M 3 Programmer's Utilities Guide for a complete description of LINK-80.

MAC

MAC, the CP/M 3 macro-assembler, reads assembly language statements from a file of type ASM, assembles the statements, and produces output files with filetypes HEX, PRN, and SYM. The HEX file contains INTEL hexadecimal format object code. The PRN contains an annotated source listing which can be examined at the console or printed. The SYM file contains a sorted list of symbols defined in the program.

PATCH

The PATCH command displays or installs patch number n to the CP/M 3 system or command files. The patch number n must be between 1 and 32 inclusive.

PIP

The file copy program PIP copies files, combines files, or transfers files between users, discs, printers, consoles, or other devices attached to the computer. If no command tail is given, PIP displays a "*" prompt and waits for commands to be entered line by line.

PUT

PUT can be used to direct console or printer output to a file for either the duration of the next command or until a PUT CONSOLE or PUT PRINTER command is entered.

SET

SET initiates password protection and time stamping of files. It also sets the file and drive attributes Read-Write, Read-Only, DIR and SYS. SET can be used to label a disc and to password-protect the label. In order to format the disc directory, INITDIR must be run before attempting to enable time-stamping.

SETDEF

SETDEF allows the user to display or define up to four drives for the program search order, the drive for temporary files, and the file type search order. The SETDEF definitions affect only the loading of programs and/or execution of SUBMIT (SUB) files. SETDEF turns on/off the system Display and Console Page modes. When on, the system displays the location and name of programs loaded or SUBmit files executed, and stops after displaying one full console screen of information.

SHOW

The SHOW command displays the following disc drive information:

Access mode and the amount of free disc space
Disc label
Current user number and
Number of files for each user number on the disc
Number of free directory entries for the disc
Drive characteristics

SID6

SID6 is a symbolic debugger for monitoring and testing programs. It uses 8080-style mnemonics and supports real-time breakpoints, fully monitored execution, symbolic disassembly and assembly, and memory display and fill functions. It can dynamically load SID6 utility programs to provide traceback and histogram facilities. SID6 has been modified by Megatel to use RST6 instead of RST7 as the breakpoint vector.

SUBMIT

The SUBMIT command lets you execute a group (batch) of commands from a SUBmit file (a file with filetype of SUB).

BDOS3.SPR

Used by the GENCPM utility to create the CPM3.SYS file for a non-banked system.

BIOSKRNL.SPR

Used by the GENCPM utility to create the CPM3.SYS file for a banked system.

CCP.COM

The CP/M Plus Console Command Processor.

CPM3.SYS

Created by the GENCPM utility. Contains BDOS and BIOS system components.

5.2 Megatel Utilities and Files

The following utilities are written by Megatel and are included on the QUARK Distribution Diskettes.

QSYS.DAT

File containing default paramaters for standard distribution diskette, which is updated with user-supplied parameters after running the QINSTALL procedure.

QINSTALL.COM (Ver 3.01)

CP/M customization and installation utility.

QINSTALL.SUB
CONTINUE.SUB
QASETUP.SUB
QGSETUP.SUB

CP/M submit files containing the commands and files to be executed during the operating system customization and installation procedure. See section 4 in the Installation Manual for details on these procedures.

BIOSKRNL.ASM

Includes jump tables, basic operating system initialization, and dispatches the character and disc I/O.

BOOT.ASM

Performs system initialization other than character and disk I/O. Loads the CCP for cold starts and reloads it for warm starts.

CHARIO.ASM

Performs all character device (except CRT drivers) initialization, input, output, and status polling.

CHR.DAT

Standard monochrome or colour graphics-mode character set supplied on the Distribution Diskettes.

CHRED.COM (Ver 3.01)

Character set editor. Can be used to modify existing character sets or create new ones. Used for monochrome alphanumeric mode, and for monochrome or colour graphics modes.

CHRLD.COM (Ver 3.01)

Monochrome character set loader (QUARK/100, /200, etc.), or default colour character set loader (QUARK/150 only). Can be used to load its imbedded default character set or a custom character set files on disc. This file is always called when the system is booted.

CHRLDR.ASM

Source code for monochrome character generator loader.

CLR.DAT

Megatel-supplied sample alphanumeric-mode colour character set, included only for the QUARK/150.

CLRED.COM (Ver 3.01)

Colour character set editor, included only for the QUARK/150.

CLRLD.COM (Ver 3.01)

Colour character set loader, included only for the QUARK/150.

CLRSET.COM (Ver 3.01)

Colour symbol table selector and refresh byte loader for the Alphanumeric Mode on the QUARK/150.

COPYSYS.COM (Ver 3.01)

COPYSYS copies the CP/M 3 system from one CP/M 3 system diskette to another diskette of the same format. This utility has been modified by Megatel so that it does not copy the CPM3.SYS file.

DRVTBL.ASM

Higher-level logical floppy disc routines. Includes translation for side and density bits, and error reporting/retries.

FDISK.ASM

Low-level floppy disc routines.

INTROUT.ASM

Interrupt routine, includes internal and screen clock updating.

LDRBIOS.ASM

Main code for loader BIOS.

LDRTAB.ASM

Some of the variables used by LDRBIOS.ASM

MOVE.ASM

Performs memory-to-memory moves and bank selects.

Q3BOOT.ASM

Boot sector routine loaded by Bootstrap PROM. The object code for this routine is located on Track 0, Sector 1. The Bootstrap PROM loads this sector at location 0080hex and then jumps to location 0080hex.

QCOPY.COM (Ver 3.01)

Diskette-to-diskette copy utility. Multiple track buffering permits rapid copying of entire diskettes with the same storage capacity and format.

QDSKTWO.COM

Program to patch the CP/M system in memory to enable the use of a second physical drive in the operating system before installation. The CP/M system as distributed allows for a one-physical-drive set up.

QGRPHDRV.ASM

Graphics-mode alphanumeric terminal driver for CP/M 3.0.

QTCONFIG.COM (Ver 3.01)

Utility for configuring the terminal control codes used by the QUARK operating system's video driver. The video driver may be configured to emulate the control codes used on the Televideo 920 terminal, the standard Megatel control codes, or the user's own codes.

QTERMDRV.ASM

Alphanumeric-mode alphanumeric terminal driver for CP/M 3.0.

QTERMDS.ASM

Variables for terminal drivers.

SCB.ASM

Contains the public definitions of the various fields in the CP/M 3 system control block, so that the BIOS can reference the public variables.

SETACIA.COM (Ver 3.01)

Utility for setting the word length, parity, and number of stop bits for communication via the full duplex serial channel.

SYM.DAT

Megatel-supplied sample symbol table for a colour graphics character set, included only for the QUARK/150. For use with SYMED.COM and SYMLD.COM.

SYMED.COM (Ver 3.01)

Colour graphics symbol table set editor, included only for the QUARK/150.

SYMLD.COM (Ver 3.01)

Colour graphics symbole table loader, included only for the QUARK/150.

VARS.ASM

Contains most of the variable declarations defined in the BIOS. All variables patched by QINSTALL.COM are here as well as variables accessed by the USERF jump table entry.

COPYSYS.ASM

Source for COPYSYS.COM utility.

5.3 CP/M Users' Group Utilities

The following utilities are CPMUG (CP/M Users' Group) utilities, and have been modified for operation on the QUARK.

DU.COM

DU is a disc dump utility.

SWEEP.COM

SWEEP is a file maintenance utility.

QMOD790.COM

QMOD790 is a version of MODEM7 configured for use with the QUARK.

6.0 QCERT.COM - The diskette formatting utility

QCERT is a diskette formatting program supplied on the QUARK System Distribution Diskette. QCERT allows new or previously-used diskettes to be formatted. After formatting a diskette, QCERT also verifies that the diskette has been formatted correctly by reading back every sector. All necessary information i.e. Interlace tables for QCERT to format diskettes is contained within the operating system.

6.1 Operation of QCERT

When QCERT is started (by typing **QCERT** in response to a CP/M prompt), it queries the user for the identity of the logical drive which is to be formatted, as shown below.

Which Drive (A,B,...P), then RETURN ?

If the diskette is formatted successfully, the following message will appear when finished.

Hit RETURN to Reboot, any other key to re-start.

To abort QCERT at any time, type the **ESC**ape key. If typing this key does not seem to have any immediate effect, type the key repeatedly until QCERT halts and displays an appropriate message.

6.2 Standard Diskette format

When operating under CP/M Plus, the utility QCERT picks up all the disc format parameters required for formatting a disc from the operating system.

This page intentionally left blank.

(QCERT errors, continued)

```
****ERROR HAS OCCURRED****
Please press the down arrow (Control J) to acknowledge
```

This error message will appear after every error to ensure that the display has been read and acknowledged. QCERT will pause until a line feed (Control-J) is entered.

6.3 QCERT Error Messages

The following list describes all the error messages that may appear during the operation of QCERT.COM.

Speed Error - x MS. per revolution

This error indicates that QCERT has detected that the rotational speed of the diskette is not within the required range. This is probably a hardware fault in the disc drive and should be corrected.

ERROR - Cannot write physical track n

This indicates that QCERT was not successful in writing track number "n". The most common reason for this fault is because the diskette is write-protected. It might also indicate that there is a fault in the floppy-disc interface hardware, in the cable connecting the QUARK board to the floppy disc drive, or in the diskette itself, and this is stopping the program from writing the data for formatting purposes. This error will only occur in the formatting phase.

Physical track x does not validate

This error occurs during the verification phase when the program finds that it cannot read back the data that it has just written during the formatting phase.

Unable to home drive - ABORTING
Unable to restore drive

These two errors are likely to occur together and indicate a hardware fault. QCERT has been unable to return the heads to the "home" position on track 00. This error will occur before any formatting has taken place.

ABORTING FORMATTER - DISK STATUS = n

This message will always appear after every error, to indicate the status of the floppy disc controller at the time the error occurred. The value of "n" is the value in the Status Register of the Floppy Disk Controller (part number 1793-02 refer to the supplied data sheet) when the error was detected.

7.0 COPYSYS.COM - The CP/M PLUS System Copy Utility

COPYSYS is a utility routine for reading and writing the CP/M Plus loader on the reserved tracks of a given logical drive. COPYSYS does not destroy or alter any files on the destination diskette, as only the reserved tracks of the destination diskette are written. It is necessary to "Copysys" only those diskettes which are to be used as "system" diskettes (i.e., those diskettes from which the QUARK is to booted). To use the new disc as a CP/M Plus system disc, the system files CPM3.SYS and CCP.COM must also be present on the new disc. These files are copied using the PIP command. The following example shows the procedure that would be followed when setting up a system diskette on logical drive B: from the diskette on logical drive A:.

```
A>COPYSYS
CP/M 3 COPYSYS - Version 3.0
Source drive name (or return for default) A
Source on A then type return <ret>
Function complete
Destination drive name (or return to reboot) B
Destination on B then type return  ret
Function complete
Destination drive name (or return to reboot) <ret>

A>PIP B:=A:CCP.COM [VO]

A>PIP B:=A:CPM3.SYS [VO]
```

This completes the copying of the System files, CCP files and the loader.

This page intentionally left blank.

8.0 Software for the Video Display Interface

The utility routine CHRLD.COM, supplied on the distribution diskettes, is used to load a character set for any of the various alphanumeric display drivers which are part of the QUARK operating system.

On the monochrome versions of the QUARK (eg.,the QUARK/100 and /200), the program CHRLD.COM is used to load a character set into the Programmable Character Generator. This utility can be used to load either the default character set imbedded within the CHRLD.COM file, or any other character set saved in a file on a diskette. (Sec. 8.1.)

On the colour versions of the QUARK (eg. the QUARK/150), the program CLRLD.COM is used to load a colour character set. (Sec. 8.4.) This utility can be used to load an arbitrary colour character set only as there is no imbedded character set.

The character set used for alphanumeric display is a full 256-character set, including both normal- and reverse-video characters for the monochrome QUARK, or a pair of 128-character colour character sets on the colour QUARK. The character set is intended to be used with ASCII codes. For monochrome character sets, each character is stored as eight contiguous bytes within the data file. For colour character sets, each character is stored as eight sets of four bytes in the data files.

For further information on the operation of the Programmable Character Generator or the Video Display Interface, refer to section T/2 of this manual. Also refer to Section 8.2, for CHRED, a utility for editing the patterns used for monochrome characters, and to Section 8.5 for CLRED, a utility for editing the patterns for colour characters.

8.1 CHRLD.COM - The monochrome Character Generator Loader

This file will load the default character set, the data for which is contained within CHRLD.COM itself. If the user wishes to load the default character set after the system is up, it is necessary only to enter the following command:

```
A>CHRLD
A>
```

To load another character set into the Programmable Character Generator without altering the CHRLD.COM file, specify the character set file name on the command line immediately following the command

CHRLD. CHRLD will find the character set file and load the specified character set file. The following is an example of this use of CHRLD.COM:

```
A>CHRLD CHR.DAT
A>
```

The character set "CHR.DAT" set will remain loaded in the Programmable Character Generator until another character set is loaded, or until power is removed. Character set files can be created or modified using the Character Set Editor CHRED.COM. (See Sec 8.2.)

To change the default character set within CHRLD.COM a desired character set file must be patched into CHRLD.COM. To perform this patch, load CHRLD.COM at address Ø1ØØH, and load the file containing the new character set with an offset of 1Ø3H. An example is presented below.

```
A>SID6 CHRLD.COM

#RCHR.DAT,42
#WCHRLD.COM,1ØØ,AØØ
#GØ

A>
```

The file CHR.DAT, the new default character set, would have been created using the Character Set Editor CHRED.COM.

