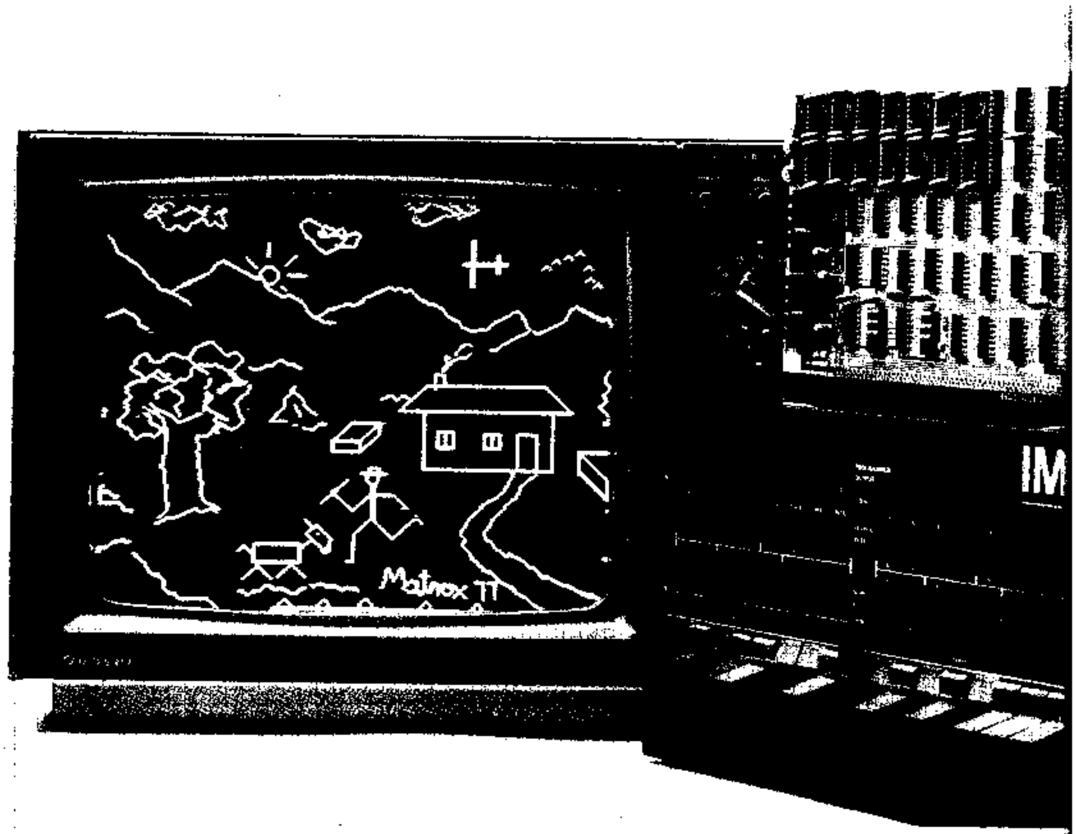
matrox
electronic systems ltd.
5800 ANDOVER AVE., T.M.R., QUE., H4T 1H4, CANADA
TEL.: 514—735-1182 TELEX: 05-825651

ALT - 256 * * 2

MTX TV CRT CONTROLLER FAMILY

GRAPHIC DISPLAY



* MATROX products covered by Canadian and foreign patent and/or patent pending.

INTRODUCTION:

The Matrox ALT-256**2 board is a fully tested, assembled and burned-in interface card which provides capability for a complete graphic system at a fraction of the cost of any other commercial graphic system. The card contains all interface electronics, a TV sync generator, and its own 65,536 X 1 bit refresh memory. It plugs directly into one slot of any S-100 bus compatible computer. The built in refresh memory allows much greater flexibility and speed since no CPU time is required to refresh the screen.

The output is a composite video signal which can be connected to any TV monitor or the video portion of a TV set. The unit produces a high resolution 256 X 256 dot raster. The complete screen can be cleared or preset by a single instruction.

The ALT-256**2 board occupies a single S-100 bus slot and requires 4 output ports and 1 input port (port address is selectable on the card with jumpers).

FUNCTIONAL DESCRIPTION:

Four output ports are used for loading of data into the display refresh memory. Assuming that output ports ØØ to Ø3 are selected (which also gives ØØ for the input port) the addressing is as follows:

The X coordinate of a given dot is loaded by outputting an 8 bit coordinate to the port H "Ø1. (instruction: OUT Ø1). The Y coordinate is outputted by OUT Ø2. These two instructions will set the cursor at the desired dot address.

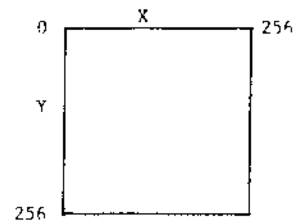
After the dot address is loaded, the dot intensity is loaded by outputting data to port H "ØØ". (OUT ØØ). Data H "ØØ" will result in a black dot; H "Ø1" will display a white dot. After the dot intensity is loaded, the ALT-256**2 will require 3.4 µsec to write the dot in the refresh memory. This is necessary to allow for internal synchronisation of the write operation, TV read scan and dynamic memory refresh. Since the CPU almost always requires more than 3.4 µsec to load the next dot address and data, the CPU can run at its full speed. Also note that regardless of the speed at which the CPU is accessing the ALT-256**2, there will be no flashes or streaks on the display, since access is internally synchronised. This results in a truly professional display. In other words, all the above three commands can be executed at full CPU speed with no image degradation.

Assuming port select bits are programmed as:

A7	A6	A5	A4	A3	A2	A1	A0
Ø	Ø	Ø	Ø	Ø	Ø	X	X

X, Y Ports

OUT Ø1 - X address
OUT Ø2 - Y address



DOT write port

OUT ØØ - Dot intensity

DØ = Ø black dot
DØ = 1 white dot
D7-D1 don't care

ERASE port

OUT Ø3 - screen clear

DØ = Ø all dots balck
DØ = 1 all dots white
D7-D1 don't care

STATUS port

IN ØØ - status

DØ = Ø display ready
DØ = 1 display being erased
D1 = Ø video portion
D1 = 1 vertical blank
D7-D2 don't care

TABLE 1: I/O PORT ADDRESSES AND FUNCTIONS OF THE ALT-256**2

The whole screen can be easily cleared by outputting H "ØØ" to the output port H "Ø3". This will result in the entire screen being black. Outputting H "Ø1" will result in all white since all 65,536 bits of the refresh memory will be simultaneously loaded with 1. This operation will require one TV frame time (33 msec max).

After the CPU outputs data to the port H "Ø3", an internal flag is set for between 16 msec to 33 msec until the screen is cleared. The testing of the flag is accomplished by inputting data from the input port H "ØØ". (INØØ). If the data bit DØ is high then the ALT-256**2 is busy; if DØ is low the ALT-256**2 is ready to accept new data.

Input port H "ØØ" provides additional information about the position of the electronic beam, (vertical blank) which is useful for dynamic motion display. (see table).

THEORY OF OPERATION:

The ALT-256**2 has four major blocks: the TV sync generator, scanning circuitry, cursor and interface electronics and 65,536 X 1 memory. (see Fig. 3 & 4) The sync generator is formed of x-tal oscillator (A 31) and a divider chain (A24, 16, 23, 6). This divider chain produces all timing signals for the memory scanning as well as horizontal and vertical sync. The TV sync generator can be programmed by jumpers for the European or American TV standard.

The scanning circuitry consists of multiplexers (A4, 12, 21, 20, 25, 26) which provide proper address and R/W signals for the RAM required.

The cursor consists of two 8 bit latches which are loaded by the CPU. A10 and A11 are X address register, A2 and A3 , Y address register. Necessary interface address and timing decoding is accomplished by A1, A9 and A51 to A55.

The refresh memory has 16, 4K dynamic memories (4096, 16 pin) organised as a 65,536 X 1 bit memory (A32-A47).

Power supplies for 5V, 600 mA; 12V, 100 mA and -5V, 10 mA are generated by A56, A57 and CR1: on board voltage regulators.

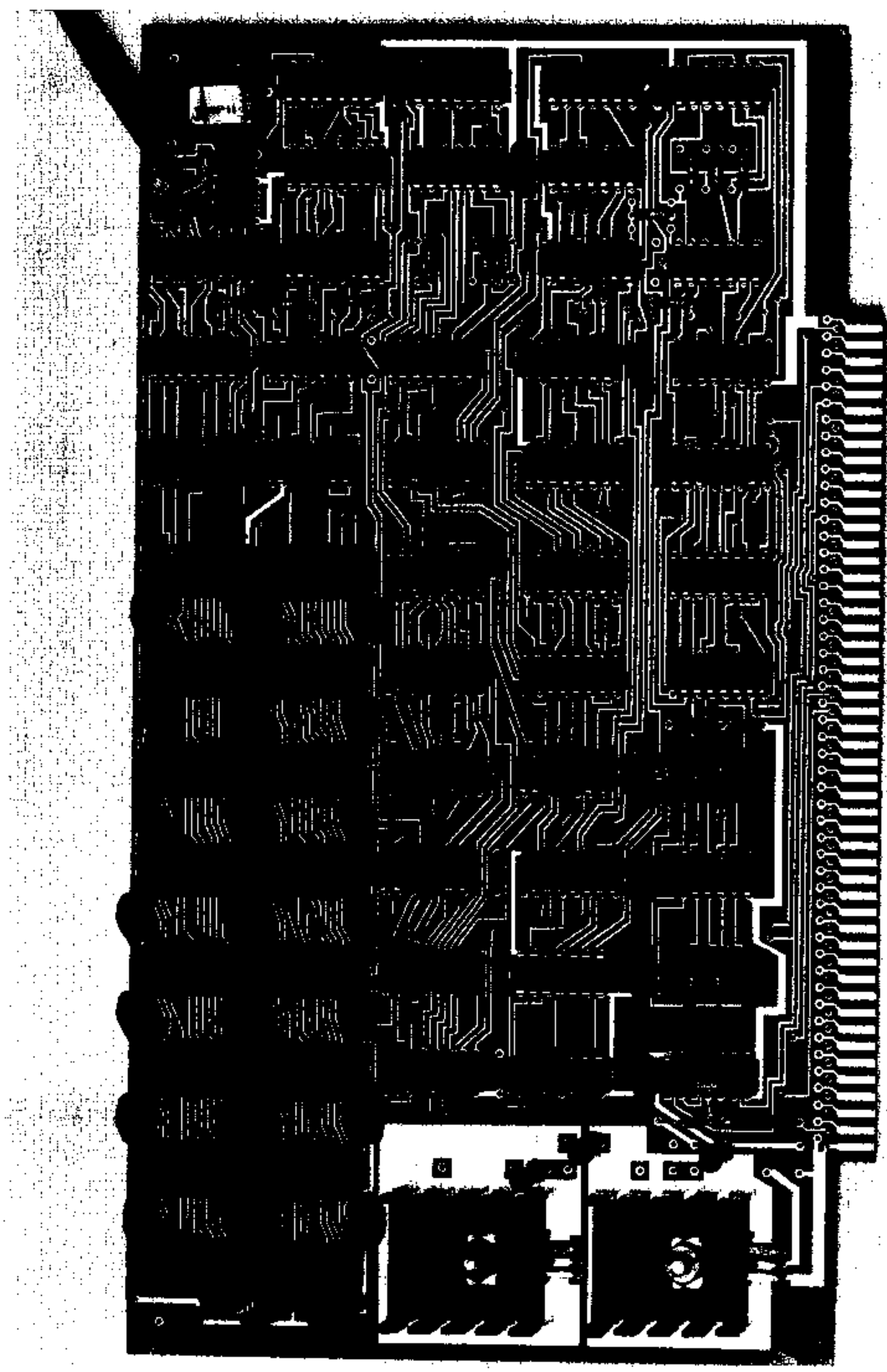


FIG. 1: PHOTOGRAPH OF THE ALT-256★★2

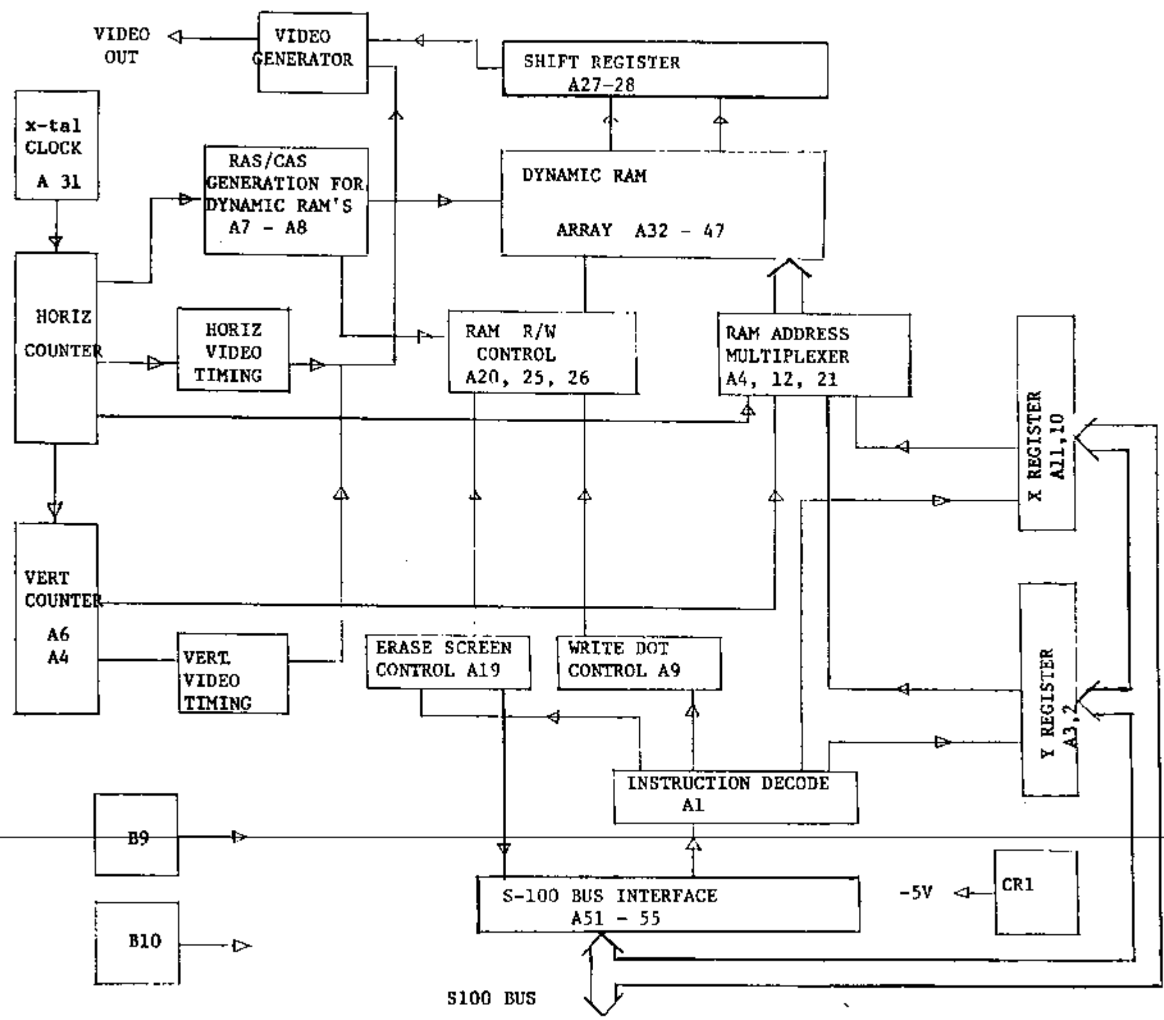


FIG. 4: BLOCK DIAGRAM FOR ALT-256**2

MAINTENANCE AND WARRANTY:

The ALT-256**2 is a fairly complex card and to understand its operation requires extensive knowledge of TV scanning, 4K dynamic memories and hardware. The complete circuit and assembly schematics are supplied to allow a competent user to troubleshoot the board if necessary. However, each board is fully tested, assembled and burned in for 24 hrs. before shipping to ensure reliability. In case of trouble, a warranty is provided.

Matrox products are warranted against defects in materials and workmanship for a period of 3 months from date of delivery. We will repair or replace products which prove to be defective during the warranty period, provided they are returned to Matrox Electronic Systems Ltd. No other warranty is expressed or implied. We are not liable for consequential damages.

Non-warranty repairs are billed at a minimum of $50 and a maximum of $100 according to time and materials required.

JUMPER OPTIONS:

The board is normally programmed for different options before shipping as specified in the purchase order. There are two things which can be programmed.

The TV sync generator can be programmed for one of 3 options. The American Standard option (60 Hz) has a 240 line X 256 dot raster (240 visible horizontal lines). The American Non-standard (60 Hz) version has a full 256 line X 256 dot raster with horizontal frequency of 16.8 KHz for a total of 280 lines. Some adjustment of the horizontal hold on the TV monitor may be necessary to allow the TV to lock. The European standard option has 312 lines (50 Hz).

JUMPERS IN:	ANS	W1 W4 W5 W9 W10 W12 W14 W17 W19 W21 W22
	AS	W2 W4 W6 W8 W11 W13 W15 W17 W18 W20 W23
	ES	W1 W3 W5 W7 W11 W12 W14 W16 W19 W20 W22

TABLE 2: TV STANDARD JUMPER OPTION

The four output ports and one input port can be located on any 4 location boundary in the 256 port address space. Address bits A2 to A7 can be selected as follows:

		JUMPER ACCROSS PINS	
ADDRESS BIT =		Ø	1
	A2	1 and 16	2 and 15
A49	A3	3,14	4,13
SOCKET	A4	5,12	6,11
	A5	1,16	2,15
A50	A6	3,14	4,13
SOCKET	A7	5,12	6,11

x jumper can be any resistor between Ø to 51 Ohm

TABLE 3: I/O ADDRESSING JUMPER OPTIONS

INSTALLATION AND TESTING:

To install the ALT-256**2, switch the Altair/Imsai 8080 type computer off and plug the card in. Connect the composite video output to the input of the TV monitor and switch the TV monitor and computer on. The display will be a random pattern of the content of the refresh memory.

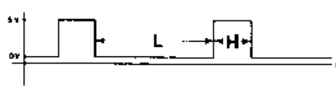
To help in testing the card a short test program listing (see Table 4) is provided. The program makes the following assumptions: The ALT-256**2 is strapped to begin its output ports at location H "ØØ". The computer front panel has input port switch register at location H "FF". If the address strapping of the ALT-256**2 or front panel ports is different from that specified, the program can be easily modified.

To test the card, toggle into memory the short test program (see Table 4). The starting address is H "Ø100". The program will first clear the entire screen depending on the setting of the input port switch (SWO). A setting of H "ØØ" will produce a black screen, H "01" will produce a white screen.

Following this, the program clears the X and Y register and writes SWO (up = 1; down = 0) into dot ØØ. (Upper left corner). Then address X is incremented, SWO is written into the next dot etc, continuously. By changing the SWO position during the computer scan, a white or black dot is written. The speed of the computer scan can be varied by the position of the switches S7-S1 on the front panel.

VIDEO SIGNALS

Composite video signal. Output impedance 75 Ohms. Short circuit protection built in.



Horizontal and Vertical Sync signals.

SIGNAL	FREQ.	HIGH	LOW	STD.
GH	16.8kHz	4.5	55	ANS
Horizontal	15.7kHz	5.8	57.8	AS
Sync	15.6kHz	4.5	59.6	ES
SV	60 Hz	238	16.43	ANS
Vertical	60 Hz	254	16.42	AS
Sync	50 Hz	256	19.74	ES
DTC	7.0536 mHz			ANS
Dot	5.5334 mHz			AS
Clock	7.0536 mHz			ES

ANS American nonstandard (280 lines, 60 Hz)
AS American standard (262 lines, 60 Hz)
ES European standard (312 lines, 50 Hz)

ADDRESS	CONTENT (Hex)	MNEMONIC	COMMENT
0100	DB	IN FF	/input front panel switch
0101	FF		
102	D3	OUT Ø3	/input SW to erase port
03	Ø3		
04	DB	LUP 1, IN ØØ	/input ALT-256**2 status
05	ØØ		
06	E6	ANI Ø1	/mask SWØ
07	Ø1		
08	C2	JNZ LUP1	/test for busy (Not Ø)
09	Ø4		
0A	Ø1		
0B	21	LXIH ØØ	/load H, L with Ø
0C	ØØ		
0D	ØØ		
0E	7D	LUP 2, MOV A, L	/move L to A
0F	D3	OUT Ø1	/output X coordinate
10	Ø1		
11	7C	MOV A, H	/move H to A
12	D3	OUT Ø2	/output Y coordinate
13	Ø2		
14	DB	IN FF	/input front panel switch
15	FF		
16	D3	OUT ØØ	/write dot to port ØØ
17	ØØ		
18	E6	SPEED, ANI FE	/mask switches S7-S1
19	FE		
1A	3C	LUP 3, INC A	/loop delay for speed
1B	C2	JN Z LUP 3	/test for zero
1C	1A		
1D	Ø1		
1E	23	INX H	/next dot
1F	C3	JMP LUP 2	/go back
20	0E		
21	Ø1		

TABLE 4 : TEST PROGRAM FOR ALT-256**2

COLOR/GREY SCALE*

Each ALT-256 card has a built-in crystal controlled sync generator. However, each card can also be synchronized to an external sync source. This feature is extremely useful for applications requiring more than one bit per pixel (grey scale or color). Multiple cards are required for this application (up to 24 cards can be synchronized).

VIDEO AND SYNC SIGNALS

All video and sync signals are available on a 16 pin plug-in socket at position S1. These signals can be used for a variety of applications:

PIN	NAME	COMMENT
12	VDO	Composite video output. Can be directly connected to a TV monitor video input via 75 ohm cable
4	ALPHA	Alphanumeric video input. (From MTX-1632SL for video mixing). Has to be low when not used. (Jumper W24 in).
13	SV	Vertical sync signal (Positive pulse). It can be used to drive a TV monitor vertical deflection circuit or it can drive an MTX-1632SL alphanumeric VRAM.
10	SH	Horizontal sync signal (positive pulse).
11	BV	Vertical blank. This signal is low during vertical retrace (3 msec). It is available to a uP by reading I/O port Ø.
7	BH	Horizontal blank (low during blank).
3	D OUT	Serial video signal (TTL level, high-white, low-black). It can drive directly a TTL compatible TV monitor video input or it can be used as one bit of video information in a color/grey scale system.
2	DOT CLOCK	Bidirectional dot clock input/output (depends on M/S control jumper W26). If W26 is in, ALT-256 is a master card and DOT CLOCK is an output. If W26 is out, ALT-256 is a slave card and DOT CLOCK is an input.
15	RESET H	Bidirectioanl horizontal reset input/output. (80 nsec negative pulse which synchronizes horizontal counters).

* For more information on color/grey scale applications consult Matrox's MTX-256 color/grey application note.

PIN	NAME	COMMENT
16	RESET V	Bidirectional vertical reset input/output. (80 nsec negative pulse) Synchronizes vertical counters.
1,8,9,16	GND	Ground

The following jumpers determine the use of the ALT-256 in color/grey scale applications.

W26 Determines if card will be a master or slave card (generates its own sync signals or accepts external sync signal from a master card). Sync signals are DOT CLOCK, Vertical reset and Horizontal reset.

W26 - in	ALT-256 is a master card
W26 - out	ALT-256 is a slave card

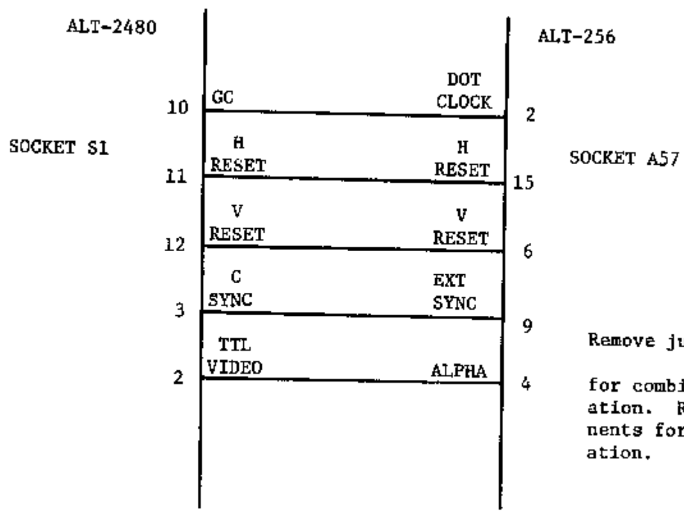
W25 Video data input bit connection.

W25 - in	Video data input bit is connected to DØ.
W25 - out	The user can connect video data bit to any of the data bus bits D0-D7 with a wire.

W24 Alpha input

W24 - in	The socket S1 pin 4 is grounded (no alpha)
W24 - out	Alphanumeric input at pin 4 of S1 is added to the graphic video.

NOTE: All ALT-256 are shipped with jumpers W24, W25, W26 in. (Standard configuration for a single level ALT-256 application).



Remove jumper W24,26, and 27 on 256 unit for combined 2480/256 operation. Restore these components for independent operation.

FIG. 5 _ COMBINING ALT-2480 and ALT-256

The ALT-2480 and -256 display cards are directly compatible with one another. This powerful feature permits generation of a combined alphanumeric/graphic display with no extra hardware. Multiple ALT-256 cards can also be slaved to a single ALT-2480 master for color/grey scale applications.

Figure 5 shows how the ALT-2480 and ALT-256 cards are connected together. The connection is accomplished via 16 pin DIP plugs on each card. Some jumpers must be altered on the ALT-256. The combined composite video output is taken from the ALT-256 output.

SUMMARY OF MTX-GRAPH
SOFTWARE PACKAGE

The MTX GRAPH software package is designed for use with the Matrox ALT-256 graphics display. The package is configured as a series of callable sub-routines. The MTX GRAPH package occupies memory locations 0104 to 04FF (Hex). The package incorporates the following features:

1. Variable Resolution: The display resolution can be selected to be 256 x 256; 128 x 128; or 64 x 64; by software command.

2. Point Plot: Any arbitrary point can be displayed by specifying X-Y coordinates. Dot size depends on the resolution selected.

3. Line Vector Graphics: Line can be drawn by specifying the two end points.

4. Alphanumeric Display: A full ASCII character generation routine is incorporated. Control characters are correctly interpreted. Characters can be positioned anywhere on the screen.

5. Animation Synchronization: This feature allows the user to generate animation synchronization at line rate.

5. Color Option: The software package will support a 3 card color/grey scale system.

More detail on the above is contained in the MTX GRAPH user manual.

A second program supplied is intended for demonstration purposes. The program utilizes the MTX GRAPH sub-routines to create a continuous live action graphics display.

The paper tapes are supplied in Imsai binary loader compatible format. A listing of the loader and instructions for its use are provided with the manual.

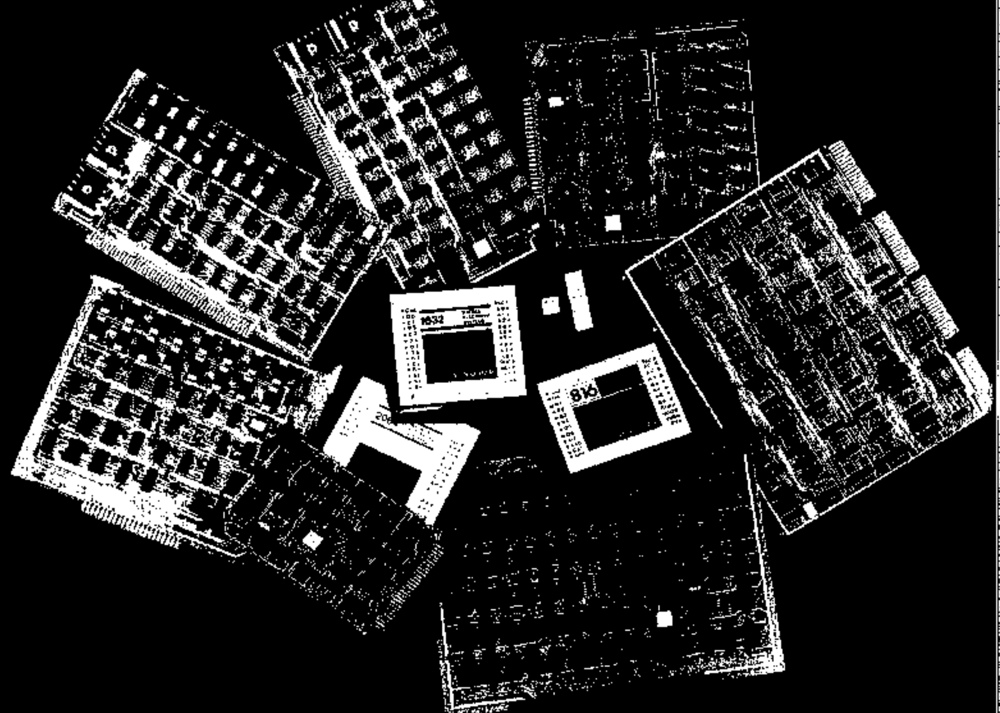
The Imsai loader should be used to load first MTX GRAPH and then the demo program. Both must be co-resident to use the demonstration program. The ALT-256 should be jumpered for address 10-13 (hex) and data bit DØ. (jumpers A2, 3, 5, 6, 7 set to Ø, A4 set to 1. See manual P9.) Start the computer at location 0500 Hex. Sit back and watch the show. The demonstration program will pause whenever the data switches are set to 01 (hex).

NOTE: Old versions of the ALT-256 require the following hardware change to work with this software package: (units shipped prior to Aug. 1/77). Tie A3 and A11 pin 4 to +5V with jumper wires soldered directly to the artwork.

PRICE: $25.00 for MTX GRAPH manual plus binary paper tapes of MTX GRAPH and demo program.

CATALOGUE – SF1

matrox
microprocessor displays



APRIL 78

matrox electronic systems ltd.

INTRODUCTION

Matrox Electronic Systems is a dynamic young electronics company. We have created a line of OEM display interface controllers that has grown along with the explosive micro-processor revolution. There are many companies specialized in data acquisition systems for microprocessors. Matrox is the only company specialized in display systems. We offer the most complete line of advanced CRT display controllers in the industry. The family of Matrox display controllers has been designed for maximum reliability, simplicity and lowest cost.

A majority of Matrox displays are designed for use with standard TV CRT monitors. Two main groups are alphanumeric and graphic video random access memories. A third type of display controller (the alpha chip) is designed for use with LED, flourescent, gas discharge, incandescent etc. 5 X 7 dot matrix or multisegment type displays.

The controllers are available in different forms such as general purpose monolithic integrated circuits, plug-in modules or PC boards for any uP, plug-in boards for specific computer bus or uP or stand alone display systems.

Matrox offers numerous models which can be used in various combinations. A wide choice of display formats, character sets, TV standard, external/internal syncs, resolutions bus compatibility etc. allow the OEM user to build a display for any application at the lowest possible cost in the minimum of time.

For applications requiring special custom designs, Matrox has the capability to design and deliver prototype and production quantity display controllers according to customer specifications in a relatively short time.

OEM users have the option to manufacture their own display controllers under Matrox licence. After the user buys 200 units, Matrox will supply all schematics, artworks, specifi-cations, and parts lists, for a flat one time charge. The OEM user can then use Matrox as a second source for his own production.

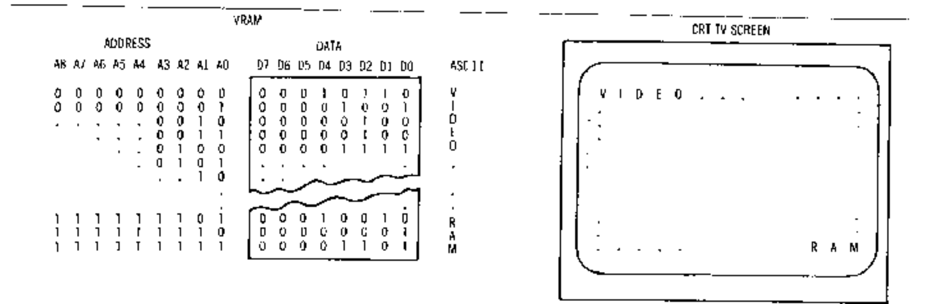
Matrox displays have been used in more than 10,000 installations in every imaginable application: from ground control displays for the Viking mission to Mars to hobby displays.

As an innovator in microcomputer displays, Matrox is fully committed to the design and manufacture of displays using the latest state of the art technology. We were the first to introduce VRAM concept to displays and we are the first to supply complete display controllers in a single chip.

Typical applications for Matrox OEM products include alphanumeric and graphics displays for sophisticated instruments and products such as scientific calculators, process control equipment, navigation equipment, medical instrumentation, industrial control, image processing, simulation, sophisticated video games, dumb or intelligent alpha-numeric and/or graphics terminals, etc. The potential for application of Matrox display products is virtually unlimited.

ALPHANUMERIC VIDEO RAM'S

On the input side, an alphanumeric VRAM looks like a static random access memory and it can be directly connected to the address and data bus of a typical microprocessor system. The output is a video signal that produces a display of alphanumeric data on a TV monitor. Each character position on the screen is equivalent to a memory location. It can be written into and read out the same way as any other RAM in the microprocessor address space. The content of a video memory location determines the character to be displayed. Various display effects can be accomplished with the availability of the read/write feature. The full microprocessor instruction set including all memory reference instructions may be used for display data manipulation at full speed.

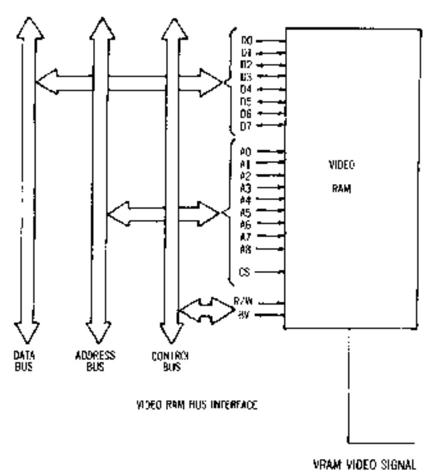


(Fig. 1) **VRAM AND CRT ORGANIZATION**

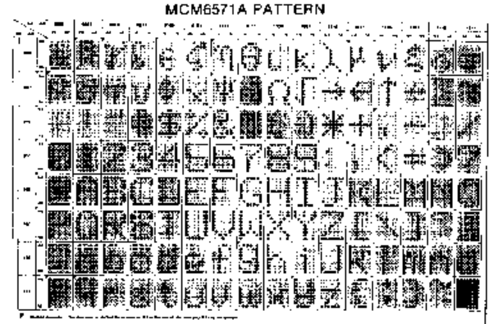
The VRAM bus interface shown in Figure 2 demonstrates the simplicity of interface to a typical bus system. Data, address and control lines can be connected without buffers. The CS (chip select) input is used to select the VRAM in microprocessor address space.

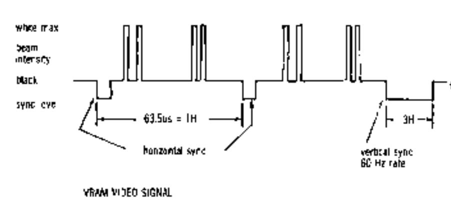
The CRT display is refreshed at a 60 Hz rate. However, there is no need for the CPU to refresh the VRAM; once written, a character is continuously displayed until a new character is rewritten into a location. This unique feature is a significant improvement over the commonly used DMA approach, since it requires no CPU time for refresh.

A VRAM produces a composite video signal which drives directly any standard TV monitor. A 75 ohm output impedance permits connection of a 75 ohm cable of up to 2,000 feet to drive up to 10 TV monitors.



MCM6571A PATTERN





(Fig. 2) **VRAM BUS INTERFACE**

MTX SERIES VRAM MODULES

The Matrox MTX series of VRAM TV CRT controller modules are designed for use in systems that require a display of alphanumeric data. The family is packaged in small self-contained modules to facilitate use as a component. It can be soldered or plugged in the user PCB. On the input side, an MTX VRAM looks like an ordinary 8 bit wide RAM and can be connected directly to the address and data bus of any bus organized system. MTX series VRAM modules are particularly suitable for use in microcomputer systems, due to their low cost, small size, modular packaging, single +5V power supply and ease of interface.

1. CRT DISPLAY CONTROLLERS (ALPHANUMERIC)

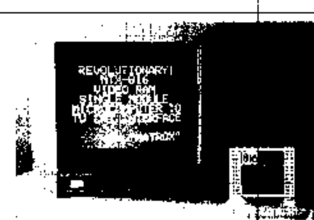
1.1 MODULES

MTX-816 — 8X16 VIDEO RANDOM ACCESS MEMORY

The MTX-816 is a TV CRT controller designed for use in systems that require display of alphanumeric data. On the input side the device is directly connected to bus organized systems and looks like a 128x8 RAM. The output is a video signal which directly drives a TV monitor to provide an 8x16 field of 128 ASCII characters.

- organized as 128x8 RAM
- 8x16 display field
- no external refresh
- bidirectional data bus
- CMOS/TTL compatible
- access time 1 usec
- flicker free display
- single +5V power supply
- low power 800 mW
- ASCII font standard (5x7)
- standard video output
- drives up to 25 TV monitors
- remote display
- electronic intensity control

Price $179/single $149/100



Dimensions 4" x 4.5" x .5" plug-in module

MTX-1632 — 16X32 VIDEO RANDOM ACCESS MEMORY

The MTX-1632 is a TV CRT controller designed for use in systems that require display of alphanumeric data. On the input side the device is directly connected to bus organized systems and looks like a 512x8 RAM. The output is a video signal which directly drives a TV monitor to provide a 16x32 field of 512 ASCII characters.

- organized as 512x8 RAM
- 16x32 display field
- no external refresh
- bidirectional data bus
- TTL compatible
- access time 550 ns
- flicker free display
- single +5V power supply
- low power
- ASCII font standard (7x9)
- standard video output
- drives up to 25 TV monitors
- character blinking
- electronic intensity control
- upper/lower case

Price: $225/single. $169/100

Dimensions: 4" x 4.5" x .5" plug-in module

MTX-1632SL — EXTERNALLY SYNCHRONIZED 16X32 VIDEO RANDOM ACCESS MEMORY

The MTX 1632SL is a TV CRT controller designed for use in systems that require display of alphanumeric data. On the input side the device is directly connected to bus organized systems and looks like a 512x8 RAM. The output is a video signal which directly drives a TV monitor to provide a 16x32 field of 512 ASCII characters. The device can be slave locked to an external source (TV sync. generator)

- external synchronization
- organized as 512x8 RAM
- 16x32 display field
- bidirectional data bus
- TTL compatible
- access time 550 ns
- broadcasting applications
- single +5V power supply
- ASCII font standard (7x9)
- standard video output
- drives up to 25 TV monitors
- character blinking
- upper/lower case

Price. $225/single. $169/100

Dimensions: 4" x 4.5" x .5" plug-in module

MTX-2064 — 20X64 VIDEO RANDOM ACCESS MEMORY

The MTX-2064 is a TV CRT controller designed for use in systems that require display of alphanumeric data. On the input side the device is directly connected to bus organized systems and looks like a 1280x8 RAM. The output is a video signal which directly drives a TV monitor to provide a 20x64 field of 1280 ASCII characters.

- organized as 1280x8 RAM
- 20x64 display field
- no external refresh
- bidirectional data bus
- TTL compatible
- access time 550 ns
- flicker free display
- single +5V power supply
- low power
- ASCII font standard (7x9)
- standard video output
- drives up to 25 TV monitors
- character blinking
- electronic intensity control
- upper/lower case

Price: $295/single $190/100

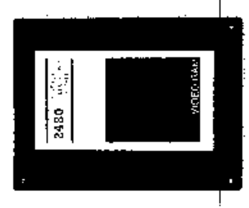
Dimensions: 4.5" x 6" x .5" plug-in module

MMD-2480 — 24X80 VIDEO RANDOM ACCESS MEMORY

The MMD-2480 is a TV CRT controller designed for use in systems that require display of alphanumeric data. On the input side the device is directly connected to bus organized systems and looks like a 4Kx8 RAM. The output is a video signal which directly drives a TV monitor to provide a 24x80 field of 1920 ASCII characters.

- organized as 4Kx8 RAM
- 24x80 display field
- no external refresh
- bidirectional data bus
- TTL compatible
- access time 500 ns
- flicker free display
- single +5V power supply
- external sync option
- ASCII font standard
- standard video output
- drives up to 25 TV monitors
- character blinking
- upper/lower case/graphics

Price: $395/single $290/100



Dimensions: 4.5" x 6" x .5" plug-in module

PLUG-IN PC BOARDS
ALPHANUMERIC CONTROLLERS

A series of plug-in alphanumeric CRT controller PC boards for most industry standard buses is available. This OEM display controller plugs-in directly into the computer bus and provides a video signal which directly drives a CRT monitor. All interface electronics, refresh memory and TV scanning is built-in. Each board has a variety of options and features which can be user programmed by jumpers. Matrox video boards allow the system designer to add display to his system in the shortest possible time at a very low cost. Software packages are also available.

MTX-1664SL — EXTERNALLY SYNCHRONIZED 16X48 OR 16X64 VIDEO RANDOM ACCESS MEMORY

The MTX-1664SL is a TV CRT controller designed for use in systems that require display of alphanumeric data. On the input side the device is directly connected to bus organized systems and looks like a 1024X8 RAM. The output is a video signal which directly drives a TV monitor to provide a 16X64 field of 1K ASCII characters. The device can be slave locked to an external source (TV sync generator).

- external synchronization
- organized as 1Kx8 RAM
- 16X64 display field or 16X48 display field
- bidirectional data bus
- TTL compatible
- access time 500 ns
- broadcasting applications
- single +5V power supply
- low power
- ASCII font standard (7x9)
- standard video output
- drives up to 25 TV monitors
- character blinking

Price: $295/single: $190/100

Dimensions: Prolog bus plug-in PCB; 4.5" x 6.5" PCB

MTX-2480 — 24X80 VIDEO RANDOM ACCESS MEMORY

The MTX-2480 is a TV CRT controller designed for use in systems that require display of alphanumeric data. On the input side the device is directly connected to bus organized systems and looks like a 4096x9 RAM. The output is a video signal which directly drives a TV monitor to provide a 24x80 field of 1920 ASCII characters.

- organized as 4096x9 RAM
- 24x80 display field
- no external refresh
- bidirectional data bus
- TTL compatible
- access time 500 ns
- flicker free display
- single +5V power supply
- low power
- ASCII font standard
- standard video output
- half intensity
- character blinking
- inverse video

Price: $395/single: $290/100

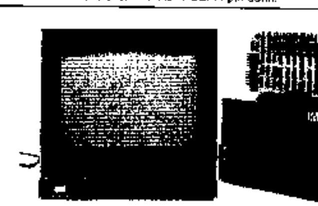
Dimensions: 7" x 7.5" PCB/44 pin conn.

ALT-2480 — 24X80 VIDEO RANDOM ACCESS MEMORY (S-100 BUS)

The ALT-2480 is a TV CRT controller designed especially for the industry standard S-100 bus. It is used in systems that require a display of alphanumeric data. On the input side, the device is directly connected to an S-100 bus organised system and looks like a 4096x8 static RAM with an access time of 500 ns. The output is a video signal which directly drives a TV monitor to provide a 24x80 field of 1920 ASCII characters. A software package, MTX-ALPHA, is available for use with the ALT-2480 to emulate an intelligent terminal.

- organised as 4096x8 RAM
- 24x80 display field
- no external refresh
- bidirectional data bus
- TTL compatible
- access time 500 ns
- flicker free display
- standard video output
- single +5V power supply
- low power
- ASCII font 5x7 or 7x9
- option of lower case characters
- character blinking
- inverse video
- drives up to 10 TV monitors
- can combine with the ALT-256 to produce a combined alphanumeric and graphic display

Price. $295/single: $265/100



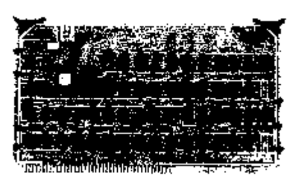
Standard S-100 bus size (5.3" x 10")

MSBC-2480 — 24X80 VIDEO RANDOM ACCESS MEMORY (SBC-80)

The MSBC-2480 is a TV CRT controller designed especially for the industry standard Intel SBC-80 bus. It is used in systems that require a display of alphanumeric data. On the input side, the device is directly connected to an SBC-80 bus organised system and looks like a 4096x8 static RAM with an access time of 500 ns. The output is a composite video signal which directly drives a TV monitor to provide a 24x80 field of 1920 ASCII characters. A software package, MTX-ALPHA, is available for use with the MSBC-2480 to emulate intelligent terminals (Lear Siegler ADM-3A and DEC DECSCOPE VT-52)

- plugs directly in SBC-80 bus
- 32 lines x 80 characters
- upper/lower case, graphics
- memory mapped (VRAM)
- built-in refresh memory
- user programmable character generator (2716 EPROM)
- external/internal sync
- looks like 4Kx8 RAM
- built-in ASCII keyboard interface
- normal/inverse/blink
- drives TV monitor directly
- software control
- single +5V power supply, 9A
- hardware scroll
- 500 nsec access time
- can be combined with MSBC 512x512 graphics

Price. $495/single: $350/100



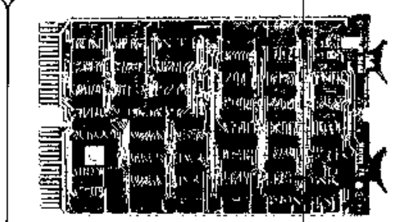
Standard SBC-80 size card (6.75" x 12")

MLSI-2480
24X80 VIDEO RANDOM ACCESS MEMORY (LSI-11 BUS)

The MLSI-2480 provides an alphanumeric video interface between an LSI-11 bus microcomputer and a TV monitor. It outputs the industry standard 24 line by 80 character display which is invaluable for professional applications such as an intelligent CRT terminal and word processor. The MLSI-2480 is compatible with the MLSI-512 graphics interface board, permitting a powerful combined alphanumeric/graphics display

- plugs directly in LSI-11 bus
- 24 lines X 80 charaters
- upper/lower case/graphics
- byte mapped (4K X 8)
- built-in R/W refresh memory
- user programmable character generator (2716 EPROM)
- full software control
- external/internal sync
- normal/inverse control
- drives TV monitor directly
- dual size
- 500 nsec access time
- can be combined with MLSI-512 X 512 graphics

Price: $495/single; $350/100 Standard Dual LSI-11 card

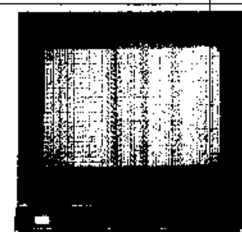


MDC-2480
24X80 VIDEO RANDOM ACCESS MEMORY (PDP-11 BUS)

The MDC-2480 provides an alphanumeric video interface between a PDP-11 bus microcomputer and a TV monitor. It outputs the industry standard 24 line X 80 character display which is invaluable for professional applications such as an intelligent CRT terminal and word processor. The MDC-2480 is compatible with the MDC-512 graphics alphanumeric/graphics display.

- plugs directly in PDP-11 bus
- 24 lines X 80 characters
- upper/lower case/graphics
- byte mapped (4K X 8)
- built-in R/W refresh memory
- user programmable character generator (2716 EPROM)
- full software control
- external/internal sync
- normal/inverse control
- drives TV monitor directly
- quad size
- 500 nsec access time
- single +5V, .9A
- can be combined with MDC-512 X 512 graphics

Price. $495/single; $350/100 Standard Quad PDP-11 card



GRAPHICS CRT CONTROLLERS

Each graphic VRAM has several registers of varying length. The registers are used to store parameters of the currently addressed dot such as X-Y coordinates, color or intensity, as well as commands such as clear display, scroll, vector plot, etc. A memory mapped I/O technique is used to address registers. This means that each register looks to the CPU like a RAM location. This feature allows extremely simple hardware and software interfacing since the graphic display can be interfaced to the CPU as a 4 or 8 location x 8 or 16 bit wide RAM. The use of an X-Y addressing scheme permits addressing up to 262,000 on board refresh memory bits using only two computer memory locations.

All Matrox graphic video RAM's are designed such that multiple units can be combined for color/grey scale applications. The Matrox graphic VRAM's are divided into two main subgroups: the 256 family and the 512 family. The 256 family is designed for lowest cost with a 256 x 256 dot matrix resolution. The 512 family incorporates a revolutionary variable resolution feature which permits user selection of 256 x 256; 256 x 512; 512 x 512; and 256 x 1025 dot matrix displays. The design is available for a number of popular buses.

2. CRT DISPLAY CONTROLLERS (GRAPHICS)

ALT-256
256X256 GRAPHIC DISPLAY CONTROLLER (S-100 BUS)

The ALT-256, directly plug compatible with the S-100 bus contains all interface electronics, a TV sync generator and its own 65,536 X 1 bit refresh memory. It outputs a composite video which can be connected to any TV monitor or the video portion of a TV set. The unit produces a high resolution 256 X 256 dot raster. The ALT-256 can combine with the ALT-2480 to produce a powerful combined alphanumeric/graphic display

A software package, MTX-GRAPH, is available for use with the ALT-256

- 256 X 256 dot raster
- each dot individually addressed
- refresh memory (64K X 1) built-in
- single instruction erase
- 3.4 us max/dot write time
- external/internal sync
- multiple boards stackable for color/grey scale applications
- powerful X-Y addressing
- American/European TV standard
- can be combined with ALT-2480
- low cost

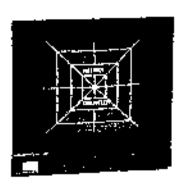
Price: $395/single $355/100 Standard S-100 bus size

MTX-256
256X256 GRAPHIC DISPLAY CONTROLLER (GENERAL PURPOSE)

The MTX-256**2 is a unique modular graphics system designed for direct interfacing with any mini or microcomputer bus. On the input side, the device looks like a 4 location X 8 bit static random access memory. The output is a composite video signal which directly drives commercial TV monitors to provide a 256 X 256 dot raster. Multiple boards can be used for color/grey scale applications. External/internal sync capability is provided. The graphics can be mixed with an alphanumeric VRAM (such as the MTX-1632SL or the MTX-2480, etc.) to obtain full alphanumeric/graphic capability.

- 256 X 256 dot raster
- directly interfaces to any microprocessor
- drives a standard TV monitor
- expansion to color/grey scale
- modular form
- individually addressed dots
- 3.6 usec cycle time/dot
- vector/point plot
- no external refresh
- erase screen capability
- graphics/alphanumerics
- flicker free display
- remote display
- American/European TV standard
- TTL compatible
- +5V, +12V power supply
- low power 5W
- low cost

Price $595/single $495/100 8.5" X 7", 44 pin dual connector



MSBC-512
VARIABLE RESOLUTION GRAPHICS CONTROLLER (SBC-80 BUS)

The MSBC-512 incorporates the revolutionary concept of variable resolution graphics on a single card. The MSBC-512 is directly plug-in compatible with the industry standard Intel SBC 80 bus. The same card can be user programmed to produce a dot matrix of 256 X 256, 256 X 512, 512 X 512, or 256 X 1024 points by using 4K, 8K or 16K plug-in compatible dynamic memories.

- variable resolutions
- multiple cards stackable for color/grey scale applications
- single command erase
- vertical scroll built-in
- external/internal sync
- can be combined with the alphanumeric MSBC-2480
- display memory read/write
- each dot individually addressed
- 1.4 μsec max/dot access time
- X-Y addressing
- American/European standard
- software package available

Price: $1395/single; $1150/100 512 x 512, 256 x 1024; $895/single; $650/100 256 x 256; $1095/single; $850/100 256 x 512

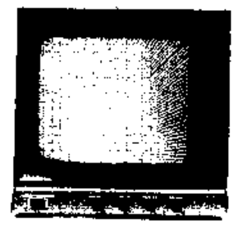
Standard SBC-80 size board (12" x 6.75")

MLSI-512
VARIABLE RESOLUTION GRAPHICS CONTROLLER (LSI-11 BUS)

The MLSI-512 incorporates the revolutionary concept of variable resolution graphics on a single card. The MLSI-512 is directly plug-in compatible with the industry standard LSI 11 bus. The same card can be user programmed to produce a dot matrix of 256 X 256, 256 X 512, 512 X 512, or 256 X 1024 points by using 4K, 8K or 16K plug-in compatible dynamic memories.

- variable resolutions
- multiple cards stackable for color/grey scale applications
- single command erase
- vertical scroll built-in
- external/internal sync
- can be combined with the alphanumeric MLSI-2480
- display memory read/write
- each dot individually addressed
- 1.4 usec max/dot access time
- X-Y addressing
- American/European standard
- software package available

Price: $1395/single; $1150/100 512 x 512, 256 x 1024; $895/single; $550/100 256 x 256; $1095/single; $850/100 256 x 512

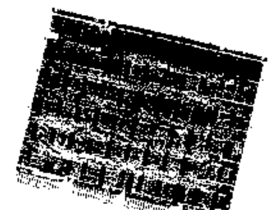

Standard LSI-11 size board (quad) (8.5" x 10.45")

MDC-512
VARIABLE RESOLUTION GRAPHICS CONTROLLER (PDP-11 BUS)

The MDC-512 incorporates the revolutionary concept of variable resolution graphics on a single card. The MDC-512 is directly plug-in compatible with the industry standard DEC PDP-11 bus. The same card can be user programmed to produce a dot matrix of 256 X 256, 256 X 512, 512 X 512, or 256 X 1024 points by using 4K, 8K or 16K plug-in compatible dynamic memories.

- variable resolutions
- multiple cards stackable for color/grey scale applications
- single command erase
- vertical scroll built-in
- external/internal sync
- can be combined with the alphanumeric MDC-2480
- display memory read/write
- each dot individually addressed
- 1.4 usec max/dot access time
- X-Y addressing
- American/European standard
- software package available

Price: $1395/single; $1150/100 512 x 512; 256 x 1024; $895/single; $650/100 256 x 256; $1095/single; $850/100 256 x 512


Standard PDP-11 size board (quad) (8.5" x 10.45")

RGB-256
SINGLE BOARD 256 X 256 X 4 COLOR CONTROLLER (SBC-80 OR GENERAL PURPOSE)

The RGB-256 is a single board graphic controller which displays a 256 X 256 raster with 4 bits/pixel. All refresh memory is built-in. The board can be used for a variety of graphic systems such as a 256 X 256, 16 level grey or a 16 color display. Each dot is individually addressed and it can be read or written into in less than 1.2 usec/dot.

- 256 X 256 X 4 raster
- 16 level color (RGB)
- composite color output
- 16 level grey scale
- composite grey scale output
- American/European standard
- external/internal sync
- expansion for multiple boards for more bits/pixel
- can be combined with optional frame grabber card for frame grabbing applications
- plugs directly into an SBC-80 type computer
- can be used with any μP
- single instruction clear
- broadcast application
- image processing

Price: $1595/single; $1250/100 256 x 256 x 4; $995/single; $680/100 256 x 256 x 1


Stretched SBC-80 board (8" x 12")

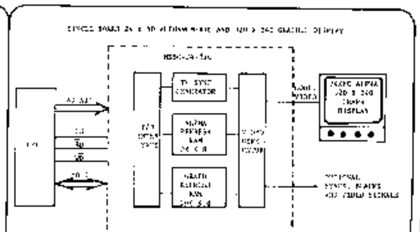
CRT DISPLAY CONTROLLERS (ALPHA AND GRAPH) COMBINED

MSBC-24/320
SINGLE BOARD 24 X 80 ALPHANUMERIC AND 320 X 240 GRAPHIC DISPLAY

The MSBC 24/320 is an SBC-80 bus compatible VRAM that integrates an alphanumeric and graphic display on the same printed circuit board. The alphanumeric section outputs a full 24 lines X 80 column text display. The graphic portion generates a dot raster of 240 X 320 dots. The graphic and alphanumeric displays are aligned and scaled to occupy the same screen area. On the input side, the card looks like a 4096 X 8 static read/write RAM with an access time of 500 ns. By being a part of the memory, the full power of the processor is available for display manipulations. Each alphanumeric character position on the screen corresponds to a unique memory location, whereas graphic data is addressed as 8-bit wide segments. The board can easily be interfaced to any micro or minicomputer.

- alphanumeric and graphic display on a single board
- 24 lines X 80 columns alphanumerics
- 320 X 240 dot raster graphics
- normal/inverse, on/off video control
- single +5V power supply
- 2716 EPROM character generator
- external/internal sync
- SBC 80 plug-in or general purpose
- refresh 12K X 8 static RAM built-in (read/write, 500 ns)
- American/European standard non-interlaced
- upper/lower case

Price: $1395/single; $1150/100


Stretched SBC 80 card size (8" x 12")

COMBINED ALPHA/GRAPH BOARDS

In addition to using single board alpha/graph combination (MSBC-24/320), the unique Matrox design allows for any combination of graphic and alphanumeric controllers. Since each VRAM has a built-in TV sync generator which can be either externally or internally synchronised, video combination can be generated by selecting one of the controllers as a master and synchronising the rest to the master. Typical combinations are the ALT-2480 and ALT-256, the MTX-2480 and MTX-256, the MSBC-2480 and MSBC-512, the MLSI-2480 and MLSI-512; etc. Module VRAMs can also be synchronised to graphics boards if required.

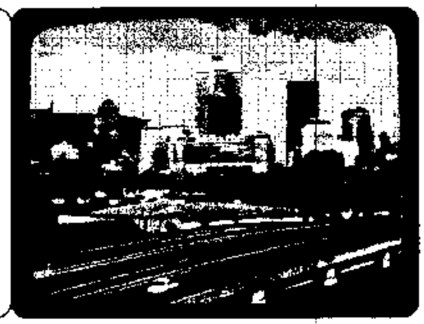
COLOR CRT CONTROLLERS

COMBINED BOARDS

The simplest graphic color system can be obtained by producing three separate video signals which then directly drive the red, green and blue guns of a color monitor. Up to a maximum of 24 graphic cards can be used in a master slave configuration to generate the red, blue and green inputs for the RGB monitor. The master card will supply three signals to all slaves: Dot clock, horizontal and vertical reset. Each card generates one bit of video information which can be used to produce a grey scale or color video signal. Typical graphic cards which can be used in this configuration include the ALT-256, MTX-256, MSBC-512, MLSI-512 and the MDC-512.

Outputs of these boards can be combined via a D/A converter for each color to obtain 2^N colors (N-number of boards). Additional RBG encoder circuits can be built to obtain composite color video if required.

The single board RGB-256 provides 4 bits per card, and can be used if 16 colors are required.



SINGLE CHIP DISPLAY/KEYBOARD CONTROLLERS

Matrox has introduced the first two chips in its single chip I/O controller family. They are alphanumeric display/keyboard controllers and are intended to drive a wide variety of the displays presently available on the market in either 5 X 7 dot matrix (MTX-A1) or 7, 14, 16 segment (MTX-B1) configurations (LED, Liquid X-tals, flourescent, incandescent, gas discharge, etc.).

The controllers are monolithic NMOS, LSI circuits packaged in a standard 40 pin DIP. They have a single +5V power supply and interface directly to any uP through an 8 bit bi-directional bus.

The I/O controller provides all timing and refresh signals for driving up to 32 character displays. An ASCII character generator ROM (64 X 5 X 7 -MTX-A1 and 64 X 16 -MTX-B1) is built-in as well as 32 X 8 refresh RAM. The keyboard portion provides all the scanning signals, debounce and decoding for any keyboard with up to 64 keys (X-Y or common pole).

Many intelligent commands such as clear display, shift left/right, cursor control, blink, read/write, etc., are featured. Display parameters such as refresh rate, display length, etc., are user programmable.

Typical applications are uP controlled instruments, equipment, POS terminals, electronic scales.

The Matrox single chip I/O controller represents a breakthrough in display technology by providing a complete intelligent alphanumeric display and keyboard controller in a single LSI chip.

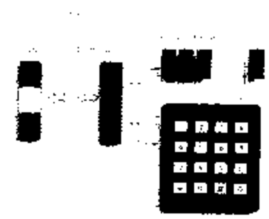
Complete assembled OEM displays are also available. They consist of single chip controller, LED displays, all driving and interface electronics and mounting hardware and filter on a small PCB. (8" X 3.25").

MTX-A1
ALPHANUMERIC (DOT) DISPLAY/KEYBOARD CONTROLLER

The MTX-A1 is a general purpose programmable alphanumeric display and keyboard interface device for use with any 8-bit microprocessor such as the 8080A, 6800, etc. The display portion provides all timing and refresh signals to drive up to 32 popular 5X7 dot matrix LED displays. The keyboard portion provides all scanning signals and debounces and decodes any keyboard with up to 64 keys. The single chip controller interfaces directly to the uP via the uP data bus. Many intelligent commands for display and keyboard manipulation are incorporated.

- Single chip controller
- drives up to 32 5X7 LED
- 64 ASCII character set
- character generator ROM (64X5X7) built-in
- refresh 32 X 8 RAM built-in
- no external refresh required
- self-test built-in
- scans up to 64 keys
- interface to any uP directly
- all parameters programmable
- single +5V, 60 mA
- low power
- many intelligent commands

Price. $49/single; $39/100. $12/10K

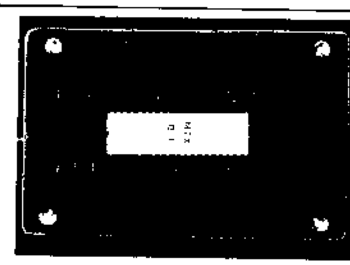


MTX-B1
ALPHANUMERIC (SEGMENT) DISPLAY/KEYBOARD CONTROLLER

The MTX-B1 is a general purpose programmable alphanumeric display and keyboard interface device for use with any 8-bit microprocessor such as the 8080A, 6800, etc. The display portion provides all timing and refresh signals to drive up to 32 popular segment (from 7 to 16) displays. The keyboard portion provides all scanning signals and debounces and decodes any keyboard with up to 64 keys. The single chip controller interfaces directly to the uP via the uP data bus. Many intelligent commands for display and keyboard manipulation are incorporated.

- single chip controller
- drives up to 32 displays
- 7 to 16 segment displays
- LED, plasma, flourescent, gas discharge, incandescent, etc.
- common anode/cathode
- alphanumeric 16 segment character generator built-in
- refresh 32X8 RAM built-in
- dual scan mode doubles duty cycle
- scans up to 64 keys
- interfaces to any uP directly
- all parameters programmable
- single +5V, 60 mA

Price: $49/single: $39/100. $12/10K



COMPLETE LED DISPLAYS

MTX-A2
COMPLETE OEM 16 CHARACTER ALPHANUMERIC DISPLAY (5X7 LED)

The MTX-A2 is a complete ready to use 16 character alphanumeric display for OEM. It contains all drivers, an MTX-A1 display controller and 16 alphanumeric .35" LED displays. All data signals are brought to a 44 pin connector.

- complete 16 character display
- expansion to 32 characters
- interfaces directly to any uP
- MTX-A1 single chip controller
- mounting hardware/red filter included
- $180 without LEDS/$280 with LEDS
- single +5V, 800 mA power supply
- variable intensity
- scans up to 64 keys
- all solid state
- 22 intelligent commands

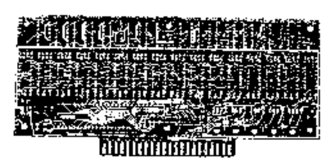
Price: $280/single; $230/100

Dimensions: 8" X 3.25" PCB

MTX-A2 EXD

The MTX-A2 EXD is a blank PC board which can be used for various applications. By plugging in 16 LEDS and shift registers, the MTX-A2 EXD can be used to expand the MTX-A2 to a 32 digit display (the MTX-A1 and the display drivers from the MTX-A2 will drive the additional LEDS). For applications requiring large characters (up to 3"), the user can plug in the MTX-A1 chip and the display drivers and build his own large 5 X 7 alpha LED display using discrete LEDS.

Prices: Blank PCB only. $28/single, LEDS & SHR $200, MTX-A1 & drivers $95



Dimensions: 8" X 3.25" PCB

MTX-B2
COMPLETE OEM 32 CHARACTER ALPHANUMERIC DISPLAY (14 SEG)

The MTX-B2 is a complete ready to use 32 character alphanumeric display using large .5", 14 segment LEDS. All drivers, LEDS and the MTX-B1 controller are included. The maximum number of characters is 32 organized in 2 lines of 16 each. However, any display format from 1 to 32 can be used by plugging in the required number of LEDS and initializing the MTX-B2 chip. Keyboard scan signals are brought out on the connector for keyboard scanning.

- complete 32 character display
- interfaces directly to uP
- MTX-B1 single chip controller
- mounting hardware/filter included
- 2 lines X 16 characters
- modular 1 to 32 character display
- single +5V power supply
- user adjusted brightness
- scans up to 64 keys
- large .5 inch LEDS (MTX-14SD)
- all solid state
- 22 intelligent commands

Prices. MTX-B2 (32) [with 32 LEDS] $380/single; $285/100
MTX-B2 (16) [with 16 LEDS] $260single; $195/100

Dimensions: 8" X 3.75"

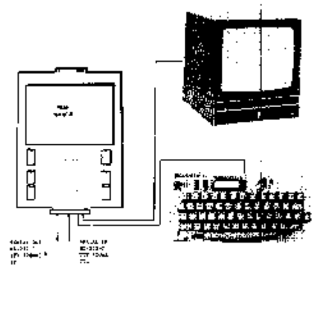
ACCESSORIES

SI-ABCD SERIAL INTERFACE ADAPTER BOARD

Serial interface adapter board (SI) allows the user to build a variety of low cost terminals with standard serial interface. By adding an ASCII keyboard, TV monitor and any of Matrox VRAM modules or boards, a versatile low cost CRT terminal with different screen formats can be built.

- serial interface (user selectable) RS-232-C; TTY (20mA, 60mA); TTL
- user selectable baud rate 110, 150, 300, 600, 1200, 2400, 4800, 9600
- works with any Matrox video RAM; MTX-816, -1632, -1632SL, -1648/64SL, -2064, -2480
- accepts ASCII keyboard inputs (8 bits and strobe)
- drives directly TV monitors, composite video, or separate video and syncs
- all video and baud rate signals X-tal controlled
- accepts CR (carriage return), LF (line feed) and clear screen commands
- external clear screen command
- synchronised VRAM writing; no flicker during VRAM write
- can be used as serial write only terminal (no keyboard)

Price: $120/single; $98/100

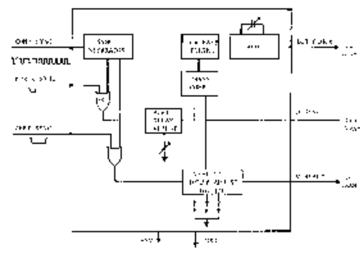


PLL-01
EXTERNAL SYNC PHASE LOCK LOOP MODULE FOR VRAMS

The PLL-01 is a 2.5" X 3" plug in module intended for use with the alphanumeric and graphic VRAMs requiring external sync capabilities. This module allows the user to synchronise any VRAM to a TV camera, master sync generator, etc., for various applications requiring video mixing.

The module requires a single +5V power supply and interfaces directly to the VRAM. (the VRAM is operating in the slave mode). The PLL-01 will accept either composite or separate syncs. The user can adjust the horizontal and vertical delay. This allows the user to position the VRAM video picture relative to the external syncs.

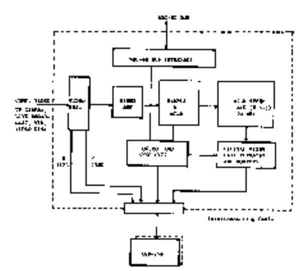
Price: $48/single; $38/100



FG-01

The FG-01 is a frame grabber card used in combination with the RBG-256 single board CRT controller. The grabber plugs into the INTEL SBC-80 bus and allows the CPU to grab a single frame of a standard TV signal. The board contains a high speed 8 bit A/D converter, and all logic required to interface with the RGB-256 card.

- grabs single TV frame
- CPU grab control
- American/European standard
- 4 bit/8 bit grey scale
- lowest cost complete video system
- broadcast quality
- can be used with any computer
- wide range of application

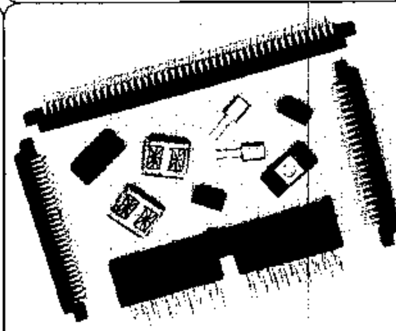


Dimensions: 6.75" X 12"

HARDWARE ACCESSORIES

MBC-01	Blank prototype board (SBC-80 or general purpose)	$68	$54
MTX-305	5X7 LED display (HL-305 equivalent)	$ 7.5	$ 6.5
MTX-14SD	Dual (2 digit) 14 seg. alphanumeric LED display	$ 9.5	$ 7.8
VCB-75	75 Ohm coax video cable (price/foot)	$ 1.5	
X-tal	X-tal (for different TV standards)	$10	
CHG	Character generator (alpha. CRT controllers)	$20	
MS-BAR	Socket bars for alphanumeric modules	$ 4	
CON-44	44-pin connector (530654-6)	$ 5	
CON-50	50-pin connector (MP-0100-25-DP-1)	$ 6	
CON-56	56-pin connector (530664-6)	$ 6	
CON-86	86-pin connector (1-530654-3) for SBC-80 boards	$ 9	
CON-DEC	DEC connector (for LSI-11 and PDP-11 boards) (H8030)	$15	

Price reference: First price per every 1 — Second price per every 100



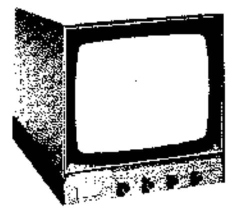
CRT MONITORS

MCRT-9, MCRT-14/MCRT-14G

The MCRT-9 and MCRT-14 are 9" and 14" (black/white P4 phosphor) solid state video monitors which have been built to international standards as established by the communications and computer industries. The MCRT-14G is similar to the MCRT-14 except that it has green phosphor (P39). These high quality monitors can be used with any of the Matrox CRT controllers.

- internal/external sync capability
- A/B video input selection
- VTR time constant switching
- pilot lamp/tally lamp
- 75 Ohm termination switch (A&B)
- UL/CSA approved
- switchable power supply 110/220V Hi Lo line 90-132V
- 15 MHz bandwidth to optimize resolution
- wide dynamic range
- excellent black level clamping and hum suppression
- EHT regulation
- up front primary and most secondary controls
- AC cord winder
- universal AC interlock
- attractive styling

Price: MCRT-9 $420 MCRT-14 $525, MCRT-14G $580

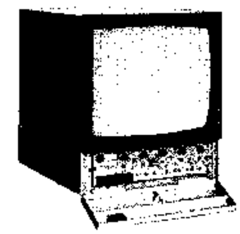


MCRT-CC19/RGB/PAL/SECAM

The MCRT-19 is a 19" high quality color TV monitor with a 6 MHz color handwidth. It can be used with any Matrox CRT controller in systems requiring color graphics.

- RGB/NTSC/PAL/SECAM convertibility through optional plug in circuit boards
- signal processing circuitry located on front accessible plug-in circuit boards
- DC coupled operating controls permitting final user to remote any or all primary functions.
- excellent tracking of all three channels throughout the operating range.
- active convergence circuitry provides full raster convergence
- excellent black level stability through high level back porch clamp.
- a multiple clamping arrangement provides hum rejection in series mode
- the power supply design allows asynchronous operation
- maintenance of full resolution at brightness levels by the use of the Negative Guardband picture tube.
- optional luminance comb filter provides full luminance response avoiding the bandwidth limitations associated with conventional passive subcarrier traps. As a second function, this comb filter reduces luminance cross color interference in the chroma display
- each model offers reduced scan

Price: $5950



SOFTWARE AND DOCUMENTATION

MTX-ALPHA SOFTWARE PACKAGE

The MTX-ALPHA software package, for use with any 2480, provides the user with the full flexibility of a software based intelligent terminal. The software package is written in 8080 assembly language. The program occupies approximately 3K of memory. The package will fully emulate the popular Lear Siegler Inc. ADM-3A and Digital Equipment Corp. DECSCOPE VT-52 interactive display terminals. Line at a time and text block input modes are available to provide the powerful text preparation features of an intelligent terminal. The package includes a detailed manual, a listing with comments and paper tape (object).

Price: $28

MTX-GRAPH SOFTWARE PACKAGE

The MTX-GRAPH software package is designed for use with any 256 graphics display. The package is configured as a series of callable sub-routines and occupies a 1K block of memory. Some features included in MTX-GRAPH are point plot and line vector graphics, variable size alphanumeric character generation, animation synchronisation and an option for color graphics. The package includes a detailed manual, a listing with comments and an object paper tape.

Price: $28

MANUALS

Manuals including schematics for various products are available. The price is $10 each if bought separately. Free if bought with product.

CUSTOM DESIGN

Custom CRT display controllers can be designed according to customer specifications. Examples of custom orders would be different screen formats, special size PCB's, VRAM, additional features, modifications of standard designs, system integration, custom character generators, etc. Send a request for quotation for your particular needs and we will be pleased to suggest solutions and quote prices and delivery. For larger OEM requirements, Matrox can licence the user to make his own VRAM. Matrox will supply all schematics, specifications and custom made integrated circuits, where required for production, for a flat one time charge.

matrox electronic systems ltd.

2795 BATES RD., MONTREAL, QUE. H3S 1B5, CANADA
TEL.: (514) 481-6838 or 735-1182 TELEX: 05-825651

Canada Post / Postes Canada
Postage paid / Port payé
Third class / Troisième classe
BM 41
MONTREAL

PRINTED MATTER

HEAD OFFICE
Matrox Electronic Systems Ltd
2795 Bates Road
Montreal, Que., Canada H3S 1B5
Tel 514-481-6838 or 514 735-1182
Telex 05-825651

U.S. SUBSIDIARY
Matrox International Corp.
Trimex Building
Mooers, N.Y. 12958

CANADIAN SALES REPRESENTATIVES
Multitek Inc
15 Grenfell Crescent, 2nd Floor
Ottawa, Ontario
K2G 0G3
Tel 613-226-2365

U.S. SALES REPRESENTATIVES

Hudd-tronics Inc
256 Engle Street
Tenafly, N.J. 07670
Tel 201-569-2230

Dynamic Sales
113 Terrace Hall Avenue
Burlington, Mass. 01803
Tel 617-272-5676

Mycrosystems Marketing Inc.
13777 N. Central Expressway
Suite 405, Dallas, Texas 75243
Tel 214-238-7157
Twx 910-867-4763

Microsystems Marketing Inc
6610 Harwin
Suite 125, Houston, Texas 77036
Tel 713-783-2900
Twx 910-881-5439

Newman Computer Exchange
1250 North Main Street
Ann Arbor, Mich. 48104
Tel. 313-994-3200
Twx 810-223-8023

Media Systems Inc
139 First Street
Cambridge, MA 02142
Tel 617 661-9424

INTERNATIONAL SALES REPRESENTATIVES

AUSTRALIA
Measuring & Control Equipment, Mece
Co. Pty. Ltd.
P.O. Box 78
Epping, N.S.W. 2121, Australia
Tel 86-4060

AUSTRIA
Kontron GES M.B.H. & Co. KG
A-1140 Wein
Amoisg. 49, Austria
Tel (0222) 94 56 46
Tlx 01-1699

ENGLAND
Shelton Instruments Ltd.
24 Copenhagen Street
London N1 OJD, England
Tel 01-278 6273
Tlx. 299441 GWI G

Techexport Inc.
Bradley House
St Paul's Lane
Bournemouth BH8 8HN, England

DENMARK
Jorgen Andersen Ingeniorfirma A.S
Produktionsvej 1
DK-2600 Glostrup, Denmark
Tel. (02) 91 86 88
Tlx 35378 JORGEX dk

FINLAND
Distep-KY
Arkadiankatu 16C
00100 Helsinki 10 Finland
Tlx 12-1307 Aspen S F

FRANCE
Metrologie
La Tour D'Asnieres
4 Ave Laurent Cely
92606 Asnieres France
Tel 791 44 44
Tlx 611 448

GERMANY
Atlantik Electronik GmbH
Hofmannstrasse 20
8000 Munchen 70, West Germany
Tel. 089/7 85 31 12, 7 85 31 68
Tlx 5 213 066

GREECE
Dimitri B. Zahos
30 Dragoumi
612 Athens, Greece
Tel. 77 70 423
Tlx. (21) 8409

HONG KONG
Shankir Trading Co. Ltd
Rm. 503-5, Kam Chung Bldg
52-54 Jaffe Road, Hong Kong BCC
Tel 5-281521-3
Tlx 74269 DIODE HX

INDIA
Summit PVT Ltd.
3 Esplanade East
Calcutta 700069, India
Tel 23-1671 2
Tlx 021-7776

ISRAEL
HN Electronics Agencies Ltd.
193 Hagolan Street
Ramat-Hachayal Tel-Aviv Israel
Tel (03) 471859
Tlx 341730 SPEED IL

ITALY
3-G Electronics s r l
Via Perugino 9
20135 Milano, Italy
Tel. 544291 5466397 543096
Tlx. 35024 TREG

JAPAN
Internix Incorporated
Narto Bldg.
7-2-8 Nishishinjuku
Shinjuku-ku, Tokyo 160, Japan
Tel 369-1101
Tlx J28497

NETHERLANDS
Famatra Benelux
P.O. Box 721
Breda, Netherlands
Tel. (076) 133457
Tlx. 54521 Fatra nl

SOUTH AFRICA
L Electron (PTY) Ltd.
P.O. Box 10544
Johannesburg 2000 South Africa
Tel 40 6296/40-8057/40-8096
Tlx 8-2333 SA

SPAIN
Neotecnica S.A.E.
Marques de Urquijo 44
Madrid 8, Spain
Tel. 2420900
Tlx. 22099 NTSAE E

SWEDEN
Satt Electronlund AB
P.O. Box 9004
200 39 Malmoe 9, Sweden
Tel 040-22 20 00
Tlx 32422 ELUND

SWITZERLAND
Kontron Electronic Ltd.
Bernerstrasse Sud 169
Zurich 8048, Switzerland
Tel 01-62 82 82
Tlx 58836 KONCO CH

TAIWAN
Multitech International Corporation
2nd Floor, 977 Min Shan E. Road
Taipei 105, Taiwan
Tel. (02) 7681232, (02) 7654092
(02) 7913940
Tlx. 23756 MULTIIC

ORDERING INFORMATION

Canadian Customers: Order directly from Matrox, Montreal. Add 12% federal sales tax if applicable. **Overseas Customers:** Order directly from Matrox, Montreal or from overseas distributors. **U.S. Customers:** Order directly from Matrox, Montreal, or you can order from our U.S. company, Matrox International Corporation, Trimex Building, Mooers, N.Y. 12958. Shipments can be made either from Canada or from Mooers, N.Y. to avoid delays in customs. Please add 5½% duty to cover customs and brokerage charges when ordering this way. Specify in your order whether you wish goods FOB Montreal or FOB Mooers, N.Y. All prices are quoted in U.S. funds, except for Canadian customers. Deliveries are typically two to four weeks ARO on all items. When ordering, specify exactly options and versions desired (consult the data sheets for detailed information). Add suffix "E" when ordering European (50 Hz, vertical scan) versions.

matrox electronic systems ltd.
2795 BATES RD., MONTREAL, QUE. H3S 1B5, CANADA
TEL.: (514) 481-6838 or 735-1182 TELEX: 05-825651

OEM PRICE LIST

Effective November 1, 1978; Supercedes all previous price lists

1. CRT DISPLAY CONTROLLERS (ALPHANUMERIC)

1.1 Modules

		1	2-9	10-24	25-99	100-199
MTX-816	8 lines X 16 columns CRT controller module	179	173	166	156	149
MTX-1632	16 lines X 32 CRT controller module	225	210	198	185	169
MTX-1632SL	16 X 32 CRT contr. module (external sync)	225	210	198	185	169
PV-1	Up to 16 X 64 user programmable VRAM (in/ex)	295	270	245	220	190
MTX-2064	20 lines X 64 columns CRT controller module	295	270	245	220	190
MMD-2480	24 lines X 80 columns CRT controller module	395	370	345	320	290

1.2 Printed Circuit Boards

		1	2-9	10-24	25-99	100-199
MTX-1648/64SL	16 X 48 or 64 (Prolog bus. ext sync)	295	270	245	220	190
MTX-2480	24 X 80 CRT controller (general purpose)	395	370	345	320	290
ALT-2480	24 X 80 CRT controller (S100 bus plug-in)	295	295	280	280	265
EXO-2480	24 X 80 CRT controller (Exorciser plug-in)	495	455	420	385	350
MSBC-2480	24 X 80 CRT controller (SBC-80 bus plug-in)	495	455	420	385	350
MLSI-2480	24 X 80 CRT controller (LSI-11 bus plug-in)	495	455	420	385	350
MDC-2480	24 X 80 CRT controller (PDP-11 bus plug-in)	495	455	420	385	350

2. CRT DISPLAY CONTROLLERS (GRAPHICS)

2.1 Modules

		1	2-9	10-24	25-99	100-199
MMD-256	256 X 256 dot raster module (+5V only)	595	570	545	520	495
MMD-256D	256 X 256 dot raster module (+5V, +12V)	495	455	420	385	350

2.2 Printed Circuit Boards

		1	2-9	10-24	25-99	100-199
MTX-256	256 X 256 dot raster (general purpose)	595	570	545	520	495
ALT-256	256 X 256 dot raster CRT contr. (S100 bus)	395	395	375	375	355
ALT-512	512 X 256 dot raster CRT contr. (S100 bus)	595	570	545	520	495
EXO-512	512 X 256 dot raster (Exorciser bus)	695	665	635	605	575
MSBC-256	256 X 256 (SBC-80 bus or general purpose)	895	795	750	695	650
MSBC-256/512	256 X 256 (SBC-80 bus or general purpose)	1095	995	950	895	850
MSBC-512	512 X 512 (SBC-80 bus or general purpose)	1395	1295	1250	1195	1150
MSBC-1024	256 X 1024 (SBC-80 bus or general purpose)	1395	1295	1250	1195	1150
NSBC-512	512 X 512 raster with vector plot (SBC-80)	1495	1395	1350	1295	1250
MLSI-256	256 X 256 dot raster (LSI-bus, plug-in)	895	795	750	695	650
MLSI-256/512	256 X 512 dot raster (LSI-11 bus, plug-in)	1095	995	950	895	850
MLSI-512	512 X 512 dot raster (LSI-11 bus)	1395	1295	1250	1195	1150
MLSI-1024	256 X 1024 dot raster (LSI-11 bus)	1395	1295	1250	1195	1150
MDC-256	256 X 256 dot raster (PDP-11 bus)	895	795	750	695	650
MDC-256/512	256 X 512 raster (PDP-11 bus)	1095	995	950	895	850
MDC-512	512 X 512 raster (PDP-11 bus)	1395	1295	1250	1195	1150
MDC-1024	256 X 1024 raster (PDP-11 bus)	1395	1295	1250	1195	1150
RGB-256/4	256 X 256 raster, 4 bit/pixel, color/grey (SBC-80 bus or gen. purpose)	1595	1480	1430	1365	1300
RGB-256/3	256 X 256/3 bit/pixel; color/grey; exp.	1395	1295	1230	1195	1150
RGB-256/2	256 X 256/2 bit/pixel; color/grey; exp.	1295	1200	1150	1100	1050
RGB-256/1	256 X 256/1 bit/pixel; color/grey; exp.	1095	995	965	940	895

3. CRT DISPLAY CONTROLLERS (ALPHA & GRAPH COMBINED)

		1	2-9	10-24	25-99	100-199
MSBC-24/320	Single board 24 X 80 alphanumeric;320X240 raster graphics CRT display controller (SBC-80 or general purpose)	1395	1295	1250	1195	1150

NOTE: Combined alphanumeric and graphic display can also be obtained by combining other standard Matrox alpha and graph controllers (i.e.: ALT-2480 and ALT-256 or ALT-512; MMD-2480 and MMD-256; MLSI-2480 and MLSI-512, etc.)