8.1.1 Changing Character Set Loaded at Cold-boot

In order to change the character set loaded after each cold-boot, the character set imbedded within the CP/M Plus loader must be altered. This can be done by creating the desired character set and then patching this file into the CP/M Plus loader. To perform this patch a file containing the CP/M Plus loader must be created by reading the loader from the reserved tracks then write it to a file. This file is patched with the new character set with an offset of 26ØØH, then written back to the reserved tracks. An example is presented below.

```
A>SAVE

A>COPYSYS
CP/M 3 COPYSYS - Version 3.0
Source drive name (or return for default) A
Source on A then type return <ret>
Function complete
Destination drive name (or return to reboot) <ret>
```

(changing the default character set, continued)

```
CP/M 3 SAVE - Version 3.0
Enter file (type RETURN to exit): CPM.LDR
Beginning hex address 1ØØØ
Ending hex address    3EØØ

A>SID6 CPM.LDR
CP/M SID - Version 3.Ø
NEXT MSZE  PC  END
2FØØ 2FØØ Ø1ØØ CDFF
#RCHR.DAT,26ØØ
NEXT MSZE  PC  END
2FØØ 2FØØ Ø1ØØ CDFF
#WCPM.LDR
ØØ5Ch record(s) written
#GØ

A>COPYSYS CPM.LDR
CP/M 3 COPYSYS - Version 3.Ø
Destination drive name (or return to reboot) A
Destination on A then type return  ret
Function complete
Destination drive name (or return to reboot)  ret
```

The file CHR.DAT, the new default character set, would have been created using the Character Set Editor CHRED.COM.

8.2 CHRED.COM - The Monochrome Character Set Editor

The Monochrome Character Set Editor utility CHRED.COM is included on the Distribution Diskettes. It can be used to modify existing character sets, or to create new ones. The Editor displays the eight-by-eight matrix of dots forming the character pattern in an eight-by-eight edit frame on the CRT. With a given set of control keys, the user may move a "dot cursor" around within the grid, stopping on any dot and turning the dot on or off. The 8-bit ASCII code which is to represent the character may be entered, and the set of characters may be scanned in ascending or descending order of codes.

The calling syntax of the Character Set Editor is as follows:

```
A>CHRED dr:filename.ext
```

where "dr:" is the optional logical drive on which the colour character set file can be found, and "filename.ext" is the name of the character set file to be edited.

After CHRLD has been loaded, it will prompt the user for the hexidecimal ASCII code for the character to be edited. When a code is entered, the pattern for that character in the character set file will be displayed in a the large 8-by-8 edit frame, as well as in a small single-character cell below the grid. Also displayed is a "painting cursor". Control keys E, X, S, and D will move the cursor up, down, left, or right within the edit frame. Positioning the cursor on any cell of the frame and typing a "space" will cause that cell to be inverted, ie., the dot will be switched on or off. Also, the corresponding pixel in the single-character cell below the edit frame will change to show the actual appearance of the character.

Typing control-C or control-R will move the edit frame onto the next or previous ASCII character. Typing a carriage return allows a new code for the character to be edited to be entered. If "ØØ" is entered as the new code, CHRLD will ask if the edited character set is to be saved. If the response to this prompt is "Y", then it will save the edited file under the filename given when CHRED was called.

A file CHR.DAT is included on the Distribution Diskettes. This file contains a character set identical to the set imbedded within the distributed copy of CHRLD.COM, and can be used as a starting point when creating new character sets.

8.3 CHRLD.COM - Its use with the Colour Video Display Interface

The utility CHRLD is used with the Colour Video Display Interface of the Quark/150 to load a character set when running in graphics mode. For more information on the operation of CHRLD refer to section 8.1.

8.3.1 CHRED.COM - Its use with the Colour Video Display Interface

The utility CHRED is used with the Colour Video Display Interface of the Quark/150 to edit a character set when running in graphics mode. For more information on the operation of CHRED refer to section 8.2.

8.4 CLRLD.COM - The Colour Character Set Loader

The program CLRLD.COM is used to load a colour character set into the Colour Translation Table of the Colour Video Display Interface of the QUARK/150. Its use is analogous to that of CHRLD.COM for the monochrome Video Display Interface, although it does not contain an imbedded character set.

CLRLD.COM will load an arbitrary colour character set file from the disc. The syntax for its use is as shown in one of the following examples:

```
A>CLRLD
A>CLRLD d:filename.ext
A>CLRLD d:filename.ext s
```

where "d:" is the optional drive specifier, "filename.ext" is the name of a colour character set, and "s" is the optional value Ø, 1, or 2, which determines into which of the two sets of symbol tables the character set is loaded.

If no command line arguments are given, CLRLD will prompt the user for the name of a file containing the colour character set to be loaded, and the code for the set of symbol tables to be loaded. Entering Ø or 1 for this prompt will load the character set file into symbol tables Ø-7 or 8-15, respectively, while entering 2 will load the file into the set of tables currently being used for display. If a character set file is specified on the command line without identifying the set of tables to be loaded, the user will be prompted to give the code for the set of tables to be used (again, Ø, 1, or 2). If a 0, 1, or 2 character follows the filename on the command line, tables 0-7, 8-15, or the current set will be loaded.

8.5 CLRED.COM - The Colour Character Set Editor

For the Colour Video Display Interface of the QUARK/150, a special character set editor utility is provided. The file CLRED.COM can be found on the Distribution Diskettes included with the QUARK/150. Similar in operation to CHRED.COM (the monochrome character set editor-see section 8.2), it allows the user to create or modify colour character files, which can be later loaded by CLRLD.COM (see sec. 8.4).

The calling syntax for CLRED.COM is as follows:

```
A>CLRED dr:filename.ext
```

where "dr:" is the logical drive on which the colour character set

file can be found, and "filename.ext" is the name of the character set file to be edited.

After CLRED has been loaded, it will display the 16-colour palette at the bottom of the screen, and prompt the user for the hexidecimal ASCII code for the character to be edited. When a code is entered, the pattern for that character in the character set file will be displayed in a large 8-by-8 edit frame, as well as in a small single-character cell below the grid. Also displayed is a "painting cursor". Typing a control-P will cause CLRED to prompt the user for the desired "paint" colour. Control keys E, X, S, and D will move the cursor up, down, left, or right within the edit frame. Positioning the cursor on any cell of the frame and typing a "space" will cause that cell to be coloured with the previously-selected paint colour. Also, the corresponding pixel in the single-character cell below the edit frame will change to show the actual appearance of the character.

Typing control-B will prompt the user to select a colour with which to fill the entire 8-by-8 character matrix. Typing control-C or control-R will move the edit frame onto the next or previous ASCII character. Typing a carriage return allows a new code for the character to be edited to be entered. If "00" is entered as the new code, CLRLD will ask if the edited character set is to be saved. If the response to this prompt is "Y", then it will save the edited character set under the filename given when CLRED was called.

A file CLR.DAT is included on the Distribution Diskettes. This file contains a character set identical to the set imbedded within the distributed copy of CHRLD.COM for the QUARK/150, and can be used as a starting point when creating new character sets. This character set, and the character set imbedded within CHRLD, also contains "colour swatches" - full 8-by-8 characters in each of the sixteen colours - which are displayed for ASCII codes 01 to 10hex. These codes are used, in particular, by CLRED to display the colour palette. Code 11hex is used as the single display character below the edit frame, and code 00 is used as the default blank character. It is recomended that these colour characters not be altered if confusion of the colours available in CLRED is to be avoided. To this end, CLRED ignores requests to edit the character codes 0 through 11(hex).

Code 12hex is the character used by CLRED as the "paint brush" in the edit frame. CLRED allows this character to be edited, so the user may wish to create a small paint brush symbol for this code.

8.6 SYMED.COM and SYMLD.COM - The Colour Symbol utilities

The program SYMED.COM is used to create a symbol table file for use with the colour graphics mode of operation of the QUARK/150. The file so created can be loaded into one of the sixteen symbol tables by SYMLD.COM.

The calling syntax for SYMED.COM is as follows:

```
A>SYMED dr:filename.ext
```

where "dr:" is the logical drive on which the colour symbol file will be written, and "filename.ext" is the name of the colour symbol character set file to be edited. If the specified file already exists on the named drive, it will be overwritten at the end of the edit session. If no file is specified on the command line, SYMED will return an error message.

SYMED gives the user a choice of the three "standard" resolution modes: 640, 320, and 160 pixels. If the "640 mode" is selected, the user will be asked to specify one of the 16 colours for use as the foreground colour, and colour for use as the background colour.

If the "320 mode" is selected, then the user will be asked to specify four pairs of colour values. Each pair of values entered will be treated as one pixel when SYMED makes up the symbol table file.

If the "160 mode" is selected, then the user can select either the "standard 160-mode pattern", or a completely user-specified pattern. The "standard 160-mode pattern" includes all possible combinations of the 16 colours. If this option is not selected, then the user is allowed to specify sixteen sets of four colours. In an analogous fashion to the "320 mode", SYMED will treat each set of four values as one pixel when making the symbol table file.

The colour symbol file created by SYMED can be loaded into one of the symbol tables on the QUARK/150 by SYMLD.COM. The calling syntax for SYMLD is as follows:

```
A>SYMLD dr:filename.ext n
```

where dr: is the logical drive on which the symbol table file filename.ext is found, and n is a number from 0 to 16 which specifies the symbol table to be loaded. If 16 is given as the symbol table number, the "default" symbol table will be loaded. The default table is determined by the value in the least-significant four bits of location 500Fhex in Bank B.

The program SYMLD.COM can be used when operating in either the Alphanumeric or Graphics modes on the QUARK/150.

8.7 CLRSET.COM - The Colour Table Selector

The program CLRSET.COM allows the symbol table (when in Graphics mode) or the set of symbol tables (when in Alphanumeric mode) to be selected. The calling syntax is as follows:

```
A>CLRSET n
```

where n is a number which determines the symbol table(s) to be used.

If operating in Alphanumeric mode, n should be 0 or 1. Entering 0 will cause symbol tables 0 to 7 to be used as the colour character generator, while entering a 1 will cause tables 8 to 15 to be used.

When operating in Graphics mode, n can be any value from 0 to 15. The value entered will determine the symbol table used for the entire screen area.

If no value for n is entered on the command line, CLRSET will prompt the user for the value.

In addition to loading the Table Control byte, CLRSET will set up the Translation Table refresh bytes in the video display memory.

8.8 GRFLD.COM - The graphics Mode Alphanumeric Video Utility

The program GRFLD is no longer required to use the graphics terminal driver for Version 3.01 of the CP/M Plus Operating System. The user can write his own utility to load a terminal emulation program which will display alphanumeric characters while operating under graphics mode of the Video Display Interface.

8.9 QTCONFIG.COM - Terminal Code configuration utility

The purpose of this utility is to allow the user to change the terminal control codes used by the terminal driver. QTCONFIG allows the standard Megatel control codes, the user's own set of codes, or the set of control codes used on a Televideo 920 terminal. To run this utility the user should enter:

```
A>QTCONFIG
```

The screen will now display:

```
Terminal Emulation Utility Vers. 3.01

    1.  Televideo 920
    2.  QUARK
    3.  User Defined

    ENTER -
```

Option 1 will configure the QUARK to emulate a Televideo 920. Option 2 will configure the QUARK with the Megatel control codes used on the Distribution Diskettes. These control codes are given in Table XI of the Appendix.

Option 3 allows the user to configure his own terminal codes or to load these codes from another file. This option will prompt the user with queries and then save the responses in a user-specified file.

When setting up the user-specified control codes, QTCONFIG will display a description of the terminal function and then allow the user to enter a two-byte sequence representing the control code to be used for that terminal function. If it is desired that a particular function have only a one-byte control code, the desired hexidecimal value for the code should be entered as the first value, and FFhex be entered as the second. Only the first byte entered will then be the control code; the value FF will not become part of the code. If a particular terminal function is not to be implemented, then the value FFhex should be entered for both parts of the code.

Note that the number of character rows displayed by the video driver will not be changed when QTCONFIG is run. Since the standard number of rows displayed on a Televideo 920 terminal is 24, application programs designed to run on a Televideo 920 should be patched to allow for number of rows installed for the "Normal" screen area of the QUARK.

With the QUARK terminal driver the control codes for high intensity will cause the characters to be displayed in reverse video.

This page intentionally left blank.

9.0 QINSTALL.COM - The Installation Program for CP/M 3.01

The installation program is written to allow the QUARK user to customize his CP/M operating system. It will prompt the user for parameters necessary to bring up an efficient system then insert these parameters into CPM3EQU.LIB. The assembler files can then be assembled to produce loadable codes.

QINSTALL.COM is only the first program called by the submit file used for the installation procedure. It does not assemble any files. For the complete installation procedure see Section I/4.0.

The command syntax for the QINSTALL program is as follows:

```
QINSTALL  dr
```

The "dr" parameter tells the install program to look for the "CPM3EQU.LIB" file on the drive specified, or on the default drive, if no "dr" parameter is sepcified.

Once the user has entered all the parameters through QINSTALL for the customized system and has continued with the installation procedure the default file (QSYS.DAT) will be updated from the distribution diskette default values to the new target system values.