4. COLOR CRT CONTROLLERS

Multiple standard graph cards can be combined to obtain color. Up to 24 bit per pixel can be obtained, 256 X 256 or 512 X 512 resolution.
NOTE: RGB-256/4 is single board 256 X 256 X 16 color CRT controller.

5. uP DISPLAY CONTROLLERS (INTEGRATED CIRCUITS)

MTX-A1	Single chip I/O display/keyboard controller 5 X 7 dot LED's up to 32 character, 64 keys	49	46.5	44	41	39
MTX-B1	Single chip I/O display/keyboard controller 14, 16 segment LED, 32 characters, 64 keys	49	46.5	44	41	39

MTX-A1 and MTX-B1 in quantities of 10K are $12 each.

6. COMPLETE LED DISPLAYS

MTX-A2	16 characters alphanumeric display (5X7 LED)	280	265	250	240	230
MTX-A2EXD	32 characters alphanumeric display (5X7 LED)	460	435	410	390	375
MTX-B2	32 characters alpha. display (14 Seg. LED)	380	360	340	325	310

7. HARDWARE ACCESSORIES

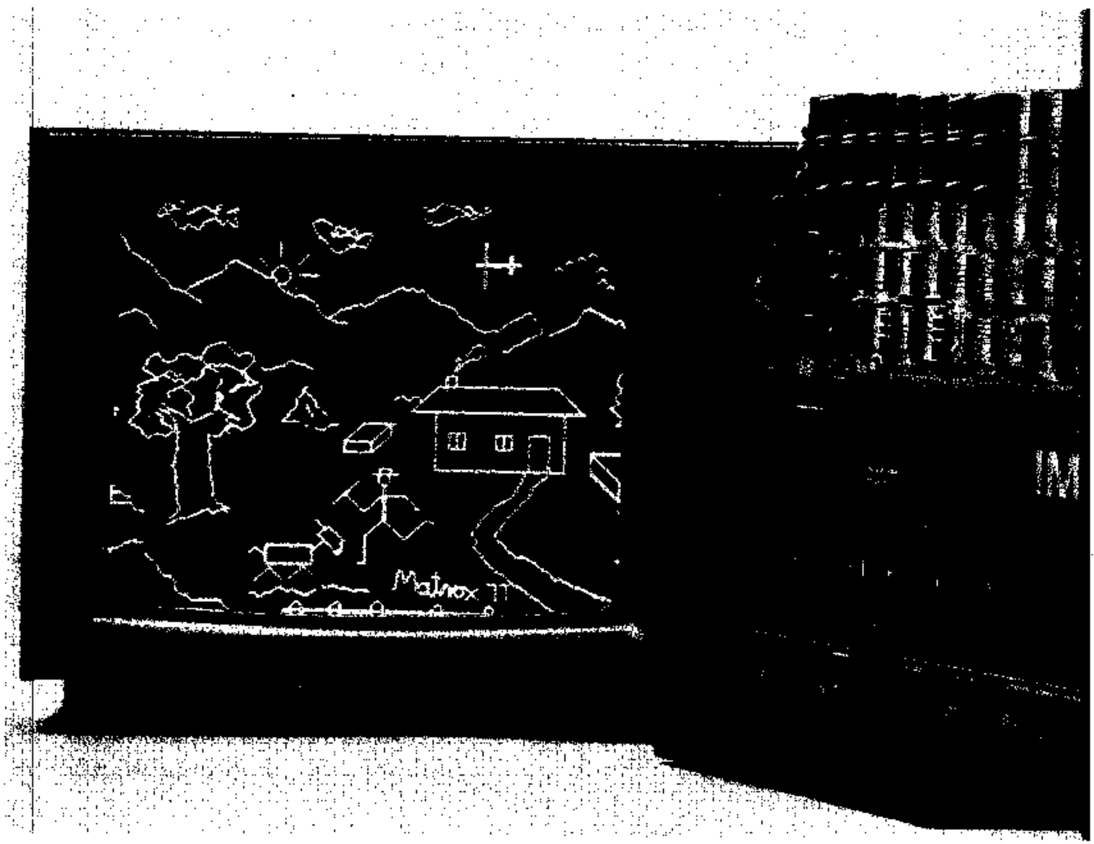
		1	2-9	9 UP
SI-ABCD	Serial interface adaptor board (alpha CRT cont.)	120	109	98
PLL-01	External sync phase lock loop module	48	43	38
PLL-SBC	Ext. sync phase lock loop board (SBC-80 plug-in)	149	139	119
MTX-A2BL	Blank MTX-A2 board	28	26	22
MBC-01	Blank prototype board (SBC-80 or general purpose)	68	62	54
MES-01	TV Sync generator PCB (for MTX-1648/64SL add-on)	38	34	32
MTX-305	5X7 LED display (TIL-305 equivalent)	7.5	7.0	6.5
MTX-14SD	Dual (2 digit) 14 seg. alphanumeric LED display	9.5	8.8	7.8
FG-01	Frame grabber (8 bit/30 mHz A/D) (SBC-80 bus)	3500	3300	3000
X-tal	X-tal (for different TV standards)	10	10	10
CHG	Character Generator (alpha. CRT controllers)	20	20	20
MCH-01	5X7 character generator (upper/lower case/graphics)IC	16	14	12
MS-BAR	Socket bars for alphanumeric modules	4	4	4
CON-44	44 pin connector (530654-6)	5	5	5
CON-50	50 pin connector (MP-0100-25-DP-1)	6	6	6
CON-56	56 pin connector (530664-6)	6	6	6
CON-86	86 pin connector (1-530654-3)	9	9	9
CON-DEC	DEC connector (for LSI-11 and PDP-11 boards)(H8030)	15	15	14

matrox electronic systems ltd.
2795 BATES RD., MONTREAL, QUE. H3S 1B5, CANADA
TEL.: (514) 481-6838 or 735-1182 TELEX: 05-825651

ALT - 256 * * 2

MTX TV CRT CONTROLLER FAMILY

GRAPHIC DISPLAY



* MATROX products covered by Canadian and foreign patent and/or patent pending.

INTRODUCTION:

The Matrox ALT-256**2 board is a fully tested, assembled and burned-in interface card which provides capability for a complete graphic system at a fraction of the cost of any other commercial graphic system. The card contains all interface electronics, a TV sync generator, and its own 65,536 X 1 bit refresh memory. It plugs directly into one slot of any S-100 bus compatible computer. The built in refresh memory allows much greater flexibility and speed since no CPU time is required to refresh the screen.

The output is a composite video signal which can be connected to any TV monitor or the video portion of a TV set. The unit produces a high resolution 256 X 256 dot raster. The complete screen can be cleared or preset by a single instruction.

The ALT-256**2 board occupies a single S-100 bus slot and requires 4 output ports and 1 input port (port address is selectable on the card with jumpers).

FUNCTIONAL DESCRIPTION:

Four output ports are used for loading of data into the display refresh memory. Assuming that output ports ØØ to Ø3 are selected (which also gives ØØ for the input port) the addressing is as follows:

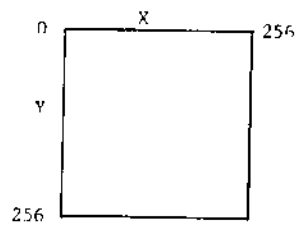
The X coordinate of a given dot is loaded by outputting an 8 bit coordinate to the port H "Ø1. (instruction: OUT Ø1). The Y coordinate is outputted by OUT Ø2. These two instructions will set the cursor at the desired dot address.

After the dot address is loaded, the dot intensity is loaded by outputting data to port H "ØØ". (OUT ØØ). Data H "ØØ" will result in a black dot; H "Ø1" will display a white dot. After the dot intensity is loaded, the ALT-256**2 will require 3.4 μsec to write the dot in the refresh memory. This is necessary to allow for internal synchronisation of the write operation, TV read scan and dynamic memory refresh. Since the CPU almost always requires more than 3.4 μsec to load the next dot address and data, the CPU can run at its full speed. Also note that regardless of the speed at which the CPU is accessing the ALT-256**2, there will be no flashes or streaks on the display, since access is internally synchronised. This results in a truly professional display. In other words, all the above three commands can be executed at full CPU speed with no image degradation.

Assuming port select bits are programmed as:

A7	A6	A5	A4	A3	A2	A1	A0
Ø	Ø	Ø	Ø	Ø	Ø	X	X

X, Y Ports



OUT Ø1 - X address
OUT Ø2 - Y address

DOT write port

OUT ØØ - Dot intensity

DØ = Ø black dot
DØ = 1 white dot
D7-D1 don't care

ERASE port

OUT Ø3 - screen clear

DØ = Ø all dots balck
DØ = 1 all dots white
D7-D1 don't care

STATUS port

IN ØØ - status

DØ = Ø display ready
DØ = 1 display being erased
D1 = Ø video portion
D1 = 1 vertical blank
D7-D2 don't care

TABLE 1: I/O PORT ADDRESSES AND FUNCTIONS OF THE ALT-256**2

The whole screen can be easily cleared by outputting H "ØØ" to the output port H "Ø3". This will result in the entire screen being black. Outputting H "Ø1" will result in all white since all 65,536 bits of the refresh memory will be simultaneously loaded with 1. This operation will require one TV frame time (33 msec max).

After the CPU outputs data to the port H "Ø3", an internal flag is set for between 16 msec to 33 msec until the screen is cleared. The testing of the flag is accomplished by inputting data from the input port H "ØØ". (INØØ). If the data bit DØ is high then the ALT-256**2 is busy: if DØ is low the ALT-256**2 is ready to accept new data.

Input port H "ØØ" provides additional information about the position of the electronic beam, (vertical blank) which is useful for dynamic motion display. (see table).

THEORY OF OPERATION:

The ALT-256**2 has four major blocks: the TV sync generator, scanning circuitry, cursor and interface electronics and 65,536 X 1 memory. (see Fig. 3 & 4) The sync generator is formed of x-tal oscillator (A 31) and a divider chain (A24, 16, 23, 6). This divider chain produces all timing signals for the memory scanning as well as horizontal and vertical sync. The TV sync generator can be programmed by jumpers for the European or American TV standard.

The scanning circuitry consists of multiplexers (A4, 12, 21, 20, 25, 26) which provide proper address and R/W signals for the RAM required.

The cursor consists of two 8 bit latches which are loaded by the CPU. A10 and A11 are X address register, A2 and A3 , Y address register. Necessary interface address and timing decoding is accomplished by A1, A9 and A51 to A55.

The refresh memory has 16, 4K dynamic memories (4096, 16 pin) organised as a 65,536 X 1 bit memory (A32-A47).

Power supplies for 5V, 600 mA; 12V, 100 mA and -5V, 10 mA are generated by A56, A57 and CR1; on board voltage regulators.

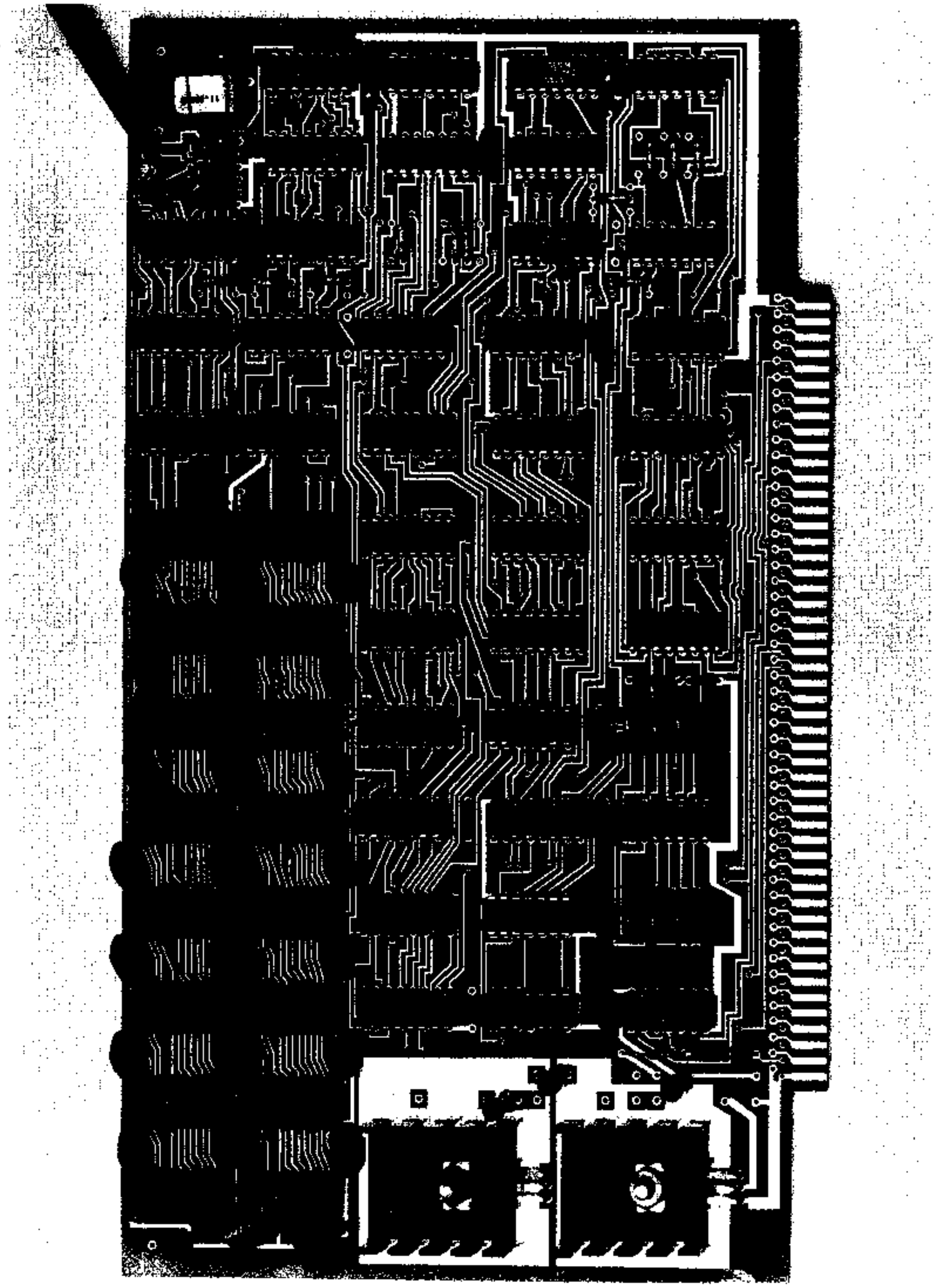


FIG. 1: PHOTOGRAPH OF THE ALT-256★★2

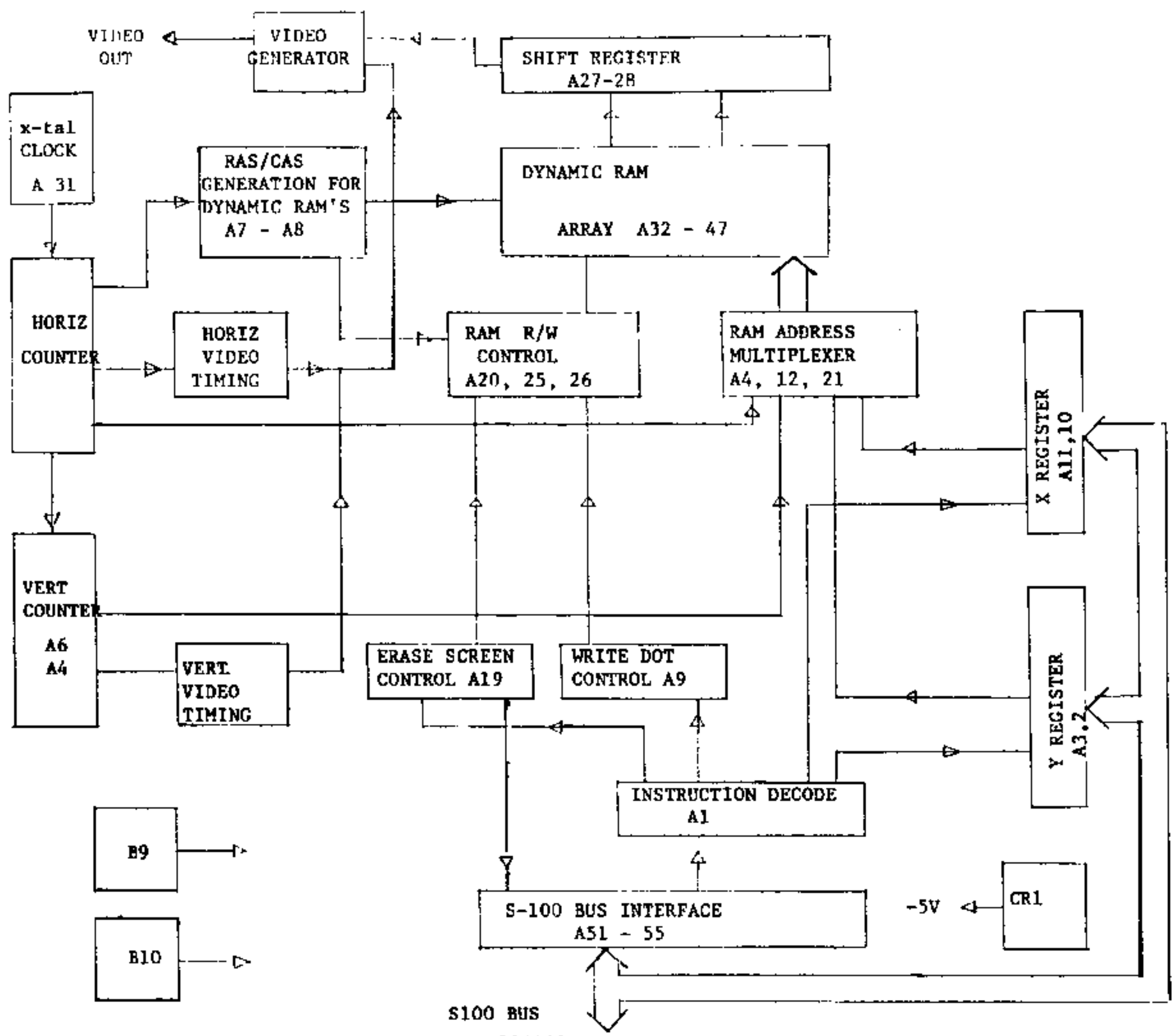


FIG. 4: BLOCK DIAGRAM FOR ALT-256**2

MAINTENANCE AND WARRANTY:

The ALT-256**2 is a fairly complex card and to understand its operation requires extensive knowledge of TV scanning, 4K dynamic memories and hardware. The complete circuit and assembly schematics are supplied to allow a competent user to troubleshoot the board if necessary. However, each board is fully tested, assembled and burned in for 24 hrs. before shipping to ensure reliability. In case of trouble, a warranty is provided.

Matrox products are warranted against defects in materials and workmanship for a period of 3 months from date of delivery. We will repair or replace products which prove to be defective during the warranty period, provided they are returned to Matrox Electronic Systems Ltd. No other warranty is expressed or implied. We are not liable for consequential damages.

Non-warranty repairs are billed at a minimum of $50 and a maximum of $100 according to time and materials required.

JUMPER OPTIONS:

The board is normally programmed for different options before shipping as specified in the purchase order. There are two things which can be programmed.

The TV sync generator can be programmed for one of 3 options. The American Standard option (60 Hz) has a 240 line X 256 dot raster (240 visible horizontal lines). The American Non-standard (60 Hz) version has a full 256 line X 256 dot raster with horizontal frequency of 16.8 KHz for a total of 280 lines. Some adjustment of the horizontal hold on the TV monitor may be necessary to allow the TV to lock. The European standard option has 312 lines (50 Hz).

JUMPERS IN:	ANS	W1 W4 W5 W9 W10 W12 W14 W17 W19 W21 W22
	AS	W2 W4 W6 W8 W11 W13 W15 W17 W18 W20 W23
	ES	W1 W3 W5 W7 W11 W12 W14 W16 W19 W20 W22

TABLE 2: TV STANDARD JUMPER OPTION

The four output ports and one input port can be located on any 4 location boundary in the 256 port address space. Address bits A2 to A7 can be selected as follows:

		JUMPER ACCROSS PINS	
ADDRESS BIT =		Ø	1
	A2	1 and 16	2 and 15
A49	A3	3,14	4,13
SOCKET	A4	5,12	6,11
	A5	1,16	2,15
A50	A6	3,14	4,13
SOCKET	A7	5,12	6,11

x jumper can be any resistor between Ø to 51 Ohm

TABLE 3: I/O ADDRESSING JUMPER OPTIONS

INSTALLATION AND TESTING:

To install the ALT-256**2, switch the Altair/Imsai 8080 type computer off and plug the card in. Connect the composite video output to the input of the TV monitor and switch the TV monitor and computer on. The display will be a random pattern of the content of the refresh memory.

To help in testing the card a short test program listing (see Table 4) is provided. The program makes the following assumptions: The ALT-256**2 is strapped to begin its output ports at location H "ØØ". The computer front panel has input port switch register at location H "FF". If the address strapping of the ALT-256**2 or front panel ports is different from that specified, the program can be easily modified.

To test the card, toggle into memory the short test program (see Table 4). The starting address is H "Ø100". The program will first clear the entire screen depending on the setting of the input port switch (SWO). A setting of H "ØØ" will produce a black screen, H "01" will produce a white screen.

Following this, the program clears the X and Y register and writes SWO (up = 1; down = 0) into dot ØØ. (Upper left corner). Then address X is incremented, SWO is written into the next dot etc, continuously. By changing the SWO position during the computer scan, a white or black dot is written. The speed of the computer scan can be varied by the position of the switches S7-S1 on the front panel.

VIDEO SIGNALS

Composite video signal. Output impedance 75 Ohms. Short circuit protection built in.

Horizontal and Vertical Sync signals.

SIGNAL	FREQ.	HIGH	LOW	STD.
SH	16.8kHz	4.5	55	ANS
Horizontal	15.7kHz	5.8	57.8	AS
Sync	15.6kHz	4.5	59.6	ES
SV	60 Hz	238	16.43	ANS
Vertical	60 Hz	254	16.42	AS
Sync	50 Hz	256	19.74	ES
DTC	7.0536 mHz			ANS
Dot	5.5334 mHz			AS
Clock	7.0536 mHz			ES

ANS American nonstandard (280 lines, 60 Hz)
AS American standard (262 lines, 60 Hz)
ES European standard (312 lines, 50 Hz)

ADDRESS	CONTENT (Hex)	MNEMONIC	COMMENT
0100	DB	IN FF	/input front panel switch
0101	FF		
102	D3	OUT Ø3	/input SW to erase port
03	Ø3		
04	DB	LUP 1, IN ØØ	/input ALT-256**2 status
05	ØØ		
06	E6	ANI Ø1	/mask SWØ
07	Ø1		
08	C2	JNZ LUP1	/test for busy (Not Ø)
09	Ø4		
0A	Ø1		
0B	21	LXIH ØØ	/load H, L with Ø
0C	ØØ		
0D	ØØ		
0E	7D	LUP 2, MOV A, L	/move L to A
0F	D3	OUT Ø1	/output X coordinate
10	ØI		
11	7C	MOV A, H	/move H to A
12	D3	OUT Ø2	/output Y coordinate
13	Ø2		
14	DB	IN FF	/input front panel switches
15	FF		
16	D3	OUT ØØ	/write dot to port ØØ
17	ØØ		
18	E6	SPEED, ANI FE	/mask switches S7-S1
19	FE		
1A	3C	LUP 3, INC A	/loop delay for speed
1B	C2	JN Z LUP 3	/test for zero
1C	1A		
1D	Ø1		
1E	23	INX H	/next dot
1F	C3	JMP LUP 2	/go back
20	0E		
21	ØI		

TABLE 4 : TEST PROGRAM FOR ALT-256**2

COLOR/GREY SCALE*

Each ALT-256 card has a built-in crystal controlled sync generator. However, each card can also be synchronized to an external sync source. This feature is extremely useful for applications requiring more than one bit per pixel (grey scale or color). Multiple cards are required for this application (up to 24 cards can be synchronized).

VIDEO AND SYNC SIGNALS

All video and sync signals are available on a 16 pin plug-in socket at position S1. These signals can be used for a variety of applications:

PIN	NAME	COMMENT
12	VDO	Composite video output. Can be directly connected to a TV monitor video input via 75 ohm cable
4	ALPHA	Alphanumeric video input. (From MTX-1632SL for video mixing). Has to be low when not used. (Jumper W24 in).
13	SV	Vertical sync signal (Positive pulse). It can be used to drive a TV monitor vertical deflection circuit or it can drive an MTX-1632SL alphanumeric VRAM.
10	SH	Horizontal sync signal (positive pulse).
11	BV	Vertical blank. This signal is low during vertical retrace (3 msec). It is available to a uP by reading I/O port Ø.
7	BH	Horizontal blank (low during blank).
3	D OUT	Serial video signal (TTL level, high-white, low-black). It can drive directly a TTL compatible TV monitor video input or it can be used as one bit of video information in a color/grey scale system.
2	DOT CLOCK	Bidirectional dot clock input/output (depends on M/S control jumper W26). If W26 is in, ALT-256 is a master card and DOT CLOCK is an output. If W26 is out, ALT-256 is a slave card and DOT CLOCK is an input.
15	RESET H	Bidirectioanl horizontal reset input/output. (80 nsec negative pulse which synchronizes horizontal counters).

* For more information on color/grey scale applications consult Matrox's MTX-256 color/grey application note.

PIN	NAME	COMMENT
16	RESET V	Bidirectional vertical reset input/output. (80 nsec negative pulse) Synchronizes vertical counters.
1,8,9,16	GND	Ground

The following jumpers determine the use of the ALT-256 in color/grey scale applications.

W26 Determines if card will be a master or slave card (generates its own sync signals or accepts external sync signal from a master card). Sync signals are DOT CLOCK, Vertical reset and Horizontal reset.

W26 - in	ALT-256 is a master card
W26 - out	ALT-256 is a slave card

W25 Video data input bit connection.

W25 - in	Video data input bit is connected to DØ.
W25 - out	The user can connect video data bit to any of the data bus bits D0-D7 with a wire.

W24 Alpha input

W24 - in	The socket S1 pin 4 is grounded (no alpha)
W24 - out	Alphanumeric input at pin 4 of S1 is added to the graphic video.

NOTE: All ALT-256 are shipped with jumpers W24, W25, W26 in. (Standard configuration for a single level ALT-256 application).

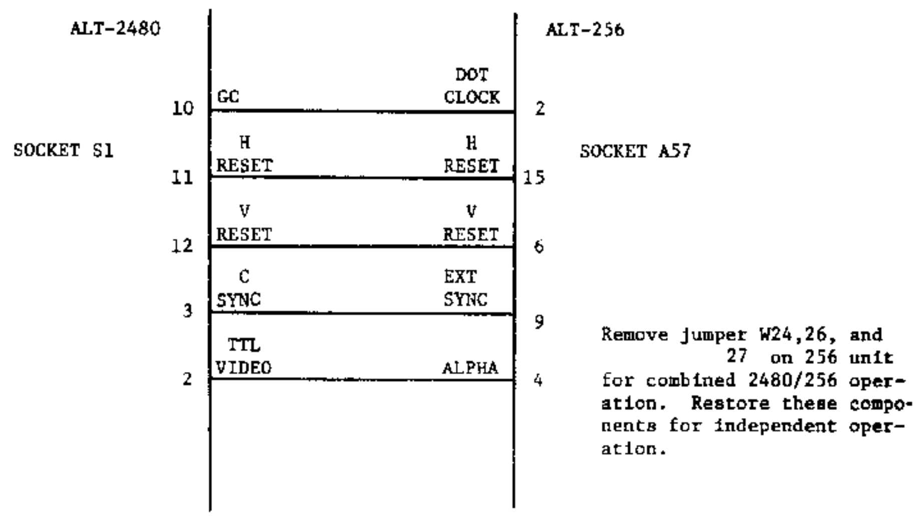


FIG. 5 _ COMBINING ALT-2480 and ALT-256

The ALT-2480 and -256 display cards are directly compatible with one another. This powerful feature permits generation of a combined alpha-numeric/graphic display with no extra hardware. Multiple ALT-256 cards can also be slaved to a single ALT-2480 master for color/grey scale applications.

Figure 5 shows how the ALT-2480 and ALT-256 cards are connected together. The connection is accomplished via 16 pin DIP plugs on each card. Some jumpers must be altered on the ALT-256. The combined composite video output is taken from the ALT-256 output.

SUMMARY OF MTX-GRAPH

SOFTWARE PACKAGE

The MTX GRAPH software package is designed for use with the Matrox ALT-256 graphics display. The package is configured as a series of callable sub-routines. The MTX GRAPH package occupies memory locations 0104 to 04FF (Hex). The package incorporates the following features:

1. Variable Resolution: The display resolution can be selected to be 256 x 256; 128 x 128; or 64 x 64; by software command.

2. Point Plot: Any arbitrary point can be displayed by specifying X-Y coordinates. Dot size depends on the resolution selected.

3. Line Vector Graphics: Line can be drawn by specifying the two end points.

4. Alphanumeric Display: A full ASCII character generation routine is incorporated. Control characters are correctly interpreted. Characters can be positioned anywhere on the screen.

5. Animation Synchronization: This feature allows the user to generate animation synchronization at line rate.

5. Color Option: The software package will support a 3 card color/grey scale system.

More detail on the above is contained in the MTX GRAPH user manual.

A second program supplied is intended for demonstration purposes. The program utilizes the MTX GRAPH sub-routines to create a continuous live action graphics display.

The paper tapes are supplied in Imsai binary loader compatible format. A listing of the loader and instructions for its use are provided with the manual.

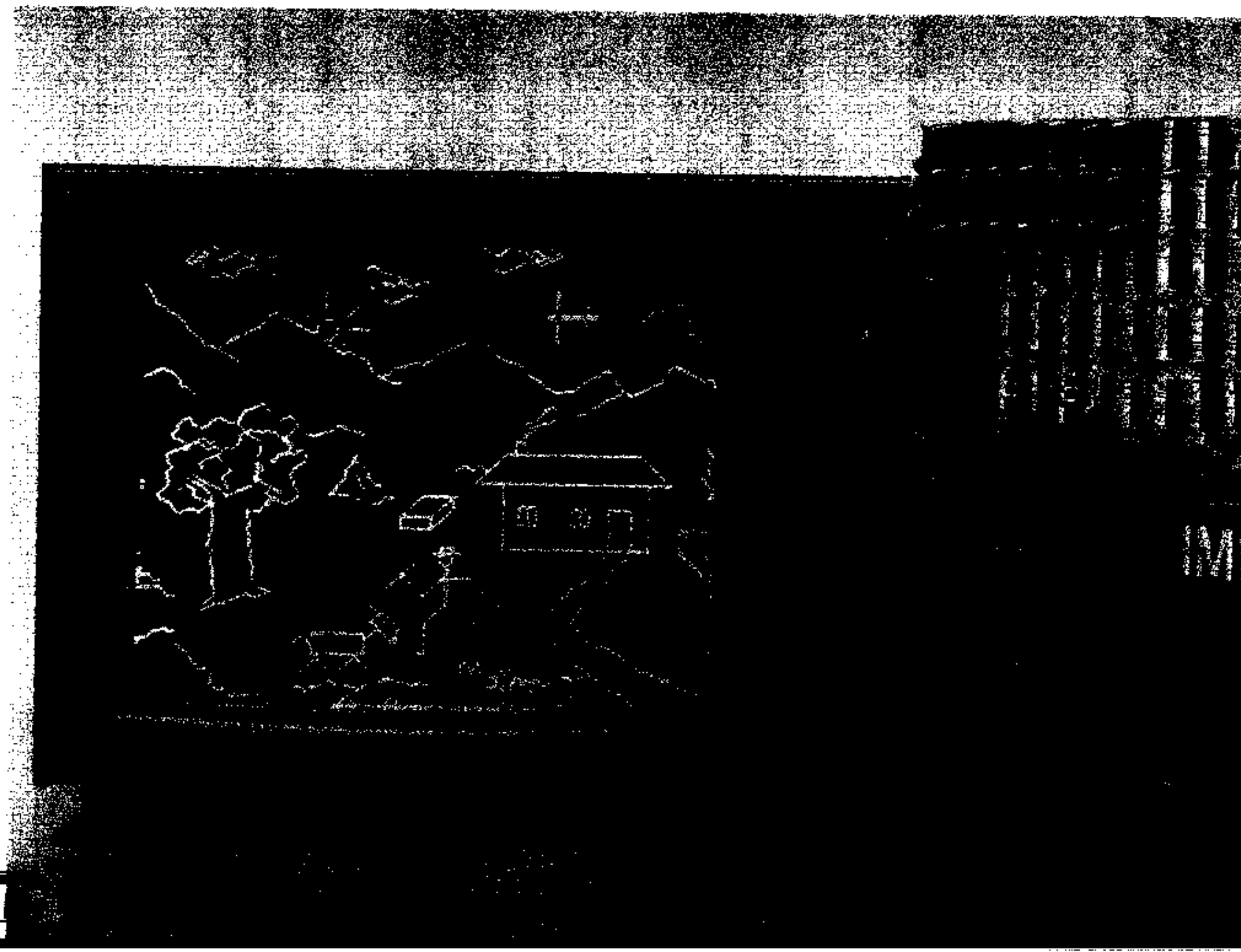
The Imsai loader should be used to load first MTX GRAPH and then the demo program. Both must be co-resident to use the demonstration program. The ALT-256 should be jumpered for address 10-13 (hex) and data bit DØ. (jumpers A2, 3, 5, 6, 7 set to Ø, A4 set to 1. See manual P9.) Start the computer at location 0500 Hex. Sit back and watch the show. The demonstration program will pause whenever the data switches are set to 01 (hex).

NOTE: Old versions of the ALT-256 require the following hardware change to work with this software package: (units shipped prior to Aug. 1/77). Tie A3 and A11 pin 4 to +5V with jumper wires soldered directly to the artwork.

PRICE: $25.00 for MTX GRAPH manual plus binary paper tapes of MTX GRAPH and demo program.

ALT — 256 ** 2

ALTAIR — IMSAI BUS COMPATIBLE GRAPHIC DISPLAY INTERFACE



SPECIFICATIONS FOR ALT-256**2 GRAPHIC DISPLAY INTERFACE

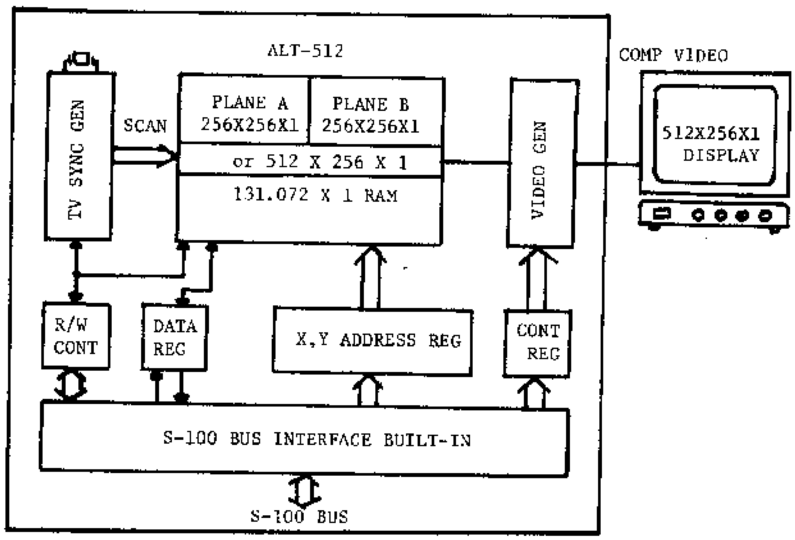
INTRODUCTION:	The ALT-256 low cost high resolution graphics interface can be used in a wide range of applications. Typical examples range from video games like electronic etch-a-sketch, to industrial uses such as computer image processing. The unit has 4 times the resolution of other S100 bus graphics and includes features such as expansion for color/grey scale, external sync capability, on card refresh memory and direct S100 bus compatibility. The ALT-256 is compatible with the ALT-2480 permitting a powerful combined alphanumeric/graphic display.
RESOLUTION:	256 X 256 dot raster.
ACCESS TIME:	3.4 usec. max /dot: each dot induvidually addressed
ERASE:	Single instruction erases screen: 33 msec. max.
REFRESH MEMORY:	Built-in on the card: 65,536 X 1 bit memory
BUS:	S100, plugs directly into Altair-Imsai bus
COLOR/GREY SCALE:	Up to 24 bits/pixel (2^{24} different colors or grey levels/dot) by using identical multiple boards.
DIMENSIONS:	9" X 5"
POWER:	8V, 600mA; 18V, 100mA; -18V, 10mA, on board regulators
OUTPUTS:	Composite video; 75 Ohm, x-tal controlled: TTL video, horizontal and vertical syncs and blanks outputs.
SYNCHRONIZATION:	Built-in TV sync generator (x-tal controlled). Free running or external sync capability.
TV STANDARD: 4	American standard (262 vertical lines, 60 Hz; 240 vertical video lines) 4:3 aspect ratio; American Non-standard(280 lines, 60 Hz; 256 video lines), horizontal freq. = 16.8 KHz. (1:1 aspect ratio); European (312, 50 H 1:1 aspect ratio). Non-interlaced picture. Standard selectable on the boar
MONITOR:	Any standard TV monitor or modified TV set.
REMOTE DISPLAY:	75 Ohm cable, up to 2,500 feet,; multiple monitors, up to 25 TV's.
ADDRESSING:	Four output ports and one input port built-in. Port address selectable on the board.
X-Y PLOT MODE:	X coordinate (output port 1, 8 bit registor); Y coordinate (output port 2, 8 bit registor). Data port (output port Ø, 8 bit registor). Outputting data H"Ø1" to port H"ØØ" writes a white dot at the point addressed by the X and Y registors. Writing data H"ØØ2 to port H"ØØ" writes a black dot.
ERASE:	Outputting data to port 3 will erase the screen. Data H"ØØ" will set all bits to 0 (black). Data H"Ø1" will set all bits to 1 (white.)
DISPLAY STATUS:	Available by reading input port Ø. Data bit DO = graphic interface busy flag; D1 = vertical blank.
DOCUMENTATION:	12 page manual; complete description, circuit schematics and a test program. Additional color/grey scale application note available on request.
WARRANTY:	90 days parts and labor.
SOFTWARE:	A complete software package is available for $28.
ORDERING:	Available directly from Matrox Electronic Systems Ltd., Montreal or from franchised distributors. Specify options desired. Delivery 2-4 weeks.

matrox electronic systems ltd.
2795 BATES RD., MONTREAL, QUE. H3S 1B5, CANADA
TEL.: (514) 481-6838 or 735-1182 TELEX: 05-825651

matrox electronic systems ltd.
2795 BATES RD., MONTREAL, QUE. H3S 1B5, CANADA
TEL.: (514) 481-6838 or 735-1182 TELEX: 05-825651

ALT-512

S100 BUS 512 x 256 GRAPHICS DISPLAY



* 512 X 512 X 1 resolution or 256 X 256 X 2 resolution
* Built-in 131K X 1 RAM
* Single instruction erase
* 1.4 usec access time
* External/internal sync
* 50/60 Hz
* Complete software control
* Grey scale mode (4 level)
* Multiple boards stackable for color/ grey applications
* Can be combined with ALT-2480 board
* Two independent 256 X 256 X 1 planes allow 8 display formats
* Ideal for live animation effects
* Displays one plane while CPU writes into other
* Works with any 8080, 8085, Z80 or equivalent (4 or 8 mHz clocks)

The ALT-512 is a complete graphics display controller on a single S100 bus plug-in board. It contains its own 131.072 bit refresh memory, TV sync and video generator, and all I/O for S100 bus. Each display dot (pixel) is addressable via X-Y registers and can either be written into or read out. The board has built-in six outputs and two input ports.

The display field consists of two 256 X 256 X 1 planes. The two plane arrangement allows the user to implement eight different display formats by software control/ (CONTROL register). Either or both planes can be displayed in various combinations.

*MATROX products covered by Canadian and foreign patent and/or patent pending.

September '78

SPECIFICATIONS ON THE ALT-512

RESOLUTION : Eight display modes (software controlled via CONTROL registers_.
256 X 256 X 1 Plane A displayed/B off
256 X 256 X 1 Plane B in/A off
256 X 256 X 1 Plane A & B superimposed (digital video or)
256 X 256 X 2 4 level grey scale display ($A=2^0$, $B=2^1$)
These four modes can be displayed with planes A and B one over the other or plane A horizontally one dot to relatively to plane B. (effectively 512 X 256 resolution when both A and B are dos-played).

ADDRESSING: : The ALT-512 has built-in six I/O ports user positioned addressed as follows:

Output ports

Address	7	6	5	4	3	2	1	0	
0	-	-	-	-	-	-	-	D0	Dot register; D0=black, D0=1 white
1	X7	.	.	.	.	.	.	X1	X register
2	Y7	.	.	.	.	.	.	Y0	Y register
3	-	-	-	-	-	-	-	D0	Clear register
4	-	-	-	-	-	-	-	X0	X0=0, plane A; X=1 plane B
5	--	-	C4	.	.	.	.	C0	Control register (display mode)

Input ports

Address	7	6	5	4	3	2	1	0	
0	-	-	-	-	-	-	-	D0	Read pixel port
1	-	-	-	-	B	L	H	V	Flag register,(blanks, light pen)

ERASE : Single instruction erases screen 33 msec maximum

REFRESH : Built-in 131,072 X 1 bit. Independent of the CPU timing.

BUS : S-100. Plugs directly into any S-100 bus systems.

COLOR ; Up to 24 bits by using multiple boards. 3 boards enough for 512 X 256 X 8 color systems or 256 X 256 X 64 color system (RGB)

DIMENSION : 9" x 5" S-100 standard 100 pin connector

POWER : 8V, 600mA; 18V, 100mA; -18V, 10mA on board regulators

OUTPUTS : Composite video; 75 ohm X-tal controlled 50/60hz, TTL video, horizontal and vertical syncs, blanks.

SYNCHRONIZATION: Built-in TV sync generator. Free running or external sync capability.

TV STANDARD : American (60 Hz) 262 lines non-interlaced. 240 displayed.
European (50 Hz) 314 lines non-interlaced. 256 displayed.

TV MONITOR : Any standard TV monitor or modified TV set.

REMOTE DISPLAY : 75 Ohm cable, up to 2500 sect, Multiple monitors up to 25 TV.

PRICE : $595 FOB Montreal; single

SOFTWARE : A complete software package is available for $28.

ORDERING : Available directly from Matrox Systems Limited, Mtl. or franchise distributors

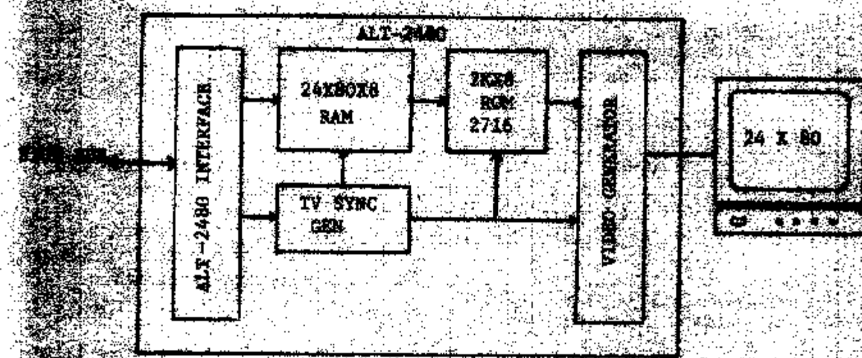
DELIVERY : Two to four weeks ARO.

[illegible] metrox electronic systems ltd.
[illegible] RD., MONTREAL, QUE. H3S 1B6 CANADA
[illegible] or 735-1182 TELEX: 05-[illegible]

ALTR-24[illegible]

[illegible] CRT CONTROLLER FAMILY — ALPHANUMERIC DISPLAY

* NEW EXCLUSIVE TRANSPARENT MEMORY FEATURE



[illegible]2480 is a unique single board video interface between S100 type computer [illegible] monitor. The board allows S100 users to add video display to their [illegible] new ones. The ALT-2480 incorporates a revolutionary display memory [illegible] is completely transparent. Alphanumeric video display boards made by [illegible] all other manufacturers suffer from interference or streaking when [illegible] accessed. This occurs because the display memory must be accessed by [illegible] and sync generator. The standard solution is to access the [illegible] or to use DMA. Both approaches add complexity and drastically [illegible]. The Metrox transparent memory is a revolutionary [illegible] problem. It is not necessary to wait for blanks, no DMA is [illegible] at full speed, and there is no streaking on the screen no matter [illegible] accesses the board.

- [illegible] directly in S100 bus
- [illegible] lower case/graphics
- [illegible] X 80 characters or 2 pages [illegible] 24 X 40 characters
- [illegible] (4K X 8)
- [illegible] memory
- [illegible] programmable character generator [illegible] EPROM)
- * External/internal sync
- * Normal/inverse/blank control
- * Drives TV monitor directly
- * [illegible]
- * [illegible]
- * [illegible] programs
- * Software package available.
- * Full software control

[illegible] SPECIFICATIONS FOR ALTE-2400 DISPLAY [illegible]

[illegible] : The ALT-2400 [illegible] S100 bus microcomputer and a TV [illegible] standard 24 line X 80 character [illegible]

[illegible] : 24 lines X 80 characters [illegible]

[illegible]

[illegible] : S100 bus plug-in.

[illegible]

BYTE May 1978
Volume 3 Number 5

Product Description:

The Matrox ALT-256 Video Board

Gary Ruple
Matrox Electronics Systems
POB 56 Ahuntsic Sta
Montreal Quebec H3L 3N5
CANADA

The ALT-256 is an Altair (S-100) bus compatible graphics card that gives a resolution of 256 by 256 dots. This display is useful for professional graphics applications such as computer aided design, simulation, business and educational displays, and plotting curves.

The display memory on the ALT-256 consists of 16 4 K dynamic memory integrated circuits in the 16 pin package. The dynamic memory refresh is handled by the video scan circuitry so that no processor time is required for this function.

Board Addressing

The board is addressed as four contiguous output ports and one input port (IO ports are built in), selectable by on board address jumpers. The dot addressing is done in X–Y fashion. Output ports 1 and 2 are used as registers for the X and Y positions, respectively, to position the cursor at the selected dot. Output 0 is used to write the intensity of the dot: 00 for black and 01 for white. After the dot intensity is loaded, the ALT-256 will require 3.4 μs to write the dot in the display memory (ie: to allow for internal synchronization of the wire operation, video read scan and dynamic memory refresh). Since the 8080 processor almost always requires more than 3.4 μs to load the next dot address and data, the processor can run at its full speed. Also, there will be no streaks or flashes on the display no matter how fast the

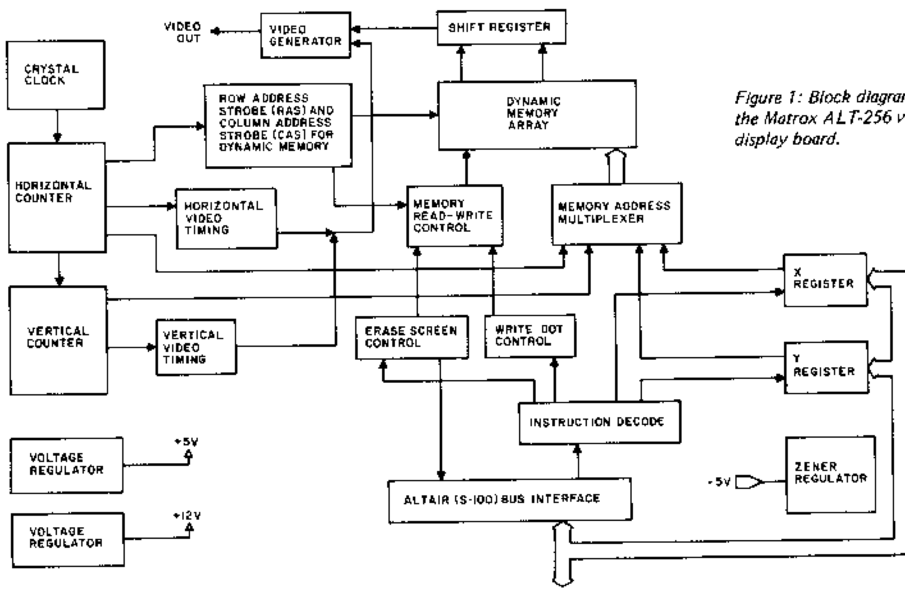


Figure 1: Block diagram of the Matrox ALT-256 video display board.

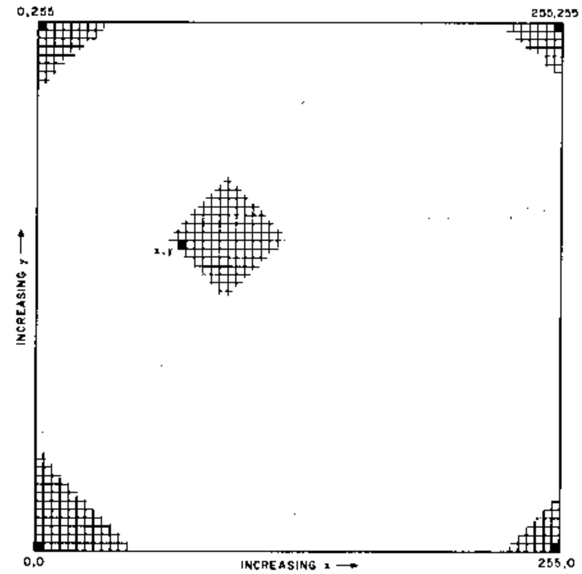


Figure 2: Display coordinate system for video displays (MTX-Graph software).

processor accesses the card since the write operation is internally synchronized.

Output port 3 is used as an erase screen command to either clear the screen or to set it all white according to data bit 0: 0 = black, 1 = white. The erase operation can take up to 33 ms. The status port (IN0) has been provided as a means of checking the status of the ALT-256. The following is a description of the status bits:

Bit 0: When 0 the display is ready.
When 1 the display is being erased and should not be accessed.

Bit 1: When 0 the display scan is in the video portion.
When 1 the display scan is in the vertical blanking period allowing access of the display when not visible. This is useful for dynamic motion and animation synchronization.

The X–Y addressing scheme provides the programmer with a way to address individual dots. A horizontal or vertical line requires only one address to be updated for each new dot. A 45° diagonal requires each address to be incremented or decremented. Using output ports as registers and as a means of addressing the card also conserves memory space, since the 65,536 directly addressable dots are equivalent to 8 K bytes of memory. (You would also have to keep track of dot position in the byte.)

Theory of Operation

The ALT-256 has four major blocks: the video sync generator, scanning circuitry, cursor and interface electronics, and 65,536 by 1 memory (see figures 1 and 2). The sync generator consists of a crystal oscillator and a divider chain. This divider chain produces all timing signals for the memory scanning as well as horizontal and vertical sync. The video sync generator can be programmed by jumpers for either the European or the American video standard.

The scanning circuitry consists of multiplexers which provide proper address, and read and write signals for the programmable memory.

The cursor consists of two 8 bit latches (the X and Y address registers) which are loaded by the processor. Necessary interface address and timing decoding is accomplished by the Altair (S-100) bus interface logic.

The refresh memory has 16 4 K dynamic memories (IC type 4096) organized as a 65,536 by 1 bit memory.

Power supplies for 5 V, 600 mA; 12 V, 100 mA, and −5 V, 10 mA are generated by on board voltage regulators.

Multiple Boards

The ALT-256 may be used in multiple board systems for color graphics or grey scale applications. Any reasonable number of boards may be used. When more than one is used, one board must be chosen as the master and the others are configured as slaves synchronized to the master. There is an on board jumper allowing use as a master or slave and a socket provided for connecting the sync signals and video between master and slaves. Single or multiple boards may also be slaved to an external sync generator such as a TV camera signal or broadcast video.

Software

MTX Graph, the software package available for the ALT-256, provides all the commonly used low level graphics routines, and is configured as a series of subroutines that occupy hexadecimal memory locations 0104 to 04FF. Multiple boards can be supported by up to a maximum of eight bits of color or grey scale information. Features of the package are described in table 1.

Photo 1a.

Photo 1b.

Photo 2a.

Photo 2b.

Photo 3a.

Photo 3b.

Photos 1 to 3: Some examples of grey scale and pseudocolor images produced by the ALT-256 video board. The grey scale photos were produced by feeding the output from a TV camera through a slow scan analog to digital converter. Next, the 3 bit digitized output was processed through three ALT-256 video boards and a digital to analog converter to give an 8 level video signal. The color pictures were produced by feeding the outputs from the video boards directly to the red, blue and green inputs of an RBG color monitor (see figure 3).

Table 1: Features of MTX GRAPH, a software package available for the ALT-256 graphics board.

1. **Variable Resolution:** The display resolution may be set to 256 by 256, 128 by 128, or 64 by 64 through software control.
2. **Point Plot:** A dot corresponding in size to the resolution selected may be displayed at any arbitrary point by specifying X-Y coordinates.
3. **Line Vector Graphics:** Lines can be drawn from the current cursor position to the endpoint specified by the user.
4. **Alphanumeric Display:** A full ASCII character generation routine is provided. Characters can be positioned anywhere on the screen. Carriage control characters are correctly interpreted. Character size is adjustable.
5. **Animation Synchronization:** This feature is used to synchronize animated display updates with the vertical scan.
6. **Color Option:** The software package supports a 3 card color or grey scale system as described later in this article.

All subroutines use standard Cartesian coordinates with the display occupying the first quadrant. As shown in figure 2, the origin (X = 0, Y = 0) is defined as the bottom left point on the display. X increases in value to a maximum of 255 at the right edge, while Y rises to a maximum of 255 at the top. If the ALT-256 is jumpered for American Standard scan (240 lines displayed), the lowest Y coordinate displayed is 16, and points with coordinates from 0 to 15 cannot be seen.

Grey Scale and Color Applications

In the grey scale configuration, multiple boards provide binary intensity information. All boards have the same address decoding and sync signals. Each board has a dot intensity bit (output port 0) which is normally tied to data bus bit D0. In a typical 3 board color or grey scale system, the intensity bit is changed on two boards to be D1 and D2 (or any other bits you choose), thus writing three binary bits for each write operation. This method allows computation of grey scale and single instruction load of all bits making up a single picture element (often contracted to "pixel" in graphics literature). The TTL video outputs from the three boards are fed to a simple 3 bit digital to analog converter.

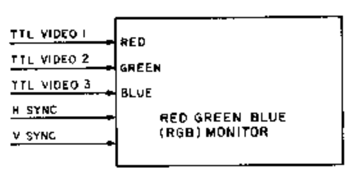


Figure 3: Connecting an RBG. (red green blue) monitor for 8 color operation using three ALT-256 boards.

RGB (Red Green Blue) Color

In a color scheme, the best results are obtained by directly driving the red, green and blue guns of a color monitor with the video signal from three boards, as in figure 3. An ordinary color TV can be modified to accept separate color inputs. This should not be undertaken by anyone who lacks an understanding of color TV and electronic design. The alternative is to generate an encoded composite color signal. An application note available from Matrox gives details on a color encoder circuit.

Photographs

The photos accompanying this article were generated by feeding a TV camera output through a slow scan analog to digital converter. The 3 bit digitized output was then fed to a 3 card ALT-256 graphics system. The grey scale pictures were produced by feeding the outputs of the three cards to a 3 bit digital to analog converter. The resulting pictures have eight discrete grey levels. The color pictures were produced by feeding the outputs of the three cards to the red, blue and green (RBG) inputs of an RBG color monitor. The resulting eight color pictures are pseudocolored. This means that a different color has been assigned arbitrarily to each grey level in the original picture. Pseudocoloring is used in many industrial and research applications (ie: many NASA space pictures are processed this way).

Conclusion

The Matrox ALT-256 represents one approach to high resolution graphics capabilities for the Altair (S-100) bus. Multiple board systems can be used for medical displays, research applications, pseudocolor imaging, fast animated displays, computer aided design, sophisticated computer games and computer generated art. For the Star Trek freak, now there is available a real (if imaginary) universe to save, rather than a slow printer banging out descriptions. For the artist, a canvas; the researcher, a window; and the kids, an electronic sketch pad.■

Note: The completely assembled, tested and burned in ALT-256 board is available for $395 from Matrox Electronic Systems, POB 56, Ahuntsic Station, Montreal, Quebec H3L 3N5 CANADA. Also available is the ALT-2480, an Altair (S-100) compatible alphanumeric generator board, which can be used in conjunction with the ALT-256 to produce *simultaneous* graphics and alphanumeric displays.

matrox electronic systems ltd.
2795 BATES RD., MONTREAL, QUE. H3S 1B5, CANADA
TEL.: (514) 481-6838 or 735-1182 TELEX: 05-825651

SUMMARY OF MTX-ALPHA

SOFTWARE PACKAGE

The MTX-ALPHA software package provides the user the full flexibility of a software based intelligent terminal. The software package has been written in 8080 assembly language and will work with most of the Matrox 2480 family of cards (MTX-2480, ALT-2480, MSBC-2480). MTX-ALPHA has been designed explicitly to support easy and reliable modification to meet varying user requirements. The program occupies approximately 3K of memory.

As supplied, the package will fully emulate the popular Lear Siegler Inc. ADM-3A and Digital Equipment Corp. DECSCOPE VT-52 interactive display terminals. In addition, line at a time and text block input modes are available to provide the powerful text preparation features of an intelligent terminal. More detail on the three basic input modes is provided below:

FULL DUPLEX MODE (FDX)

In this mode no processing is performed on input. For a typed in character to appear on the display, it must be echoed by the user program. Characters are passed unaltered to the user program as soon as they are input. This mode mimics the operation of a dumb terminal or teletype.

HALF DUPLEX MODE (HDX)

Half duplex mode buffers characters as they are input until a full line is typed. A full line consists of either 80 characters or 0 through 79 characters followed by a CR, ESC, or LF. Rubout will delete the last character in the buffer (and on the screen), while ^U(control U) will cancel the entire line. Once a line is complete, the buffer may be passed on to the user program.

BLOCK MODE

In block mode, the user can generate an entire block of data using all the editing capabilities of the intelligent terminal system. By inputing the XMIT (end of text) code (^D), all the data on the screen entered since the last XMIT code is sent to the user program. Examples of editing commands include:

- Programmable wrap around or scroll
- Insert/delete line
- Insert/delete character
- Horizontal and vertical tab
- Cursor motion and cursor home commands
- Programmable display line length.

matrox
electronic systems ltd.
2795 BATES RD., MONTREAL, QUE. H3S 1B5, CANADA
TEL.: (514) 481-6838 or 735-1182 TELEX: 05-825651

SUMMARY OF MTX-GRAPH
SOFTWARE PACKAGE

The MTX GRAPH software package is designed for use with the Matrox ALT-256 graphics display. The package is configured as a series of callable sub-routines. The MTX GRAPH package occupies memory locations 0104 to 04FF (Hex). The package incorporates the following features:

1. Variable Resolution: The display resolution can be selected to be 256 x 256; 128 x 128; or 64 x 64; by software command.

2. Point Plot: Any arbitrary point can be displayed by specifying X-Y coordinates. Dot size depends on the resolution selected.

3. Line Vector Graphics: Line can be drawn by specifying the two end points.

4. Alphanumeric Display: A full ASCII character generation routine is incorporated. Control characters are correctly interpreted. Characters can be positioned anywhere on the screen.

5. Animation Synchronization: This feature allows the user to generate animation synchronization at line rate.

5. Color Option: The software package will support a 3 card color/grey scale system.

More detail on the above is contained in the MTX GRAPH user manual.

A second program supplied is intended for demonstration purposes. The program utilizes the MTX GRAPH sub-routines to create a continuous live action graphics display.

The paper tapes are supplied in Imsai binary loader compatible format. A listing of the loader and instructions for its use are provided with the manual.

The Imsai loader should be used to load first MTX GRAPH and then the demo program. Both must be co-resident to use the demonstration program. The ALT-256 should be jumpered for address 10-13 (hex) and data bit DØ. (Jumpers A2, 3, 5, 6, 7 set to Ø, A4 set to 1. See manual P9.) Start the computer at location 0500 Hex. Sit back and watch the show. The demonstration program will pause whenever the data switches are set to 01 (hex).

NOTE: Old versions of the ALT-256 require the following hardware change to work with this software package: (units shipped prior to Aug. 1/77). Tie A3 and A11 pin 4 to +5V with jumper wires soldered directly to the artwork.

PRICE: $25.00 for MTX GRAPH manual plus binary paper tapes of MTX GRAPH and demo program.

matrox
electronic systems ltd.
2795 BATES RD., MONTREAL, QUE. H3S 1B5, CANADA
TEL.: (514) 481-6838 or 735-1182 TELEX: 05-825651

MTX-256

MTX TV CRT CONTROLLER FAMILY

GRAPHIC DISPLAY

GRAPHICS

COLOR / GRAY SCALE

APPLICATION NOTE

by
B. MATIC and L. TROTTIER

April 1977

*MATROX products covered by Canadian and foreign patent and/or patent pending.

MTX-256 GRAPHIC DISPLAY APPLICATION NOTES

INTRODUCTION

The MTX-256 is a single PC board which can be used for a variety of black and white or color graphic systems. Very low cost, versatility, high speed and extreme simplicity in interfacing in hardware and software makes the MTX-256 an ideal solution for any uP or minicomputer application requiring a graphic display.

Each card has its own built-in TV sync generator. The sync generator can also be synchronized to external TV sync signals. This feature is extremely useful for applications requiring more than one bit per pixel (grey scale or color). Multiple cards are required for this application (up to 24 identical cards).

Each card provides a number of video signals which can be used to drive a TV monitor or TV set directly. Jumpers on the card allow conversion between American and European TV standards.

Interfacing to a particular uP or minicomputer is very simple and requires a minimum of additional hardware. Furthermore, a number of different versions of the MTX-256 are available which plug directly into some of the most popular computer buses (Altair-Imsai, PDP-11, SBC-80). The MTX-512 family of graphics cards features resolutions of 256 X 256; 256 X 512; 512 X 512 and 256 X 1024. Bus compatibility for the Digital Equipment PDP-11, LSI-11 and Intel SBC-80 is featured. Although written specifically for the MTX-256 card, the principles described in this application note may be applied to all Matrox graphics products.

THEORY OF OPERATION

The MTX-256 has four major functional blocks: The TV sync generator, scanning circuitry, cursor and interface electronics, and a 65,536 X 1 internal refresh memory. (See Fig. 1).

The sync generator is formed of a crystal oscillator and a divider chain which produces all timing signals for the memory scanning as well as horizontal and vertical sync. The TV sync generator can be programmed by jumpers for the European or American TV standard.

The scanning circuitry consists of address multiplexers which provide the proper address, R/W and clock signals required for the refresh memory.

The refresh memory has 16 4K dynamic memories (4096, 16 pin) organized as a 65,536 X 1 bit memory.

The cursor control section consists of two 8 bit latch/up/down counters which are loaded by the uP. The interface electronics incorporates an instruction decoder which determines the internal operation to be executed. The uP can load data into the MTX-256 without interference regardless of the position of the TV electron beam, since the TV scan, dynamic memory refresh and write are internally synchronized.

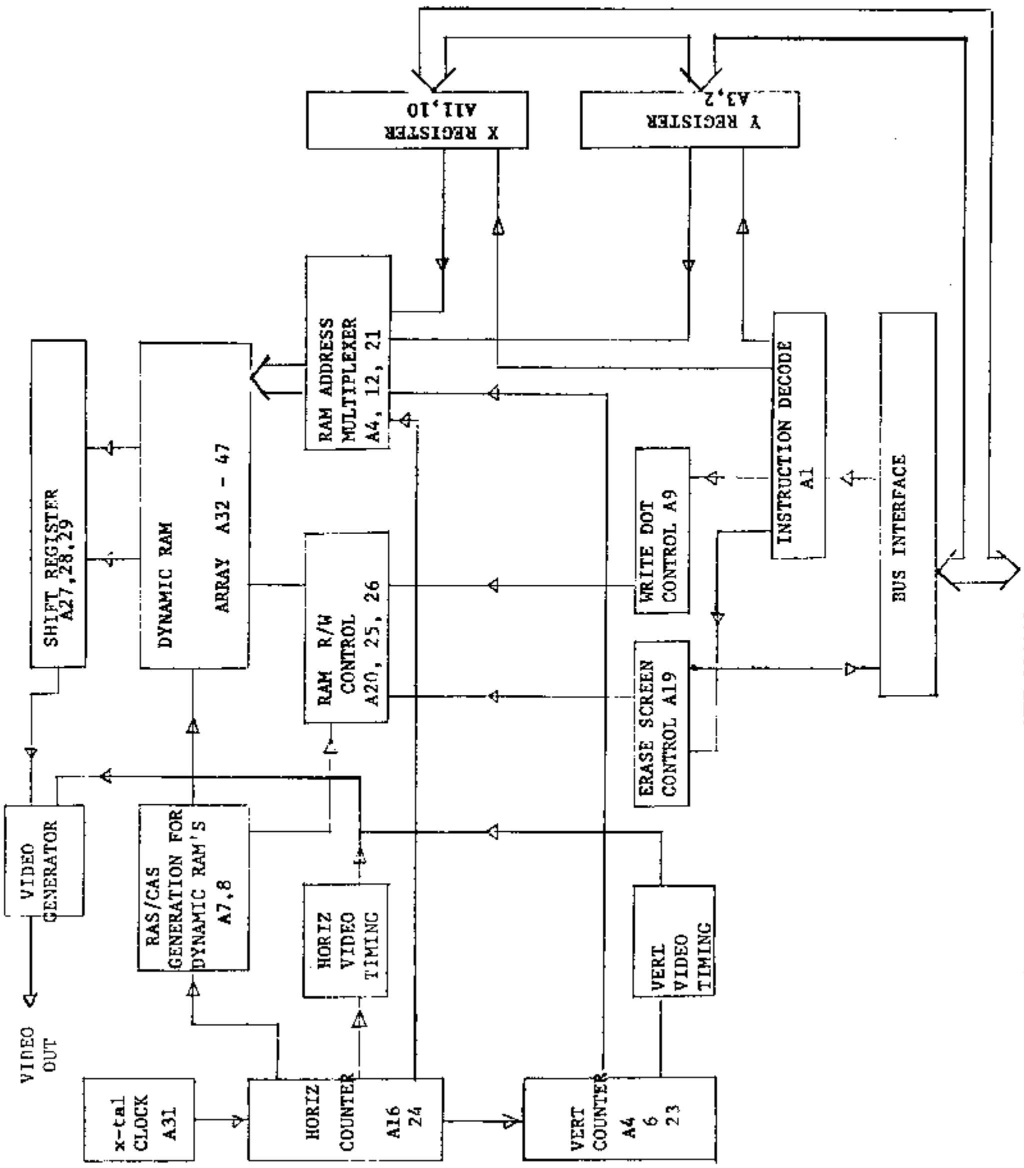


FIG. 1: BLOCK DIAGRAM FOR MTX-256**2

PIN ASSIGNMENT FOR STANDARD MTX-256**2 (44 PIN CONNECTOR)

PIN	NAME	COMMENT
		SIGNALS FROM uP
N	D0	Data input bit Ø. Directly connected to the uP data bus.
12	D1	Data input bit 1.
P	D2	Data input bit 2.
13	D3	Data input bit 3.
R	D4	Data input bit 4.
14	D5	Data input bit 5.
S	D6	Data input bit 6.
15	D7	Data input bit 7.
6	DD0	Refresh memory video data bit input. Can be connected by jumper to one of data bus bits. For a single MTX-256 is normally connected to D0.
17	CS	Chip select (high), selects the MTX-256 in the uP address space.
U	$\overline{CS}$	Chip select (low).
V	MAØ	MTX-256 Address bit Ø.
18	MA1	MTX-256 Address bit 1.
T	R/$\overline{W}$	Read/write pulse (low for write)
		SIGNALS TO uP
M	READY	Flag signal from MTX-256. Goes low during a write operation (write dot) into location Ø. (3.3 usec max). Can be used as a busy flag if required.
11	READY LOAD	Flag signal output. Goes low during a write operation (clear screen) into location 3 (33 msec max). Can be used as a busy flag if required.
		CONTROL SIGNALS
A	M/S	Master/slave control input. When grounded, the MTX-256 will become a master card and it will supply its internal clock and sync reset outputs to all other slave cards. When open (high) the MTX-256 becomes a slave card and it will accept external control signals. (M/S must be grounded if only one MTX-256 is used).

PIN	NAME	CONTROL SIGNALS
B	Dot Clock	Bi-directional dot clock input/output (7 mHz) (depends on the M/S control input.).
2	RESET H	Bi-directional horizontal reset input/output. (80 nsec negative pulse which synchronizes horizontal counters on all slaves.
4	RESET V	Bi-directional vertical reset input/output. (80 nsec negative pulse which synchronizes vertical counters on all slaves).
1	VE	Vector plot enable input. When grounded enables vector plot. When high (open) disables vector plot. Allows user to use all 8 data bits as dot intensity/color if vector plot is disabled.
		SIGNAL TO TV MONITOR (COMPOSITE)
8	VDO	Composite video output. Can be directly connected to TV monitor video input via 75 ohm cable.
		SIGNALS TO TV MONITOR (SEPARATED)
5	TTL VIDEO	Serial video signal from refresh memory. (TTL level, high-white, low-black). It can drive directly a TTL compatible TV monitor video input or it can be used as bit of video information in a color/grey scale system.
K	SH	Horizontal sync (positive TTL pulse). It can be used to drive the TV monitor horizontal deflection circuit or it can drive an MTX-1632SL alphanumeric VRAM.
9	SV	Vertical sync (positive TTL pulse). It can be used to drive the TV monitor vertical deflection circuit or it can drive an MTX-1632SL VRAM.
		ADDITIONAL VIDEO SIGNALS
10	BV	Vertical blank output. This signal is low during vertical retrace (about 3 msec.) It can be used as a flag for special video effects. (dynamic animation, frame switching, etc) .
H	ALPHA	Alphanumeric TTL video input from MTX-1632SL. If used, alphanumeric data from the MTX-1632SL is superimposed on the graphic picture. (Grounded when not used. Otherwise screen will be blank).

PIN	NAME	POWER SUPPLY
X,20	+5V	
E	+12V	
3,C	GND	

The above pin assignments differ somewhat for the other cards in the Matrox graphics family. Consult ALT-256 and MTX-512 family data sheets for exact pin assignments. (See Fig. 2,3, and 4 for typical MTX-256 interfaces to the 8080, 6800 and HP 21XX computers).

MIXING ALPHANUMERIC AND GRAPHIC VIDEO

Certain applications require an alphanumeric text in addition to graphics. This can be done in two ways.

By using software, the uP can plot characters from its ROM character generator (part of the uP program). This method allows great flexibility in character set, size and position of text. However, it requires a considerable uP overhead (time and memory) since a character has to be plotted dot by dot, the method is most useful for special alphabets such as Chinese, Japanese, Arabic, etc. The Matrox 8080 based MTXGRAPH software package already includes this capability.

The second method uses the MTX-1632SL alphanumeric VRAM. The VRAM is slaved to the MTX-256 and its video is added to the graphic video to produce a combined graphic and alphanumeric picture. (See Fig. 5). The MTX-256 can itself be slaved to the MTX-2480 text display board (see Fig. 5A). The result is a powerful high resolution graphics/alphanumeric display. Multiple graphic cards can be synched to a single 2480 for color/grey scale applications.

This method gives very fast writing and minimum uP overhead. The method is limited to the extent that the character set and position screen position is fixed. The MTX-1632SL must also be interfaced separately to the host computer system. (See Fig. 6).

INTERFACING MULTIPLE MTX-256 CARDS TO A uP

Interfacing multiple MTX-256 to a uP is very similar to interfacing a single MTX-256. The main difference is the way in which data is being written into the refresh memory. There are three basic ways:

a) All cards have the same chip select but the DDO (refresh memory data bit) video bit is connected to different uP data bus bits for each card. This method allows a single instruction load of all bits making up a single pixel. Data is written into all cards simultaneously. If the vector plot function is disabled, the 4MSB can be used for DDO. This allows up to eight cards to share one chip slect decoding.

b) Each card has a different chip select decoding. (DDO can be connected to the same data bus bit). This method allows the uP to write into each card independently one at a time. The method can be useful for superimpostion of several images or for video effects such as a background change or motion.

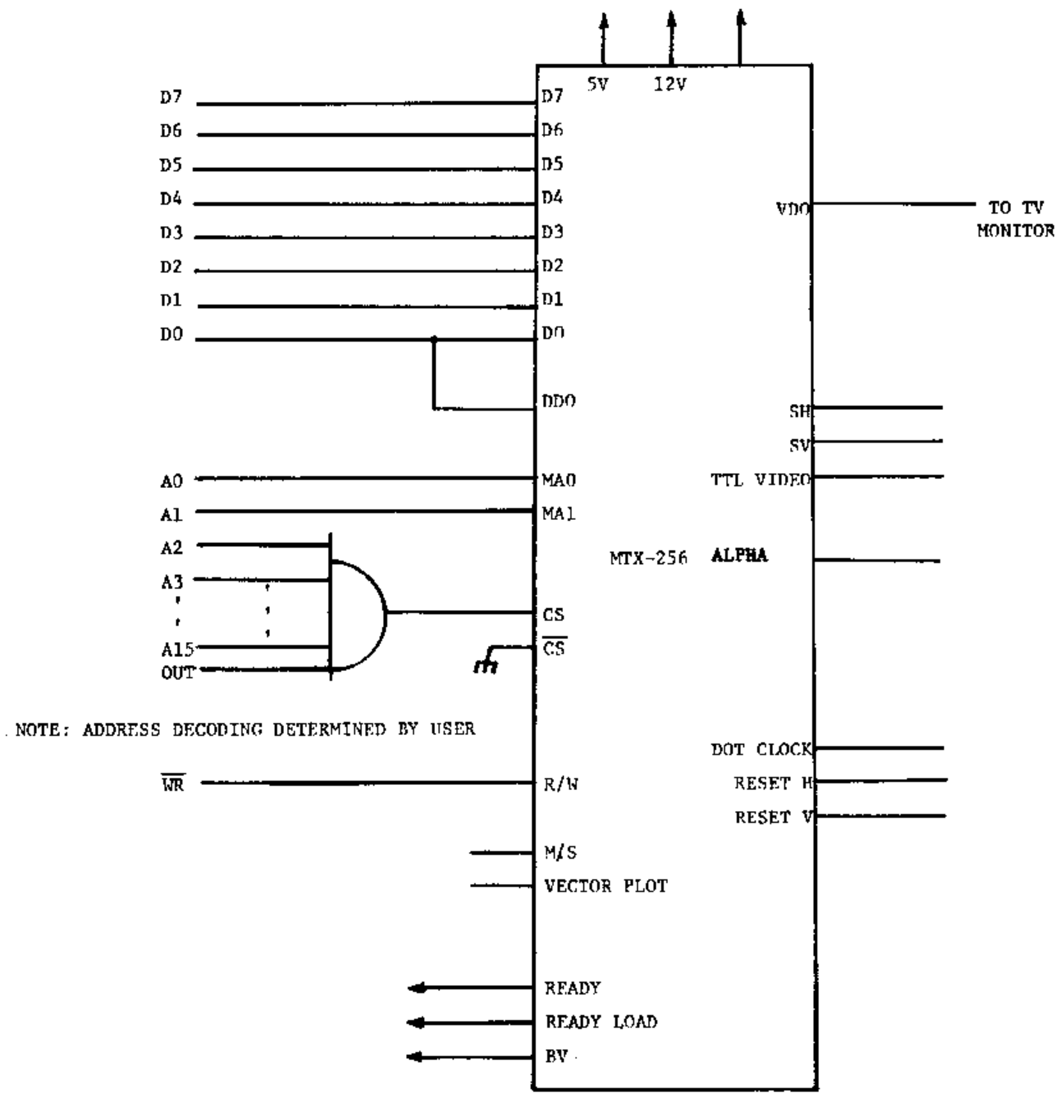


FIG. 2 8080A - MTX-256 INTERFACE

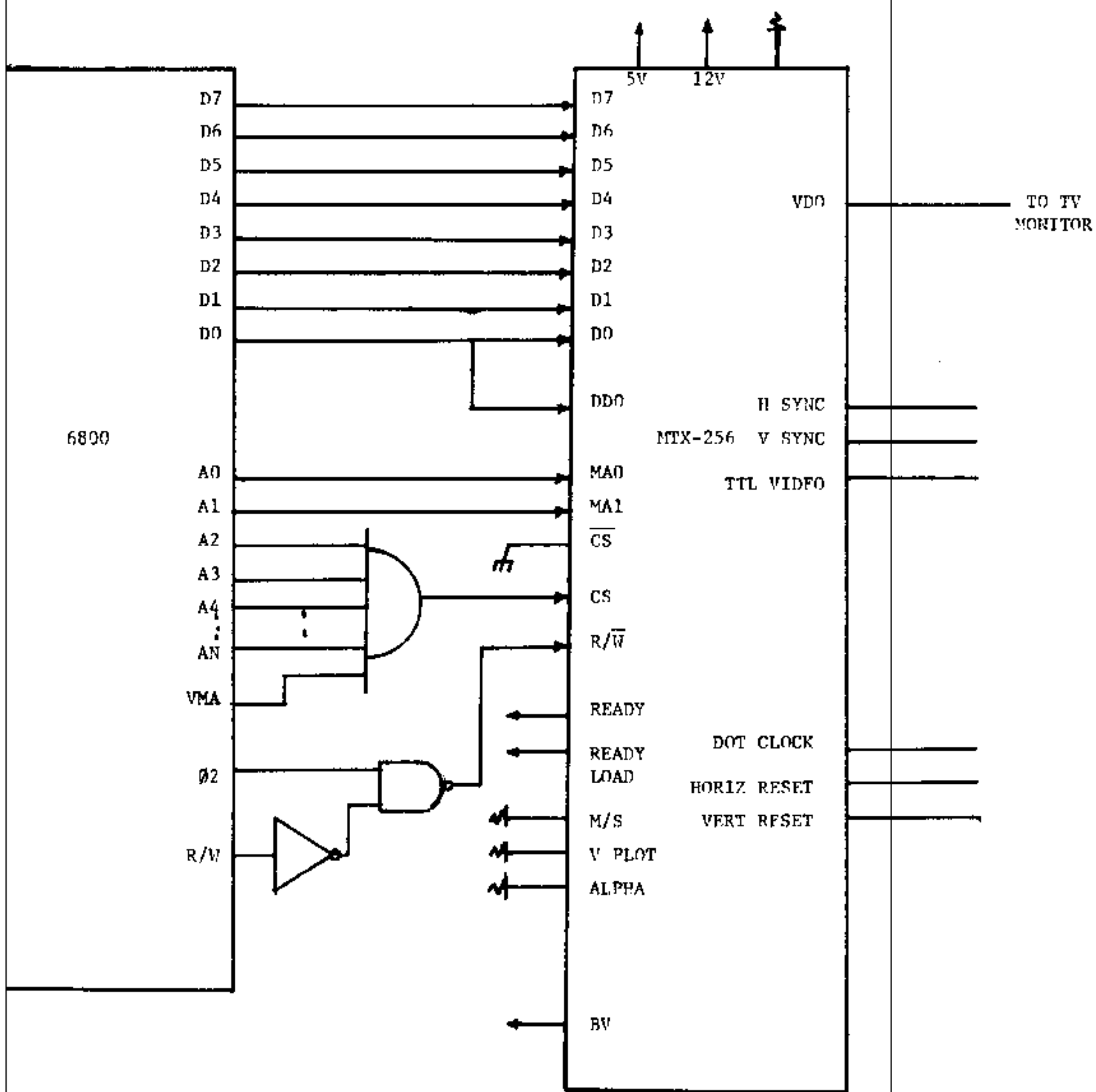


FIG. 3 6800 – MTX-256[2] INTERFACE

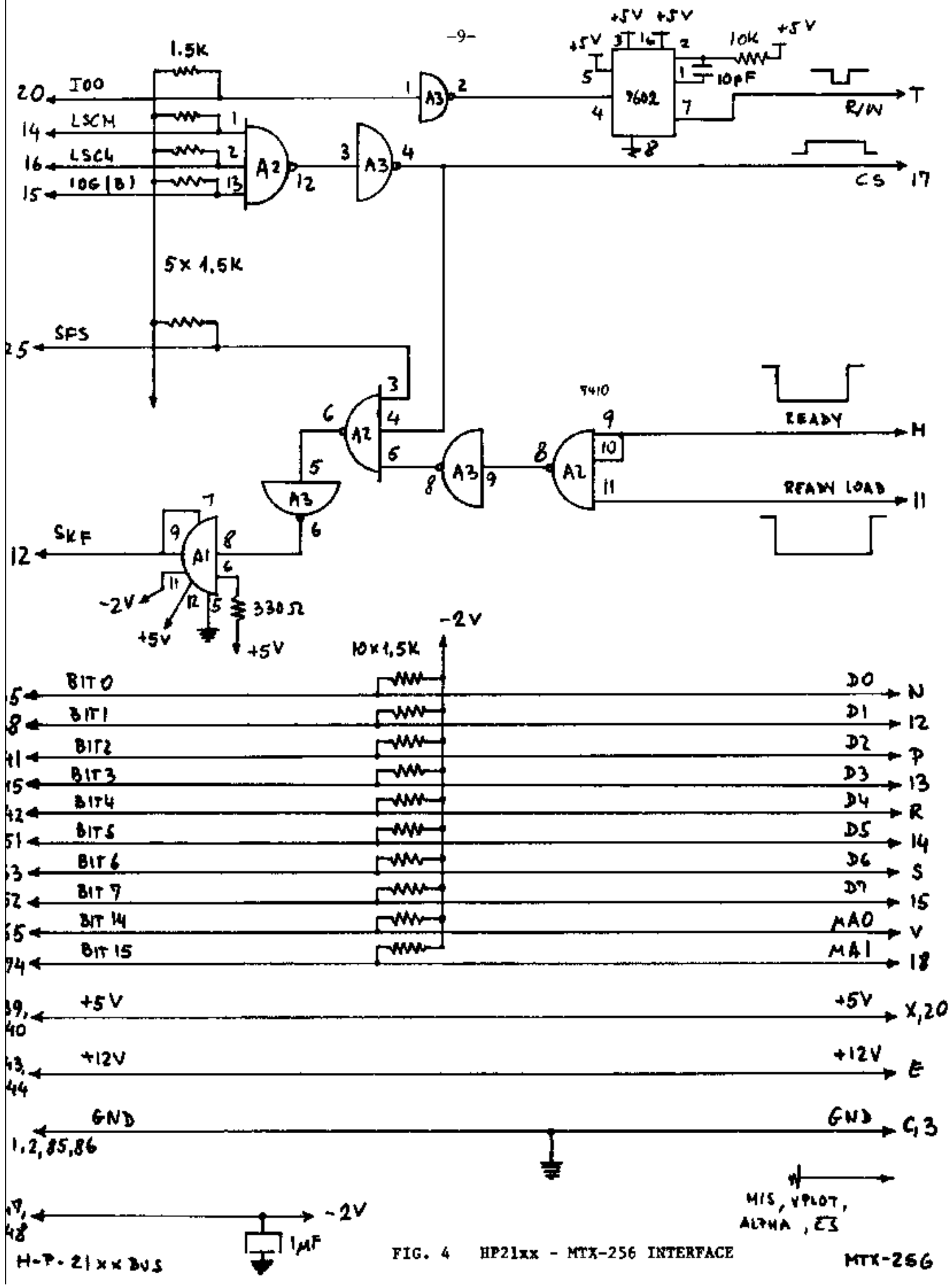


FIG. 4 HP21xx - MTX-256 INTERFACE

Fig. 5A - MTX-1632SL MTX-256 SYNCHRONIZATION

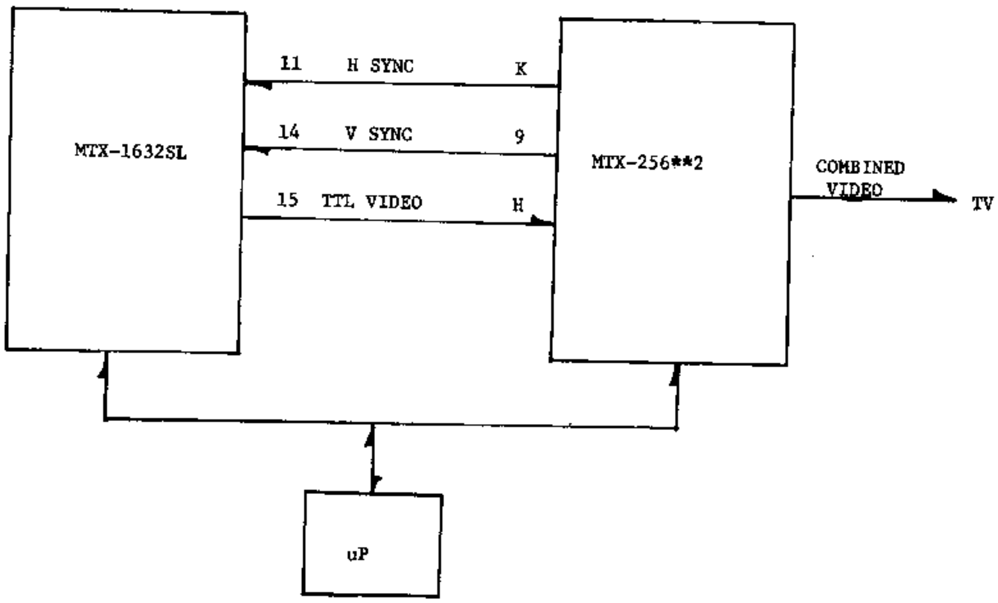
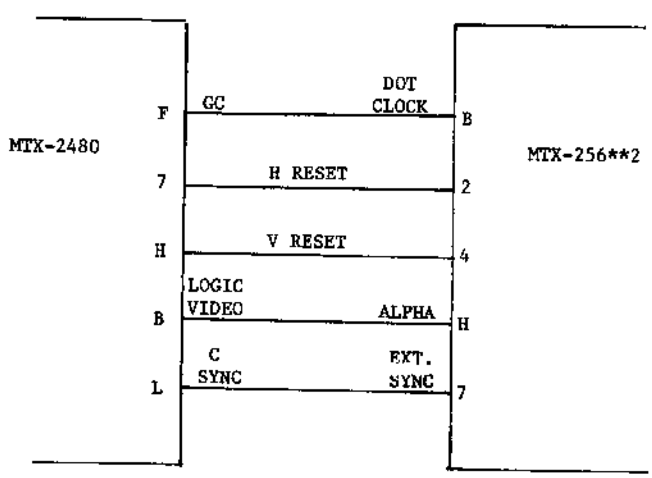
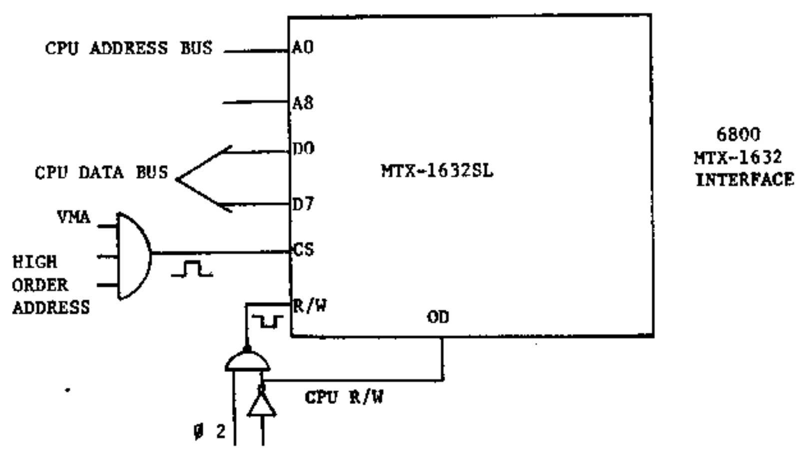