9.0.1 QASETUP.COM and QGSETUP.COM for CP/M 3.01

QASETUP and QGSETUP are submit files used in the second half of the installation procedure. Depending on the type of video display interface chosen during the installation, one of these submits will be used after the temporary system is invoked. If Alphanumerics mode is installed, the QASETUP is used, if graphics mode is installed then QGSETUP is used. The end result of either submit file is:

```
CPM3.NEW - system file for the target system
CPM.LDR  - loader file for the target system
```

To create these files the QSETUP submit will modify the CPM3EQU.LIB library. Using the parameters in CPM3EQU.LIB the QSETUP submit assembles and links all appropriate files to create BNKBIOS3.SPR (banked bios).

At this point, the GENCPM utility is invoked under automatic mode to create the system file CPM3.SYS.

The next step involves renaming the system files to preserve the new system file, while returning the status of the "work disc" to its original form (before the Installation). CPM3.SYS is renamed to

CPM3.NEW and CPM3.OLD is renamed back to CPM3.SYS.

9.1 QSYS.DAT

The system parameter file QSYS.DAT initially contains the default system parameters. These default parameters will be displayed by QINSTALL when the Installation procedure is run for the first time. If the user chooses to accept the displayed default parameter for any of QINSTALL's queries, then this displayed parameter will be the value used in configuring the operating system. If the user enters another value in response to any query, the entered value, rather than the default value, will be used. QINSTALL will modify the contents of QSYS.DAT with the system parameters entered by the user during Installation. QSYS.DAT is an ASCII file and can be edited using a text editor, although because of its complicated structure this practice is not recommended.

When running QINSTALL, the values displayed at the query prompt will be the value stored in the QSYS.DAT file on the disc at the time that QINSTALL was called. The responses to the queries will remain in memory until either function D is chosen (to continue with the INSTALL procedure) or function X is chosen (to abort). If function D is chosen, the INSTALL procedure will update the QSYS.DAT file on the disc with the new parameters. Function X will clear the values in memory and the distribution default values will remain.

QSYS.DAT file layout is as follows:

COLUMN	DESCRIPTION
1	Refers to the question that first appears in the main menu of the INSTALL procedure 0=DISC DRIVE HARDWARE SPECIFICATIONS 1=DISKETTE FORMAT SPECIFICATIONS 2=OTHER PERIPHERALS CONFIGURATION
2-3	Refers to the specific question asked under one of the above question types.
4	Refers to the type of response being entered, the values that can be used are: 0=true or false 1=integer response/integer 0 is not allowed 2=integer response/the character 'X'=0 3=integer corresponding to a value in a table 4=integer response interpreted in milliseconds/0 is not allowed 5=integer response interpreted in milliseconds/the character 'X'=0
5-9	Refers to the default values

10-14	Minimum value allowed
15-19	Maximum value allowed
20-24	Description of units
25-27	BIOS parameter number
28-29	Total number of table entries
30-78	Comments

If the response is a table selection then the line following the query must contain the table values. The table entries can be a maximum of five characters and each value must start at every fifth position.

CONTENTS OF QSYS.DAT

```
00031                                4 NUMBER OF DRIVES
01   02   03   04                      THE 4 ENTRIES
0011127  95   255          050  LOW WRITE-CURRENT PORT ADDRESS
002208   1    255          052  MASK TO BE USED FOR LOW WRITE-CURRENT
003116   1    128          420  MASK TO BE USED FOR SIDE SELECT
00401                      053  ENABLE 8 / 5-1/4 SELECTION
0051127  1    255          054  AUTO SELECT PORT ADDRESS
006216   1    255          056  MASK TO BE USED FOR AUTO SELECT
00702                      114  8 INCH OR 5.25 INCH
00832                      1132 48 OR 96 T.P.I.
96   48                      THE 2 ENTRIES
00901                      111  HARDWARE OR SOFTWARE STEPPING RATES
01034                      1104 HARDWARE STEPPING RATES
00   01   02   03                 THE 4 ENTRIES ARE
011115   3    300  msec 109  SOFTWARE STEPPING RATES
0125250  0.1  250  msec 104  MOTOR START TIME IN MILLISECONDS
0134100  0.5  100  msec 102  WAIT TIME FOR HEAD LOAD WITH MAX & MIN VALUES
0144100  0.1  100  msec 103  HEAD SETTLING TIME WITH MAX & MIN VALUES
0155250  0.1  1000 msec 105  DELAY BETWEEN DRIVE SELECTS WITH MAX & MIN
01643    0.5  10   msec 106  TUNNEL ERASE DELAY WITH MAX & MIN VALUES
017243   0    159  trks 101  TRK THAT LOW WRITE-CURRENT INVOKED.  MAX & MIN
018131   1    255          107  READ/WRITE RETRIES
019110   1    255          108  RETRIES FOR VERIFICATION
02002                      134  8 INCH OR 5.25 INCH
02132                      1332 48 OR 96 T.P.I.
96   48                      THE 2 ENTRIES
02201                      131  HARDWARE OR SOFTWARE STEPPING RATES
02334                      1304 HARDWARE STEPPING RATES
00   01   02   03                 THE 4 ENTRIES ARE
024115   3    300  msec 129  SOFTWARE STEPPING RATES
0255250  0.1  250  msec 124  MOTOR START TIME IN MILLISECONDS
0264100  0.5  100  msec 122  WAIT TIME FOR HEAD LOAD WITH MAX & MIN VALUES
0274100  0.1  100  msec 123  HEAD SETTLING TIME WITH MAX & MIN VALUES
0285250  0.1  1000 msec 125  DELAY BETWEEN DRIVE SELECTS WITH MAX & MIN
02943    0.5  10   msec 126  TUNNEL ERASE DELAY WITH MAX & MIN VALUES
030243   0    159  trks 121  TRK THAT LOW WRITE-CURRENT INVOKED.  MAX & MIN
031131   1    255          127  READ/WRITE RETRIES
032110   1    255          128  RETRIES FOR VERIFICATION
03302                      154  8 INCH OR 5.25 INCH
03432                      1532 48 OR 96 T.P.I.
96   48                      THE 2 ENTRIES
03501                      151  HARDWARE OR SOFTWARE STEPPING RATES
03634                      1504 HARDWARE STEPPING RATES
00   01   02   03                 THE 4 ENTRIES ARE
037115   3    300  msec 149  SOFTWARE STEPPING RATES
0385250  0.1  250  msec 144  MOTOR START TIME IN MILLISECONDS
0394100  0.5  100  msec 142  WAIT TIME FOR HEAD LOAD WITH MAX & MIN VALUES
0404100  0.1  100  msec 143  HEAD SETTLING TIME WITH MAX & MIN VALUES
0415250  0.1  1000 msec 145  DELAY BETWEEN DRIVE SELECTS WITH MAX & MIN
```

CONTENTS OF QSYS.DAT (continued)

```
04243     0.5   10    msec 146   TUNNEL ERASE DELAY WITH MAX & MIN VALUES
043243    0     159   trks 141   TRK THAT LOW WRITE-CURRENT INVOKED.  MAX & MIN
044131    1     255        147   READ/WRITE RETRIES
045110    1     255        148   RETRIES FOR VERIFICATION
04602                      174   8 INCH OR 5.25 INCH
04732                      1732  48 OR 96 T.P.I.
96   48                          THE 2 ENTRIES
04801                      171   HARDWARE OR SOFTWARE STEPPING RATES
04934                      1704  HARDWARE STEPPING RATES
00   01   02   03                   THE 4 ENTRIES ARE
050115    3     300   msec 169   SOFTWARE STEPPING RATES
0515250   0.1   250   msec 164   MOTOR START TIME IN MILLISECONDS
0524100   0.5   100   msec 162   WAIT TIME FOR HEAD LOAD WITH MAX & MIN VALUES
0534100   0.1   100   msec 163   HEAD SETTLING TIME WITH MAX & MIN VALUES
0545250   0.1   1000  msec 165   DELAY BETWEEN DRIVE SELECTS WITH MAX & MIN
05543     0.5   10    msec 166   TUNNEL ERASE DELAY WITH MAX & MIN VALUES
056243    0     159   trks 161   TRK THAT LOW WRITE-CURRENT INVOKED.  MAX & MIN
057131                     167   READ/WRITE RETRIES
058110                     168   RETRIES FOR VERIFICATION
10031                      1994  MEGATEL DISTRIBUTION FORMAT SELECTED BY SEL ?
0    1    2    3                  THE 4 ENTRIES
10102                      201   SINGLE OR DOUBLE DENSITY?
10201                      200   SINGLE OR DOUBLE SIDED?
103177    10    255   trks 203   NUMBER OF TRACKS PER SIDE
10402                      202   DOUBLE TRACKING Y/N?
10533                      2044  PHYSICAL SECTOR SIZE
128  256  512  1024               THE 4 ENTRIES
106117    1     48         210   PHYSICAL SECTORS PER TRACK
10712     1     8          206   SKEW FACTOR
108120    4     31    bytes207   GAP SIZE
10932                      2115  BLOCK SIZE
1024 2048 4096 8192 16384        THE 5 ENTRIES ARE
11023     1     4          213   NUMBER OF RESERVED TRACKS
1111128   64    256        212   NUMBER OF DIRECTORY ENTRIES
11231                      2084  SELECTION BY SEL
0    1    2    3                  THE 4 ENTRIES
11302                      221   SINGLE OR DOUBLE DENSITY?
11401                      220   SINGLE OR DOUBLE SIDED?
115177    10    255   trks 223   NUMBER OF TRACKS PER SIDE
11602                      222   DOUBLE TRACKING Y/N?
11733                      2244  PHYSICAL SECTOR SIZE
128  256  512  1024               THE 4 ENTRIES
118117    1     48         230   PHYSICAL SECTORS PER TRACK
11912     1     8          226   SKEW FACTOR
120120    4     31    bytes227   GAP SIZE
12132                      2315  BLOCK SIZE
1024 2048 4096 8192 16384        THE 5 ENTRIES ARE
12223     1     4          233   NUMBER OF RESERVED TRACKS
1231128   64    256        232   NUMBER OF DIRECTORY ENTRIES
12431                      2284  SELECTION BY SEL
0    1    2    3                  THE 4 ENTRIES
```

CONTENTS OF QSYS.DAT (continued)

```
12502                       241  SINGLE OR DOUBLE DENSITY?
12601                       240  SINGLE OR DOUBLE SIDED?
127177   10   255  trks    243  NUMBER OF TRACKS PER SIDE
12802                       242  DOUBLE TRACKING Y/N?
12933                       2444 PHYSICAL SECTOR SIZE
128  256  512  1024                THE 4 ENTRIES
130117   1    48            250  PHYSICAL SECTORS PER TRACK
13112    1    8             246  SKEW FACTOR
132120   4    31   bytes    247  GAP SIZE
13332                       2515 BLOCK SIZE
1024 2048 4096 8192 16384        THE 5 ENTRIES ARE
13423    1    4             253  NUMBER OF RESERVED TRACKS
1351128  64   256           252  NUMBER OF DIRECTORY ENTRIES
13631                       2484 SELECTION BY SEL
0    1    2    3                THE 4 ENTRIES
13702                       261  SINGLE OR DOUBLE DENSITY?
13801                       260  SINGLE OR DOUBLE SIDED?
139177   10   255  trks    263  NUMBER OF TRACKS PER SIDE
14002                       262  DOUBLE TRACKING Y/N?
14133                       2644 PHYSICAL SECTOR SIZE
128  256  512  1024                THE 4 ENTRIES
142117   1    48            270  PHYSICAL SECTORS PER TRACK
14312    1    8             266  SKEW FACTOR
144120   4    31   bytes    267  GAP SIZE
14532                       2715 BLOCK SIZE
1024 2048 4096 8192 16384        THE 5 ENTRIES ARE
14603    1    4             273  NUMBER OF RESERVED TRACKS
1471128  64   256           272  NUMBER OF DIRECTORY ENTRIES
14831                       2684 SELECTION BY SEL
0    1    2    3                THE 4 ENTRIES
14902                       281  SINGLE OR DOUBLE DENSITY?
15001                       280  SINGLE OR DOUBLE SIDED?
151177   10   255  trks    283  NUMBER OF TRACKS PER SIDE
15202                       282  DOUBLE TRACKING Y/N?
15333                       2844 PHYSICAL SECTOR SIZE
128  256  512  1024                THE 4 ENTRIES
154117   1    48            290  PHYSICAL SECTORS PER TRACK
15512    1    8             286  SKEW FACTOR
156120   4    31   bytes    287  GAP SIZE
15732                       2915 BLOCK SIZE
1024 2048 4096 8192 16384        THE 5 ENTRIES ARE
15823    1    4             293  NUMBER OF RESERVED TRACKS
1591128  64   256           292  NUMBER OF DIRECTORY ENTRIES
16031                       2884 SELECTION BY SEL
0    1    2    3                THE 4 ENTRIES
20033                       4023 CONSOLE INPUT
16384512  16896                  THE 3 ENTRIES
20133                       4033 CONSOLE OUTPUT
163843276849152                  THE 3 ENTRIES
20231                       4052 AUXILIARY INPUT
16384512                         THE 2 VALUES
```

CONTENTS OF QSYS.DAT (continued)

```
20331                       4064 AUXILIARY OUTPUT
163848192 1024 32768               THE 4 VALUES
20433                       4044 LIST DEVICE
4096 2048 1024 32768               THE 4 VALUES
2052X     0     255  nulls411  NULLS REQUIRED FOR EOL TO NEXT-LINE
20637                       4087 SIMPLEX SERIAL PORT BAUD RATE TABLE
14   13   12   11   10   9    8                    THE 7 ENTRIES
20737                       40714FULL DUPLEX SERIAL PORT BAUD RATE TABLE
14   13   12   11   10   9    8    7    6    5    4    3    2    1    0
20801                       030  ALPHA-NUMERIC MODE?
20921     0                 028  NUMBER OF ROWS FOR DISPLAY SCREEN
210127    1                 022  NUMBER OF ROWS FOR MAIN SCREEN
21122                       025  NUMBER OF ROWS FOR STATUS SCREEN
21201                       032  DISPLAY CLOCK?
21301                       033  WHERE TO DISPLAY CLOCK (TRUE=TOP, FALSE=BOTTOM)
2142X     1     7           432  COLUMN OFFSET IF GRAPHICS
2152X     1     16          430  COLOUR GRAFICS FORGROUND COLOUR
2162X     1     16          431  COLOUR GRAFICS BACKGROUND COLOUR
```

This page intentionally left blank.