Fig. 5B - COMBINING THE MTX-2480 AND MTX-256 DISPLAYS





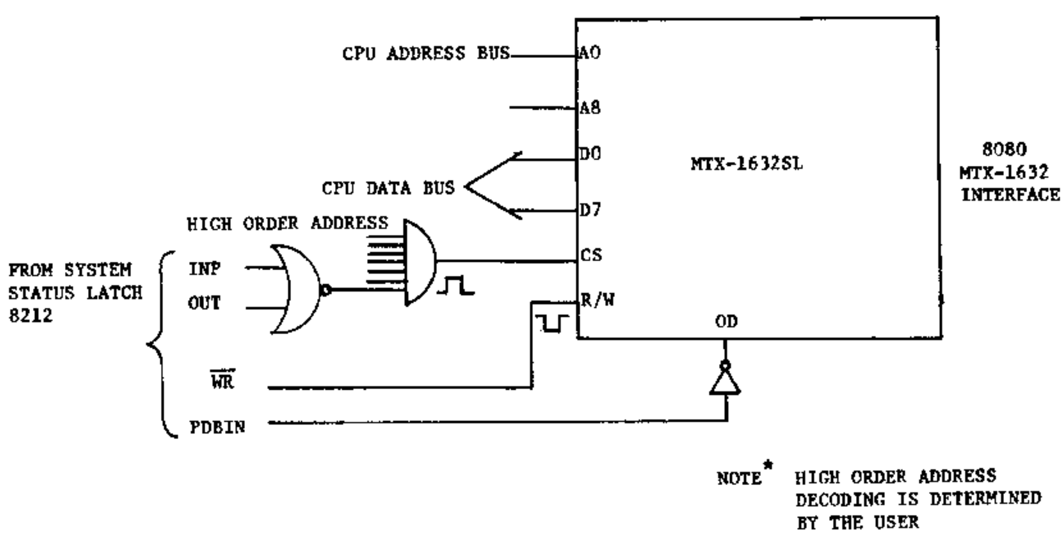


FIG. 6 MTX-1632SL TO 6800 AND 8080 uP INTERFACE

c) A combination of methods a) and b).

MASTER/SLAVE CONFIGURATION

When multiple cards are used, they have to be synchronized to a common set of sync signals. This is easily accomplished by selecting one MTX-256 to be master and all others as slaves (M/S control input).

The master card will supply three signals to all slaves: Dot clock, horizontal and vertical reset. (Up to 23 cards can be directly driven by one master card). These three lines are bi-directional input/output controlled by the M/S control input.

Fig. 7 shows an example of three cards in a Master/Slave configuration. (Note that all three cards have a common chip select, but the DD0 signals of each card go to different data bus bits).

Each card generates one bit of video information which can be used to produce a grey/scale or color video signal.

GREY/SCALE GRAPHIC SYSTEM

The circuit in Fig. 7 can form the basis for an eight level grey scale system when combined with the circuit in Fig. 8. Three bits of video information are converted to an analog voltage and horizontal and vertical sync is added to the resulting signal. The circuit in Fig. 8 is satisfactory for up to three bits of video information. The circuit is basically a buffer amplifier which sums the TTL video signals from three boards with different weights. Note that weighting resistors R1, R2, and R3 can be adjusted for the desired grey scale steps.

For more bits a digital to analog converter should be used. (Fig. 9).

RGB COLOR GRAPHIC SYSTEM

The simplest graphic color system can be obtained by producing three separate video signals which then directly drive the red, green and blue guns of a color monitor. RGB monitors have separate R, G, B inputs. Color monitors designed for an encoded composite video signal can usually be modified to drive the guns directly. (see fig. 10, 11.)

GENERATING A COMPOSITE COLOR GRAPHIC SYSTEM

The method for generating color graphics described in the preceeding section applies only to RGB monitors. RGB monitors are quite expensive. Most monitors and TV sets are designed to accept a standard composite color signal and are less expensive. While a standard monitor can be modified for RGB operation, this is sometimes quite difficult. It may be more practical to generate an encoded composite color signal. The circuits in figures 12 and 14 illustrate how to do this.

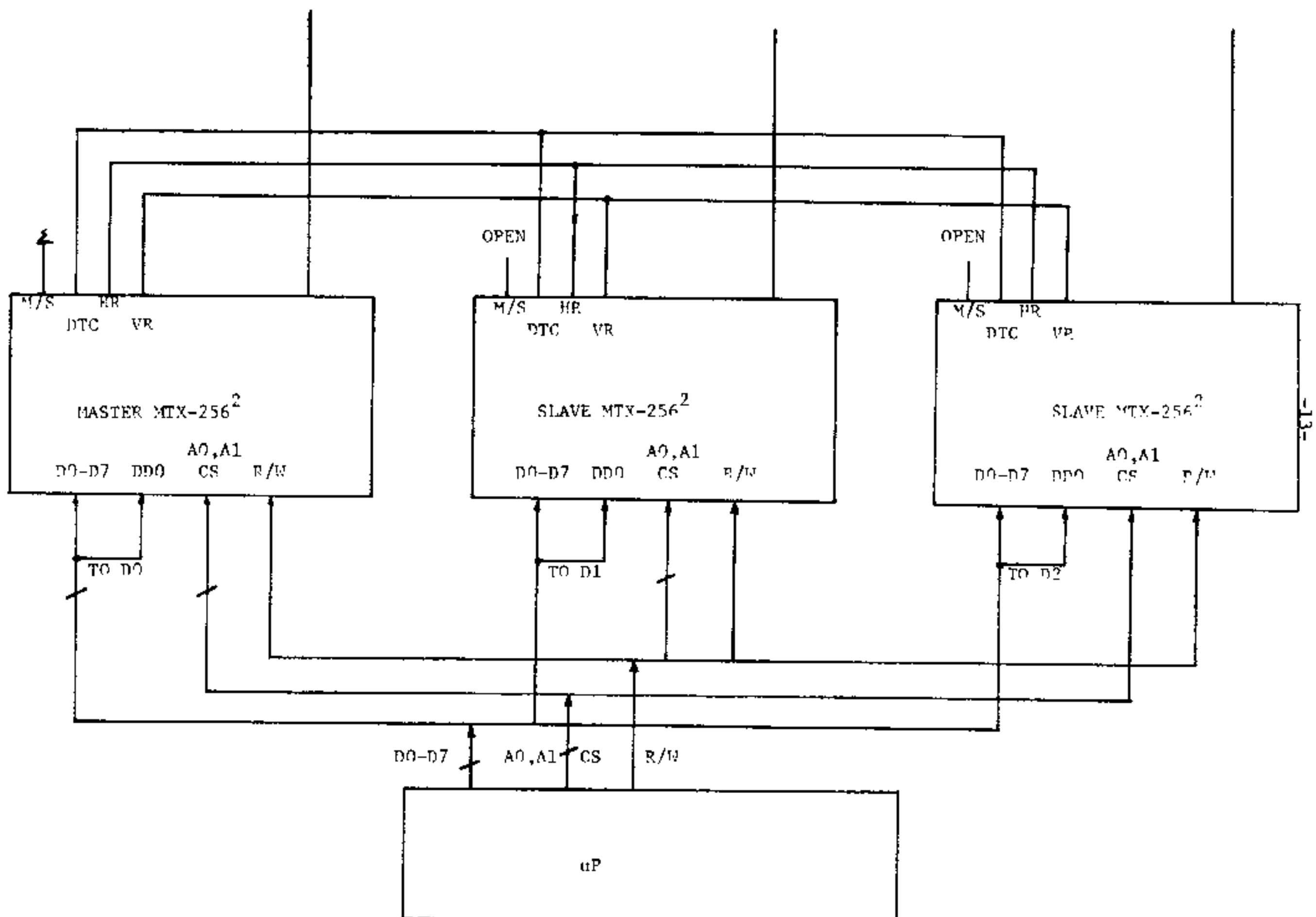


FIG. 7 THREE MTX-256^2 CARDS IN A MASTER/SLAVE CONFIGURATION

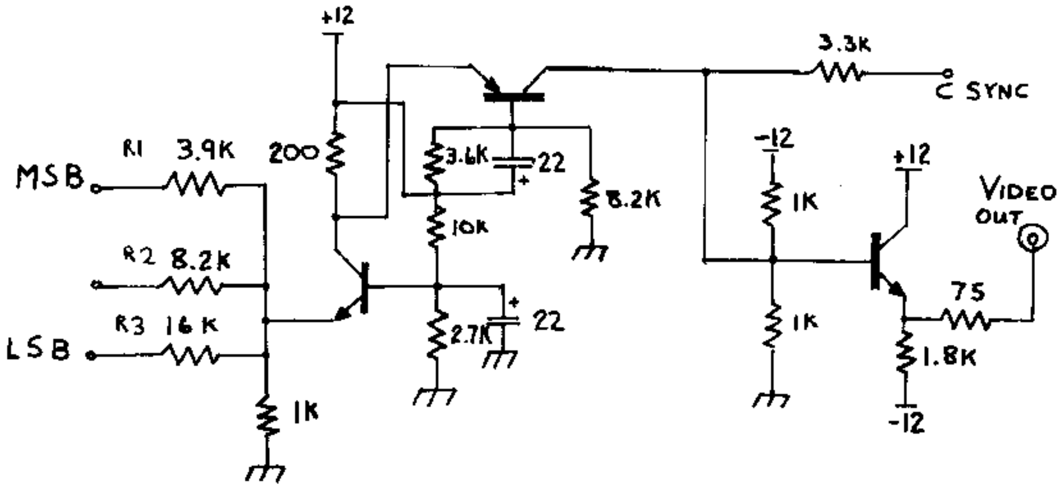


FIGURE 8 - HIGH SPEED 3 BIT VIDEO D/A CONVERTER FOR GREY SCALE IMAGING

7 mHz DATA RATE

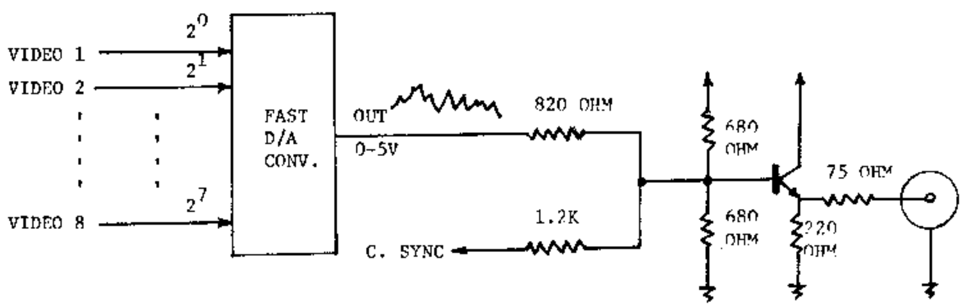


FIG. 9 256 LEVEL GREY SCALE

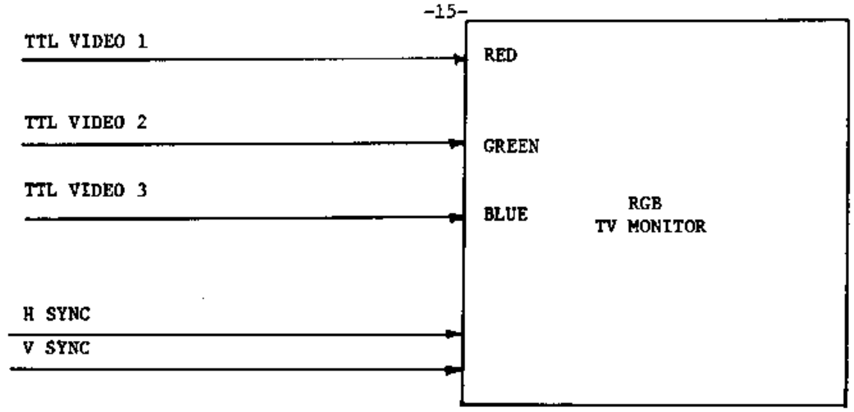


FIG. 10 8 COLOR GRAPHIC SYSTEM

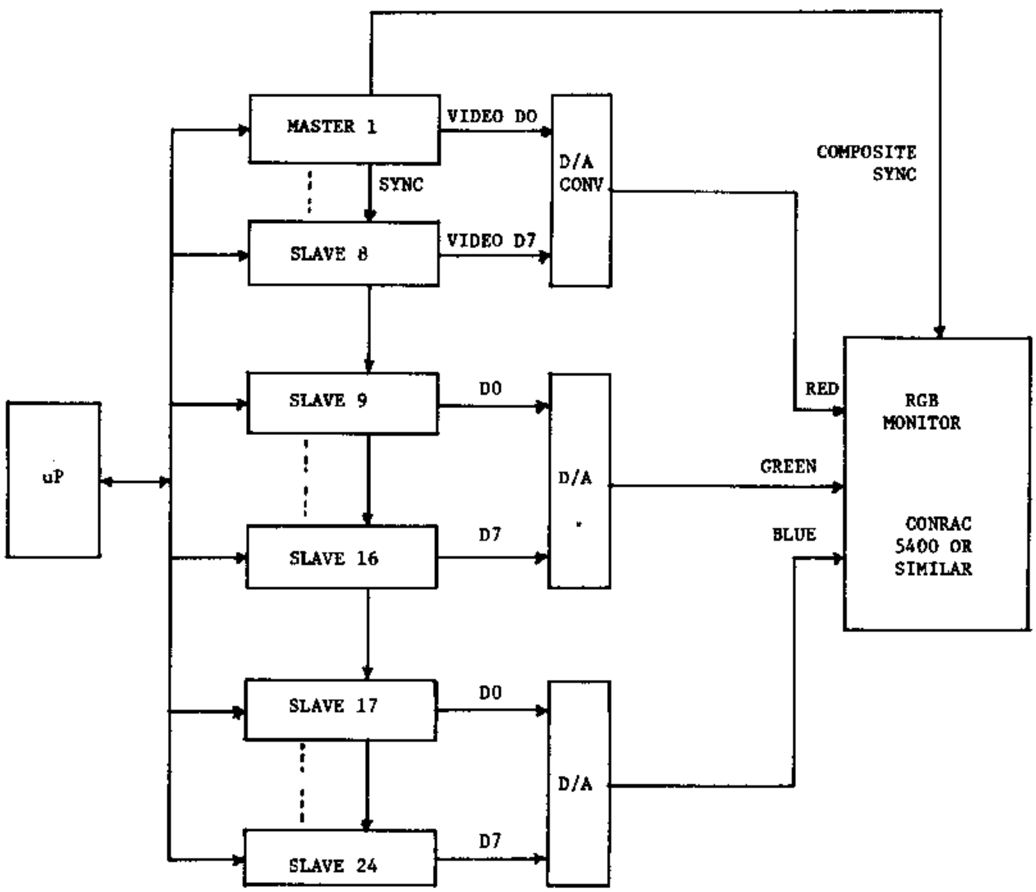


FIG. 11 RGB COLOR GRAPHIC SYSTEM (24 BITS/PIXEL)

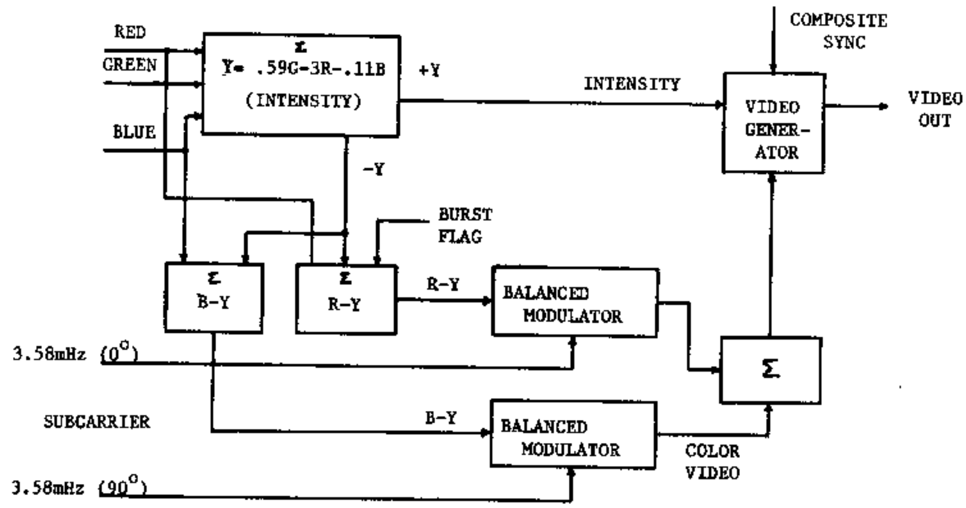


FIGURE 12 - BASIC NTSC CHROMA ENCODER

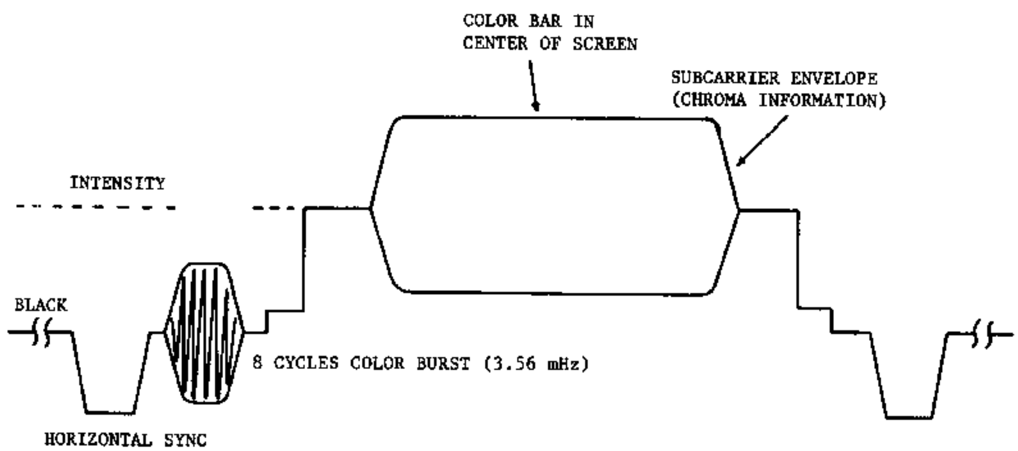


FIGURE 13 - ONE LINE OF COLOR VIDEO SIGNAL

Note that color resolution is sacrificed to some extent by using an encoded composite color signal, due to bandwidth reduction. The first step is to form the standard Y (Luminance) R-Y and B-Y signals. (See Fig. 12). The Y is generated according to the formula Y = .59G + .30R + .11B. The R-Y and B-Y signals are fed to dual balanced modulators. The modulators are supplied with a 3.579545 mHz color subcarrier. The B-Y signal should have a burst flag pulse superimposed on it. The summed output of the two balanced modulators is the chroma component of the composite video signal. The addition of luminance plus sync completes the composite color signal. (See Fig. 13).

A detailed discussion of color television principles is beyond the scope of this application note. More information can be found in numerous reference texts such as:

Color Television Fundamentals
M.S. Kiver
McGraw Hill 1964

A complete diagram for generating a composite color signal from RGB inputs is shown in Figure 14. Note that the entire color encoding is done by one chip: the LM1889 from National Semiconductor.

The circuit accepts 0-4 volt, R,G,B analog inputs and TTL level composite sync and burst flag signals. The subcarrier input requires a 1-5V p-p signal at the chroma subcarriers frequency of 3.579545 mHz.

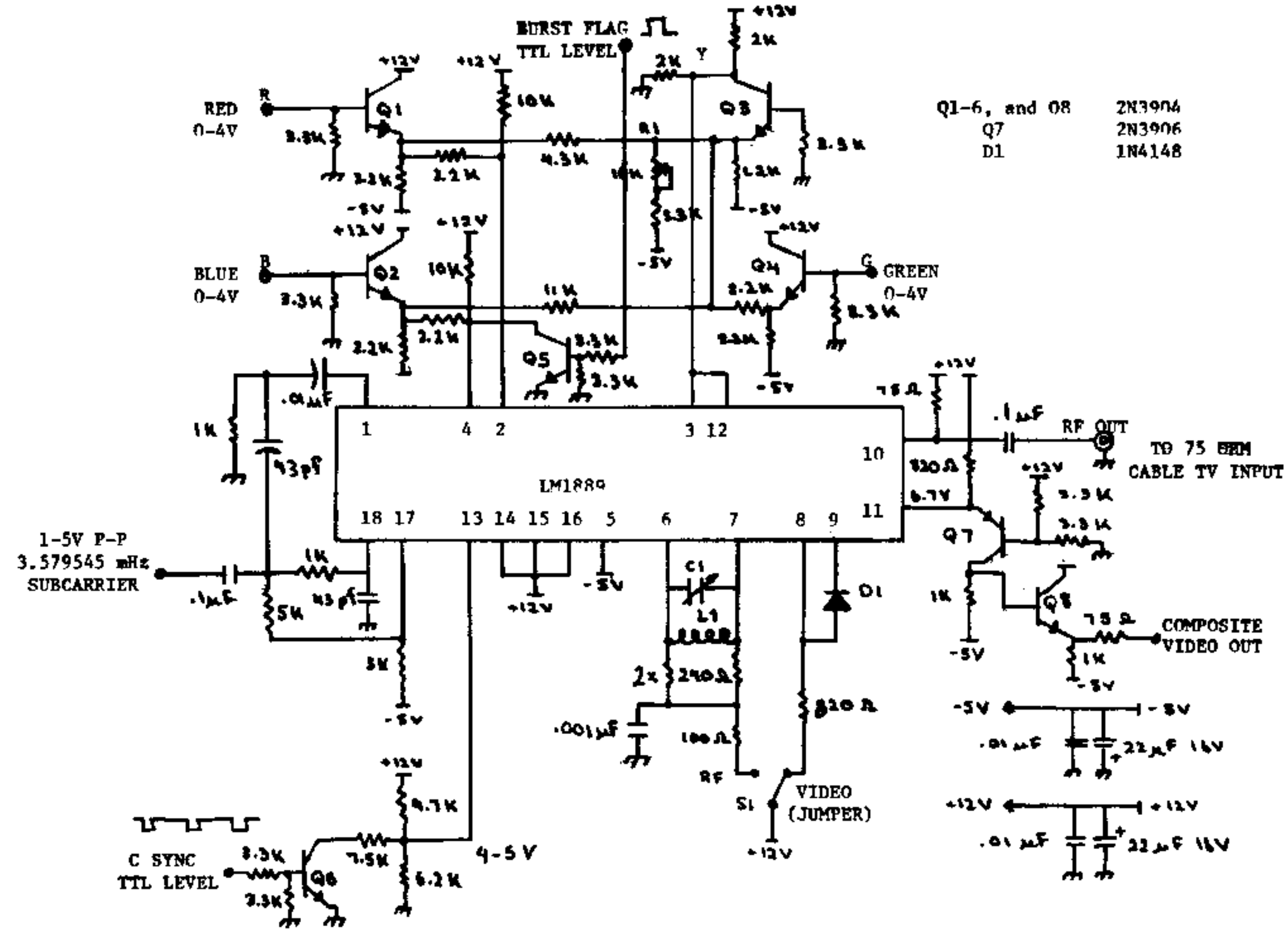
Two outputs are provided: a standard composite color video signal and a modulated RF signal. The latter can be fed directly via 75 Ohm cable to the cable input of an ordinary color TV set.

The theory of operation of the color encoder is as follows. The red and blue inputs are buffered, level shifted by Q1 and Q2 and fed directly to pins 2 and 4 of the LM1889. The luminance signal Y is formed by adding the weighted R, G,B inputs in common base stage Q3 and fed to pin 3 of the LM1889. The chroma input is phase shifted $\pm 45^{\circ}$ and fed to pins 1 and 18 of the LM1889. The LM1889 contains dual balanced modulators which generate the R-Y and B-Y chroma signals using the above inputs. Note that the color burst is generated by impressing a negative burst flag signal on pin 4.

Sync and luminance signals are added to the chroma signals through pins 13 and 12. The composite color video signal output at pin 11 is fed to common base stage Q7 and level shifted to drive emitter follower output driver Q8.

Switch S1 selects either the composite video or RF output. The RF output can be tuned to Channel 2 or 3 by adjusting C1 or the turns spacing of coil L1 (3 or 4 turns or regular solid wire, coil diameter around 3/8 inch).

To test out the circuit connect all required input signals. Observe the video output on a scope. A signal similar to that of Fig. 13 should be observed with the black level at 0V. Adjust R1 to null the subcarrier on the sync portion of the signal.



Y = .59G +.3R +.11B

FIG. 14 RGB COLOR ENCODER

For best results, the following points should be observed. The graphics cards should be synchronized to an NTSC standard color sync generator. Sync and subcarrier signals for the color encoder should come from this sync generator. A monitor with a video input is preferred to a TV set. An RGB monitor gives best results where full color resolution is required (due to wider bandwidth than with the composite video type monitor).

SYNCHRONIZING MTX-256 USING EXTERNAL PHASE LOCK LOOP

The MTX-256 built-in sync generator does not conform to the NTSC standard for broadcast video signals. The built-in sync generator generates 262 lines per field non-interlaced in the American Standard version. For broadcasting or video mixing applications, the MTX-256 can be externally synchronized. The MTX-256 must be operated at either the NTSC or PAL standard for best results with the color encoder in Figure 14.

A block diagram for an external phase lock loop is given in Figure 16. The dot clock is replaced by a VCO (voltage controlled oscillator). The VCO is controlled by a phase comparator driven by the MTX-256 horizontal sync and external horizontal sync. This forms a phase locked loop synchronizing the horizontal scanning circuits of the MTX-256 to external horizontal sync. The vertical sync circuits of the MTX-256 are synchronized to a reset pulse derived from external vertical sync. A schematic diagram implementing this approach is given in Figure 17.

All MTX-256 cards in an external sync system are jumpered as slaves. One card is selected as a quasi-master and its horizontal sync signal is used as an input for the PLL. The H reset signal is taken directly from U30 pin 6 of the quasi-master and fed via buffer U3-11 in the external sync circuit to all the H reset inputs of the graphics cards.

In the case where the MTX-256 is synchronized to an NTSC or PAL standard interlaced sync generator, the composite video output (VDO pin 8) of the MTX-256 cannot be used directly. The TTL video output of the MTX-256 should be mixed with the EXTERNAL composite sync using a circuit similar to Figure 8 or 9.

Alternatively the composite video circuit built into the MTX-256 can be used for the same purpose but the internal composite sync has to be disconnected. (The internal sync coming from pin 8 of A31 has to be disconnected from R7 and the external interlaced sync fed to R7). Composite video output VDO can then be used.

C1 varies the frequency of the VCO and should be adjusted for lockup of the PLL. C1 and R1 adjust the horizontal centering of the graphics video. The vertical centering can be adjusted by connecting U5 to different taps off counter U6. (Digitally controlled vertical delay.)

Note that the PLL VCO is sensitive to noise induced jitter. Follow good layout practices including short signal paths and proper grounding and power supply bypassing.

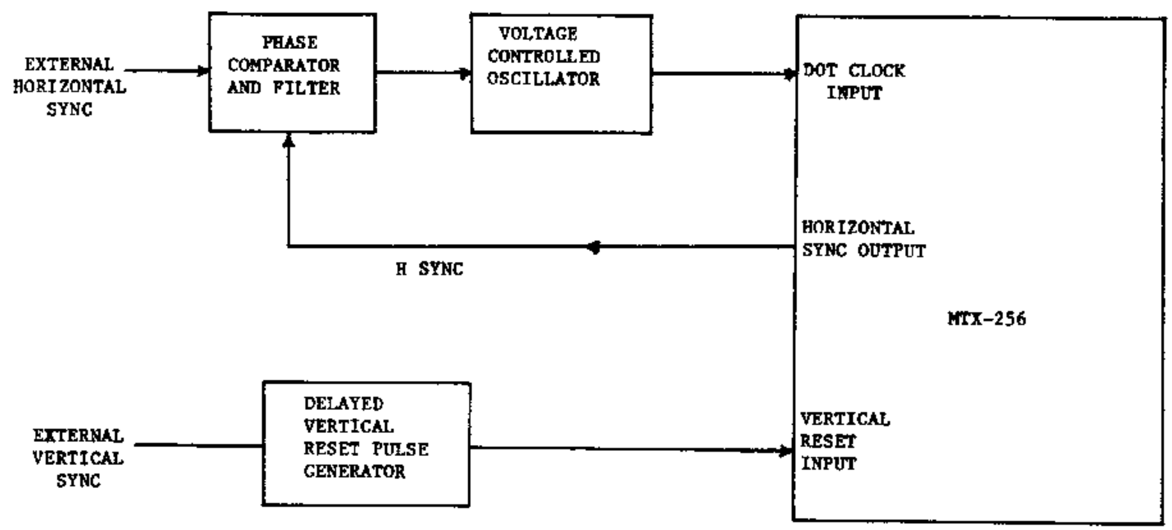
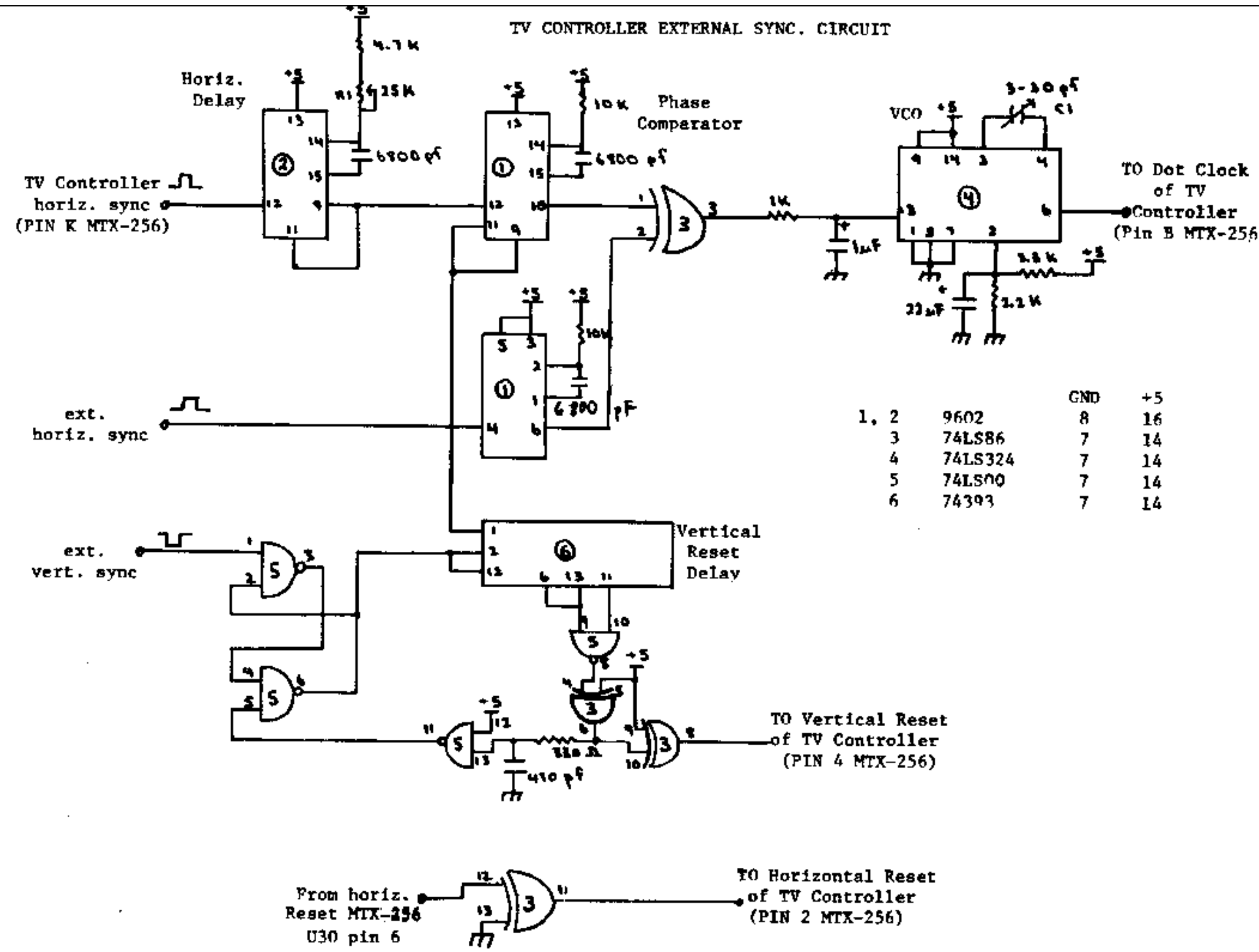


FIG. 16 SYNCHRONIZING MTX-256 USING EXTERNAL PHASE LOCK LOOP - BLOCK DIAGRAM



		GND	+5
1, 2	9602	8	16
3	74LS86	7	14
4	74LS324	7	14
5	74LS00	7	14
6	74393	7	14

FIG. 17 SYNCHRONIZING MTX-256 USING EXTERNAL PHASE LOCK LOOP - SCHEMATIC

LIGHT PEN SYSTEM USING MTX-256

A variety of low cost, extremely flexible light pen graphic systems can be built using the MTX-256 and a simple inexpensive light pen. There are two ways a light pen system can be built.

HARDWARE METHOD

Using the Dot clock and horizontal and vertical reset signals, a high speed light pen system can be built which offers a minimum of uP overhead time and fast data rate. (Fig. 18) X (9 bit) and Y (8 bit) counters track the position of the electronic beam. When the light pen detects a white dot at its position, a pulse is generated. This pulse stops the X and Y counters at the present dot address and at the same time interrupts the uP. This method requires a fast light pen (100 nsec for response time) due to the high dot frequency (7 mHz).

SOFTWARE METHOD

Any low cost, low speed phototransistor or photodiode can be used as a light pen if the raster is generated by the uP. The scanning speed is in effect controlled by software. The uP selectively illuminates one dot at a time and it checks the light pen output to see if the sensor is in the front of that dot. The computer scans all possible screen positions until the position of the light pen is found. This method requires a minimum of hardware and allows great flexibility in the scanning algorithm.

JUMPER OPTIONS

Each MTX-256 can be programmed for American, European, or American Non-standard configuration.

European and American Non-standard will produce a 256 X 256 dot raster with aspect ratio 1:1 (same distance between two dots in horizontal and vertical direction). The raster is positioned in the middle of the screen. The dot clock is 7.0536 mHz. American standard produces a 256 X 240 (vertical dot raster with a 4:3 aspect ratio. (The raster fills the entire screen). The dot clock is 5.5334 mHz.

Jumper Positions: for the MTX-256**2 standard selection (W1 - W23)

American Standard: (262 vertical lines; 60 Hz; 240 vertical video lines)

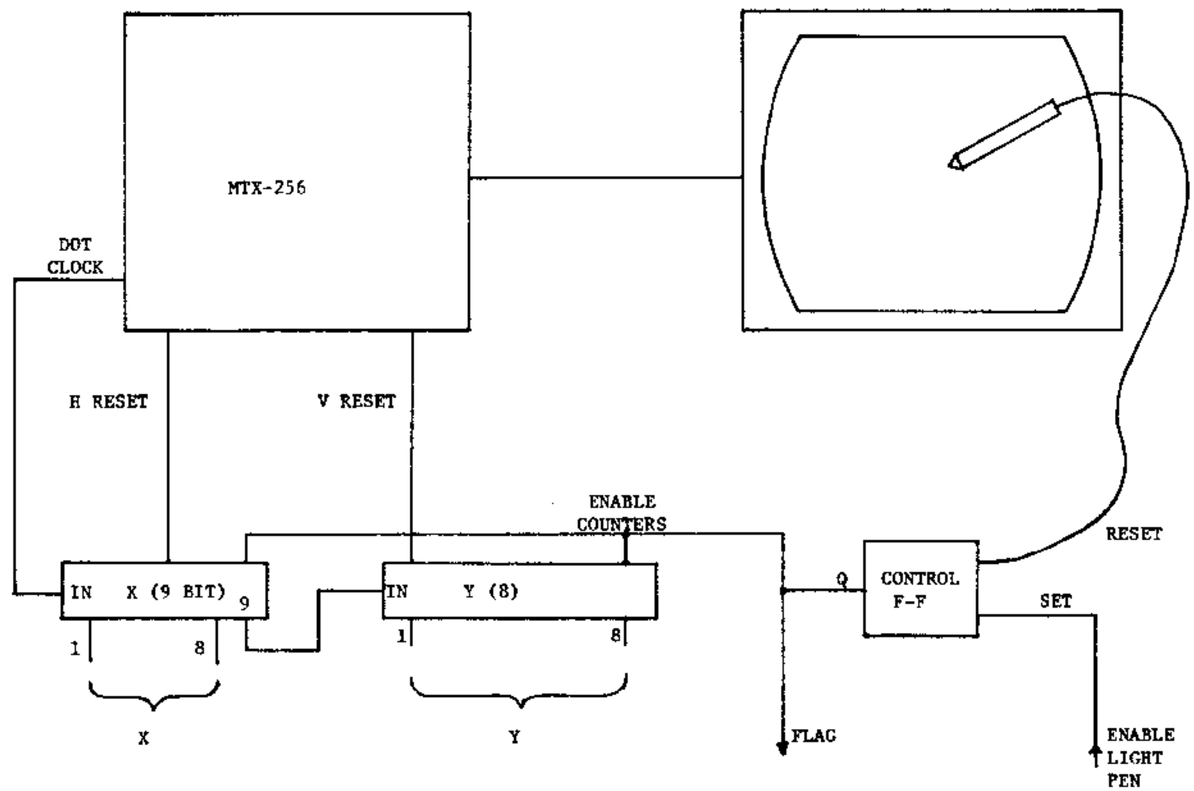
Jumpers in:	W2	W4	W6	W8	W11	W13	W15	W17	W18	W20	W23
Jumpers out:	W1	W3	W5	W7	W9	W10	W12	W14	W16	W21	W22

American Non-Standard: (280 vertical lines; 60 Hz; 256 vertical video lines)

Jumpers in:	W1	W4	W5	W9	W10	W12	W14	W17	W19	W21	W22	
Jumpers out:	W2	W3	W6	W7	W8	W11	W13	W15	W16	W18	W20	W23

European Standard: (312 lines; 50 Hz; 256 video lines)

Jumpers in:	W1	W3	W5	W7	W11	W12	W14	W16	W19	W20	W22	
Jumpers out:	W2	W4	W6	W8	W9	W10	W13	W15	W17	W18	W21	W23



NOTE* DISPLAYED DOT X ADDRESS + 12 = LIGHT PEN X ADDRESS

FIG. 18 LIGHT PEN SYSTEM USING MTX-256

OTHER PRODUCTS AVAILABLE

The Matrox 512 graphics family incorporates the revolutionary new concept of a variable resolution graphics on a single controller card. Any card in the 512 family can be user programmed to produce a dot matrix of 256 X 256; 256 X 512; 512 X 512; or 256 X 1024 points. The variable resolution feature is possible because of the new generation of compatible 16 pin 4K, 8K and 16K dynamic RAM's. The resolution is changed by simply plugging in the correct set of RAM's and a couple of PROM's. The first cards in the 512 family are designed for the following buses: the Digital Equipment PDP-11 (MDC); LSI-11 (MLSI) bus and the Intel SBC-80 (MSBC) bus. The manuals provide sufficient data such that either card can be interfaced to any mini or micro computer.

Other important features of the 512 family include the ability to stack multiple cards for color/grey scale applications, X-Y addressing, single command erase, refresh memory read, and scroll. A compatible 24 X 80 text display is available for the LSI-11 and SBC-80 buses. A complete software package is available for the MSBC-256 operated in 256 X 256 resolution mode. Typical applications of the 256 and 512 family include process control systems, computer aided design, business and educational displays, medical displays, curve plotting and image processing. The 512 cards can also be used to make intelligent stand alone graphics terminals when combined with LSI-11 or SBC-80 mainframes.

A complete software package for 8080, 8085, and Z-80 microcomputers for 256 X 256 resolution Matrox graphics controllers is available. These include specifically the ALT-256 and MSBC-256 (S-100 bus and Intel SBC-80 bus respectively) cards. The software package features point and vector plot, software selectable resolution, alphanumeric generation and animation synchronization. A similar software package for the full range of resolutions will be available for 8080 and PDP-11 based systems in the first quarter of '78.

The versatile Matrox graphics family can be used in virtually any application where CRT graphics are required. The low cost of the family is a fraction of that of competing systems.

Typical application areas include process control systems, computer aided design, business and educational displays, displays for research applications, curve plotting, medical displays and image processing.

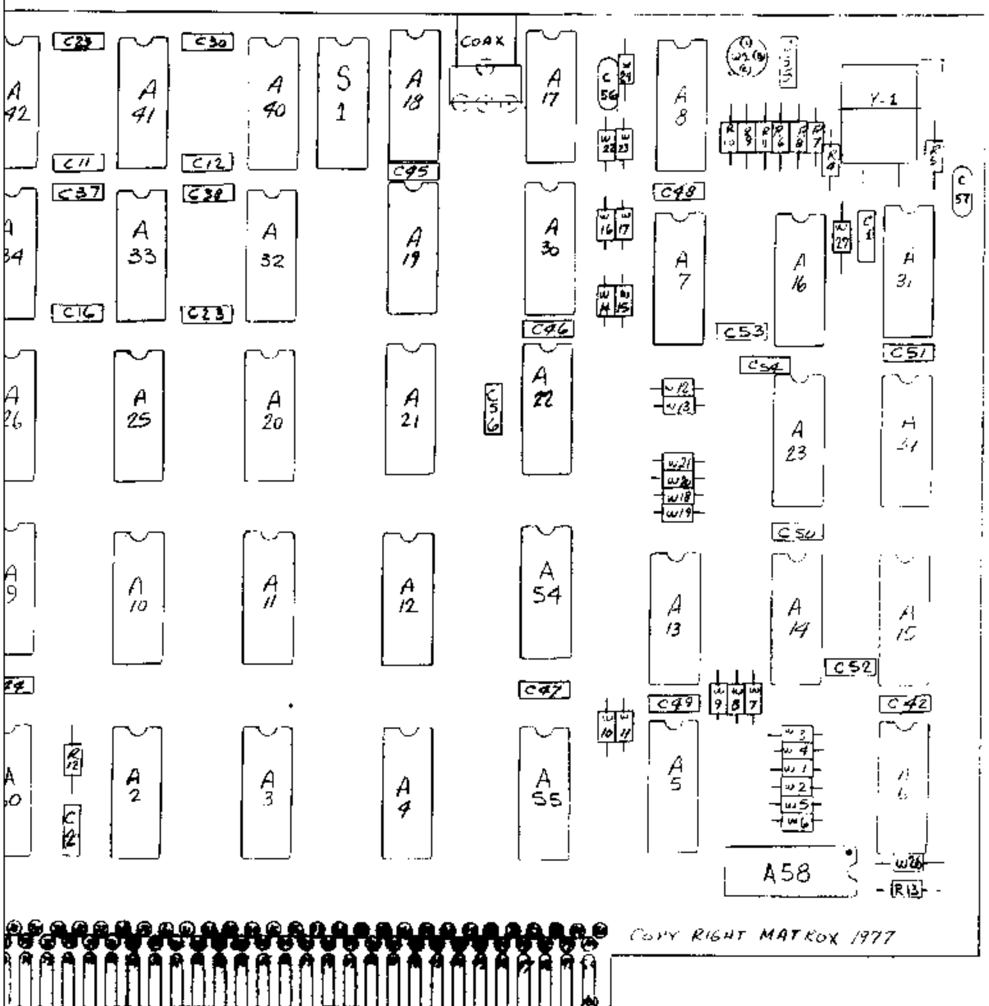
Matrox offers a complete line of alphanumeric and graphic display interfaces for micro and minicomputers. We have a product for almost any application.

2 | 1

REVISIONS

ZONE	LTR	DESCRIPTION	DATE	APPROVED

D



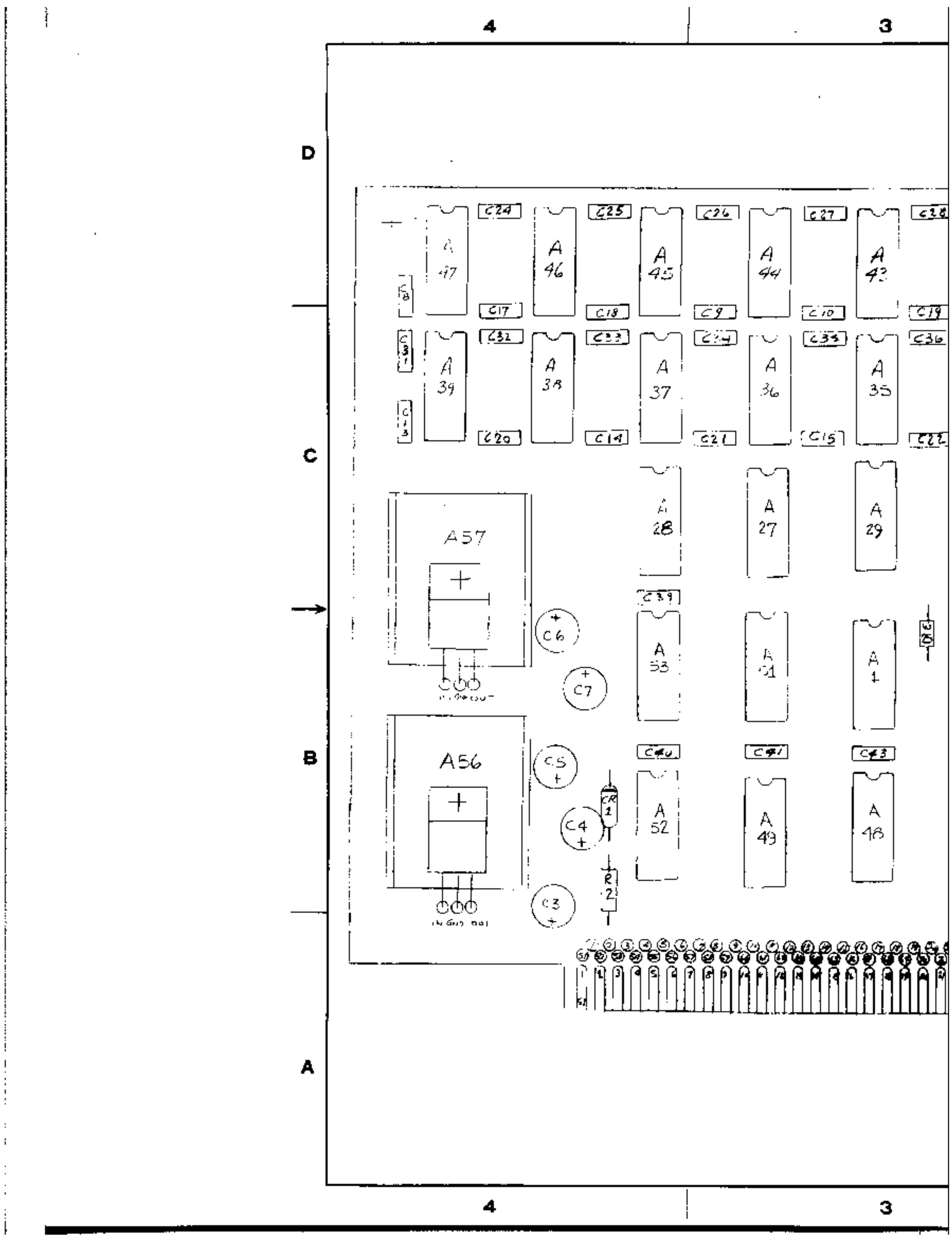
C

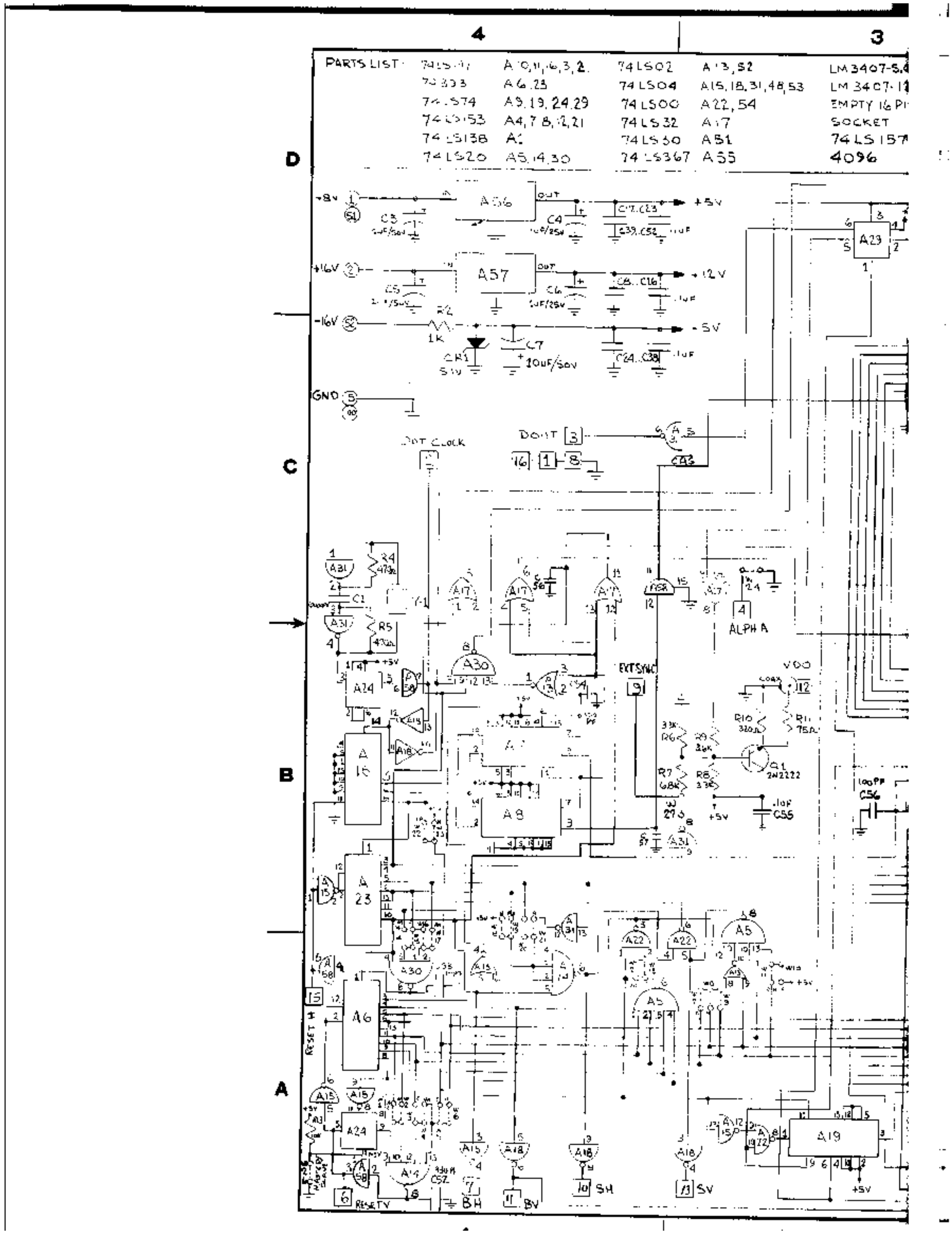
B

A

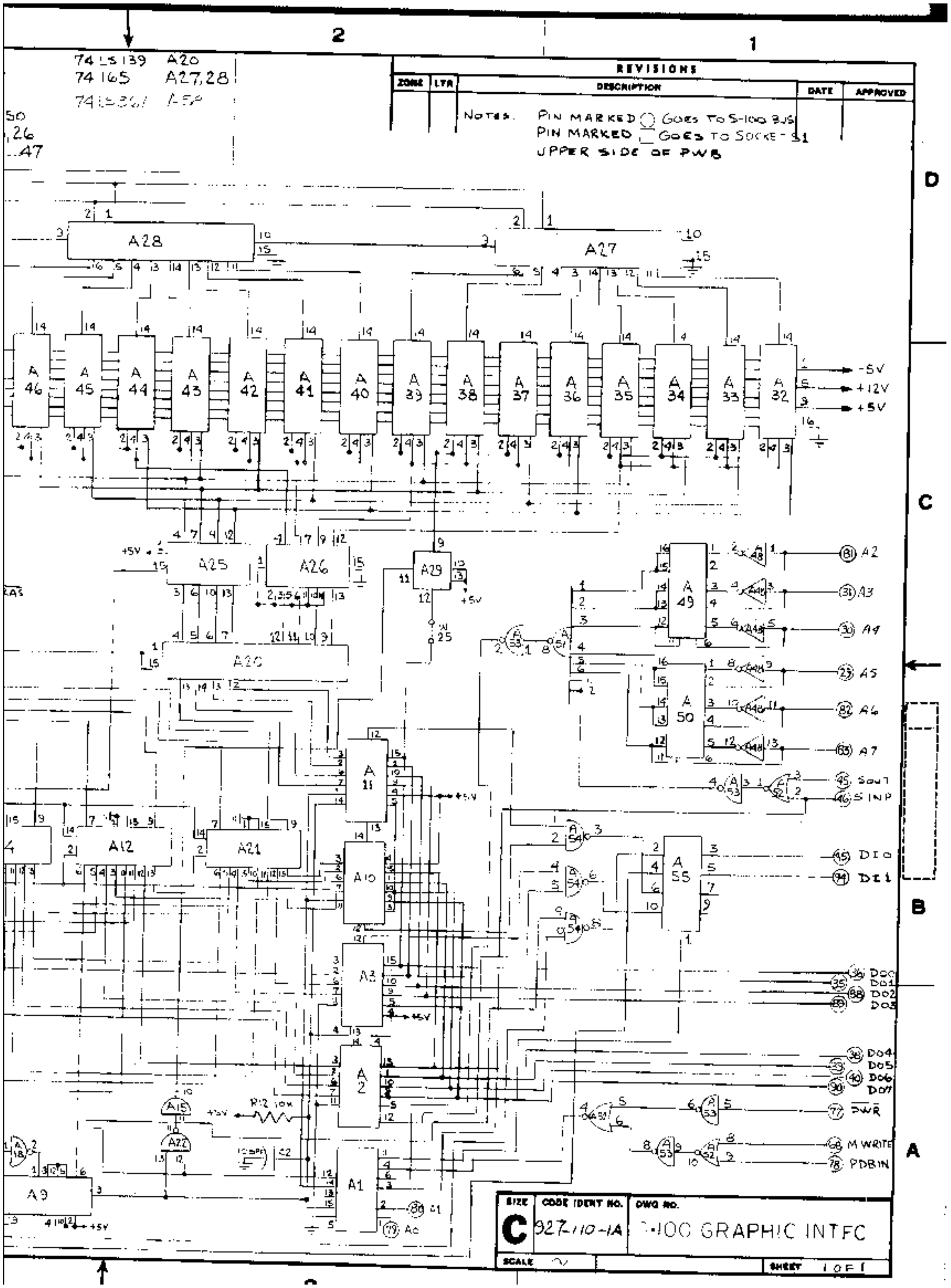
SIZE	CODE IDENT NO.	DWG NO.
C	220.110.1A	S-100 GRAPHIC INTFC ASS'Y
SCALE 2/1		SHEET 1 OF 1

2 | 1


4
3
D
C
B
A
A57
A56
A 47
A 46
A 45
A 44
A 43
A 39
A 38
A 37
A 36
A 35
A 28
A 27
A 29
A 53
A 51
A 1
A 52
A 49
A 48
C24
C25
C26
C27
C17
C18
C9
C10
C19
C32
C33
C34
C35
C36
C20
C14
C21
C15
C39
C40
C41
C43
C6
C7
C5
C4
C3
CR 1
R 2



4
3
PARTS LIST
A6,23
74LS74 A9,19,24,29
74LS153 A4,7,8,12,21
74LS138 A1
74LS20 A5,14,30
74LS02 A13,52
74LS04 A15,18,31,48,53
74LS00 A22,54
74LS32 A17
74LS30 A51
74LS367 A55
LM3407-12
EMPTY 16 PIN
SOCKET
74LS157
4096
D
C
B
A
+8V
+16V
-16V
GND
A56
A57
+5V
+12V
-5V
10UF/50V
DOT CLOCK
DOUT
ALPHA
EXT SYNC
VDD
2N2222
RESET
BH
BV
SH
SV
A19
A23
A24
A6
A8


74LS139 A20
74165 A27,28
74LS367 A58
REVISIONS
ZONE LTR DESCRIPTION DATE APPROVED
NOTES. PIN MARKED ○ GOES TO S-100 BUS
PIN MARKED ☐ GOES TO SOCKET S1
UPPER SIDE OF PWB
A28
A27
A46 A45 A44 A43 A42 A41 A40 A39 A38 A37 A36 A35 A34 A33 A32
-5V
+12V
+5V
A25
A26
A29
A20
A49
A50
A55
A11
A12
A21
A10
A3
A2
A1
A9
A15
A22
R12 10K
A2 A3 A4 A5 A6 A7
SOUT
SINP
DIO
DI1
DO0 DO1 DO2 DO3
DO4 DO5 DO6 DO7
PWR
M WRITE
PDBIN
A1
A0
SIZE C
CODE IDENT NO. 927-110-1A
DWG NO. S-100 GRAPHIC INTFC
SCALE
SHEET 1 OF 1

ALT — 256 ** 2

ALTAIR — IMSAI BUS COMPATIBLE GRAPHIC DISPLAY INTERFACE

MTXGRAPH

USER'S

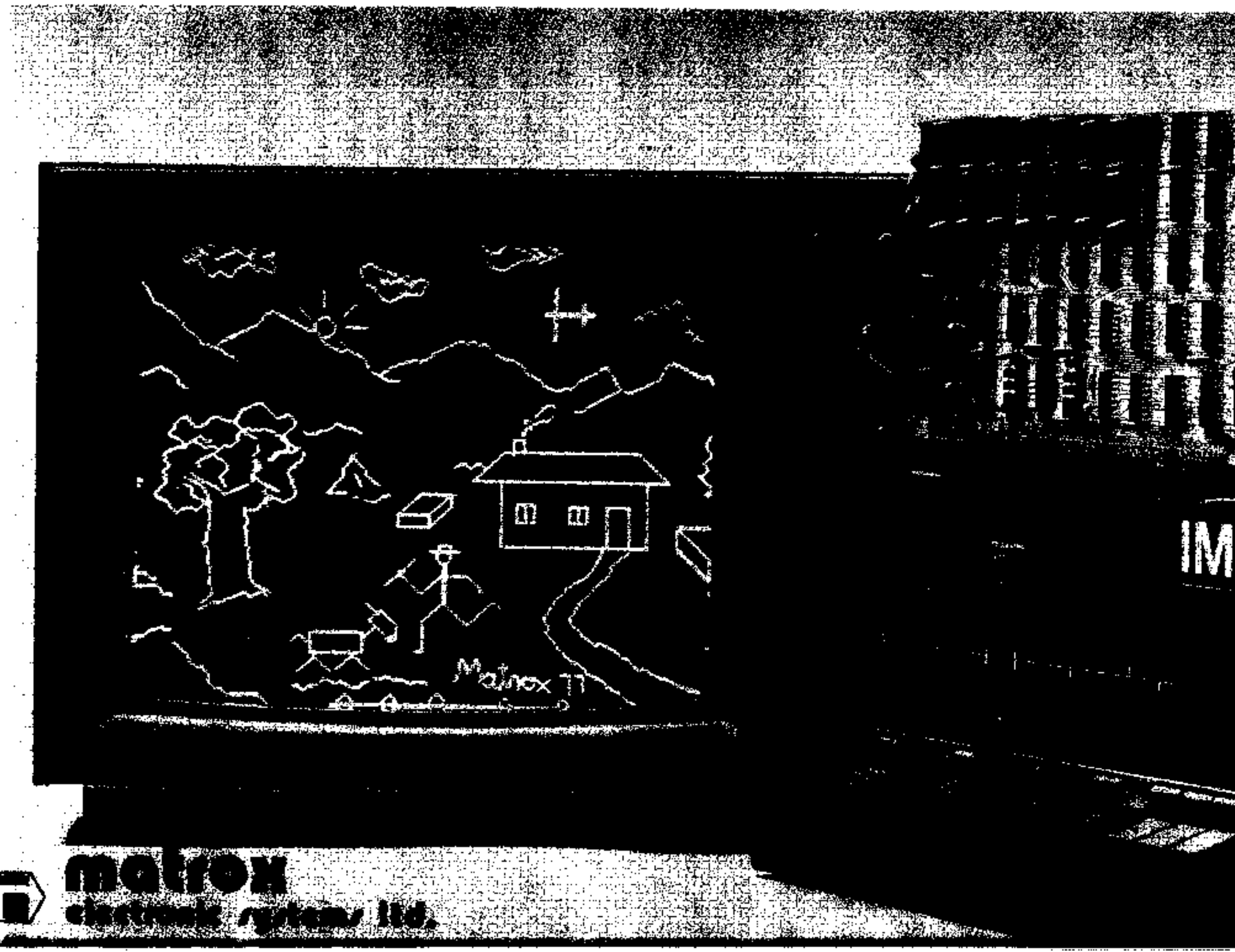
MANUAL

AUGUST 1977

matrox electronic systems ltd.
2795 BATES RD. MONTREAL, QUE. H3S 1B5 CANADA
TEL: (514) 481-6838 or 735-1182 TELEX 05-825651

ALT — 256 ** 2

ALTAIR — IMSAI BUS COMPATIBLE GRAPHIC DISPLAY INTERFACE



matrox
electronic systems ltd.

SPECIFICATIONS FOR ALT-256**2 GRAPHIC DISPLAY INTERFACE

INTRODUCTION:	The ALT-256 low cost high resolution graphics interface can be used in a wide range of applications. Typical examples range from video games like electronic etch-a-sketch, to industrial uses such as computer image processing. The unit has 4 times the resolution of other S100 bus graphics and includes features such as expansion for color/grey scale, external sync capability, on card refresh memory and direct S100 bus compatibility. The ALT-256 is compatible with the ALT-2480 permitting a powerful combined alphanumeric/graphic display.
RESOLUTION:	256 X 256 dot raster.
ACCESS TIME:	3.4 usec. max /dot: each dot induvidually addressed
ERASE:	Single instruction erases screen: 33 msec. max.
REFRESH MEMORY:	Built-in on the card: 65,536 X 1 bit memory
BUS:	S100, plugs directly into Altair-Imsai bus
COLOR/GREY SCALE:	Up to 24 bits/pixel (2^{24} different colors or grey levels/dot) by using identical multiple boards.
DIMENSIONS:	9" X 5"
POWER:	8V, 600mA; 18V, 100mA; -18V, 10mA, on board regulators
OUTPUTS:	Composite video; 75 Ohm, x-tal controlled: TTL video, horizontal and vertical syncs and blanks outputs.
SYNCHRONIZATION:	Built-in TV sync generator (x-tal controlled). Free running or external sync capability.
TV STANDARD: 4	American standard (262 vertical lines, 60 Hz; 240 vertical video lines) 4:3 aspect ratio; American Non-standard(280 lines, 60 Hz; 256 video lines), horizontal freq. = 16.8 KHz. (1:1 aspect ratio); European (312, 50 Hz 1:1 aspect ratio). Non-interlaced picture. Standard selectable on the board
MONITOR:	Any standard TV monitor or modified TV set.
REMOTE DISPLAY:	75 Ohm cable, up to 2,500 feet,; multiple monitors, up to 25 TV's.
ADDRESSING:	Four output ports and one input port built-in. Port address selectable on the board.
X-Y PLOT MODE:	X coordinate (output port 1, 8 bit registor); Y coordinate (output port 2, 8 bit registor). Data port (output port Ø, 8 bit registor). Outputting data H"Ø1" to port H"ØØ" writes a white dot at the point addressed by the X and Y registors. Writing data H"ØØ2 to port H"ØØ" writes a black dot.
ERASE:	Outputting data to port 3 will erase the screen. Data H"ØØ" will set all bits to 0 (black). Data H"Ø1" will set all bits to 1 (white.)
DISPLAY STATUS:	Available by reading input port Ø. Data bit DO = graphic interface busy flag; D1 = vertical blank.
DOCUMENTATION:	12 page manual; complete description, circuit schematics and a test program. Additional color/grey scale application note available on request.
WARRANTY:	90 days parts and labor.
SOFTWARE:	A complete software package is available for $28.
ORDERING:	Available directly from Matrox Electronic Systems Ltd., Montreal or from franchised distributors. Specify options desired. Delivery 2-4 weeks.

matrox
electronic systems ltd.
2795 BATES RD MONTREAL QUE H3S 1B5 CANADA
TEL (514) 481-6838 or 735-1182 TELEX 05-825651

TABLE OF CONTENTS

1. INTRODUCTION

MTXGRAPH is a complete graphics support software subroutine package for the Matrox ALT-256**2 graphic display. It provides for initialization, screen erase, single point display and erase, endpoint vector line generation and deletion, and character generation and deletion. Total memory required is less than 1K (1024) bytes plus program stack. The standard package is configured to be totally compatible with the 8080/Z80 Graphics Software Protocol published in _______ Byte magazine. Multiple board combinations of ALT-256**2's can be supported up to a maximum of eight bits of color/grey scale information.

1.1 Display Coordinates

For greatest flexibility, all subroutine interfaces use standard Cartesian Coordinates with the display occupying the first quadrant. As shown in figure 1-1, the origin (X = 0, Y = 0) is defined as the bottom left point on the display. X increases in value to a maximum of 255 at the right edge while Y rises to a maximum of 255 at the top. If the ALT-256**2 is jumpered for American Standard scan (240 lines displayed) the lowest Y coordinate displayed is 16 and points with Y coordinates from 0 to 15 cannot be seen.

1.2 Functions Provided

Seven subroutines are provided for user programs. They are:

1) INITG - Initialize the graphics software subsystem to standard defaults.
2) PAGE - Next page, i.e. erase the entire screen.
3) CURSOR (X,Y) - Position the cursor at the point X,Y.
4) DOT - Set the point (or points if in a lower resolution mode) defined by the cursor to the currently selected color.
5) LINE (X,Y) - Set the "DOT"s along the line connecting the current cursor position to the point X,Y to the currently selected color. Leave the cursor set to X,Y.

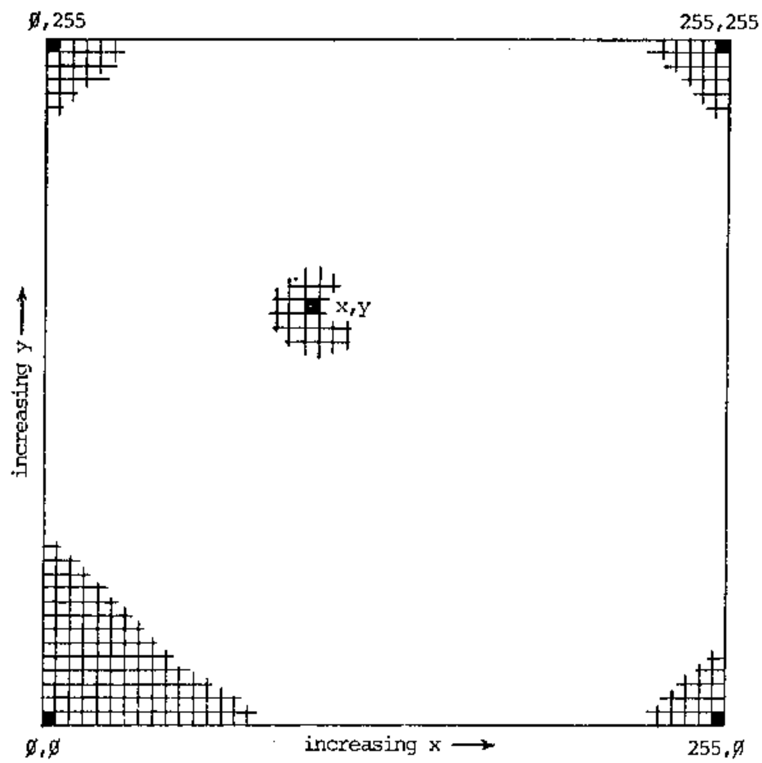


Figure 1-1 Standard Display Coordinate System

6) CHAR (VAL) - Display the character whose ASCII value is VAL at the current cursor position using the currently selected color. Leave the cursor at the next character position. Control characters provide for setting the current color, selecting fixed or proportional character spacing, setting "DOT" size, and all carriage controls.

7) ANIMAT - Pause until the start of the next vertical blanking period.

2. USING MTXGRAPH

The MTXGRAPH graphics support subroutine package provides all of the commonly used low level graphics routines. With its small size and comprehensive, easily utilized routines, it will quickly become an integral part of every program you write that requires graphics. As an added bonus, it is totally compatible with the graphics software protocol standard published in ______ Byte, allowing the use of all applications software that meets the standard regardless of what display the original software was written for.

2.1 Loading and Calling Conventions

MTXGRAPH is loaded into memory locations 0104 to 04FF(hex). This provides a standard location for the package regardless of memory size. (A lower starting address is not used to avoid conflict with monitors and programs which utilize thefirst 256 bytes of memory.) The first twenty-one bytes (0104 to 0118 hex) are the entry points to the different routines as indicated in table 2-1. Except for the INITG routine which overwrites registers H and L, the contents of all registers and flags are preserved.

The routine INITG may be called with the address of the first unused memory location above the program in register pair H,L to indicate available space for refresh buffers. While Matrox displays do not require this information, it is normally included for compatibility. The contents of H and L are replaced by INITG with two bytes describing the display being used (all other registers and flags are left undisturbed). The format for these bytes is given in figure 2-1. The available colors and scale factor fields in the H register describe the display when maximum resolution mode is selected, the same fields in the L register describe the maximum color selection mode.

The available colors field gives the number of colors a point can be written to other than white. If this field is zero, it means the

ROUTINE	VECTOR ADDRESS (hex)	PARAMETERS
INITG	104	Returns display description in H,L
PAGE	107	None
CURSOR	10A	H = X coord; L = Y coord
DOT	10D	None
LINE	110	H = X end coord; L = Y end coord
CHAR	113	A = ASCII value of character
ANIMAT	116	None

Table 2-1 MTXGRAPH Entry Vector Addresses

H								L							
ANIM	MRCOLS				MRSCLF			COL	MCCOLS				MCSCLF		
D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
MAXRD								MAXCD							

ANIM = 1 - Delay to start of vertical blanking.
       0 - Double buffered animation supported.

COL = 1 - Display is in color.
      0 - Display is black and white.

MRCOLS - Colors (grey shades) in MAXR mode.

MCCOLS - Colors (grey shades) in MAXC mode.

MRSCLF - $\frac{256}{\text{Display resolution}}$ in MAXR mode.

MCSCLF - $\frac{256}{\text{Display resolution}}$ in MAXC mode.

Figure 2-1 Display Parameter Definitions only way to erase what has been written is to page the display. The scale factor field indicates the physical size of display points in standard coordinates. If the X and Y scale factors differ, the larger of the two is used. For example, if a display has 64 lines with 100 points on each, the scale factor would be four, based on the Y axis resolution. For the ALT-256**2, the scale factors are one and each board is one bit in the colors field.

The animation and color fields apply to all display modes. If the animation field is one, then the display supports double buffered animation. Since Matrox displays do not support building one display scene while another is displaying, this field is zero. Therefore, the ANIMAT routine is a delay until the start of vertical blanking to permit synchronizing updates. The color/BW field is self-explanatory. If one, the display is in color, otherwise it is black (grey) and white. Note that this field has no meaning if the number of available colors is zero or one.

2.2 Display Controls

To the maximum extent possible, MTXGRAPH emulates a standardized display device. This standard device displays 256 lines with 256 points on each line. Each addressable point may be set to black, white, red, yellow, green, cyan, blue, or magenta. That is, any of the eight possible combinations of the three primary colors. A three board system therefore is a hardware implementation of the ideal standard display. While MTXGRAPH will support up to eight boards, it is impractical to use the CHAR routine for color control with more than four boards. Larger systems require direct manipulation of the color byte (see section 4.1).

Four standard display modes and one special mode are implemented (table 2-2). Mode zero (MAXR) requests the maximum possible resolution while mode one (MAXC) requests the maximum choice of colors. This allows for displays which offer a trade-off between resolution and color selection. For the ALT-256**2 they are identical. Modes two (R128), three (R64),

MNEMONIC	ASCII	HEX	STANDARD FUNCTION	SINGLE ALT-256**2
			Display Mode	Selection
MAXR	NUL	00	Maximum resolution	256x256 B/W
MAXC	SOH	01	Maximum colors	256x256 B/W
R128	STX	02	128 by 128	128x128 B/W
R64	ETX	03	64 by 64	64x64 B/W
RXXX	EOT	04	Undefined	64x64 DOT size with 256x256 placement resolution
			Carriage	Control
BS	BS	08	Backspace (optional)	X = X - 6
HT	HT	09	Hor. Tab (optional)	X = (X + 32) MOD 32
LF	LF	0A	Line Feed	Y = Y - 8
VT	VT	0B	Vert. Tab (optional)	Y = (Y MOD 32) - 8
FF	FF	0C	Form Feed	X = 0; Y = -6
CR	CR	0D	Carriage Return	X = 0
			Character	Spacing
SO	SO	0E	Undefined	Fixed
SI	SI	0F	Undefined	Proportional
			Current Color	Selection
BLK	DLE	10	Black	Black
RED	DC1	11	Red	White
BLU	DC2	12	Blue	White
MAG	DC3	13	Magenta	White
GRN	DC4	14	Green	White
YEL	NAK	15	Yellow	White
CYN	SYN	16	Cyan	White
WHI	ETB	17	White	White
NONE	ETX to GS	18 to 1F	Eight optional colors	White

Table 2-2 Control Character Functions

and four (RXXX) provide the ability to deliberately select larger size "DOT"s. As is implied by the mnemonics, R128 is 128 by 128 resolution with four points in each "DOT" and R64 is 64 by 64 resolution with sixteen points in each "DOT". RXXX combines the 64 by 64 "DOT" size with full 256 by 256 dot placement resolution. This is a nonstandard mode and characters generated in this mode are unreadable. All modes use the full 256 by 256 coordinate system with modes R128 and R64 ignoring the low order bit and two bits respectively. Cursor position is not maintained when shifting to or from modes R64 or R128. (Note that the standard does not require the cursor to be maintained with any display mode change.)

To enhance the readability of textual output, a choice between fixed and proportional character spacing is provided. Default mode (set by INITG) is proportional spacing. This provides a higher character density while actually promoting legibility. Like a newspaper, narrow characters use less space than normal while wide characters take more. Fixed character spacing requires every character to use the maximum width, but also results in vertical alignment of the characters in each line. This is particularly beneficial when using the display as a TV typewriter.

Most of the carriage controls are self-explanatory, however, some of them do have nonstandard effects. Backspace backs up the width of a maximum width character. Consequently it is not generally useable when in proportional spacing mode. Similarly, horizontal tab stops do not match character spacing in either mode. Tabs are set at even multiples of thirty-two "DOT"s across the screen. Finally, form-feed does not erase the screen but only homes the cursor to the upper left character position.

As hinted earlier, there are sixteen controls for setting the current color, one for each of a maximum of sixteen colors. The first eight are the defined standards, the second eight are user assigned. The standard colors are arranged so that regardless of the number of boards implemented, at least one will be set for any color other than black (see section 4.1).

3. SUBROUTINE DESCRIPTIONS

3.1 INITG

```
LHLD    [FIRST-FREE]    ;Refresh buffer area (ignored)
CALL    INITG
MOV     [MAXCD], L      ;Save display description
MOV     [MAXRD], H
```

The INITG routine serves several functions. As an aid to the user, the display software is initialized to the standard configuration: the cursor is positioned at X = 0, Y = 0; the current color is set to white; the display is cleared; proportional character spacing is selected; and the display mode is set for maximum resolution (mode 0). All special options are disabled so that general purpose programs do not need to know about them to function correctly. Because this routine performs all initialization functions required by the other display routines, it must be called before any other graphics package routines are used. INITG also returns the display parameters in register pair H,L. The interpretation of these parameters is described in section 2.1. INITG is the only routine which returns any values to the calling program.

3.2 PAGE

```
CALL PAGE
```

The PAGE routine clears the display screen. No other changes are made to the state of the display. In particular, the cursor is not moved, the current color is not changed and the display mode is unaffected.

3.3 CURSOR

```
MOV     H,[X]
MOV     L,[Y]
CALL CURSOR
```

The CURSOR routine resets the software display cursor to a particular point on the screen. This establishes the starting location for routines DOT, LINE, and CHAR. Coordinates are always interpreted using the full

256 by 256 coordinate system regardless of the resolution in use. When in a lower resolution mode, the low order bits of the position requested are ignored. For example, when in 128 by 128 resolution mode (mode 2), the points (8,4), (8,5), (9,4), and (9,5) will all be interpreted as the same "DOT" (the low order bit in each coordinate has no effect). When changing between display modes, cursor position is not always maintained by the graphics package. To avoid erroneous results, all changes to display mode should be followed by a cursor positioning command.

3.4 DOT

```
CALL DOT
```

The DOT routine sets the display point(s) indicated by the cursor to the currently selected color. This results in several points being written to form each dot when in lower resolution modes. For example, 16 hardware points are affected for every "DOT" when in 64 by 64 resolution mode.

3.5 LINE

```
MOV     H,[XF]          ;X coord of destination
MOV     L,[YF]          ;Y coord of destination
CALL LINE
```

The LINE routine generates the line connecting the point defined by the cursor to the point whose coordinates are in H and L. Both endpoints are included in the line. A line of zero length therefore is equivalent to a call to DOT. When erasing or otherwise changing the color of an existing line, care must be exercised as the "DOT"s making up the line from X_1,Y_1 to X_2,Y_2 may not be the same "DOT"s used when the line is drawn from X_2,Y_2 to X_1,Y_1. The cursor is left positioned at the coordinates specified in H,L.

3.6 CHAR

```
MOV     A,[VAL]         ;ASCII character
CALL CHAR
```

The CHAR routine provides the capability to display alphanumeric, as well as graphical data. Control characters provide for cursor positioning and control over display mode, spacing mode, and current color as discussed in section 2.2. Control characters not recognized are ignored. Note that form feed only positions the cursor, it does not erase the screen.

Characters are positioned so that the cursor defines the lower left corner of a standard character (characters with descenders will extend below the cursor position). The cursor is left positioned at the next character position. No check is made to detect characters off the edge of the screen. Parity is ignored. Lower case characters are converted to upper case.

3.7 ANIMAT

CALL ANIMAT

The ANIMAT routine provides for flicker free changes in the display by permitting the user to synchronize refresh buffer updates with the vertical blanking period. Each call to ANIMAT results in a delay until the start of the next vertical blanking period. Upon return from a call to ANIMAT, the caller is guaranteed a full vertical blanking period to make any desired display changes. Note that the standard's ANIMAT allows for double buffering. To write totally compatible software requires examination of the animation support bit returned by INITG to determine the animation mode supported by the particular display.

4. AVAILABLE USER ADJUSTMENTS

The software as supplied assumes a single ALT-256**2 strapped for I/O ports 10 thru 13(hex) and responding to data bit 0. The following modifications to the code allow customizing the graphics package to suit individual requirements. Before making any changes, be sure to verify that the software (check paper tape label) and the patch table (table 4-1) version numbers correspond.

4.1 Multiboard Systems

As mentioned previously, the basic package will support systems with up to eight ALT-256**2 boards slaved in sync, as long as they all share the same I/O port addresses. To permit general purpose applications programs to fully utilize the additional capability, it is recommended that the two bytes of display parameters returned by INITG be modified to accurately reflect the hardware available.