10.0 Floppy disc interface error messages (CP/M 3.01)

Error messages returned from the floppy disc handler are usually in the form indicated below:

```
BIOS Error on d: T-00000, S-00000 oo,ee
 Retry (Y/N)?
```

where d — represents the logical drive name on which the error occured

T-00000 — represents the number of the track on which the error occured

S-00000 — represents the number of the sector on which the error occured

oo — is a description of the operation being executed at the time of the error

ee — description of the error

The possible operations (oo) with descriptions are as follows:

READ	- attempting to read the disc
WRITE	- attempting to write to the disc
SEEK WITH READ	- attempting to locate a specific track and sector with a read pending
SEEK WITH WRITE	- attempting to locate a specific track and sector with a write pending

The possible errors (ee) with descriptions are as follows:

Write Protected	- could not write to the disc because it is write protected
Write Fault	- could not write to the desired track and sector
Read Fault	- could not read the desired track and sector
Verification Fault	- the data read did not match the data in memory
Lost Data	- the computer did not respond to the data request in one byte time
Data Request Fault	- the DR is full on a read operation or the DR is empty on a write operation
Busy	- another command is under execution

This page intentionally left blank

11.0 Miscellaneous Software Notes for Megatel QUARK CP/M 3.0

1. For release 3.01, sections of the BIOS source file located within "IF...ENDIF" assembly switches which are turned OFF in the distributed copy of the source are not guaranteed to be correct.

2. For release 3.01 a screen print/dump feature is available as part of the console input routine for alphanumeric mode in the BIOS. To use this feature enter "control\". This will output all printable characters that are being displayed on the screen to the LST: device. Any items displayed on the STATUS or DISPLAY lines are not printed.

APPENDIX

ELECTRICAL SPECIFICATIONS

Parameter	60Hz models	50Hz models	units
Master clock frequency	23.86176	24.80000	MHz
Z-80B clock frequency	5.96544	6.20000	MHz
Z-80B T-state period	167.63223	161.29032	ns
E-clock frequency	1.49136	1.55000	MHz
Write-precompensation:			
8-inch drives	125	125	ns
5-1/4 inch drives	250	250	ns
Horizontal sync frequency	15.540	16.150	kHz
Horizontal sync period	64.35	61.92	us
H-sync pulse-width	21.45	20.64	us
Horizontal sync polarity	positive	positive	
V-sync pulse-width	187.7	180.6	us
Vertical sync polarity	negative	negative	
Length of Video data	53.6	51.6	us
Percentage line utilization	83.3	83.3	%
Video output amplitude	4.0	4.0	Vp-p±3dB
H-sync output amplitude	4.0	4.0	Vp-p±3dB
V-sync output amplitude	4.0	4.0	Vp-p±3dB
H-sync pulse width	8.0	8.0	us
Front porch (data to start H-sync)	1.8	1.8	us
Back porch (end H-sync to data)	0.7	0.7	us
Composite video amplitude	1.0	1.0	Vp-p±3dB
Composite sync level relative to black level	-0.5	-0.5	V

TABLE I I/O ADDRESSES AND FUNCTIONS FOR THE QUARK

I/O ADDRESS	DEVICE & REGISTER	FUNCTION (READ / WRITE)
00-3F	Character generator (Q/100, Q/200) Translation Tables (QUARK/150)	Special procedure must be invoked to write (sec T/2)
40-5E	I/O alias	Not recommended for use
5F	Parallel printer output port	-VIA CA1: parallel printer Acknowledge input -VIA CA2: parallel printer Data Strobe output
60	VIA ORB/IRB	Output register B / Input Register B -PB0-5: General-purpose I/O lines PB0 configured for bell output -PB6: Simplex port protocol/data input -PB7: ACIA transmitter/transmitter & receiver clock output
61	" ORA/IRA	Output register A / Input Register A -PA0-7: General-purpose I/O lines
62	" DDRB	Data direction register B
63	" DDRA	Data direction register A
64	" T1C-L	T1 low-order counter / T1 low-order latch
65	" T1C-H	T1 high-order cntr / T1 high-order latch/cntr
66	" T1L-L	T1 low-order latch
67	" T1L-H	T1 high-order latch -T1 counter used as ACIA baud rate generator
68	" T2C-L	T2 low-order counter / T2 low-order latch
69	" T2C-H	T2 high-order counter -T2 counter used as baud rate generator for VIA Shift Register (simplex serial port), or for full-duplex serial port transmitter with split baud rates, or as a timer
6A	" SR	Shift register for simplex serial port -CB1: SR clock from PB7 if J3 & J4 installed -CB2: Tx DATA output for simplex serial port
6B	" ACR	Auxiliary control register
6C	" PCR	Peripheral control register -CA1: Parallel printer acknowledge input -CA2: Parallel printer data strobe output
6D	" IFR	Interrupt flag register
6E	" IER	Interrupt enable register
6F	" ORA/IRA	Same as address 61 except no "handshake"

TABLE I I/O ADDRESSES AND FUNCTIONS FOR THE QUARK (CONTINUED)

I/O ADDRESS	DEVICE & REGISTER	FUNCTION (READ / WRITE)
70-73	I/O alias	Not recommended for use
74	PIA PA or DDRA	Peripheral reg A or data direction reg A -PA0-7: 8-bit encoded keyboard input or general-purpose I/O lines
75	" CRA	Control register A -CA1: Vertical Sync interrupt input -CA2: External interrupt input, or keyboard strobe, or bell output
76	" PB or DDRB	Peripheral reg B or data direction reg B -PB0-2: Floppy Disc SEL 0-2 outputs(act. high) -PB3: Floppy Disc SEL 3/LOW CURRENT(act. high) -PB4: Floppy Disc SIDE select output -PB5: Floppy Disc SNGL/DBLE select (high=SNGL) -PB6: GRAPHICS/ALPHA mode bit (high=ALPHA) -PB7: Full-duplex port DTR output (active low)
77	" CRB	Control register B -CB1: Floppy Disc Controller INTERRUPT REQUEST -CB2: Boot Mode: low = normal operation high= PROM selected, RAM deselected
78	ACIA SR/CR	Status register / Control register
79	" RDR/TDR	Receive data register / Transmit data register -VIA Timer 1 sets baud rate on transmitter and receiver under non-split baud rates, receiver-only under split baud rates -VIA Timer 2 sets transmitter baud rate under split baud rates
7A-7F	I/O alias	Not recommended for use
80	FDC STR/CR	Status register / Control register -FDC interrupts are sent to PIA CB1 input
81	" TR	Track register
82	" SR	Sector register
83	" DR	Data register

TABLE I I/O ADDRESSES AND FUNCTIONS FOR THE QUARK (CONTINUED)

I/O ADDRESS	DEVICE & REGISTER	FUNCTION (READ / WRITE)
84-8F	I/O alias	Not recommended for use
90	LAN STS/CMD (Q/200 only)	Local-area network status/command register -stat reg bit 7: LAN READY flag (active high) -control block address for Omninet interface
91-97	I/O alias	Not recommended for use.
98	LAN INTERRUPT (Q/200 only)	LAN interrupt status/control register -status reg bit 7=1: LAN interrupt pending -control reg bit 7=1: LAN interrupts enabled -control reg bit 7=0: LAN interrupts disabled -writing to this port clears LAN interrupt
99-BF	I/O alias	Not recommended for use.
C0-FF	User /CS output	Active low output on pin A-14 of the ESIC connector. Use to select a single external peripheral device, or to qualify decoding of address lines for multiple external peripheral devices.

Notes:

1. For software compatibility between current and future QUARK single-board computers, it is recommended that I/O addresses indicated as "Reserved" or "I/O alias" not be used in programming the QUARK.
2. The Parallel Printer Data Strobe output is driven an inverting TTL buffer on version 04R01 and earlier QUARK/100s, and on all QUARK/150s and /200s earlier than 02R00. On revision 05R00 and later QUARK/100s, this output is driven by a non-inverting buffer.

TABLE IIa TIMER-1 BAUD RATES FOR FULL-DUPLEX INTERFACE - 50HZ VERSION

BAUD RATE	DIVIDE BY 1 VALUE	(HEX)	ERROR	DIVIDE BY 16 VALUE	(HEX)	ERROR	DIVIDE BY 64 VALUE	(HEX)	ERROR
19200	39	0027	-.95	1	0001	-8.26	-1	000-1	-15.91
9600	79	004F	-.03	3	0003	6.22	0	0000	-27.92
7200	106	006A	-.10	5	0005	-.33	0	0000	-3.89
4800	160	00A0	-.18	8	0008	3.50	1	0001	-8.26
3600	214	00D6	-.22	C	000C	-2.15	2	0002	-10.30
2400	321	0141	.05	18	0012	2.19	3	0003	6.22
1800	429	01AD	-.05	25	0019	.60	5	0005	-.33
1200	644	0284	.01	39	0027	-.95	8	0008	3.50
600	1290	050A	-.01	79	004F	-.03	18	0012	1.55
300	2582	0A16	-.02	160	00A0	-.18	39	0027	-.95
150	5165	142D	.00	321	0141	.05	79	004F	-.03
134.5	5760	1680	.01	358	0166	.11	88	0058	.31
110	7044	1B84	.00	439	01B7	-.09	108	006C	.31
75	10332	285C	.00	644	0284	.01	160	00A0	-.18
50	15498	3C8A	.00	967	03C7	.00	240	00F0	.18

* Not recommended. Use the Divide-by-16 mode.

TABLE IIb TIMER-1 BAUD RATES FOR FULL-DUPLEX INTERFACE - 60HZ VERSION

BAUD RATE	DIVIDE BY 1 VALUE	(HEX)	ERROR	DIVIDE BY 16 VALUE	(HEX)	ERROR	DIVIDE BY 64 VALUE	(HEX)	ERROR
19200	37	0025	.23%	1	0001	-11.73%	--	--	--
9600	76	004C	-.10	3	0003	2.20	--	--	--
7200	102	0066	-.18	5	0005	-4.10	--	--	--
4800	154	009A	-.26	8	0008	-.42	1	0001	-11.73
3600	205	00CD	.19	B	000B	1.54	2	0002	-13.69
2400	309	0135	-.02	18	0012	-1.68	3	0003	2.20
1800	413	019D	-.12	24	0018	.55	5	0005	-4.10
1200	620	026C	-.06	37	0025	.23	8	0008	-.42
600	1241	04D9	.03	76	004C	-.10	17	0011	.23
300	2484	09B4	-.01	154	009A	-.26	37	0025	.23
150	4969	1369	.01	309	0135	-.02	76	004C	-.10
134.5	5542	15A6	.01	345	0159	-.07	85	0055	-.14
110	6777	1A79	.00	422	01A6	-.02	104	0068	.16
75	9941	26D5	.00	620	026C	-.06	154	009A	-.26
50	14912	3A40	.00	930	03A2	.04	231	00E7	.12

* Not recommended. Use the Divide-by-16 mode.

Notes for Table II(a) and II(b)

1. The frequency of the Timer 1-generated clock output on PB7 of the VIA is given by

$$f_E/(2N+3.5),$$

where f_E is the frequency of the E-clock and N is the value in the Timer 1 latch. The baud rate is this frequency divided by 1, 16, or 64, according to the divide ratio bits in the ACIA.

2. Baud rates other than those shown above are possible by loading the Timer 1 latches whith the value determined by the equation above. Consult the data sheets for the VIA (6522A) and the ACIA (68A50).

3. Baud rate errors exceeding out 5% may be unacceptable in some applications. If this is the case, Timer-2 may be used to generate the Transmit and Receive clocks for the Full-duplex port. To do this, install jumpers J3 and J4, set PB7 of the VIA to act as an input, and use the Timer-2 generated baud rates (Table IIIa) for the ACIA Transmit and Receive clocks.

4. In the divide-by-one mode, the ACIA receiver clock should be synchronized with the incoming data. The ACIA transmitter will operate normally in this mode.