The positions of the various fields in the display description bytes are defined in figure 2-1. The fields which normally need modification are the color/BW bit and the available colors fields. If the system implemented is a color system, bit 7 (the higher order bit) of location MAXCD should be set to one. Regardless of whether the implementation is color or grey scale, set bits 3 and 4 of both MAXCD and MAXRD to one for a two board system, bits 3, 4, and 5 to one for a three board system, and bits 3, 4, 5, and 6 to one for a four or more board system. Except for the high order bit, MAXCD and MAXRD should be identical. For example, a three board full color system would have MAXCD equal to B9(hex) and MAXRD equal to 39(hex).

For two board systems, the hardware should be strapped so that one board responds to data bit 1 and the other to data bit 2. For best results with color systems, board 2 should be the complementary color for board 1. For grey scale applications, bit 2 is the high order bit.

For three board systems, the boards should be strapped to bits 3, 4, and 5. The standard assumes board 3 is red, board 4 is blue, and

Name	Address (hex)	Original Contents (hex)
MAXCD	012D	09
MAXRD	012E	09
ANTWRN	011F	00
COLORS	04C1 - 04D0	See Table 4.2
COLOR	04F3	Variable
MTRO	0135, 0171, 018D, 0194, 019B, 01A2, 01BA, 0397, 03CE, 03D5	10
MTRX	0168, 0185, 0198, 01B1, 0393	11
MTRY	016C, 018A, 0191, 019F, 01B7, 038E	12
MTRB	0133	13

Table 4-1 Patch Addresses for MTXGRAPH Version _____

board 5 is green. Grey scale will be linear if board 5 is twice as bright as board 4 which is twice as bright as board 3. (Board numbers refer to the data bit the board is connected to, not the number of boards in the system.)

A fourth board would be strapped to data bit 6 in a grey scale system. For color use, there is no preferred position. The values of each data bit for each of the sixteen selectable colors/grey shades is given in table 4-2. If different bit patterns are required, the lookup table COLORS may be altered as required. The first entry (black) is not required to be zero by the software, however, a non-zero value would result in black and clear screen using different colors.

To make maximum use of more than four boards, it is necessary to directly manipulate the color byte at location COLOR. However, this does reduce the hardware and software independence gained by using MTXGRAPH. In particular, COLOR is not a guaranteed location and may change in future releases of MTXGRAPH.

4.2 Animation Warning Message

As a service to users of general applications software which sometimes utilizes double buffered animation without first checking for its availability, the routine ANIMAT displays a warning message the first time it is called after initialization. If this warning is objectionable, it may be eliminated by changing location ANIWRN from 00 to FF(hex).

4.3 Changing I/O Port Assignments

MTXGRAPH is normally supplied to use I/O ports 10, 11, 12, and 13(hex). If this conflicts with your system configuration, it is necessary to change all the IN and OUT commands in the package. All the addresses listed in table 4-1 under MTRO sould be modified to contain the I/O port number actually assigned to the ALT-256**2(s) control port. Similarly, all locations under MTRX get the new X cursor port, those under MTRY get the Y cursor port and those under MTRB get the page set port.

ARG TO CHAR (hex)	STD COLOR	DATA BIT 7 6 5 4	3 2 1 0	1 BD note 1	2 BDS BW note 2	2 BDS COL note 3	3 BDS BW note 4	3 BDS COL note 5	4 BDS BW note 6
10	Blk	0 0 0 0	0 0 0 0	Blk	Blk	Blk	Blk	Blk	Blk
11	Red	1 0 0 0	1 0 1 1	Whi	G5	Red	G2	Red	G1
12	Blu	1 0 0 1	0 0 1 1	Whi	G5	Red	G4	Blu	G2
13	Mag	1 0 0 1	1 0 1 1	Whi	G5	Red	G6	Mag	G3
14	Grn	1 0 1 0	0 1 0 1	Whi	G10	Cyn	G8	Grn	G4
15	Yel	1 0 1 0	1 1 0 1	Whi	G10	Cyn	G10	Yel	G5
16	Cyn	1 0 1 1	0 1 0 1	Whi	G10	Cyn	G12	Cyn	G6
17	Whi	1 0 1 1	1 1 1 1	Whi	Whi	Whi	Whi	Whi	G7
18	-	0 1 0 0	0 1 0 1	Whi	G10	Cyn	Blk	Blk	G8
19	-	0 1 0 0	1 1 0 1	Whi	G10	Cyn	G2	Red	G9
1A	-	0 1 0 1	0 1 0 1	Whi	G10	Cyn	G4	Blu	G10
1B	-	0 1 0 1	1 1 0 1	Whi	G10	Cyn	G6	Mag	G11
1C	-	0 1 1 0	0 0 1 1	Whi	G5	Red	G8	Grn	G12
1D	-	0 1 1 0	1 0 1 1	Whi	G5	Red	G10	Yel	G13
1E	-	0 1 1 1	0 0 1 1	Whi	G5	Red	G12	Cyn	G14
1F	-	0 1 1 1	1 1 1 1	Whi	Whi	Whi	Whi	Whi	Whi

notes:
1) Single board responding to data bit 0.
2) Grey 5 on bit 1; Grey 10 on bit 2.
3) Red on bit 1; Cyan on bit 2.
4) Grey 2 on bit 3; Grey 4 on bit 4; Grey 8 on bit 5.
5) Red on bit 3; Blue on bit 4; Green on bit 5.
6) Grey 1 on bit 3; grey 2 on bit 4; Grey 4 on bit 5; Grey 8 on bit 6.

Table 4-2 Color/Grey Scale Assignments

5. SAMPLE USER PROGRAM

The demonstration program provided with the package is more than just a hardware checkout test. It also illustrates how to use the various graphics subroutines and take full advantage of whatever capabilities the display system hardware has to offer. In addition, it indicates some of the techniques useful when writing general purpose programs designed to use with any display meeting the software protocol standard.

6. LOADING & RUNNING MTX GRAPH WITH DEMO PROGRAM

The MTX GRAPH software package is designed for use with Matrox ALT-256 graphics display. The package is configured as a series of callable sub-routines. The MTX GRAPH package occupies memory locations 0104 to 04FF.

A second program supplied is intended for demonstration purposes. The program utilizes the MTX GRAPH sub-routines to create a continuous live action graphics display.

The tapes are supplied in Imsai binary loader compatible format. A listing of the loader and instructions for its use are given in Section 7.

The Imsai loader should be used to load first MTX GRAPH and then the demo program. Both must be co-resident to use the demonstration program. The ALT-256 should be jumpered for address 10-13 (hex) and data bit DØ. (jumper A2, 3, 5, 6, 7 set to Ø, A4 set to 1. See manual P9). Start the computer at location 0500 Hex. Sit back and watch the show. The demonstration program will pause whenever the dot switches are set to 01 (hex).

NOTE: Old versions of the ALT-256 require the following hardware change to work with this software package: (units shipped prior to Aug. 1/77) Tie A3 and A11 pin 4 to +5V with jumper wires soldered directly to the artwork.

7 - USING THE IMSAI BINARY LOADER

To use this loader, first key it in, starting at location 1000H. Then mount the tape in the reader on the teletype, set the address switches to 1000H, then press 'stop', 'reset', 'examine', and 'run' (i.e. start the program at 1000H). Then start the teletype reader. NOTE: for reasons of brevity, this loader does not check the checksum on the tape.

A detailed listing of the Imsai loader is attached. An abbreviated list in straight hex code is given below. The original Imsai code is for a USART chip at ports 02 and 03 hex. A second abbreviated listing is given for a UART at port 00 and 01 hex. The code may be modified by the user for the port addressing of his serial interface.

For USART at ports 02 - 03, Normal Imsai Loader.

```
1ØØØ  3E  CE  D3  Ø3  3E  17  D3  Ø3  3E  11  D3  Ø2  31  67  1Ø  CD
1Ø1Ø  51  1Ø  FE  3A  C2  ØF  1Ø  CD  37  1Ø  B7  CA  36  1Ø  47  CD
1Ø2Ø  37  1Ø  67  CD  37  1Ø  6F  CD  37  1Ø  CD  37  1Ø  77  23  Ø5
1Ø3Ø  C2  2A  1Ø  C3  ØF  1Ø  76  CD  44  1Ø  87  87  87  87  57  CD
1Ø4Ø  44  1Ø  B2  C9  CD  51  1Ø  FE  3A  FA  4E  1Ø  C6  Ø9  E6  ØF
1Ø5Ø  C9  DB  Ø3  E6  Ø2  CA  51  1Ø  DB  Ø2  E6  7F  C9
```