TABLE IIIa TIMER-2 BAUD RATES FOR FULL-DUPLEX RECEIVER IN SPLIT BAUD MODE - 50HZ VERSION

BAUD RATE	DIVIDE BY 1 VALUE	(HEX)	ERROR	DIVIDE BY 16 VALUE	(HEX)	ERROR	DIVIDE BY 64 VALUE	(HEX)	ERROR
19200	38	26	.91	1	01	-15.61	0*	00	-68.47
9600	79	4F	-.33	3	03	.91	0*	00	-36.93
7200	106	6A	-.33	5	05	-3.89	0*	00	-15.91
4800	159	9F	.28	8	08	.91	1*	01	-15.91
3600	213	D5	.12	11	0B	3.89	1	01	12.31
2400	--	--	--	18	12	.91	3	03	.91
1800	--	--	--	25	19	-.33	5	05	-3.89
1200	--	--	--	38	26	.91	8	08	.91
600	--	--	--	79	4F	-.33	18	12	.91
300	--	--	--	159	9F	.28	38	26	.91
150	--	--	--	321	--	--	79	4F	-.33
134.5	--	--	--	358	--	--	88	58	.04
110	--	--	--	438	--	--	108	6C	.08
75	--	--	--	644	--	--	159	9F	.28
50	--	--	--	967	--	--	240	F0	.08

* Not recommended. Use the Divide-by-16 mode.

TABLE IIIb TIMER-2 BAUD RATES FOR FULL-DUPLEX RECEIVER IN SPLIT BAUD MODE - 60HZ VERSION

BAUD RATE	DIVIDE BY 1 VALUE	(HEX)	ERROR	DIVIDE BY 16 VALUE	(HEX)	ERROR	DIVIDE BY 64 VALUE	(HEX)	ERROR
19200	37	25	-.42	1	01	-19.09	0*	00	-69.66
9600	76	4C	-.42	3	03	-2.91	0*	00	-39.32
7200	102	66	-.42	5	05	-7.53	0*	00	-19.02
4800	153	99	-.23	8	08	-2.91	0*	00	21.37
3600	205	CD	-.06	11	0B	-.42	1	01	7.88
2400	--	--	--	17	11	2.20	3	03	-2.91
1800	--	--	--	24	18	-.42	5	05	-7.53
1200	--	--	--	37	25	-.42	8	08	-2.91
600	--	--	--	76	4C	-.42	17	11	2.20
300	--	--	--	153	99	.23	37	25	-.42
150	--	--	--	309	--	--	76	4C	-.42
134.5	--	--	--	345	--	--	85	55	-.43
110	--	--	--	422	--	--	104	68	-.08
75	--	--	--	619	--	--	153	99	.23
50	--	--	--	930	--	--	231	E7	.01

* Not recommended. Use the Divide-by-16 mode.

Notes for Table III(a) and III(b)

1. The frequency of the Timer-2 clock output on CB1 of the VIA is given by

$$f_E/(2N+4),$$

where f_E is the frequency of the E-clock and N is the value in the Timer-2 latch. The baud rate is this frequency divided by 1, 16, or 64, according to the divide ratio bits (bits 0 and 1) in the ACIA control register.

2. For the split baud rate mode, J3 must be opened and J4 closed. Timer 1 is used to generate a square wave on PB7, the frequency of which determines the Full-duplex port transmitter baud rate. (Use the values given in Table II for these transmitter baud rates). With the Shift Register in the free-running output mode, the receiver clock frequency is determined by Timer 2.

TABLE IVa SIMPLEX SERIAL PORT BAUD RATE SELECTION - 50HZ

BAUD RATE	VALUE FOR VIA TIMER 2 LATCH VALUE	(HEX)	MODE	ERROR-%
9600	79	4F	1 bit/bit	-.33
4800	159	8F	1 bit/bit	.28
9600	38	26	2 bits/bit	.91
4800	79	4E	2 bits/bit	-.33
2400	159	8F	2 bits/bit	.28
19200	8	8	4 bits/bit	.91
9600	18	12	4 bits/bit	.91
7200	25	19	4 bits/bit	-.33
4800	38	26	4 bits/bit	.91
3600	52	34	4 bits/bit	-.33
2400	79	4E	4 bits/bit	-.33
1800	106	6A	4 bits/bit	-.33
1200	159	8F	4 bits/bit	.28

TABLE IVb SIMPLEX SERIAL PORT BAUD RATE SELECTION - 60HZ

BAUD RATE	VALUE FOR VIA TIMER 2 LATCH VALUE	(HEX)	MODE	ERROR-%
9600	76	4C	1 bit/bit	-.42
4800	153	99	1 bit/bit	.23
9600	37	25	2 bits/bit	-.42
4800	76	4C	2 bits/bit	-.42
2400	153	99	2 bits/bit	.23
19200	8	09	4 bits/bit	-2.9
9600	17	11	4 bits/bit	2.2
7200	24	18	4 bits/bit	-.42
4800	37	25	4 bits/bit	-.42
3600	50	32	4 bits/bit	-.42
2400	76	4C	4 bits/bit	-.42
1800	102	66	4 bits/bit	-.42
1200	153	99	4 bits/bit	.23

Notes for Table IV

1. "Mode" indicates the number of bits in the VIA Shift Register which are used to generate one "bit" of output. The expansion of bits in this manner must be handled in software. The Simplex Serial Port drivers in the QUARK operating systems use the "4 bits/bit" mode.

2. The actual shift frequency is given by

$$f_E/(2N+4),$$

where f_E is the frequency of the E-clock and N is the value in the Timer 2 latch.

3. Connecting jumpers J3 and J4 will connect the PB7 I/O line from the VIA to the CB1 control line of the VIA. This allows Timer 1, normally used to generate the transmit and receive clocks for the full-duplex serial port, to generate the simplex serial port clock as well. However, split baud rates on the full-duplex channel are not possible when Timer 1 is used in this way.

TABLE V SYNCHRONOUS ADDRESS MULTIPLEXER ADDRESS ASSIGNMENTS

ADDRESS (HEX)	CONTROL REGISTER BIT	SET/CLEAR	REMARKS
FF80	V0	CLR	Normally cleared
FF81	V0	SET	
FF82	V1	CLR	Normally cleared
FF83	V1	SET	
FF84	V2	CLR	Cleared for Alphanumeric Mode
FF85	V2	SET	Set for Graphics Mode
FF86	F0	CLR	Start address bit 10
FF87	F0	SET	
FF88	F1	CLR	Start address bit 11
FF89	F1	SET	
FF8A	F2	CLR	Start address bit 12
FF8B	F2	SET	
FF8C	F3	CLR	Start address bit 13
FF8D	F3	SET	
FF8E	F4	CLR	Normally set
FF8F	F4	SET	
FF90	F5	CLR	Start address bit 14
FF91	F5	SET	
FF92	F6	CLR	Start address bit 15
FF93	F6	SET	
FF94	P1	CLR	Page bit-see Section T/1.1
FF95	P1	SET	
FF96	R0	CLR	Normally cleared
FF97	R0	SET	
FF98	R1	CLR	Normally cleared
FF99	R1	SET	
FF9A	M0	CLR	Normally cleared
FF9B	M0	SET	
FF9C	M1	CLR	Normally set
FF9D	M1	SET	
FF9E	TY	CLR	Map type-see sec T/1.1
FF9F	TY	SET	(Normally set)

Notes:

1. To set or clear any of the bits in SAM registers, load the address in the above table corresponding to the bit to be set or cleared into the HL register, and then execute a CALL to the subroutine at location 0Bhex. (Note that this routine alters the contents of the C register.)

TABLE VI SUGGESTED VALUES FOR THE SAM CONTROL REGISTER

f_{VERT}	MODE	F6	F5	F4	F3	F2	F1	F0	VIDEO MEMORY ADDRESS RANGE
60Hz	ALPHA	0	0	1	1	1	0	0	3000-3FFF
60Hz	ALPHA	0	1	1	1	1	0	0	7000-7FFF
60Hz	ALPHA	1	0	1	1	1	0	0	B000-BFFF
60Hz	ALPHA	1	1	1	1	1	0	0	F000-FFFF
60Hz	GRAPHICS	0	0	1	1	0	0	0	2000-7FFF
60Hz	GRAPHICS	0	1	1	1	0	0	0	6000-BFFF
60Hz	GRAPHICS	1	0	1	1	0	0	0	A000-FFFF
50Hz	ALPHA	0	0	1	1	0	1	1	2C00-3FFF
50Hz	ALPHA	0	1	1	1	0	1	1	6C00-7FFF
50Hz	ALPHA	1	0	1	1	0	1	1	AC00-BFFF
50Hz	ALPHA	1	1	1	1	0	1	1	EC00-FFFF
50Hz	GRAPHICS	0	0	1	0	0	1	0	0800-7FFF
50Hz	GRAPHICS	0	1	1	0	0	1	0	4800-BFFF
50Hz	GRAPHICS	1	0	1	0	0	1	0	8800-FFFF

Notes:

1. The starting address is calculated from the polynomial

   $SA=(F6)*2^{15}+(F5)*2^{14}+(F3)*2^{13}+(F2)*2^{12}+(F1)*2^{11}+(F0)*2^{10}$.

   Thus the starting address is the binary number

   (F6)(F5)(F3)(F2) (F1)(F0)00 0000 0000.

2. The final address is the first 16k boundary following the starting address in Alphanumeric mode, or the second 16k boundary following the starting address in Graphics mode.

3. Bit F4 in the SAM Control Register must always be one.

4. On Quarks with 128k memory, the Video Memory is located in the memory bank determined by bit 0 of the I register. See section T/1.3 on Video Display Memory.

5. In Graphics mode, the starting address must be on a 3k boundary.

TABLE VII QUARK PIN CONNECTIONS AND FUNCTIONS

PIN	GROUP	DESCRIPTION
A-1	CRT	Ground
A-2	CRT	Vertical sync output (RED output on QUARK/150)
A-3	FULL-DUP	RS-232C transmit data from ACIA (full-duplex port)
A-4	SIMPLEX	RS-232C transmit data from VIA (see sec. T/3.4)
A-5	FULL-DUP	Ground
A-6	P3	VIA PB1 parallel I/O line → BUTTON
A-7	P3	VIA PB3 parallel I/O line → BACKWARD } Joystick
A-8	P3	VIA PB5 parallel I/O line → RIGHT
A-9	PAR PTR	Parallel printer output bit 7
A-10	PAR PTR	Parallel printer output bit 5
A-11	PAR PTR	Parallel printer output bit 2
A-12	PAR PTR	Parallel printer output bit 0
A-13	P/S PTR	Ground for parallel/serial printer
A-14	EXP BUS	User chip select output (active low)
A-15	EXP BUS	E-CLK output
A-16	DISC	Ground
A-17	DISC	Step output to floppy disc drive (active low)
A-18	DISC	Write gate output to floppy disc drive (active low)
A-19	DISC	Floppy disc drive side select (active low)
A-20	DISC	Ground
A-21	P1	PIA PA2 parallel I/O line (KBD2)
A-22	P1	PIA PA4 parallel I/O line (KBD4)
A-23	P1	PIA PA6 parallel I/O line (KBD6)
A-24	EXP BUS	Ground
A-25	EXP BUS	Z-80B data bus D1
A-26	EXP BUS	Z-80B data bus D7
A-27	EXP BUS	Z-80B data bus D6
A-28	EXP BUS	Z-80B data bus D3
A-29	FLOPPY	Floppy disc index signal input (active low)
A-30	RESERVED	Reserved for future hard disc version
A-31	POWER	Ground return line
A-32	POWER	Ground return line

Joystick Pin Outs (DB 9)

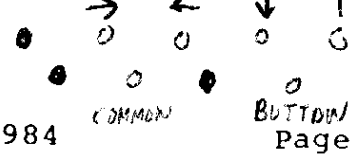


TABLE VII QUARK PIN CONNECTIONS AND FUNCTIONS (CONTINUED)

PIN	GROUP	DESCRIPTION
B-1	CRT	TTL video signal out (GREEN output on QUARK/150)
B-2	SIMPLEX	Printer busy input (see sec T/3.4)
B-3	FULL-DUP	RS-232C RTS output from ACIA
B-4	FULL-DUP	RS-232C DSR input to ACIA
B-5	P2	VIA PA0 parallel I/O line
B-6	P2	VIA PA1 parallel I/O line
B-7	P2	VIA PA2 parallel I/O line
B-8	P2	VIA PA3 parallel I/O line
B-9	P2	VIA PA4 parallel I/O line
B-10	P2	VIA PA5 parallel I/O line
B-11	P2	VIA PA6 parallel I/O line
B-12	P2	VIA PA7 parallel I/O line
B-13	PAR PTR	Parallel printer acknowledge input to VIA
B-14	EXP BUS	Z-80B address bus A2
B-15	EXP BUS	Z-80B address bus A4
B-16	DISC	Ground
B-17	DISC	Use for Disc Size Mode (user modification)
B-18	DISC	Floppy disc drive select #3 output (active low)
B-19	DISC	Floppy disc drive select #0 output (active low)
B-20	DISC	Ground
B-21	P1	PIA PA0 parallel I/O line (KBD0)
B-22	P1	PIA PA7 parallel I/O line (KBD7)
B-23	P1	PIA CA2 control line (KB STROBE)
B-24	EXP BUS	Z-80B address bus A1
B-25	EXP BUS	Power-on-reset output (active low)
B-26	EXP BUS	Z-80B WR control line (active low)
B-27	EXP BUS	Z-80B RD control line (active low)
B-28	EXP BUS	Z-80B INT input line (active low)
B-29	FLOPPY	Track 00 sense input (active low)
B-30	RESERVED	Reserved for future hard disc version
B-31	POWER	+5V regulated input
B-32	POWER	+5V regulated input

TABLE VII QUARK PIN CONNECTIONS AND FUNCTIONS (CONTINUED)

PIN	GROUP	DESCRIPTION
C-1	CRT	Horizontal sync output (BLUE output on QUARK/150)
C-2	FULL-DUP	RS-232C RxD serial data input to ACIA
C-3	FULL-DUP	RS-232C DTR output (driven by PIA PB7-see sec T/3.3)
C-4	FULL-DUP	RS-232C CTS input to ACIA
C-5	CRT	Composite video output (COMPOSITE SYNC on QUARK/150)
C-6	P3	VIA PB2 parallel I/O line — FORWARD
C-7	P3	VIA PB0 parallel I/O line — BELL/SOUND } Joystick + Sound.
C-8	P3	VIA PB4 parallel I/O line — LEFT
C-9	PAR PTR	Parallel printer interface output bit 4
C-10	PAR PTR	Parallel printer interface output bit 3
C-11	PAR PTR	Parallel printer interface output bit 6
C-12	PAR PTR	Parallel printer interface outputbit 1
C-13	PAR PTR	Parallel printer interface data strobe output
C-14	EXP BUS	Z-80B address bus A3
C-15	EXP BUS	Z-80B address bus A5
C-16	DISC	Read data input from floppy disc (active low)
C-17	DISC	Direction control output to floppy (active low)
C-18	DISC	SEL 2 floppy disc drive select (active low)
C-19	DISC	SEL 1 floppy disc drive select (active low)
C-20	DISC	Write data output to floppy disc drive (active low)
C-21	P1	PIA PA1 parallel I/O line (KBD1)
C-22	P1	PIA PA3 parallel I/O line (KBD3)
C-23	P1	PIA PA5 parallel I/O line (KBD5)
C-24	EXP BUS	Z-80B address bus A0
C-25	EXP BUS	Z-80B data bus D0
C-26	EXP BUS	Z-80B data bus D2
C-27	EXP BUS	Z-80B data bus D4
C-28	EXP BUS	Z-80B data bus D5
C-29	DISC	Floppy disc write protect sense input (active low)
C-30	RESERVED	Reserved for future hard disc version
C-31	RESET	Reset input (active low)
C-32	POWER	+12 V regulated input

1. P1, P2, P3 refer to general-purpose parallel ports 1, 2, and 3. In standard Quark operating systems port P1 is configured as a standard encoded-keyboard data input, with CA2 of the PIA as the keyboard data strobe input.
2. EXP BUS refers to the set of connections to the Z-80 address, data, and control lines, and to the E-clock and external chip-select lines. These lines form the Peripheral Expansion Bus.
3. DISC refers to connections for the floppy disc drives.
4. All inputs and outputs for the Parallel Printer (PAR PTR) interface are TTL-compatible.
5. The Full-duplex Serial Port (FULL-DUP) is implemented using the ACIA and PB7 of the PIA.
6. The Simplex Serial Port (SIMPLEX) is implemented using the VIA.
7. For proper operation, the composite video output should be terminated by a bridging 75 ohm load at the monitor.
8. Bit 0 is the least-significant, bit 7 the most-significant.

TABLE VIII QUARK PERIPHERAL CONNECTIONS

TABLE VIIIa FULL-DUPLEX PORT CONNECTIONS

DB-25S PIN (MODEM)	DB-25S PIN (TERMINAL)	FUNCTION	QUARK PIN	REMARKS
1	1	PROT GND	---	(Opt) Chassis ground
2	3	Tx DATA	A-3	Twist with ground wire
3	2	Rx DATA	C-2	Twist with ground wire
4	5	RTS	B-3	
5	4	CTS	C-4	
6	20	DSR	B-4	
7	7	GROUND	A-5	Ground
20	6	DTR	C-3	

Notes:
1. The pin numbers in the column labelled "MODEM" represent the pinouts for connecting to industry-standard modems using a DB-25S connector.
2. The pin numbers in the column labelled "TERMINAL" represent the pinouts for connecting to a standard computer terminal with an RS-232C serial interface using a DB-25S connector. Use these pinouts for connecting an external terminal to be used as the Console device. (See sec. I/2.0 on installation).
3. If a serial printer is to be connected on the Full-Duplex Serial Port, the DB-25S pinout given in the column labelled "TERMINAL" should be used.

TABLE VIIIb PARALLEL PRINTER INTERFACE CONNECTIONS

PRINTER PIN	FUNCTION	QUARK PIN	REMARKS
1	DATA STROBE	C-13	Data strobe, Acknowledge,
2	BIT 0 (LSB)	A-12	and Bits 0-7 may each be
3	BIT 1	C-12	twisted with ground wires
4	BIT 2	A-11	
5	BIT 3	C-10	
6	BIT 4	C-9	
7	BIT 5	A-10	
8	BIT 6	C-11	
9	BIT 7 (MSB)	A-9	
10	ACKNOWLEDGE	B-13	
19-29	GROUND	A-13	

Notes:
1. Printer pin numbers shown for the Parallel Printer connections are the pin numbers for a Centronics 739 printer connector.

TABLE VIII QUARK PERIPHERAL CONNECTIONS (CONTINUED)

TABLE VIIIc 5-1/4-INCH FLOPPY DISC DRIVE CONNECTIONS

EDGE CONNECTOR	FUNCTION	QUARK PIN	REMARKS
1	GROUND	----	
2	OPT	----	
3	GROUND	----	
4	BUSY	----	
5	GROUND	----	
6	SEL 3	B-18	
7	GROUND	A-20	Read Data, Write Data, Index,
8	INDEX	A-29	and Step may each be twisted
9	GROUND	----	with ground wires.
10	SEL 0	B-19	All disc drive lines are
11	GROUND	----	active low.
12	SEL 1	C-19	
13	GROUND	----	
14	SEL 2	C-18	
15	GROUND	----	
16	MOTOR ON	----	
17	GROUND	----	
18	DIRECTION	C-17	
19	GROUND	A-16	
20	STEP	A-17	
21	GROUND	B-20	
22	WRITE DATA	C-20	
23	GROUND	----	
24	WRITE GATE	A-18	
25	GROUND	----	
26	TRACK 00	B-29	
27	GROUND	----	
28	WRITE PROTECT	C-29	
29	GROUND	B-16	
30	READ DATA	C-16	
31	GROUND	----	
32	SIDE	A-19	
33	GROUND	----	
34	SPARE	----	
	RESERVED	B-17	Use for Disc Size Mode
	RESERVED	A-30	Reserved for hard disc version
	RESERVED	B-30	" " " " "
	RESERVED	C-30	" " " " "

Notes:

1. Pin numbers in the above table refer to standard edge connector pin numbers for a 5-1/4 inch floppy disc drive.
2. If the Motor control line must be used, it could be controlled by on of the QUARK parallel I/O lines (if suitably buffered), or by one of the Drive Select lines. Either of these configurations would require a patch to the BIOS.

TABLE VIII QUARK PERIPHERAL CONNECTIONS (CONTINUED)

TABLE VIIId 8-INCH FLOPPY DISC DRIVE INTERFACE

EDGE CONNECTOR	FUNCTION	QUARK PIN	REMARKS
1	GROUND	----	
2	LOW CURRENT	B-18	See Sec T/3.8 for use
3	GROUND	----	
4,6,8,10	N/C	----	
5,7,9,11	GROUND	----	
12	DISC CHANGE	----	
13	GROUND	----	
14	SIDE	A-19	
15	GROUND	----	
16	IN USE	----	
17	GROUND	----	
18	HEAD LOAD	----	
19	GROUND	----	
20	INDEX	A-29	Read Data, Write Data, Index,
21	GROUND	A-20	and Step may each be twisted
22	READY	----	with ground wires.
23	GROUND	----	All disc drive lines are
24	SECTOR	----	active low.
25	GROUND	----	
26	SEL 0	B-19	
27	GROUND	----	
28	SEL 1	C-19	
29	GROUND	----	
30	SEL 2	C-18	
31	GROUND	----	
32	SEL 3	B-18	
33	GROUND	----	
34	DIRECTION	C-17	
35	GROUND	----	
36	STEP	A-17	
37	GROUND	A-16	
38	*WRITE DATA*	*C-20*	
39	GROUND	B-20	
40	WRITE GATE	A-18	
41	GROUND	----	
42	TRACK 00	B-29	
43	GROUND	----	
44	WRITE PROTECT	C-29	
45	GROUND	----	
46	READ DATA	C-16	
47	GROUND	B-16	
48	SEP DATA	----	
49	GROUND	----	
50	SEP CLK	----	

Notes:

1. Pin numbers in the above table refer to standard edge connector pin numbers for an 8-inch floppy disc drive.

TABLE VIII QUARK PERIPHERAL CONNECTIONS (CONTINUED)

TABLE VIIIe ASCII-ENCODED PARALLEL-OUTPUT KEYBOARD CONNECTIONS

FUNCTION	QUARK PIN	REMARKS
KEYBOARD D0	B-21	D0 is the least significant bit of the ASCII code
KEYBOARD D1	C-21	
KEYBOARD D2	A-21	
KEYBOARD D3	C-22	
KEYBOARD D4	A-22	
KEYBOARD D5	C-23	
KEYBOARD D6	A-23	
KEYBOARD D7	B-22	D7 is ignored by the BIOS
KEYBOARD STROBE	B-23	STROBE is initialized active low

Notes:
1. The keyboard input uses Port A of the PIA and CA2.
2. All of the keyboard input lines are TTL-compatible.
3. If a keyboard with an active high STROBE output is to be connected to the QUARK, the STROBE input line may have to be inverted until the console input routine in the BIOS is patched to recognize an active high STROBE.

TABLE VIIIf DIRECT-DRIVE DATA DISPLAY MONITOR CONNECTIONS

FUNCTION	QUARK PIN	REMARKS
Video input on monitor	B-1	The video output from the QUARK can be twisted with a ground wire.
Horizontal sync input	C-1	
Vertical sync input	A-2	
Signal ground on monitor	A-1	

TABLE VIIIg COMPOSITE VIDEO DATA DISPLAY MONITOR CONNECTIONS

FUNCTION	QUARK PIN	REMARKS
Video input on monitor	C-5	The video output from the QUARK can be twisted with a ground wire.
Signal ground on monitor	A-1	

TABLE VIIIh ANALOG RGB COLOUR DISPLAY MONITOR CONNECTIONS

FUNCTION	QUARK PIN	REMARKS
GREEN video output	B-1	Each video output from the
BLUE video output	C-1	QUARK/150 can be twisted with
RED video output	A-2	a ground wire.
COMPOSITE SYNC output	C-5	
Signal ground on monitor	A-1	

TABLE IX JUMPER OPTIONS

JUMPER ID	FUNCTION	USE
J1	MM PAL RAS bypass	Connect for 64k-only
J2	8" / 5-1/4" floppy	Connect for 8" floppy
J3	Non-split baud rate	Connect for non-split
J4	Split baud rate	Connect for split
J5	A-4 GP serial out	Connect for VIA CB1 to A-4
J6	A-4 GP serial out	Connect for VIA CB2 to A-4
J7	B-2 GP serial in	Connect for B-2 to VIA CB2
J8	B-2 GP serial in	Connect for B-2 to VIA PB6
J9	68A50 CS2	Connect for 68A50
J10	68A52 Reset	Connect for 68A52
J11	68A50 CS1	Connect for 68A50
J12	68A52 CS	Connect for 68A52
J13	68A50 CS0	Connect for 68A50
J14	POSITIVE SYNC (Q/150)	Connect for positive sync
J15	NEGATIVE SYNC (Q/150)	Connect for negative sync

Notes:

1. The standard installed jumper configuration is: J2(if 8-inch diskettes are ordered), J3, J6, J8, J9, J11, J13.
2. Connect both J3 and J4 for Timer 1-generated baud rates on the simplex serial port as well as on transmitter and receiver of full-duplex serial port.
3. Pin A-4 is driven by an RS-232C driver output. Pin B-2 drives an RS-232C receiver input.
4. The 68A52 SSDA is available by special order only. Contact the factory for further information.
5. J1 is installed at the factory on 64k memory versions.
6. J14 and J15 apply on to the QUARK/150, where they determine the polarity of the composite sync output to the RGB monitor. Only one of J14 and J15 should be connected at any time.

TABLE X SOURCES FOR MATING 96 PIN CONNECTORS

NAME	ADDRESS	PART NUMBER	
SCHROFF connectors available through			
A.C. Simmonds & Sons, Ltd.	975 Dillingham Road Pickering, Ontario, Canada	69001-679	female w/w
Schroff Inc.	170 Commerce Drive Warwick, Rhode Island, 02886		
Panduit Canada, Ltd	140 Amber Street Markham, Ontario, Canada	100-096-4 100-096-553P	female w/w female p.c.
Panduit Corp.	17301 Ridgeland Ave. Tinley Park, Illinois, 60477		
Robinson Nugent, Inc.	800 E Eighth Street New Albany, Indiana, 47150	RNE-96CS-W-T630	f/m w/w
Amp Canada	20 Esna Park Drive Markham, Ontario, Canada	164303-1 164302-1	female w/w female solder
Amp Inc.	Harrisburg, Pennsylvania 17105		

TABLE XI TERMINAL CONTROL CODES

This table gives the hexidecimal values to be sent to the Quark's Terminal Emulator (or video driver) to perform the various functions it supports. The "Megatel" codes are the codes recognized by the terminal emulator used with any of the distributed operating systems. These codes will be the control codes used in all installed operating systems when the are first booted. QTCONFIG (see below) can be used to change the control codes used.

The "Televideo* 920" codes are an alternative set of control codes which can be used instead of the standard Megatel codes. These codes are a subset of the terminal control codes used by a Televideo 920 terminal. Only some of the functions performed by that terminal are supported by the Quark operating system, so codes other than those given below will be ignored. When running programs which have been designed to run on this terminal, it may be more convenient to load the Televideo 920 code set rather than to modify the program to use the Megatel codes.

The utility program QTCONFIG.COM can be used to load either sets of codes after the system is booted. This utility also allows a set of user-defined codes to be created. This set can be saved as a disc file which can be subsequently loaded through QTCONFIG.COM. For more information about this routine see section 8.9 in the Technical Manual.

A special feture of the Megatel Quark CP/M 3 BIOS is that the console input routine recognizes a "control-backlash" character (=1C hex) as a "Print Screen" command. Interception of this character in the console input data stream will cause the data then being displayed on the Video Display Interface to be be copied to the CP/M list (LST:) device as seven-bit ASCII character data. This feature operates correctly only when an **alhanumeric-mode** terminal driver has been installed in the operating system. If a graphics-mode terminal driver is installed, control-backlash characters should not appear in the console input data, as meaningless data will be produced. Megatel Quark CP/M 2.21 and 2.22 operating systems do not implement the Print Screen function.

* "Televideo" is a registered trademark of Televideo Systems, Inc.

TABLE XI TERMINAL CONTROL CODES

FUNCTION	"MEGATEL" CODE		"TELEVIDEO 920" CODE	
	LEAD-IN (HEX)	CONTROL (HEX)	LEAD-IN (HEX)	CONTROL (HEX)
Bell	none	07	none	07
Cursor Down (line feed)	none	0A	none	0A
Cursor Up (vertical tab)	none	0B	none	0B
Cursor Left (backspace)	none	08	none	08
Cursor Right	none	0C	none	0C
Carriage Return	none	0D	none	0D
Clear Screen (same as Clear Screen & Home)	n/a	n/a	1B	2B
Clear Screen and Home	none	1A	n/a	n/a
Cursor Home	none	1E	none	1E
Reverse Video Off	1B	19 ^Y	1B	6B
Reverse Video On	1B	1F ^_	1B	6A
Half-intensity Off (same as Reverse Off)	n/a	n/a	1B	28
Half-intensity On (same as Reverse On)	n/a	n/a	1B	29
Clear to End of Line (includes character at cursor position)	1B	15 ^U	1B	54
Clear to End of Page (includes characters at or right of cursor position)	1B	16	1B	59
Cursor Addressing (see note 1 for order of co-ordinates)	1B	3D	1B	3D
Cursor On	1B	11	1B	11
Cursor Off	1B	12	1B	12

TABLE XI (continued) Terminal control codes	"MEGATEL" CODE		"TELEVIDEO 920" CODE	
FUNCTION	LEAD-IN (HEX)	CONTROL (HEX)	LEAD-IN (HEX)	CONTROL (HEX)
Insert Line	1B	13 ^S	1B	45
Delete Line	1B	14 ^T	1B	52
Use STATUS screen	1B	17	1B	17
Use MAIN screen	1B	18	1B	18
Use DISPLAY screen	1B	1D	1B	1D
No-Check Mode Off (interprets control codes)	1B	09	1B	09
No-Check Mode On (displays control codes)	1B	10	1B	10
Load Character Pattern (for monochrome alpha mode, for monochrome graphics mode, for colour graphics mode) followed by: -ASCII character code (0-FF) -eight pattern bytes First pattern byte is top line of character pattern.	1B	01	1B	01
Load Character Pattern (for colour alphanumeric mode) followed by: -character set (0,1,or 2) (2=current character set) -ASCII character code (0-FF) -64 pixel nibbles (32 bytes) Each nibble specifies colour value (0-F) for corresponding pixel. First eight nibbles are top line of character pattern.	1B	01	1B	01
Load Symbol (Q/150 only) followed by: -symbol table number (0-F) -symbol table entry (0-FF) -8 pixel nibbles (4 bytes) Each nibble specifies colour value (0-F) for corresponding pixel.	1B	02	1B	02

TABLE XI TERMINAL CONTROL CODES (continued)

TABLE XI (continued) Terminal control codes	"MEGATEL" CODE		"TELEVIDEO 920" CODE	
FUNCTION	LEAD-IN (HEX)	CONTROL (HEX)	LEAD-IN (HEX)	CONTROL (HEX)
Select character set (Q/150 only) for alphanumeric mode followed by: -0 for tables 0-7, or -1 for tables 8-15	1B	05	1B	05
Select table (Q/150 only) for graphics mode followed by: -0 to F for table number	1B	05	1B	05

* Print Screen - 1C hex - This is not a terminal driver function but a special trap in the console input routine.

1. If the Megatel control codes are being used, the cursor addressing character sequence is

   1B 3D xx yy

   where xx and yy are the one-byte values for the x-(horizontal) and y-(vertical) co-ordinates. If the Televideo 920 control codes are in use, the the cursor addressing sequence is

   1B 3D yy xx

   In the CP/M 2.2 terminal drivers, there are 27 lines (0-26) and 80 columns (0-79) on the 60Hz video driver, and 35 (0-34) lines and 80 columns in the 50Hz video driver. In CP/M 3.0 terminal drivers, the vertical size of the screens are defined during system installation. The top left-hand corner of the in-use screen area (DISPLAY, MAIN, or STATUS) is address 0,0 (ie, there is no offset in the addresses). Cursor addresses outside the area of the screen will leave the cursor at the boundary of the screen. For instance, if the x-co-ordinate is outside the range 0 to 80, the cursor will be left at the left edge of the screen at the line specified by the y-co-ordinate. Bit 7 of the address bytes is ignored, offsets of 80(HEX) do not change the address value.

Some language processors will not properly address column 9 or line 9 when using the Cursor Address facility, because they automatically expand a "09" code into a TAB, and insert several spaces. The suggested solution is to add an offset of 80(HEX) to both the x- and y-co-ordinates, since the terminal emulator ignores the top bit of the address.

2. Cursor Up will have no effect if the cursor is positioned on line 0.

3. Cursor Down will perform a scroll (ie. entire screen up one line) if the cursor is positioned on the bottom line of the in-use screen.

4. Cursor Right has no effect if the cursor is positioned in column 80.

5. Cursor Left will have no effect if the cursor is positioned in column 0 of any line.

6. In the CP/M 2.2 terminal drivers, there are 27 lines (0-26) and 80 columns (0-79) on the 60Hz video driver, and 35 (0-34) lines and 80 columns in the 50Hz video driver. In CP/M 3.0 terminal drivers, the vertical sizes of the three screens are defined during system installation.

7. The Televideo 920 "Half-intensity" control codes perform the same effects as the "Reverse Video" control codes. The Televideo 920 "Clear Screen" code performs a "Clear Screen and Home".

8. Programs using these control codes should be made flexible as additional control codes may be implemented in the future.

TABLE XII INSTRUCTION CYCLE TIMING

This table gives the actual number of T-states, including wait states, required for the execution of each instruction on the Quark computer. The folowing rule is used in generating wait states on all models of the Quark.:

1. All machine cycles are extended to mod 4, unless they precede an internal CPU operation.
2. Internal operation T-states are added to those in the preceding cycle, and the combination is extended to mod 4.
3. An additional 4 T-states are added to input or output operations, after rules (1) and (2) are applied.

Memory cycles include opcode fetches, operand reads, data reads or writes, and stack reads or writes. Internal CPU operations are invoked when computing effective addresses in indexed addressing, when performing block move or I/O instructions, etc. I/O instructions are those involving port reads or writes.

The legend used in the presentation of the information here is given below.

A	Accumulator	BC	register pair BC
B	register B	DE	register pair DE
C	register C	HL	register pair HL
D	register D	SP	stack pointer
E	register E	PC	program counter
H	register H	IX	index register X
L	register L	IY	index register Y
R	refresh register	I	interrupt register
M	(HL)		
n	single byte	nn	double byte (word)
r	single register	rr	register pair
d	single byte displacment	addr	double byte value
(nn)	pointer to location nn	$	(PC) (i.e. current address)

The first part of this table gives the number of T-states for each of the "8080-compatible" instructions, that is, those 78 instructions for the Z-80 which are a subset of the 8080 instruction set.

The second part of the table gives the same information for the "Z-80-only" instructions, the eighty additional instructions implemented on the Z-80 but not the 8080.

The first column of the table states the mnemonics and syntax for the 8080-compatible instructions as recognized by Digital Research utility programs, such as DDT or MAC. The second column lists the equivalent mnemonis used by Zilog, Inc., in their own literature and programs for the Z-80.

TABLE XII INSTRUCTION CYCLE TIMING (continued)

The third column gives a breakdown of the number of T-states, including wait states, in each machine cycle used in the execution of an instruction. For example, the entry "448" indicates that the first machine cycle entered when executing that instruction will be 4 T-states long (including wait states), the second will be 4, and the last will be 8. When two figures appear in brackets, it indicates the the second figure is an internal operation, and that it is added to the preceeding cycle before extending the combination to mod 4. If execution of a cycle is dependent upon a flag or the outcome of a test, then the number of states (again including wait states) for that cycle are indicated in parentheses, along with the condition for entering the cycle.

The last column shows the total number of T-states for the instruction, with the number of conditional states shown separately following the "+" sign. To calculate the execution time of an instruction, multiply the total number of T-states by the length of one T-state. A T-state is one cycle of the Z-80B clock, which is 167.6 ns on 60Hz models, and 161.3 ns on 50Hz models. When conditional T-states are involved, the execution time will depend on whether the conditional machine cycles are executed.

TABLE XII INSTRUCTION CYCLE TIMING (continued)

Part 1. 8080-compatible instructions

Digital Research mnemonic		Zilog mnemonic		# states by cycle	Total
ACI	nn	ADC	A,n	44	8
ADC	m	ADC	A,(HL)	44	8
ADC	r	ADC	A,r	4	4
ADD	m	ADD	A(HL)	44	8
ADD	R	ADD	A,R	4	4
ADI	NN	ADD	A,N	44	8
ANA	M	AND	(HL)	44	8
ANA	R	AND	R	4	4
ANI	NN	AND	N	44	8
CALL	ADDR	CALL	NN	44444	20
CC	ADDR	CALL	C,NN	444(+44 if C=1)	12+8
CM	ADDR	CALL	M,NN	444(+44 if M=1)	12+8
CMA		CPL		4	4
CMC		CCF		4	4
CMP	M	CP	(HL)	44	8
CMP	R	CP	R	4	4
CNC	ADDR	CALL	NC,NN	444(+44 if C=0)	12+8
CNZ	ADDR	CALL	NZ,NN	444(+44 if Z=0)	12+8
CP	ADDR	CALL	P,NN	444(+44 if M=0)	12+8
CPE	ADDR	CALL	PE,NN	444(+44 if P=1)	12+8
CPI	NN	CP	N	44	8
CPO	ADDR	CALL	PO,NN	444(+44 if P=0)	12+8
CZ	ADDR	CALL	Z,NN	444(+44 if Z=1)	12+8
DAA		DAA		4	4
DAD	B	ADD	HL,BC	444	12
DAD	D	ADD	HL,DE	444	12
DAD	H	ADD	HL,HL	444	12
DAD	SP	ADD	HL,SP	444	12
DCR	M	DEC	(HL)	444	12
DCR	R	DEC	R	4	4
DCX	B	DEC	BC	8	8
DCX	D	DEC	DE	8	8
DCX	H	DEC	HL	8	8
DCX	SP	DEC	SP	8	8
DI		DI		4	4
EI		EI		4	4
HALT		HLT		4	4

Part 1. 8080-compatible instructions (continued)

Digital Research mnemonic		Zilog mnemonic		# states by cycle	Total
IN	N	IN	A,(N)	448	16
INR	M	INC	(HL)	444	12
INR	R	INC	R	4	4
INX	B	INC	BC	8	8
INX	D	INC	DE	8	8
INX	H	INC	HL	8	8
INX	SP	INC	SP	8	8
JC	ADDR	JP	C,NN	444	12
JM	ADDR	JP	M,NN	444	12
JMP	ADDR	JP	NN	444	12
JNC	ADDR	JP	NC,NN	444	12
JNZ	ADDR	JP	NZ,NN	444	12
JP	ADDR	JP	P,NN	444	12
JPE	ADDR	JP	PE,NN	444	12
JPO	ADDR	JP	PO,NN	444	12
JZ	ADDR	JP	Z,NN	444	12
LDA	ADDR	LD	A,(NN)	4444	16
LDAX	B	LD	A,(BC)	44	8
LDAX	D	LD	A,(DE)	44	8
LHLD	ADDR	LD	HL,(NN)	44444	20
LXI	B,NNNN	LD	BC,NN	444	12
LXI	D,NNNN	LD	DE,NN	444	12
LXI	H,NNNN	LD	HL,NN	444	12
LXI	SP,NNNN	LD	SP,NN	444	12
MOV	M,R	LD	(HL),R	44	8
MOV	R,M	LD	R,(HL)	44	8
MOV	R1,R2	LD	R1,R2	4	4
MVI	M,N	LD	(HL),N	444	12
MVI	R,N	LD	R,N	44	8
NOP		NOP		4	4
ORA	M	OR	(HL)	44	8
ORA	R	OR	R	4	4
ORI	N	OR	N	44	8
OUT	N	OUT	(N),A	448	16
PCHL		JP	(HL)	4	4
POP	B	POP	BC	444	12
POP	D	POP	DE	444	12
POP	H	POP	HL	444	12
POP	PSW	POP	AF	444	12
PUSH	B	PUSH	BC	844	16
PUSH	D	PUSH	DE	844	16
PUSH	H	PUSH	HL	844	16
PUSH	PSW	PUSH	AF	844	16

Part 1. 8080-compatible instructions (continued)

Digital Research mnemonic		Zilog mnemonic		# states by cycle	Total
RAL		RLA		4	4
RAR		RRA		4	4
RC		RET	C	8(+44 if C=1)	8+8
RET		RET		444	12
RLC		RLCA		4	4
RM		RET	M	8(+44 if S=1)	8+8
RNC		RET	NC	8(+44 if C=0)	8+8
RNZ		RET	NZ	8(+44 if Z=0)	8+8
RP		RET	P	8(+44 if S=0)	8+8
RPE		RET	PE	8(+44 if P=1)	8+8
RPO		RET	PO	8(+44 if P=0)	8+8
RRC		RRCA		4	4
RST		RST	P	844	16
RZ		RET	Z	8(+44 if Z=0)	8+8
SBB	M	SBC	A,(HL)	44	8
SBB	R	SBC	A,R	4	4
SBI	N	SBC	A,N	44	8
SHLD	ADDR	LD	(NN),HL	44444	20
SPHL		LD	SP,HL	8	8
STA	ADDR	LD	(NN),A	44	8
STAX	B	LD	(BC),A	44	8
STAX	D	LD	(DE),A	44	8
STC		SCF		4	4
SUB	M	SUB	(HL)	44	8
SUB	R	SUB	R	4	4
SUI	NN	SUB	N	44	8
XCHG		EX	DE,HL	4	4
XRA	M	XOR	(HL)	44	8
XRA	R	XOR	R	4	4
XRI	NN	XOR	N	44	8
XTHL		EX	(SP),HL	44448	24

Part 2. Z-80-only instructions

Digital Research mnemonic (Z80.LIB)	Zilog mnemonic	# states by cycle	Total
LDX r,d	LD r,(IX+d)	44(35)4	20
Load register from indexed memory (with IX)			
LDY r,d	LD r,(IY+d)	44(35)4	20
Load register from indexed memory (with IY)			
STX r,d	LD (IX+d),r	44(35)4	20
Store register to indexed memory (with IX)			
STY r,d	LD (IY+d),r	44(35)4	20
Store register to indexed memory (with IY)			
MVIX nn,d	LD (IX+d),nn	44(35)4	20
Move immediate to indexed memory (with IX)			
MVIY nn,d	LD (IY+d),nn	44(35)4	20
Move immediate to indexed memory (with IY)			
LDAI	LD A,I	48	12
Move I to A			
LDAR	LD A,R	48	12
Move R to A			
STAI	LD I,A	48	12
Move A to I			
STAR	LD R,A	48	12
Move A to R			
LXIX nnnn	LD IX,nnnn	4444	16
Load IX immediate (16 bits)			
LXIY nnnn	LD IY,nnnn	4444	16
Load IY immediate (16 bits)			
LBCD nnnn	LD BC,(nnnn)	444444	24
Load BC direct (from memory at nnnn)			
LDED nnnn	LD DE,(nnnn)	444444	24
Load DE direct			
LSPD nnnn	LD SP,(nnnn)	444444	24
Load SP direct			
LIXD nnnn	LD IX,(nnnn)	444444	24
Load IX direct			
LIYD nnnn	LD IY,(nnnn)	444444	24
Load IY direct			
SBCD nnnn	LD (nnnn),BC	444444	24
Store BC direct (to memory at nnnn)			
SDED nnnn	LD (nnnn),DE	444444	24
Store DE direct			
SSPD nnnn	LD (nnnn),SP	444444	24
Store SP direct			
SIXD nnnn	LD (nnnn),IX	444444	24
Store IX direct			
SIYD nnnn	LD (nnnn),IY	444444	24
Store IY direct			

Part 2. Z-80-only instructions (continued)

Digital Research mnemonic (Z80.LIB)	Zilog mnemonic		# states by cycle	Total
SPIX	LD	SP,IX	48	12
Copy IX to the SP				
SPIY	LD	SP,IY	48	12
Copy IY to the SP				
PUSHIX	PUSH	IX	4844	20
Push IX into the stack				
PUSHIY	PUSH	IY	4844	20
Push IY into the stack				
POPIX	POP	IX	4444	16
Pop IX from the stack				
POPIY	POP	IY	4444	16
Pop IY from the stack				
EXAF	EX	AF,AF'	4	4
Exchange AF and the alternate, AF'				
EXX	EXX		4	4
Exchange BC DE HL with BC' DE' HL'				
XTIX	EX	(SP),IX	444448	28
Exchange IX with the top of the stack				
XTIY	EX	(SP),IY	444448	28
Exchange IY with the top of the stack				
LDI	LDI		4448	20
Move m(HL) to m(DE), increment DE and HL, decrement				
LDIR	LDIR		4448(+4 if B≠0)	20+4
Repeat LDI until BC = 0				
LDD	LDD		4448	20
Move m(HL) to m(DE), decrement HL, DE, and BC				
LDDR	LDDR		4448(+4 if B≠0)	20+4
Repeat LDD until BC = 0				
CCI	CPI		4448	20
Compare A with m(HL), increment HL, decrement BC				
CCIR	CPIR		4448(+4 if B≠0)	20+4
Repeat CCI until BC = 0 or A = m(HL)				
CCD	CPD		4448	20
Compare A with m(HL), decrement HL and BC				
CCDR	CPDR		4448(+4 if B≠0)	20+4
Repeat CCD until BC = 0 or A = m(HL)				

Part 2. Z-80-only instructions (continued)

Digital Research mnemonic (Z80.LIB)	Zilog mnemonic	# states by cycle	Total
ADDX d	ADD (IX+d)	44(35)4	20
Indexed add to A			
ADDY d	ADD (IY+d)	44(35)4	20
Indexed add to A			
ADCX d	ADC (IX+d)	44(35)4	20
Indexed add with carry			
ADCY d	ADC (IY+d)	44(35)4	20
Indexed add with carry			
SUBX d	SUB (IX+d)	44(35)4	20
Indexed subtract			
SUBY d	SUB (IY+d)	44(35)4	20
Indexed Subtract			
SBCX d	SBC (IX+d)	44(35)4	20
Indexed subtract with "borrow"			
SBCY d	SBC (IY+d)	44(35)4	20
Indexed subtract with borrow			
ANDX d	AND (IX+d)	44(35)4	20
Indexed logical and			
ANDY d	AND (IY+d)	44(35)4	20
Indexed logical and			
XORX d	XOR (IX+d)	44(35)4	20
Indexed logical exclusive or			
XORY d	XOR (IY+d)	44(35)4	20
Indexed logical exclusive or			
ORX d	OR (IX+d)	44(35)4	20
Indexed logical or			
ORY d	OR (IY+d)	44(35)4	20
Indexed logical or			
CMPX d	CP (IX+d)	44(35)4	20
Indexed compare			
CMPY d	CP (IY+d)	44(35)4	20
Index compare			
INRX d	INC (IX+d)	44(35)4	20
Increment memory at m(IX+d)			
INRY d	INC (IY+d)	44(35)44	24
Increment memory at m(IY+d)			
DCRX d	INC (IX+d)	44(35)44	24
Decrement memory at m(IX+d)			
DCRY d	DEC (IY+d)	44(35)44	24
Decrement memory at m(IX+d)			

Part 2. Z-80-only instructions (continued)

Digital Research mnemonic (Z80.LIB)	Zilog mnemonic	# states by cycle	Total
NEG	NEG	44	8
Negate A (two's complement)			
IM0	IM0	44	8
Set interrupt mode 0			
IM1	IM1	44	8
Set interrupt mode 1			
IM2	IM2	4	4
Set interrupt mode 2			
DADC rr	ADC HL,rr	4444	16
Add with carry rr to HL			
DSBC rr	SBC HL,rr	4444	16
Subtract with "borrow" rr from HL			
DADX rr	ADD IX,rr	4444	16
Add rr to IX (rr may be BC, DE, SP, IX)			
DADY rr	ADD IY,rr	4444	16
Add rr to IY (rr may be BC, DE, SP, IY)			
INXIX	INC IX	48	12
Increment IX			
INXIY	INC IY	48	12
Increment IY			
DCXIX	DEC IX	48	12
Decrement IX			
DCXIY	DEC IY	48	12
Decrement IY			
BIT b,r	BIT b,r	44	8
Test bit b in register r			
SETB b,r	SET b,r	44	8
Set bit b in register r			
RES b,r	RES b,r	44	8
Reset bit b in register r			
BITX b,d	BIT b,(IX+d)	44(35)4	20
Test bit b in memory at m(IX+d)			
BITY b,d	BIT b,(IY+d)	44(35)4	20
Test bit b in memory at m(IY+d)			
SETX b,d	SET b,(IX+d)	44(35)44	24
Set bit b in memory at m(IX+d)			
SETY b,d	SET b,(IY+d)	44(35)44	24
Set bit b in memory at m(IY+d)			
RESX b,d	RES b,(IX+d)	44(35)44	24
Reset bit b in memory at m(IX+d)			
RESY b,d	RES b,(IY+d)	44(35)44	24
Reset bit b in memory at m(IY+d)			

Part 2. Z-80-only instructions (continued)

Digital Research mnemonic (Z80.LIB)	Zilog mnemonic	# states by cycle	Total
JR addr	JR addr-$	4(35)	8
Jump relative unconditional			
JRC addr	JR C,addr-$	44(+4 if C=1)	8+4
Jump relative if Carry indicator true			
JRNC addr	JR NC,addr-$	44(+4 if C=0)	8+4
Jump relative if Carry indicator false			
JRZ addr	JR Z,addr-$	44(+4 if Z=1)	8+4
Jump relative if Zero indicator true			
JRNZ addr	JR NZ,addr-$	44(+4 if Z=0)	8+4
Jump relative if Zero indicator false			
DJNZ addr	DJNZ addr-$	84(+4 if Z=0)	12+4
Decrement B, jump relative if non-zero			
PCIX	JMP (IX)	44	8
Jump to address in IX ie, Load PC from IX			
PCIY	JMP (IY)	44	8
Jump to address in IY			
RETI	RETI	4444	16
Return from interrupt			
RETN	RETN	4444	16
Return from non-maskable interrupt			
INP r	IN r,(C)	448	16
Input from port C to register r			
OUTP r	OUT (C),r	448	16
Output from register r to port (C)			
INI	INI	4884	24
Input from port (C) to m(HL), increment HL, decrement B			
INIR	INIR	4884(+4 if B≠0)	24+4
Repeat INI until B=0			
OUTI	OUTI	4848	24
Output from m(HL) to port (C), increment HL, decrement B			
OUTIR	OTIR	4848(+8 if B≠0)	24+4
Repeat OUTI until B = 0			
IND	IND	4884	24
Input from port (C) to m(HL), decrement HL & B			
INDR	INDR	4884(+4 if B≠0)	24+4
Repeat IND until B = 0			
OUTD	OUTD	4848	24
Output from m(HL) to port (C), decrement HL & B			
OUTDR	OTDR	4848(+8 if B≠0)	
Repeat OUTD until B = 0			

Part 2. Z-80-only instructions (continued)

Digital Research mnemonic (Z80.LIB)	Zilog mnemonic	# states by cycle	Total
RLCR r	RLC r	44	8
Rotate left circular register			
RLCX d	RLC (IX+d)	44(35)44	24
Rotate left circular indexed memory			
RLCY d	RLC (IY+d)	44(35)44	24
Rotate left circular indexed memory			
RALR r	RL r	44	8
Rotate left arithmetic register			
RALX d	RL (IX+d)	44(35)44	24
Rotate left arithmetic indexed memory			
RALY d	RL (IY+d)	44(35)44	24
Rotate left arithmetic indexed memory			
RRCR r	RRC r	44	8
Rotate right circular register			
RRCX d	RRC (IX+d)	44(35)44	24
Rotate right circular indexed			
RRCY d	RRC (IY+d)	44(35)44	24
Rotate right circular indexed			
RARR r	RR r	44	8
Rotate right arithmetic register			
RARX d	RR (IX+d)	44(35)44	24
Rotate right arithmetic indexed memory			
RARY d	RR (IY+d)	44(35)44	24
Rotate right arithmetic indexed memory			
SLAR r	SLA r	44	8
Shift left register			
SLAX d	SLA (IX+d)	44(35)44	24
Shift left indexed memory			
SLAY d	SLA (IY+d)	44(35)44	24
Shift left indexed memory			
SRAR r	SRA r	44	8
Shift right arithmetic register			
SRAX d	SRA (IX+d)	44(35)44	24
Shift right arithmetic indexed memory			
SRAY d	SRA (IY+d)	44(35)44	24
Shift right arithmetic indexed memory			
SRLR r	SRL r	44	8
Shift right logical register			
SRLX d	SRL (IX+d)	44(35)44	24
Shift right logical indexed memory			
SRLY d	SRL (IY+d)	44(35)44	24
Shift right logical indexed memory			
RLD	RLD	44444	20
Rotate left digit			
RRD	RRD	44444	20
Rotate right digit			