For UART at port 00 - 01, Modified Imsai Loader

```
1ØØØ  ØØ  ØØ  ØØ  ØØ  ØØ  ØØ  ØØ  ØØ  ØØ  ØØ  ØØ  ØØ  31  67  1Ø  CD
1Ø1Ø  51  1Ø  FE  3A  C2  ØF  1Ø  CD  37  1Ø  B7  CA  36  1Ø  47  CD
1Ø2Ø  37  1Ø  67  CD  37  1Ø  6F  CD  37  1Ø  CD  37  1Ø  77  23  Ø5
1Ø3Ø  C2  2A  1Ø  C3  ØF  1Ø  76  CD  44  1Ø  87  87  87  87  57  CD
1Ø4Ø  44  1Ø  B2  C9  CD  51  1Ø  FE  3A  FA  4E  1Ø  C6  Ø9  E6  ØF
1Ø5Ø  C9  DB  ØØ  E6  Ø1  C2  51  1Ø  DB  Ø1  E6  7F  C9
```

```
                ;
                ; *** SCS LOADER  REV 1. ***
                ;
                ;
                ;
                ; TO USE THIS LOADER, FIRST KEY IT IN, STARTING
                ; AT LOCATION 1000H.  THEN MOUNT THE TAPE IN THE READER
                ; ON THE TELETYPE, SET THE ADDRESS SWITCHES TO 1000H,
                ; THEN PRESS 'STOP', 'RESET', 'EXAMINE', AND 'RUN'.
                ; THE TELETYPE READER SHOULD START AUTOMATICALLY.
                ;
                ; NOTE: FOR REASONS OF BREVITY, THIS LOADER DOES NOT
                ; CHECK THE CHECKSUMS ON THE TAPE.
                ;
0000                    ORG     1000H
                ;
0002            TTY     EQU     2       ;TELETYPE DATA PORT
0003            TTS     EQU     3       ;TETELYPE STATUS PORT
0002            TTR     EQU     2       ;TELETYPE READY BIT
0011            RON     EQU     11H     ;READER ON BIT
                ;
                ; THE LOADER BEGINS HERE
                ;
1000 3ECE       LOAD:   MVI     A,0CEH  ;GET MODE COMMAND
1002 D303               OUT     TTS     ;ISSUE IT
1004 3E17               MVI     A,17H   ;GET COMMAND
1006 D303               OUT     TTS     ;ISSUE IT
1008 3E11               MVI     A,RON   ;GET 'X-ON' CHAR
100A D302               OUT     TTY     ;START THE READER
100C 316710             LXI     6,PEND+10 ;SET UP THE STACK
                ;
                ; GO THROUGH THIS LOOP ONCE FOR EACH RECORD
                ;
100F CD5110     LOOP1:  CALL    INCH    ;GET A CHARACTER
1012 FE3A               CPI     ':'     ;IS IT A COLON
1014 C20F10             JNZ     LOOP1   ;WAIT FOR COLON
1017 CD3710             CALL    GETBT   ;GET THE COUNT
101A B7                 ORA     A       ;SET FLAGS
101B CA3610             JZ      EOF     ;BRANCH IF EOF RECORD
101E 47                 MOV     B,A     ;ELSE, PUT COUNT INTO B REG
101F CD3710             CALL    GETBT   ;GET HI BYTE OF ADDR
1022 67                 MOV     H,A     ; INTO H
1023 CD3710             CALL    GETBT   ;GET LO BYTE OF ADDRESS
1026 6F                 MOV     L,A     ; INTO L
1027 CD3710             CALL    GETBT   ;GET TYPE BYTE AND IGNORE
                ;
                ; GO THROUGH THIS LOOP ONCE FOR EACH DATA BYTE IN
                ; A RECORD
                ;
102A CD3710     LOOP2:  CALL    GETBT   ;GET A DATA BYTE
102D 77                 MOV     M,A     ;STORE IT
102E 23                 INX     H       ;BUMP ADDRESS
102F 05                 DCR     B       ;DECREMENT COUNT
1030 C22A10             JNZ     LOOP2   ;DO IT AGAIN
1033 C30F10             JMP     LOOP1   ;GO GET NEXT RECORD
```

```
                  ;
1036 76          EOF:   HLT
                  ;
                  ;
                  ; THIS ROUTINE READS TWO CHARACTERS FROM
                  ; THE TAPE, AND ASSEMBLES THEM INTO A BYTE, WHICH
                  ; IS RETURNED IN THE A REGISTER
                  ;
1037 CD4410      GETBT:  CALL    INDIG     ;GET A DIGITI
103A 87                  ADD     A         ;SHIFT IT ONE BIT
103B 87                  ADD     A         ;   TWO BITS
103C 87                  ADD     A         ;   THREE BITS
103D 87                  ADD     A         ; AND FOUR BITS
103E 57                  MOV     D,A       ;SAVE IT IN D
103F CD4410              CALL    INDIG     ;GET ANOTHER DIGIT
1042 B2                  ORA     D         ;OR IN LAST DIGIT
1043 C9                  RET               ;AND RETURN
                  ;
                  ; THIS ROUTINE READS A HEX DIGIT FROM THE
                  ; TAPE. NOTE THAT IT DOES NO VALIDITY CHECKING.
                  ;
1044 CD5110      INDIG:  CALL    INCH      ;GET A CHAR FROM THE TAPE
1047 FE3A                CPI     '9'+1     ;CHECK FOR NUMERIC
1049 FA4E10              JM      IND1      ;SKIP IF NUMERIC
104C C609                ADI     9         ;ELSE FUDGE, SO 'A'-'F' => 4A -
104E E60F        IND1:   ANI     0FH       ;MASK OFF LO 4 BITS
1050 C9                  RET               ;AND RETURN
                  ;
                  ; THIS ROUTIEN READS A CHARACTER FROM THE TELETYPE
                  ; PAPER TAPE READER
                  ;
1051 DB03        INCH:   IN      TTS       ;GET TELETYPE STATUS
1053 E602                ANI     TTR       ; WAIT TILL READY
1055 CA5110              JZ      INCH
1058 DB02                IN      TTY       ;GET THE CHAR
105A E67F                ANI     7FH       ;KILL THE PARITY BIT
105C C9                  RET
                  ;
105D             PEND    EQU     $
0000                     END
```

```
                ;       GRAPHICS PACKAGE DEMONSTRATION PROGRAM
                ;           VERSION 2.06M  <>  AUG 25,1977
                ;
                ;                  COPYRIGHT 1977
                ;               DR. VINCENT C. JONES
                ;               11017 BENNINGTON AVE
                ;               KANSAS CITY MO 64134
                ;
0500                    ORG     500H    ;START AFTER GRAPHICS PACKAGE
                ;;
                ;DEFININTION OF GRAPHICS PACKAGE ENTRY POINTS
                ;
0104 =                  INITG   EQU     104H
0107 =                  PAGE    EQU     107H
010A =                  CURSOR  EQU     10AH
010D =                  DOT     EQU     10DH
0110 =                  LINE    EQU     110H
0113 =                  CHAR    EQU     113H
0116 =                  ANIMAT  EQU     116H

                ;
                ;    AND STANDARD COLORS
                ;
0010 =                  BLK     EQU     10H     ;DELETE
0011 =                  RED     EQU     11H     ;RED
0015 =                  YEL     EQU     15H     ;YELLOW
0014 =                  GRN     EQU     14H     ;GREEN
0016 =                  CYN     EQU     16H     ;CYAN
0012 =                  BLU     EQU     12H     ;BLUE
0013 =                  MAG     EQU     13H     ;MAGENTA
0017 =                  WHI     EQU     17H     ;WHITE

                ;
                ;    AND TIMES FOR PAUSE (ASSUMES 2MHZ CLOCK)
                ;
0001 =                  HALF    EQU     1       ;HALF A SEC
0002 =                  ONE     EQU     2
0004 =                  TWO     EQU     4
000A =                  FIVE    EQU     0AH
0014 =                  TEN     EQU     14H

                ;
                ;     AND DISPLAY MODES
                ;
0000 =                  MAXR    EQU     00H     ;MAXIMUM RESOLUTION
0001 =                  MAXC    EQU     01H     ;MAXIMUM COLOR CHOICE
0002 =                  R128    EQU     02H     ;128 BY 128
0003 =                  R64     EQU     03H     ;64 BY 64
0004 =                  RXXX    EQU     04H     ;WHO KNOWS
                ;
                ;       HERE STARTS THE ACTUAL PROGRAM
                ;
0500 312D09     DEMO:   LXI     SP,STACK ;INIT STACK
0503 212E09             LXI     H,STACK+1 ;1ST FREE LOC
0506 CD0401             CALL    INITG   ;GET THE DISPLAY GOING
0509 222B00             SHLD    MAXCD   ;SAVE DISPLAY PARAMETERS
                ;
```

```
                  ;DEMO #1:   HIGH RESOLUTION LINE DEMO
                  ;
050C 3E03         DEM1:   MVI     A,R64    ;64 BY 64 MODE
050E CD1301               CALL    CHAR
0511 21A444               LXI     H,44A4H  ;PUT UP LOGO
0514 CD0A01               CALL    CURSOR
0517 21BB07               LXI     H,STR00  ;MATROX
051A CD9107               CALL    STRING
051D 210034               LXI     H,3400H
0520 CD0A01               CALL    CURSOR
0523 21C707               LXI     H,STR01  ;GRAPHICS
0526 CD9107               CALL    STRING
0529 215C3C               LXI     H,3C5CH
052C CD0A01               CALL    CURSOR
052F 21D207               LXI     H,STR02  ;SYSTEMS
0532 CD9107               CALL    STRING
0535 0605                 MVI     B,5      ;LET PEOPLE READ IT
0537 CDA107               CALL    PAUSE
053A 3E00                 MVI     A,MAXR   ;SHIFT TO MAX RES
053C CD1301               CALL    CHAR
053F 01FF00               LXI     B,00FFH  ;EXTREMES OF DISPLAY
0542 50                   MOV     D,B      ;D = CURRENT STEP
0543 79           DEM01:  MOV     A,C      ;E = MAX-CURRENT
0544 92                   SUB     D
0545 5F                   MOV     E,A
0546 62                   MOV     H,D      ;STARTING POINT
0547 68                   MOV     L,B      ; IS D,0
0548 CD0A01               CALL    CURSOR
054B 61                   MOV     H,C      ;TO 255,D
054C 6A                   MOV     L,D
054D CD1001               CALL    LINE
0550 63                   MOV     H,E      ;TO 255-D,255
0551 69                   MOV     L,C
0552 CD1001               CALL    LINE
0555 60                   MOV     H,B      ;TO 0, 255-D
0556 6B                   MOV     L,E
0557 CD1001               CALL    LINE
055A 62                   MOV     H,D      ;AND FINALLY
055B 68                   MOV     L,B      ;  BACK TO D,0
055C CD1001               CALL    LINE
055F 3A2C08               LDA     MAXRD    ;DETERMINE NEXT D
0562 E607                 ANI     07H      ;WIDTH OF A POINT
0564 C26805               JNZ     DEM12    ;MAKE SURE AT LEAST ONE
0567 3C                   INR     A
0568 17           DEM12:  RAL              ;MOVE 8 DISPLAY POINTS
0569 17                   RAL
056A 17                   RAL
056B 82                   ADD     D        ;D = D+RES*8
056C 57                   MOV     D,A
056D D24305               JNC     DEM01    ;MORE TO GO
0570 0614                 MVI     B,TEN
0572 CDA107               CALL    PAUSE    ;SHOW IT OFF A BIT
                  ;
                  ;DEMO #2:   SHOW OFF CHARACTER SET
                  ;
0575 CD0701               CALL    PAGE
0578 3E03                 MVI     A,R64    ;USE BIGGEST CHARACTERS POSSIBLE

057A CD1301               CALL    CHAR
```

```
057D 0E07                MVI     C,7      ;INIT COLOR COUNTER
057F 3E10       DEM2:    MVI     A,BLK    ;START OUT WHITE
0581 81                  ADD     C
0582 CD1301              CALL    CHAR
0585 0620                MVI     B,' '    ;FIRST CHAR IS 'BLANK'
0587 21E808              LXI     H,08E8H  ;UPPER LEFT CORNER
058A CD0A01     DEM20:   CALL    CURSOR   ;POSITION CHARACTER
058D 78                  MOV     A,B      ;  AND DISPLAY IT
058E CD1301              CALL    CHAR
0591 3E20                MVI     A,20H    ;MOVE OVER TO NEXT POSITION
0593 84                  ADD     H
0594 67                  MOV     H,A
0595 D29F05              JNC     DEM21    ;PAST END OF LINE?
0598 7D                  MOV     A,L      ;MOVE DOWN TO NEXT
0599 D620                SUI     20H
059B 6F                  MOV     L,A
059C DAA305              JC      DEM22    ;DONE IF OFF BOTTOM
059F 04         DEM21:   INR     B        ;NEXT CHARACTER
05A0 C38A05              JMP     DEM20
05A3 0602       DEM22:   MVI     B,ONE    ;LOOK AT IT A BIT
05A5 CDA107              CALL    PAUSE    ;AREN'T THEY PRETTY?
05A8 0D                  DCR     C        ;TRY A NEW COLOR
05A9 F27F05              JP      DEM2     ;GO UNTIL ERASE
05AC CD0701              CALL    PAGE     ;JUST IN CASE A TEK 4010
05AF 21C04D              LXI     H,4DC0H  ;MUST BE IN R64 AT THIS POINT
05B2 CD0A01              CALL    CURSOR
05B5 210208              LXI     H,STR06  ;'LARGE' + SHIFT TO MAXR
05B8 CD9107              CALL    STRING
05BB 218069              LXI     H,6980H
05BE CD0A01              CALL    CURSOR
05C1 210A08              LXI     H,STR07  ;'AND SMALL' + SHIFT TO R128
05C4 CD9107              CALL    STRING
05C7 21404D              LXI     H,4D40H
05CA CD0A01              CALL    CURSOR
05CD 211508              LXI     H,STR08  ;'CHARACTERS'
05D0 CD9107              CALL    STRING
05D3 0614                MVI     B,TEN    ;LET THAT SOAK IN
05D5 CDA107              CALL    PAUSE    ;  FOR A WHILE
05D8 CD0701              CALL    PAGE     ;CLEAR FOR NEXT DEMO
                ;
                ;DEMO #3:   FULL COLOR CONTROL
                ;
05DB 3E03                MVI     A,R64    ;LARGE LABELS
05DD CD1301              CALL    CHAR
05E0 21985C              LXI     H,5C98H
05E3 CD0A01              CALL    CURSOR
05E6 21DD07              LXI     H,STR03  ;'FULL' + COLOR SELECT
05E9 CD9107              CALL    STRING
05EC 3A2B08              LDA     MAXCD    ;COLOR CHOICE AVAILABLE?
05EF E670                ANI     70H      ;2 OR MORE?
05F1 0E11                MVI     C,BLK+1  ;ASSUME NOT
05F3 CA1906              JZ      DEM3X    ;GOOD ASSUMPTION
05F6 0E17                MVI     C,WHI    ;ASSUME 8 COLOR
05F8 E640                ANI     40H      ;MORE THAN 8?
05FA CAFF05              JZ      DEM3X1   ;  NO.  GOOD ASSUMPTION
05FD 0E1F                MVI     C,BLK+15 ;GO FOR 16 COLORS
05FF 217450     DEM3X1:  LXI     H,5074H
0602 CD0A01              CALL    CURSOR
0605 3A2B08              LDA     MAXCD    ;COLOR OR MONOCHROME?
```

```
0608 E680              ANI     80H      ;  CHECK THE BIT
060A C21306            JNZ     DEM3X2   ;COLOR!
060D 21EE07            LXI     H,STR04G  ;'TONAL'
0610 C31606            JMP     DEM3X3
0613 21E307  DEM3X2:   LXI     H,STR04  ;'COLOR' IN COLOR
0616 CD9107  DEM3X3:   CALL    STRING
0619 21503C  DEM3X:    LXI     H,3C50H
061C CD0A01            CALL    CURSOR
061F 21F907            LXI     H,STR05  ;'CONTROL' IN WHITE
0622 CD9107            CALL    STRING
0625 1600              MVI     D,00H    ;INIT COUNTER
0627 3E01              MVI     A,MAXC   ;MAXIMUM COLORS (GREY SHADES)
0629 CD1301            CALL    CHAR
062C 59      DEM30:    MOV     E,C      ;START OUT WHITE
062D 26FF    DEM3:     MVI     H,0FFH   ;FROM TOP RIGHT SIDE
062F 7A                MOV     A,D      ;WHICH STEP?
0630 E63F              ANI     3FH      ;NORMALIZE TO 64
0632 17                RAL              ;AND SCALE BACK
0633 17                RAL              ;  TO 256
0634 6F                MOV     L,A
0635 CD0A01            CALL    CURSOR
0638 210000            LXI     H,0000H  ;TO LOWER LEFT CORNER
063B CD1001            CALL    LINE
063E 2EFF              MVI     L,0FFH   ;UP TO TOP EDGE
0640 67                MOV     H,A
0641 CD1001            CALL    LINE
0644 14                INR     D        ;NEXT STEP
0645 CA9706            JZ      DEM4     ;256 STEPS AND ALL DONE
0648 7A                MOV     A,D      ;WHAT STEP ARE WE ON?
0649 FE40              CPI     40H
064B DA6406            JC      DEM31    ;FULL COLOR, SLOW
064E CA7806            JZ      DEM32    ;SHIFT TO R64
0651 FE80              CPI     80H
0653 DA6406            JC      DEM31    ;R64, SLOW
0656 CA8D06            JZ      DEM33    ;BACK TO FULL COLOR
0659 FEC0              CPI     0C0H
065B DA6906            JC      DEM34    ;FULL COLOR, FULL SPEED
065E CA9206            JZ      DEM35    ;SHIFT TO ERASE
0661 C32D06            JMP     DEM3     ;FULL SPEED ERASE
0664 0601    DEM31:    MVI     B,HALF   ;SLIGHT PAUSE
0666 CDA107            CALL    PAUSE
0669 1D      DEM34:    DCR     E
066A 7B                MOV     A,E      ;BLACK YET?
066B FE10              CPI     BLK
066D F27106            JP      DEM37    ;NO
0670 59                MOV     E,C      ;START OVER
0671 7B      DEM37:    MOV     A,E      ;NEW COLOR
0672 CD1301  DEM38:    CALL    CHAR     ;  (OR MAYBE MODE)
0675 C32D06            JMP     DEM3
0678 3E03    DEM32:    MVI     A,R64    ;SHIFT TO 64 BY 64
067A CD1301            CALL    CHAR
067D 3E04              MVI     A,RXXX   ;TRY FOR SOMETHING SPECIAL!
067F 0604    DEM36:    MVI     B,TWO    ;PAUSE BETWEEN MODE CHANGES
0681 CDA107            CALL    PAUSE
0684 CD0701            CALL    PAGE     ;CLEAR SCREEN
0687 CD1301            CALL    CHAR     ;AND FINALLY CHANGE MODE
068A C32C06            JMP     DEM30
068D 3E01    DEM33:    MVI     A,MAXC   ;SHIFT TO MAX COLORS
068F C37F06            JMP     DEM36
```

```
0692 3E10        DEM35:  MVI     A,BLK    ;TIME TO ERASE
0694 C37206              JMP     DEM36
                 ;
                 ;DEMO #4: ANIMATION
                 ;
0697 3E03        DEM4:   MVI     A,R64    ;BIG CHARACTERS
0699 CD1301              CALL    CHAR
069C CD0701              CALL    PAGE
069F 218028              LXI     H,2880H ;POSITION TITLE
06A2 CD0A01              CALL    CURSOR
06A5 212008              LXI     H,STR09 ;'ANIMATION'
06A8 CD9107              CALL    STRING
06AB CD1601              CALL    ANIMAT   ;SWITCH BUFFERS
06AE CD0701              CALL    PAGE     ;CLEAR OTHER BUFFER
                                          ;  (OR ERROR MESSAGE IF ONE)
06B1 218028              LXI     H,2880H ;ADD TITLE THERE TOO
06B4 CD0A01              CALL    CURSOR
06B7 212008              LXI     H,STR09
06BA CD9107              CALL    STRING
06BD 3E02                MVI     A,R128   ;NEED SOME RESOLUTION
06BF CD1301              CALL    CHAR
06C2 0604                MVI     B,TWO    ;LET THIS SINK IN
06C4 CDA107              CALL    PAUSE
06C7 211200              LXI     H,18D    ;NEED SOME TERRA FIRMA
06CA CD0A01              CALL    CURSOR
06CD 26FF                MVI     H,0FFH
06CF CD1001              CALL    LINE
06D2 CD1601              CALL    ANIMAT   ;PUT IN BOTH BUFFERS
06D5 CD0A01              CALL    CURSOR
06D8 2600                MVI     H,00H
06DA CD1001              CALL    LINE
06DD 0610                MVI     B,16D    ;STARTING POSITION
06DF CD0E07      DEM40:  CALL    MAN      ;DRAW NEW FRAME
06E2 CD1601              CALL    ANIMAT   ;TRY EVEN IF NOT SUPPORTED
06E5 3E10                MVI     A,BLK    ;ERASE OLD ONE
06E7 CD1301              CALL    CHAR
06EA 3A2C08              LDA     MAXRD    ;DOUBLE BUFFERED?
06ED E680                ANI     80H      ;TEST BIT
06EF CAF306              JZ      DEM41    ;DON'T BACK UP TO ERASE
06F2 05                  DCR     B        ;BACK TO PREVIOUS FRAME
06F3 CD0E07      DEM41:  CALL    MAN      ;ALL GONE
06F6 3E17                MVI     A,WHI    ;BACK TO WHITE
06F8 CD1301              CALL    CHAR
06FB 3A2C08              LDA     MAXRD    ;DID WE DECREMENT BEFORE?
06FE E680                ANI     80H
0700 CA0407              JZ      DEM42    ;NO, SO DON'T DOUBLE INCR
0703 04                  INR     B        ;ADVANCE TO NEXT FRAME
0704 04          DEM42:  INR     B
0705 78                  MOV     A,B
0706 FEF0                CPI     0F0H     ;DONE YET?
0708 DADF06              JC      DEM40    ;KEEP ON TRUCKING
070B C30005              JMP     DEMO     ;START ALL OVER AGAIN
                 ;
                 ;SUBROUTINE TO DRAW A LITTLE MAN
                 ;
070E 78          MAN:    MOV     A,B      ;X IS IN B, 0 TO 255
070F E60F                ANI     0FH
0711 D608                SUI     08H      ;ARM SWING (AS)
0713 4F                  MOV     C,A      ;C = AS
```

```
0714 07              RLC               ;TIMES 2
0715 57              MOV     D,A       ;  FOR LEG SWING (LS)
0716 78              MOV     A,B       ;FIND FIXED LEG
0717 E6F0            ANI     0F0H
0719 C608            ADI     08H       ;R=[X AND F0]+8
071B 5F              MOV     E,A       ;E = R
071C 78              MOV     A,B       ;L = X+LS
071D 82              ADD     D
071E 57              MOV     D,A       ;D = L
071F 78              MOV     A,B
0720 3C              INR     A         ;A = X+1
0721 05              DCR     B         ;B = X-1
                ;ALL SET TO DRAW ALL BUT HEAD
0722 2E14            MVI     L,20      ;START WITH LEGS
0724 63              MOV     H,E
0725 CD0A01          CALL    CURSOR    ;R, 20
0728 67              MOV     H,A
0729 2E24            MVI     L,36
072B CD1001          CALL    LINE      ;X+1, 36 >> RT LEG
072E 60              MOV     H,B
072F CD1001          CALL    LINE      ;X-1, 36 >> HIPS
0732 62              MOV     H,D
0733 2E14            MVI     L,20
0735 CD1001          CALL    LINE      ;L, 20 >> LEFT LEG
0738 2E25            MVI     L,37      ;LEFT SIDE OF BODY
073A 67              MOV     H,A
073B CD0A01          CALL    CURSOR    ;X+1, 37
073E 2E30            MVI     L,48
0740 CD1001          CALL    LINE      ;X+1, 48 >> LEFT BODY
0743 2E26            MVI     L,38
0745 91              SUB     C
0746 67              MOV     H,A
0747 CD1001          CALL    LINE      ;X+1-AS, 38 >> LEFT ARM
074A 78              MOV     A,B       ;SAME FOR RIGHT SIDE
074B 67              MOV     H,A
074C 2E25            MVI     L,37
074E CD0A01          CALL    CURSOR    ;X-1, 37
0751 2E30            MVI     L,48
0753 CD1001          CALL    LINE      ;X-1, 48 >> RIGHT BODY
0756 2E26            MVI     L,38
0758 81              ADD     C
0759 67              MOV     H,A
075A CD1001          CALL    LINE      ;X-1+AS, 38 >> RT ARM
075D 04              INR     B         ;FINALLY DO HEAD
075E 2E30            MVI     L,48      ;NOTE: B IS BACK TO
                                       ;   ORIGINAL VALUE = X
0760 60              MOV     H,B
0761 CD0A01          CALL    CURSOR
0764 2E34            MVI     L,52
0766 CD1001          CALL    LINE      ;X, 52 >> NECK
0769 78              MOV     A,B
076A D602            SUI     2
076C 67              MOV     H,A
076D CD1001          CALL    LINE      ;X-2, 52
0770 2E38            MVI     L,56
0772 CD1001          CALL    LINE      ;X-2, 56
0775 78              MOV     A,B
0776 C602            ADI     2
0778 67              MOV     H,A
```

```
0779 CD1001           CALL    LINE      ;X+2, 56
077C 2E34             MVI     L,52
077E CD1001           CALL    LINE      ;X+2, 52
0781 60               MOV     H,B
0782 CD1001           CALL    LINE      ;X, 52  >> END OF HEAD
0785 2E36             MVI     L,54
0787 C602             ADI     2
0789 67               MOV     H,A
078A CD0A01           CALL    CURSOR    ;X+4, 54
078D CD0D01           CALL    DOT       ;THE NOSE!
0790 C9               RET

                ;
                ;COMMON SUBROUTINES
                ;
                ;       EXCEPT AS NOTED ALL REGISTERS ARE PRESERVED
                ;
                ;SUBOUTINE STRING
                ;       DISPLAY THE STRING OF ASCII CHARACTERS POINTED
                ;          TO BY H,L ERMINATING WITH "$".
                ;       EXITS WITH H,L POINTING TO THE "$".
0791 F5         STRING: PUSH    PSW       ;SAVE A AND FLAGS
0792 7E         STR000: MOV     A,M       ;GET CHAR
0793 FE24               CPI     '$'       ;TERMINATOR
0795 CA9F07             JZ      STR001
0798 CD1301             CALL    CHAR      ;OK SO DISPLAY IT
079B 23                 INX     H         ;NEXT CHAR
079C C39207             JMP     STR000
079F F1         STR001: POP     PSW       ;RESTORE A
07A0 C9                 RET
                ;
                ;SUBROUTINE PAUSE
                ;       DELAY A BIT AS DETERMINED BY REGISTER B.
                ;       EXITS WITH B=0.
                ;       WARNING: B = 0 IS MAX, NOT ZERO, DELAY.
                ;       SENSE SWITCHS SET TO 01H CAUSE INFINITE PAUSE.
                ;
07A1 E5         PAUSE:  PUSH    H         ;NEED A COUNT DOWN REGISTER
07A2 F5                 PUSH    PSW
07A3 210000             LXI     H,0000H   ;MAKE UNIT DELAY SHORT
07A6 2D         PAUS00: DCR     L         ;A 3 REGISTER COUNT DOWN
07A7 C2A607             JNZ     PAUS00
07AA 25                 DCR     H
07AB C2A607             JNZ     PAUS00
07AE 05                 DCR     B
07AF C2A607             JNZ     PAUS00
07B2 DBFF       PAUS01: IN      0FFH      ;READ SENSE SW
07B4 3D                 DCR     A         ;ARE THEY SET TO ONE?
07B5 CAB207             JZ      PAUS01    ;  YES
07B8 F1                 POP     PSW
07B9 E1                 POP     H
07BA C9                 RET
                ;
                ;STRING DEFINITIONS
                ;
07BB 174D41545253TR00:  DB      WHI,'MATROX$$$$$'
07C7 4752415048STR01:   DB      'GRAPHICS$$$'
07D2 5359535445STR02:   DB      'SYSTEM$$$$$'
07DD 174655404CSTR03:   DB      WHI,'FULL$'
```

```
07E3 1143154F145TR04:  DB     RED,'C',YEL,'O',GRN,'L',CYN,'O',BLU,'R$'

07EE 1154124F13STR04G: DB     11H,'T',12H,'O',13H,'N',14H,'A',15H,'L$'

07F9 17434F4E54STR05:  DB     WHI,'CONTROL$'
0802 174C415247STR06:  DB     WHI,'LARGE',MAXR,'$'
080A 414E442053STR07:  DB     'AND SMALL',R128,'$'
0815 4348415241STR08:  DB     'CHARACTERS$'
0820 17414E494DSTR09:  DB     WHI,'ANIMATION$'

                  ;
                  ;VARIABLE STORAGE AREA
                  ;
082B              MAXCD:   DS      1          ;MAXC DISPLAY DESCRIPTOR
082C              MAXRD:   DS      1          ;MAXR DISPLAY DESCRIPTOR
082D                       DS      100H       ;STACK
092D              STACK:   DS      1
092E                       END     DEMO
```

```
                  ;                THE VCJ GRAPHICS PACKAGE
                  ;                  8080/MATROX VERSION
                  ;              VERSION 1.05  <>  SEPT 3,1977
                  ;
                  ;                   COPYRIGHT 1977
                  ;                DR. VINCENT C. JONES
                  ;                11017 BENNINGTON AVE
                  ;                KANSAS CITY MO 64134
                  ;
                  ;
                  ;JUMP TABLE TO DEFINE STANDARD ENTRY POINTS
                  ;
0104                       ORG     104H    ;START OF STANDARD SPACE
0104 C31901                JMP     INITG   ;INITIALIZE GRAPHICS
0107 C33601                JMP     PAGE    ;CLEAR THE SCREEN
010A C33C01                JMP     CURSOR  ;GO TO A POINT ON THE SCREEN
010D C35A01                JMP     DOT     ;DISPLAY A POINT ON THE SCREEN
0110 C3C601                JMP     LINE    ;DRAW A LINE BETWEEN POINTS
0113 C36602                JMP     CHAR    ;DISPLAY AN ASCII CHARACTER
0116 C37903                JMP     ANIMAT  ;ANIMATION SEMI-SUPPORTED


                  ;DEFINE THE ALT-256**2 PORTS

00E0 =            MTR0     EQU     0E0H    ;CONTROL PORT
00E1 =            MTRX     EQU     MTR0+1  ;X COORDINATE
00E2 =            MTRY     EQU     MTR0+2  ;Y COORDINATE
00E3 =            MTRB     EQU     MTR0+3  ;BULK MODE

                  ;
                  ;ROUTINE INITG
                  ;        INITIALIZE THE MATROX TO 256 BY 256 B/W MODE,
                  ;           PROPORTIONAL SPACING, X = 0, Y = 0,
                  ;           SCREEN CLEARED & CURRENT COLOR SET TO WHITE.
                  ;        H,L CONTAINS FIRST AVAILABLE ADDRESS FOR
                  ;           REFRESH BUFFERS  (NOT USED).
                  ;        RETURNS DISPLAY CHARACTERISTICS IN H,L
                  ;
0119 F5           INITG:   PUSH    PSW     ;SAVE ACCULULATOR
011A AF                    XRA     A       ;CLEAR A
011B 21EF04                LXI     H,ANIM  ;START OF VARIABLE AREA
011E 3600                  MVI     M,00H   ;NO ATTEMPTS AT ANIMATION
                                           ;  CHANGE ARGUMENT TO FF(H) TO
                                           ;  INHIBIT ANIMATION WARNING.
0120 23                    INX     H       ;REAIM AT YPOS
0121 77                    MOV     M,A     ;   WHICH IS ZERO
0122 23                    INX     H       ;REAIM AT XPOS
0123 77                    MOV     M,A     ;  WHICH IS ZERO
0124 23                    INX     H       ;REAIM AT CHARACTER MODE
0125 360F                  MVI     M,0FH   ;  WHICH IS PROPORTIONAL SPACING

0127 23                    INX     H       ;REAIM AT CURRENT COLOR
0128 36FF                  MVI     M,0FFH  ;  WHICH IS WHITE
012A 23                    INX     H       ;REAIM AT MODE
012B 77                    MOV     M,A     ;  WHICH IS 256 BY 256
012C 210909                LXI     H,0909H ;LET THE USER KNOW WHAT'S HERE
012F F1                    POP     PSW     ;RESTORE A
```

```
                                        ;AND CLEAR THE SCREEN
                ;*****WARNING*****  THIS ROUTINE MUST BE IMMEDIATELY
                ;                   FOLLOWED BY THE PAGE ROUTINE.

                ;
                ;ROUTINE PAGE
                ;       CLEAR THE SCREEN
                ;
0130 F5         PAGE:   PUSH    PSW     ;SAVE THE USER WORLD
0131 AF                 XRA     A       ;NEED A ZERO
0132 D3E3               OUT     MTRB    ;SET ALL MEMORIES TO ZERO
0134 DBE0       P000:   IN      MTR0    ;DONE YET?
0136 1F                 RAR
0137 DA3401             JC      P000    ;KEEP TRYING
013A F1                 POP     PSW     ;
013B C9                 RET             ;ALL DONE

                ;
                ;ROUTINE CURSOR
                ;       POSITION THE CURSOR AT X,Y IN H,L
                ;           OR...
                ;       CONVERT THE COORDINATES IN H,L FROM IDEAL
                ;       COORDINATES (0 THRU 255 ON EACH AXIS) TO THE
                ;       COORDINATES CURRENTLY IN USE BY THE DISPLAY.
                ;
013C F5         CURSOR: PUSH    PSW     ;SAVE THE WORLD
013D C5                 PUSH    B       ;  OR AT LEAST THE
013E E5                 PUSH    H       ;      AFFECTED PART
013F CD4901             CALL    CU000   ;CONVERT TO COORD IN USE
0142 22F004             SHLD    YPOS    ;AND SAVE FOR OTHER PEOPLE
0145 E1                 POP     H       ;RESTORE THE WORLD
0146 C1                 POP     B       ;
0147 F1                 POP     PSW     ;
0148 C9                 RET             ;ALL DONE

                ;INTERNAL SUBROUTINE CU000
                ;       CONVERT THE X,Y COORDINATE PAIR IN H,L TO
                ;         THE COORDINATE SYSTEM CURRENTLY IN USE
                ;
                ;       USES REGISTERS A, B, H, AND L
                ;
0149 3AF404     CU000:  LDA     MODE    ;GET MODE
014C 47                 MOV     B,A     ;AND SAVE IT
014D 05         CU001:  DCR     B       ;-1 YET?
014E F8                 RM              ;YES, ALL DONE
014F AF                 XRA     A       ;MOVE H TO A WITH CY CLEAR
0150 B4                 ORA     H       ;
0151 1F                 RAR             ;DIVIDE BY 2
0152 67                 MOV     H,A     ;AND RESTORE
0153 AF                 XRA     A       ;DO SAME WITH Y
0154 B5                 ORA     L       ;
0155 1F                 RAR             ;DIVIDE BY 2
0156 6F                 MOV     L,A     ;H, L = X/2, Y/2
0157 C34D01             JMP     CU001   ;DONE YET?

                ;ROUTINE DOT
                ;       DISPLAY THE POINT AT THE CURSOR POSITION
                ;          MAY BE A SINGLE PIXEL (MODE = 0),
                ;          4 PIXELS (MODE = 1), OR 16 PIXELS (MODE = 2).
```

```
                 ;
                 ;BLOCK #1:  TEST MODE, DISPLAY IF NORMAL (0)
                 ;
015A F5          DOT:    PUSH    PSW      ;SAVE REGISTERS
015B E5                  PUSH    H
015C 2AF004              LHLD    YPOS     ;GET CURSOR POSITION
015F 3AF404              LDA     MODE     ;AND MODE
0162 B7                  ORA     A        ;NORMAL MODE?
0163 C27501              JNZ     D200     ;   NO, MODE R128 OR R64
0166 7C                  MOV     A,H      ;SET X
0167 D3E1                OUT     MTRX
0169 7D                  MOV     A,L      ;AND Y
016A 2F                  CMA              ; WHICH NEEDS INVERTING
016B D3E2                OUT     MTRY
016D 3AF304              LDA     COLOR    ;WHATEVER COLOR
0170 D3E0                OUT     MTR0     ;GO DO IT
0172 E1                  POP     H        ;AND RESTORE
0173 F1                  POP     PSW
0174 C9                  RET
                 ;
                 ;BLOCK 2:   128 BY 128 RESOLUTION "DOT"
                 ;
0175 C5          D200:   PUSH    B        ;NEED ANOTHER REGISTER
0176 4F                  MOV     C,A      ;SAVE MODE (= 1)
0177 3AF304              LDA     COLOR    ;DO SAME FOR COLOR
017A 47                  MOV     B,A
017B 0D                  DCR     C        ;MODE 1, 2 OR -1?
017C FAA001              JM      D30X     ; R64X
017F 29                  DAD     H        ;MULTIPLY X AND Y BY 2
0180 C2A701              JNZ     D300     ;  MODE 2
0183 7C                  MOV     A,H      ;DISPLAY THE FOUR POINTS
0184 D3E1                OUT     MTRX     ;1ST X COORD
0186 7D                  MOV     A,L      ;1ST Y COORD
0187 2F                  CMA              ; DON'T FORGET TO INVERT
0188 6F                  MOV     L,A
0189 D3E2                OUT     MTRY     ;
018B 78                  MOV     A,B      ;GET COLOR
018C D3E0                OUT     MTR0     ;1ST POINT
018E 2C                  INR     L        ;BUMP Y BY 1
018F 7D                  MOV     A,L      ;
0190 D3E2                OUT     MTRY     ;2ND Y, SAME X
0192 78                  MOV     A,B      ;
0193 D3E0                OUT     MTR0     ;2ND POINT
0195 24                  INR     H        ;2ND X, SAME Y
0196 7C                  MOV     A,H
0197 D3E1                OUT     MTRX     ;
0199 78                  MOV     A,B      ;
019A D3E0                OUT     MTR0     ;3RD POINT
019C 2D                  DCR     L        ;ORIGINAL Y, 2ND X
019D 7D                  MOV     A,L
019E D3E2                OUT     MTRY
01A0 78                  MOV     A,B      ;
01A1 D3E0                OUT     MTR0     ;LAST POINT
01A3 C1                  POP     B        ;RESTORE REGS
01A4 E1                  POP     H
01A5 F1                  POP     PSW      ;
01A6 C9                  RET              ;  AND RETURN
                 ;
```

```
                    ;BLOCK #3:   64 BY 64 RESOLUTION "DOT"
                    ;
01A7 29             D300:   DAD     H       ;MULTIPLY X AND Y BY 2
01A8 D5             D30X:   PUSH    D       ;NEED ALL THE REGISTERS
01A9 7D                     MOV     A,L     ;INVERT Y
01AA 2F                     CMA
01AB 6F                     MOV     L,A
01AC 0E03                   MVI     C,03H   ;COUNTER FOR X COORD
01AE 7C             D301:   MOV     A,H     ;X BASE ADDRESS
01AF 81                     ADD     C       ;ADD OFFSET
01B0 D3E1                   OUT     MTRX    ;SET X
01B2 1603                   MVI     D,03H   ;Y COUNTER
01B4 7D             D302:   MOV     A,L     ;Y BASE ADDRESS
01B5 82                     ADD     D       ;OFFSET
01B6 D3E2                   OUT     MTRY    ;SET Y COORD
01B8 78                     MOV     A,B     ;USE CURRENT COLOR
01B9 D3E0                   OUT     MTR0    ;PUT UP THE POINT
01BB 15                     DCR     D       ;ADJUST Y COUNTER
01BC F2B401                 JP      D302    ;OFFSET .GE. 0, MORE POINTS
01BF 0D                     DCR     C       ;ADJUST X COUNTER
01C0 F2AE01                 JP      D301    ;AS BEFORE
                    ;
                    ;*****WARNING*****  THIS REGISTER RESTORE ROUTINE IS
                    ;                   ALSO USED BY LINE AND CHAR.
                    ;
01C3 D1             D402:   POP     D
01C4 C1                     POP     B
01C5 E1                     POP     H
01C6 F1                     POP     PSW
01C7 C9                     RET
                    ;
                    ;
                    ;ROUTINE LINE
                    ;       GENERATE THE LINE FROM THE CURRENT CURSOR
                    ;          POSITION TO THE POINT X, Y IN H,L.
                    ;       USES DOT TO ACTUALLY DISPLAY THE POINTS.
                    ;
                    ;BLOCK 1:  PRELIMINARIES
                    ;
                    ;   1.1--SECTOR DETERMINATION
                    ;
01C8 F5             LINE:   PUSH    PSW     ;SAVE THE WORLD
01C9 E5                     PUSH    H       ;   NOTE: ORDER IS SET BY
01CA C5                     PUSH    B       ;   RESTORE IN DOT
01CB D5                     PUSH    D       ;
01CC CD4901                 CALL    CU000   ;COORDINATES NEED CHANGING
01CF 3AF104                 LDA     XPOS    ;GET THE CURRENT CURSOR POSITION

01D2 BC                     CMP     H       ;WHICH IS BIGGER?
01D3 DADC01                 JC      L100    ;XF
01D6 94                     SUB     H       ;NEED A-H
01D7 0600                   MVI     B,00H   ;SET SECTOR CODE TO ZERO
01D9 C3E101                 JMP     L101    ;AND CONTINUE
01DC 2F             L100:   CMA             ;NEED H-A
01DD 3C                     INR     A       ;  WHICH REQUIRES 2'S COMPLIMENT

01DE 84                     ADD     H       ;  AND AN ADD
01DF 0604                   MVI     B,04H   ;SECTOR CODE GETS 4
01E1 57             L101:   MOV     D,A     ;XP GOES IN D
```

```
01E2 3AF004          LDA     YPOS     ;DO THE SAME FOR Y
01E5 BD              CMP     L        ;WHICH IS LARGER
01E6 DAEE01          JC      L102     ;YF IS
01E9 95              SUB     L        ;YC IS
01EA 5F              MOV     E,A      ;SAVE IT
01EB C3F601          JMP     L103     ;AND CONTINUE
01EE 2F      L102:   CMA              ;AGAIN GET 2'S COMPLIMENT
01EF 3C              INR     A        ;
01F0 85              ADD     L        ;  TO FIND YF-YC
01F1 5F              MOV     E,A      ;  AND SAVE IT
01F2 3E02            MVI     A,02H    ;INCR SECTOR CODE BY 2
01F4 80              ADD     B        ;
01F5 47              MOV     B,A      ;NEW SECTOR VALUE
01F6 7A      L103:   MOV     A,D      ;IS XP < YP
01F7 BB              CMP     E        ;IF SO THEY NEED EXCHANGING
01F8 D2FE01          JNC     L104     ;  OK AS THEY ARE
01FB 53              MOV     D,E      ;XP = YP
01FC 5F              MOV     E,A      ;  AND YP = OLD XP
01FD 04              INR     B        ;AND SECTOR CODE GETS ONE MORE
             ;
             ;    1.2--PARAMETER INITIALIZATION
             ;
01FE 2E00    L104:   MVI     L,00H    ;XT = 0
0200 62              MOV     H,D      ;XP
0201 E5              PUSH    H        ;XP, XT
0202 65              MOV     H,L      ;0,0
0203 E5              PUSH    H        ;TA = 0
0204 6B              MOV     L,E      ;H,L = YP
0205 22E504          SHLD    DY       ;DY = +YP
0208 7A              MOV     A,D      ;DETERMINE DX
0209 2F              CMA              ;  WHICH IS 2'S COMPLIMENT
020A 6F              MOV     L,A      ;  OF XP
020B 26FF            MVI     H,0FFH   ;  I.E. DX = -XP
020D 23              INX     H        ;
020E 22E704          SHLD    DX       ;SAVE FOR LOOOP
0211 37              STC              ;T0 = 1/2 DX
0212 7C              MOV     A,H      ;ARITH SHIFT RIGHT
0213 1F              RAR              ;  OF H,L
0214 67              MOV     H,A      ;HIGH  BYTE DONE
0215 7D              MOV     A,L      ;NOW DO LOW BYTE
0216 1F              RAR              ;
0217 6F              MOV     L,A      ;ALL DONE
0218 E5              PUSH    H        ;SAVE T0
             ;
             ;   1.3--SET UP COORDINATE TRANSFORMATION TABLE
             ;
0219 21DD03          LXI     H,MXT    ;CALCULATE CORRECT MOVES
021C 78              MOV     A,B      ;OFFSET INTO TABLE
021D 07              RLC              ;EACH ENTRY IS FOUR BYTES
021E 07              RLC              ;
021F 5F              MOV     E,A      ;ADD TO BASE ADDRESS
0220 AF              XRA     A        ;GET A ZERO
0221 57              MOV     D,A      ;  FOR NOW AND FOR LATER
0222 19              DAD     D        ;H,L IS NOW ADDRESS OF M0X
0223 5E              MOV     E,M      ;GET M0X
0224 23              INX     H        ;AIM AT M0Y
0225 56              MOV     D,M      ;AND GET IT TOO
0226 EB              XCHG             ;SHIFT TO H,L
0227 22E904          SHLD    M0X      ;AND STORE IN MOVE ZERO
```

```
022A EB                 XCHG                ;NOW GET 'ONE' MOVE
022B 23                 INX     H           ;WHICH ARE THE NEXT 2 ENTRIES
022C 5E                 MOV     E,M         ;M1X
022D 23                 INX     H           ;
022E 56                 MOV     D,M         ;M1Y
022F EB                 XCHG                ;GET SET
0230 22EB04             SHLD    M1X         ;  AND STORE
               ;
               ;BLOCK #2:  THE ACTUAL LINE GENERATION LOOP
               ;
               ;    2.1--DISPLAY THE CURRENT POINT
               ;
0233 CD5A01    L200:    CALL    DOT         ;DISPLAY THE CURRENT POINT
               ;
               ;    2.2--TEST FOR DONE
               ;
0236 C1                 POP     B           ;B,C=T0
0237 D1                 POP     D           ;D,E = TA
0238 E1                 POP     H           ;H,L = XP, XT
0239 7D                 MOV     A,L         ;XT
023A BC                 CMP     H           ;XP
023B D2C301             JNC     D402        ;ALL DONE, GO RESTORE
023E 2C                 INR     L           ;XT = XT + 1
023F E5                 PUSH    H           ;SAVE FOR NEXT ITERATION
               ;
               ;    2.3--DETERMINE NEXT MOVE
               ;
0240 2AE504             LHLD    DY          ;GET DY
0243 19                 DAD     D           ;TA = TA + DY
0244 E5                 PUSH    H           ;SAVE FOR NEXT ITERATION
0245 09                 DAD     B           ;TA + T0
0246 DA5002             JC      L240        ;IF POSITIVE
               ;
               ;    2.4--MAKE THE REQUIRED MOVE
               ;
0249 C5        L242:    PUSH    B           ;T0 IS UNCHANGED WITH A MOVE ZER
0
024A 2AE904             LHLD    M0X         ;M0X IN L, M0Y IN H
024D C35802             JMP     L241        ;  GO MOVE
0250 2AE704    L240:    LHLD    DX          ;MOVE ONE INCREMENTS T0
0253 09                 DAD     B           ;T0 = T0 + DX
0254 E5                 PUSH    H           ;SAVE FOR NEXT ITERATION
0255 2AEB04             LHLD    M1X         ;M1X IN L, M1Y IN H
0258 EB        L241:    XCHG                ;MAKE ROOM FOR AN ADDRESS
0259 21F004             LXI     H,YPOS      ;UPDATE Y FIRST
025C 7A                 MOV     A,D         ;M?Y
025D 86                 ADD     M           ;  IS ADDED TO YPOS
025E 77                 MOV     M,A         ;NEW YPOS
025F 23                 INX     H           ;DO THE SAME FOR XPOS
0260 7B                 MOV     A,E         ;
0261 86                 ADD     M           ;
0262 77                 MOV     M,A         ;
0263 C33302             JMP     L200        ;END OF LINE GENERATION LOOP
               ;
               ;ROUTINE CHAR
               ;        GENERATE THE ASCII CHARACTER IN REGISTER A
               ;        CHARACTERS ARE BASED ON A VARIABLE WIDTH
               ;           4 BY 5 DOT MATRIX.
               ;        CURSOR DEFINES THE LOWER LEFT CORNER
```

```
                        ;          OF THE DOT MATRIX.
                        ;       CURSOR IS MOVED TO THE NEXT CHARACTER POSITION.
                        ;       LOWER CASE IS CONVERTED TO UPPER CASE.
                        ;       PARITY IS IGNORED.
                        ;       THE FOLLOWING CONTROL CHARACTERS ARE RECOGNIZED:

                        ;MEMONIC ASCII  HEX      FUNCTION
                        ;MAXR    NUL    00       DISPLAY MODE - 256 BY 256
                        ;MAXC    SOH    01       DISPLAY MODE - 256 BY 256
                        ;R128    STX    02       DISPLAY MODE - 128 BY 128
                        ;R64     ETX    03       DISPLAY MODE - 64 BY 64
                        ;R64X    EOT    04       DISPLAY MODE - 64 BY 64 DOT SIZE

                        ;                          WITH 256 BY 256 RESOLUTION.
                        ;                          NOTE: CHARACTERS ILLEGIBLE
                        ;BS      BS     08       BACKSPACE: XPOS = XPOS - 6
                        ;HT      HT     09       HOR TAB: XPOS = (XPOS+32)MOD 32
                        ;LF      LF     0A       LINE FEED: YPOS = YPOS - 8
                        ;VT      VT     0B       VERT TAB: ((YPOS-32) MOD 32) - 6

                        ;FF      FF     0C       FORM FEED: XPOS = 0, YPOS = 122
                        ;CR      CR     0D       CARRIAGE RETURN: XPOS = 0
                        ;
                        ;SO      SO     0E       PROPORTIONAL CHARACTER SPACING
                        ;SI      SI     0F       FIXED CHARACTER SPACING
                        ;
                        ;DLE            10       COLOR SELECTION FOR SYSTEMS UP T
0
                        ;   TO            TO       FOUR BOARDS WIDE.  SEE LOOKUP
                        ;      US           IF     TABLE "COLORS" FOR DEFINITIONS

                        ;
                        ;
                        ;BLOCK 1:  CHARACTER TYPE DETERMINATION
                        ;
0266 F5                 CHAR:   PUSH    PSW     ;SAVE THE WORLD
0267 E5                         PUSH    H       ;NOTE: ORDER IS SET BY
0268 C5                         PUSH    B       ;   RESTORE IN DOT
0269 D5                         PUSH    D       ;
026A 01C301                     LXI     B,DA02  ;FAKE A CALL FROM THE
026D C5                         PUSH    B       ;  REGISTER RESTORE SEQUENCE
026E E67F                       ANI     7FH     ;CLEAR PARITY BIT
0270 21F104                     LXI     H,XPOS  ;GET X CURSOR ADDRESS
                                                ;  IN CASE CONTROL CHAR
0273 FE20                       CPI     20H     ;COMPARE TO A BLANK
0275 DA0103                     JC      C501    ;   A CONTROL CHAR
0278 FE60                       CPI     60H     ;COMPARE TO ACCENT GRAVE
027A DA7F02                     JC      C100    ;   UPPER CASE
027D E65F                       ANI     5FH     ;CONVERT LOWER CASE TO UPPER
027F 2AF004             C100:   LHLD    YPOS    ;GET CURRENT CURSOR POSITION
0282 EB                         XCHG            ;  BUT IN D,E
                        ;
                        ;BLOCK 2:  CALCULATE THE CHARACTER MATRX ADDRESS
                        ;       A = ASCII CHARACTER     D,E = XPOS, YPOS
                        ;
0283 21FD03                     LXI     H,CHRX  ;BASE ADDRESS OF CHAR TABLE
0286 D620                       SUI     20H     ;ZEROTH ENTRY IN TABLE IS BLANK
0288 4F                         MOV     C,A     ;3 BYTES PER ENTRY
0289 0600                       MVI     B,00H   ;  SO MULTIPLY OFFSET BY 3
```

```
028B 09                 DAD     B       ;ONCE
028C 09                 DAD     B       ;  TWICE
028D 09                 DAD     B       ;    THRICE
028E 3AF204             LDA     CMODE   ;GET SPACING MODE
0291 47                 MOV     B,A     ;  AND SAVE FOR LATER
0292 7E                 MOV     A,M     ;GET BYTE 0 WITH FLAGS
0293 E603               ANI     03H     ;ISOLATE WIDTH FIELD
0295 FE03               CPI     03H     ;
0297 CAD702             JZ      C400    ;YES, GIVE SPECIAL TREATMENT
029A 4A                 MOV     C,D     ;SAVE STARTING XPOS
029B C603               ADI     03H     ;WIDTH OF CHAR + 1
029D A0                 ANA     B       ;WHAT SPACING MODE?
029E C2A302             JNZ     C202    ;  PROPORTIONAL. OK AS IS
02A1 3E06       C201:   MVI     A,06H   ;  FIXED. MAKE IT SIX WIDE
02A3 82         C202:   ADD     D       ;XPOS OF NEXT CHARACTER
02A4 57                 MOV     D,A     ;D,E IS NEXT CHAR POSITION
02A5 D5                 PUSH    D       ;SAVE UNTIL DONE
02A6 51                 MOV     D,C     ;RESTORE CURRENT POSITION
02A7 7E                 MOV     A,M     ;ONE LAST FLAG TO TEST
02A8 07                 RLC             ;IS THIS A DESCENDING CHAR?
02A9 D2AE02             JNC     C300    ;  NO. GO GENERATE IT
02AC 1D                 DCR     E       ;  YES. DOWN TWO ON Y
02AD 1D                 DCR     E       ;
                ;
                ;BLOCK 3:  GENERATE THE ACTUAL CHARACTER
                ;       A = MASK FOR BOTTOM ROW
                ;       D,E = XPOS, YPOS
                ;       H,L = ADDRESS OF FIRST BYTE OF CHAR TABLE ENTRY
                ;
02AE EB         C300:   XCHG            ;GET REGISTERS IN POSITION
02AF CDC502             CALL    C310    ;DO BOTTOM ROW OF CHAR
02B2 CDC302             CALL    C305    ;SECOND ROW
02B5 CDC502             CALL    C310    ;THIRD ROW
02B8 CDC302             CALL    C305    ;FOURTH ROW
02BB CDC502             CALL    C310    ;AND TOP ROW
02BE E1                 POP     H       ;RETRIEVE PRECALCULATED CURSOR
02BF 22F004             SHLD    YPOS    ;AND UPDATE CURSOR
02C2 C9                 RET             ;ALL DONE
02C3 13         C305:   INX     D       ;NEXT BYTE IN TABLE
02C4 1A                 LDAX    D       ;  GOES IN A
02C5 0604       C310:   MVI     B,04H   ;COLUMNS PER ROW
02C7 E5                 PUSH    H       ;SAVE STARTING POSITION
02C8 07         C311:   RLC             ;SHOULD POINT BE ON?
02C9 22F004             SHLD    YPOS    ;UPDATE CURSOR
02CC DC5A01             CC      DOT     ;PUT UP THE POINT IF REQUIRED
02CF 24                 INR     H       ;NEXT X
02D0 05                 DCR     B       ;COUNT DOWN
02D1 C2C802             JNZ     C311    ;MORE TO GO
02D4 E1                 POP     H       ;RESTORE X
02D5 2C                 INR     L       ;UP ONE ON Y
02D6 C9                 RET             ;END OF LOCAL SUBROUTINE
                ;
                ;BLOCK 4:  GENERATE FIRST COLUMN OF 5 WIDE CHARACTERS
                ;       A = 03H         C = CHAR - 32
                ;       D,E = XPOS, YPOS
                ;       H,L = ADDR OF 1ST BYTE OF CHAR TABLE ENTRY
                ;
02D7 7E         C400:   MOV     A,M     ;MOV A,M
02D8 D5                 PUSH    D       ;SAVE STARTING CURSOR
```

```
02D9 E604              ANI     04H       ;AUXILIARY LOOKUP REQUIRED
02DB C2F502            JNZ     C410      ;  YES. GO DO IT
02DE 2F                CMA               ;FIRST COLUMN IS ALL ON ('M' & '
W')
02DF 0605      C411:   MVI     B,05H     ;5 POINTS TO A COLUMN
02E1 07        C401:   RLC               ;SHOULD THE POINT BE ON?
02E2 EB                XCHG              ;GET X,Y IN H,L
02E3 22F804            SHLD    YPOS      ;CURRENT CURSOR POSITION
02E6 DC5A01            CC      DOT       ;DISPLAY AS REQUIRED
02E9 EB                XCHG              ;BACK TO NORMALCY
02EA 1C                INR     E         ;NEXT YPOS
02EB 05                DCR     B         ;TEST FOR DONE
02EC C2E102            JNZ     C401      ;NOT YET
02EF D1                POP     D         ;ORIGINAL CURSOR POSITION
02F0 4A                MOV     C,D       ;SAVE A COPY FOR NORMAL
02F1 0C                INR     C         ;FIX UP TO DO COLUMNS 2-5
02F2 C3A102            JMP     C201      ;AND RETURN TO MAINSTREAM
02F5 E5        C410:   PUSH    H         ;SAVE CHAR TABLE ENTRY
02F6 21BA04            LXI     H,CHRA-3  ;AUXILIARY TABLE ADDR
02F9 0600              MVI     B,00H     ;  FOR CHARS #, $, %, AND &
02FB 09                DAD     B         ;NOTE: C HAS CHAR - 20H
02FC 7E                MOV     A,M       ;GET THE FIRST COLUMN
02FD E1                POP     H         ;AND RESTORE TABLE ENTRY
02FE C3DF02            JMP     C411      ;DISPLAY THE RETRIEVED COLUMN
               ;
               ;BLOCK 5:  CONTROL CHARACTERS
               ;       A = ASCII CONTROL CHARACTER
               ;       H,L = ADDRESS OF X CURSOR (XPOS)
               ;
0301 FE04      C501:   CPI     04H       ;MODE CHANGE?
0303 CA1203            JZ      C507      ;R64X REQUIRES SPECIAL
0306 D21603            JNC     C510      ;  NO
0309 3D                DCR     A         ;MODE TYPE
030A F20E03            JP      C506      ;  IS NOW CORRECT
030D 3C                INR     A         ;  OPPS... TOO FAR
030E 32FA04    C506:   STA     MODE      ;RECORD NEW MODE
0311 C9                RET
0312 2F        C507:   CMA               ;MAKE MODE NEGATIVE
0313 C30E03            JMP     C506      ;  AND USE IT
0316 D608      C510:   SUI     08H       ;NORMAL CONTROL CHAR?
0318 D8                RC                ;TOO LOW, FORGET IT
0319 C22103            JNZ     C511      ;WHAT IS IT?
031C 7E                MOV     A,M       ;  BACKSPACE
031D D606              SUI     06H       ;XPOS = XPOS - 6
031F 77                MOV     M,A       ;
0320 C9                RET               ;
0321 3D        C511:   DCR     A         ;HORIZONTAL TAB?
0322 C22C03            JNZ     C512      ;  NO
0325 7E                MOV     A,M       ;  YES
0326 C620              ADI     20H       ;XPOS + 32
0328 E6E0              ANI     0E0H      ;  MODULO 32
032A 77                MOV     M,A       ;  IS NEW XPOS
032B C9                RET               ;
032C 3D        C512:   DCR     A         ;LINE FEED?
032D C23603            JNZ     C513      ;  NO
0330 2B                DCX     H         ;  YES
0331 7E                MOV     A,M       ;YPOS
0332 D608              SUI     08H       ;  - 8
0334 77                MOV     M,A       ;  IS NEW YPOS
```

```
0335 C9                 RET                 ;
0336 3D         C513:   DCR     A           ;VERTICAL TAB?
0337 C24203             JNZ     C514        ;  NO
033A 2B                 DCX     H           ;  YES
033B 7E                 MOV     A,M         ;YPOS
033C E6E0               ANI     0E0H        ;  MODULO 32
033E D606               SUI     06H         ;  MINUS 6
0340 77                 MOV     M,A         ;  IS NEW YPOS
0341 C9                 RET                 ;
0342 3D         C514:   DCR     A           ;FORM FEED?
0343 C25403             JNZ     C515        ;  NO
0346 21FF00             LXI     H,00FFH     ;UPPER LEFT CORNER
0349 CD4901             CALL    CU000       ;  IN INTERNAL COORD
034C 7D                 MOV     A,L         ;MOVE DOWN ONE PRINT LINE
034D D606               SUI     06H
034F 6F                 MOV     L,A         ;FIRST POSITION ON TOP
0350 22F004             SHLD    YPOS        ;  LINE IS NEW CURSOR
0353 C9                 RET
0354 3D         C515:   DCR     A           ;CARRIAGE RETURN?
0355 C25B03             JNZ     C516        ;  NO
0358 3600               MVI     M,00H       ;  YES
035A C9                 RET                 ;XPOS = 0
035B 21F204     C516:   LXI     H,CMODE     ;GET MODE AND COLOR BYTE ADDRESS
ES
035E 3D                 DCR     A           ;PROPORTIONAL SPACING REQUEST?
035F C26503             JNZ     C517        ;  NO
0362 360F               MVI     M,0FH       ;  YES, SET FLAG
0364 C9                 RET                 ;
0365 3D         C517:   DCR     A           ;FIXED SPACING REQUEST?
0366 C26C03             JNZ     C518        ;  NO, MUST BE COLOR SELECT
0369 3600               MVI     M,00H       ;  YES, SET FLAG
036B C9                 RET                 ;
036C 3D         C518:   DCR     A           ;A HAS COLOR DESIRED
036D 23                 INX     H           ;REAIM AT COLOR BYTE
036E EB                 XCHG                ;SAVE IN D
036F 21C104             LXI     H,COLORS    ;COLORS LOOKUP TABLE
0372 4F                 MOV     C,A         ;OFFSET
0373 0600               MVI     B,00H
0375 09                 DAD     B           ;DESIRED COLOR
0376 7E                 MOV     A,M         ; GET IT
0377 12                 STAX    D           ;  INTO COLOR BYTE
0378 C9                 RET

                ;
                ;ROUTINE ANIMAT
                ;       ANIMATION IS NOT SUPPORTED
                ;         ANIM = 0          DISPLAY WARNING
                ;         ANIM =-1          WAIT FOR VERTICAL BLANKING
                ;
0379 F5         ANIMAT: PUSH    PSW         ;SAVE SOME REGS
037A 3AEF04             LDA     ANIM        ;CHECK FLAG
037D 3C                 INR     A           ;IS IT -1?
037E CACD03             JZ      A200        ;  YES, WAIT FOR V/B
0381 E5                 PUSH    H           ;1ST TIME,   GIVE WARNING
0382 2AF004             LHLD    YPOS
0385 E5                 PUSH    H           ;SAVE CURSOR
0386 2AF304             LHLD    COLOR       ;  AND COLOR
0389 E5                 PUSH    H
038A 260F               MVI     H,0FH       ;GET SOME BACKGROUND
```

```
038C 7C         A101:   MOV     A,H       ;SET Y COORD
038D D3E2               OUT     MTRY
038F 2EFE               MVI     L,0FEH    ;INIT X COUNTER
0391 7D         A102:   MOV     A,L       ;SET X COORD
0392 D3E1               OUT     MTRX      ;
0394 3E01               MVI     A,01H     ;SET WRITE
0396 D3E0               OUT     MTR0      ;
0398 2D                 DCR     L         ;ADJUST X COUNTER
0399 C29103             JNZ     A102      ;MORE TO GO?
039C 25                 DCR     H         ;ADJUST Y COUNTER
039D C28C03             JNZ     A101      ;MORE TOGO
03A0 217917             LXI     H,1779H ;CENTER (IF PROPORTIONAL)
03A3 22F004             SHLD    YPOS
03A6 21D104             LXI     H,ANIMER  ;ANIMATION ERROR MESSAGE
03A9 7E         A103:   MOV     A,M       ;GET A CHAR
03AA FE24               CPI     '$'       ;LAST LETTER?
03AC CAB603             JZ      A104      ;YES.  DONE
03AF CD6602             CALL CHAR         ;DISPLAY IT
03B2 23                 INX     H         ;NEXT CHAR
03B3 C3A903             JMP     A103
03B6 3EFF       A104:   MVI     A,0FFH    ;SET ANIMATION FLAG
03B8 32EF04             STA     ANIM
03BB CD7903     A105:   CALL    ANIMAT    ;AND WAIT 256 FRAMES
03BE 3D                 DCR     A         ;  TO GIVE TIME TO READ
03BF C2BB03             JNZ     A105      ;  THE BAD NEWS.
03C2 E1                 POP     H         ;RESTORE DISPLAY SETTINGS
03C3 22F304             SHLD    COLOR     ;COLOR & RESOLUTION
03C6 E1                 POP     H         ;CURSOR POSITION
03C7 22F004             SHLD    YPOS      ;
03CA E1                 POP     H         ;RESTORE REGS
03CB F1                 POP     PSW
03CC C9                 RET
                ;
                ;BLOCK #2:  DELAY FOR VERTICAL BLANKING TO START
                ;
03CD DBE0       A200:   IN      MTR0      ;VER BLANK IN PROGRESS?
03CF E602               ANI     02H
03D1 C2CD03             JNZ     A200      ;YES.  WAIT TILL DONE
03D4 DBE0       A201:   IN      MTR0      ;THEN WAIT FOR VERT
03D6 E602               ANI     02H       ;  BLANKING TO START
03D8 CAD403             JZ      A201      ;
03DB F1                 POP     PSW       ;RESTORE ACCUM
03DC C9                 RET
                ;
                ;               END OF EXECUTABLE PROGRAM CODE
                ;
                ;               ******************************
                ;
                ;                       LOOKUP TABLES
                ;
                ;
                ;MOVE TABLE FOR THE LINE GENERATOR
                ;
03DD FF00FFFF   MXT:    DB      0FFH,000H,0FFH,0FFH       ;SECTOR 5
03E1 00FFFFFF           DB      000H,0FFH,0FFH,0FFH       ;SECTOR 6
03E5 FF00FF01           DB      0FFH,000H,0FFH,001H       ;SECTOR 4
03E9 0001FF01           DB      000H,001H,0FFH,001H       ;SECTOR 3
03ED 010001FF           DB      001H,000H,001H,0FFH       ;SECTOR 8
03F1 00FF01FF           DB      000H,0FFH,001H,0FFH       ;SECTOR 7
```

```
03F5 01000101           DB      001H,000H,001H,001H      ;SECTOR 1
03F9 00010101           DB      000H,001H,001H,001H      ;SECTOR 2
                ;
                ;
                ;CHARACTER MATRIX TABLE
                ;       EACH ENTRY IS 3 BYTES
                ;       BIT >>    7   6   5   4   3   2   1   0
                ;BYTE   0         U2  @   R   S   T   W5  W2  W1
                ;BYTE   1         M   N   O   P   I   J   K   L
                ;BYTE   2         E   F   G   H   A   B   C   D
                ;
                ;       A  B  C  D            FLAGS
                ;       E  F  G  H      U2:  DESCENDERS, MOVE DOWN 2
                ;       I  J  K  L      W2, W1:  WIDTH OF CHARACTER - 2
                ;       M  N  O  P      W5:  FOR FIVE WIDE FIGURES...
                ;       @  R  S  T          0 - FIRST COLUMN ALL ONES
                ;                           1 - FIRST COLUMN FROM CHRA
                ;       REPRESENTS 2ND THRU 5TH COLUMNS
                ;          OF FIVE COLUMN WIDE CHARACTERS
                ;
                CHRX:
D03FD 0200004008DB      02H,00H,00H,40H,08H,88H,01H,00H,0AAH      ;  ! "
 0406 57FAFA775EDB      57H,0FAH,0FAH,77H,5EH,4EH,1FH,0B4H,0A9H   ;# $ %
 040F 6F25CC0000DB      6FH,25H,0CCH,00H,00H,88H,20H,88H,84H      ;& ' (
 0418 40444801A4DB      40H,44H,48H,01H,0A4H,0A0H,01H,4EH,40H     ;) * +
 0421 808800010EDB      80H,88H,00H,01H,0EH,00H,40H,00H,00H       ;, - .
 042A 42421032DBDB      42H,42H,10H,32H,0DBH,96H,71H,44H,4CH      ;/ 0 1
 0433 7A42967216DB      7AH,42H,96H,72H,16H,1EH,0AH,1FH,99H       ;2 3 4
 043C 721E0F329EDB      72H,1EH,0FH,32H,9EH,86H,42H,42H,1FH       ;5 6 7
 0445 3296967217DB      32H,96H,96H,72H,17H,96H,40H,08H,00H       ;8 9 :
 044E 8088081148DB      80H,88H,08H,11H,48H,42H,01H,0E0H,0E0H     ;; < =
 0457 4142482202DB      41H,42H,48H,22H,02H,96H,62H,0BBH,9FH      ;> ? @
 0460 4AF996729EDB      4AH,0F9H,96H,72H,9EH,9EH,32H,98H,96H      ;A B C
 0469 72999E7A8EDB      72H,99H,9EH,7AH,8EH,8FH,42H,8EH,0FH       ;D E F
 0472 329B864A9FDB      32H,9BH,86H,4AH,9FH,99H,71H,44H,4EH       ;G H I
 047B 3291114AACDB      32H,91H,11H,4AH,0ACH,0A9H,7AH,88H,88H     ;J K L
 0484 0B55B14A9BDB      0BH,55H,0B1H,4AH,9BH,0D9H,32H,99H,96H     ;M N O
 048D 428E9E3AB9DB      42H,8EH,9EH,3AH,0B9H,96H,4AH,0AEH,9EH     ;P @ R
 0496 7216871222DB      72H,16H,87H,12H,22H,2FH,32H,99H,99H       ;S T U
 049F 3269990BB5DB      32H,69H,99H,0BH,0B5H,51H,4AH,66H,99H      ;V W X
 04A8 2226997A84DB      22H,26H,99H,7AH,84H,2FH,60H,88H,8CH       ;Y Z [
 04B1 0212486044DB      02H,12H,48H,60H,44H,4CH,02H,00H,96H       ;\ ] ↑
 04BA F80000    DB      0F8H,00H,00H                              ;←

                ;AUXILIARY LOOKUP TABLE
                ;       FIRST COLUMN OF #, $, %, AND &
                ;
04BD 50109860   CHRA:   DB      50H,10H,98H,60H ;# $ % &

                ;
                ;COLOR LOOKUP TABLE
                ;
                ;       BOARDS  MNEMONIC ASCII  COLOR     ! ALT-256
                COLORS:
04C1 00             DB  00H ;   BLK     DLE     BLACK     BLACK
04C2 8B             DB  8BH ;   RED     DC1     RED       WHITE
04C3 93             DB  93H ;   BLU     DC2     BLUE      WHITE
04C4 9B             DB  9BH ;   MAG     DC3     MAGENTA   WHITE
04C5 A5             DB  0A5H ;  GRN     DC4     GREEN     WHITE
```

```
04C6 AD             DB   0ADH ;   YEL       NAK      YELLOW   WHITE
04C7 B5             DB   0B5H ;   CYN       SYN      CYAN     WHITE
04C8 BF             DB   0BFH ;   WHI       ETB      WHITE    WHITE
04C9 45             DB    45H ;   N  A      CAN      N  A     WHITE
04CA 4D             DB    4DH ;   O  S      EM       O  S     WHITE
04CB 55             DB    55H ;   N  S      SUB      N  S     WHITE
04CC 5D             DB    5DH ;   E  I      ESC      E  I     WHITE
04CD 63             DB    63H ;      G      FS          G     WHITE
04CE 6B             DB    6BH ;      N      GS          N     WHITE
04CF 73             DB    73H ;      E      RS          E     WHITE
04D0 7F             DB    7FH ;      D      US          D     WHITE
                 ;
                 ;
                 ;ANIMATION ERROR MESSAGE
                 ;
04D1 02104C494DANIMER: DB 02H,10H,'LIMITED ANIMATIONS'
                 ;;
                 ;                    END OF ROMABLE SEGMENT OF PROGRAM
                 ;
                 ;                  **************************************
                 ;
                 ;                  START OF RAM (VARIABLE) STORAGE AREA
                 ;
                 ;SCRATCH PAD STORAGE FOR THE LINE GENERATOR
                 ;        THESE LOCATIONS MAY BE ALTERED AT ANY TIME A
                 ;           LINE IS NOT ACTUALLY BEING GENERATED.
                 ;        VARIABLES MUST BE IN THE ORDER GIVEN.
                 ;
04E5             DY:      DS      2        ;+YP
04E7             DX:      DS      2        ;- XP
04E9             M0X:     DS      1        ;X INCR FOR A ZERO MOVE
04EA             M0Y:     DS      1        ;Y INCR FOR A ZERO MOVE
04EB             M1X:     DS      1        ;X INCR FOR A ONE MOVE
04EC             M1Y:     DS      1        ;Y INCR FOR A ONE MOVE
                 ;
                 ;
                 ;GLOBAL STORAGE AREA FOR THE GRAPHICS PACKAGE
                 ;        THESE LOCATIONS MUST BE PRESERVED BETWEEN
                 ;           CALLS TO THE GRAPHICS ROUTINES.
                 ;        THEY ARE INITIALIZED BY INITG.
                 ;        VARIABLES MUST BE IN THE ORDER GIVEN.
                 ;
04ED             FIRST:   DS      2        ;REFRESH BUFFER ADDRESS
04EF             ANIM:    DS      1        ;ANIMATION WARNING GIVEN FLAG
04F0             YPOS:    DS      1        ;Y CURSOR VALUE
04F1             XPOS:    DS      1        ;X CURSOR VALUE
04F2             CMODE:   DS      1        ;CHARACTER SPACING MODE
04F3             COLOR:   DS      1        ;CURRENT COLOR BYTE
04F4             MODE:    DS      1        ;DISPLAY MODE

04F5                      END
```

matrox electronic systems
P.O. BOX 56, AHUNTSIC STN., MONTREAL, QUE. H3L 3N5
TEL.: (514) 481-6838 TELEX 05 825651

MTX-ALPHA SOFTWARE

ALT-2480 SOFTWARE

User's Manual

ALT-2480 SOFTWARE PACKAGE

INTRODUCTION

The ALT-2480 Software Package provides the user the full flexibility of a software driven video display with the implementation ease of a stand-alone terminal. The Software Package has been designed explicitly to support easy and reliable modification to meet varying user requirements. Wherever possible, parameters and definitions are not tested until run time to permit maximum flexibility without requiring user written code modifications. Although the input routines are set up to run using 'skip' I/O, the display routines (OUTCHR and ECHOCH) are explicitly written to be useable at interrupt level.

As supplied, the package will fully emulate the popular Lear Siegler, inc. ADM-3A and Digital Equipment Corp. DECSCOPE VT-52 interactive display terminals. In addition, line at a time and text block input modes are available to provide the powerful text preparation features of an intelligent terminal.

USER'S GUIDE

This section explains the keyboard functions available under the ALT-2480 Software Package. All key codes are interpreted by software, so the ASCII code(s) associated with any function(s) can be changed as desired (see Software Interfacing Guide). The input key codes for ATTN, XON, XOFF, and block mode ESC may also be changed dynamically under keyboard or program control.

Input Modes

There are three basic input modes that can be used depending on the degree of input processing desired. The least sophisticated mode is the full duplex (FDX) mode. In this mode no processing is performed on input. For a typed in character to appear on the display, it must be echoed by the user program. (If the ALT-2480 Software Package is being used as part of a system monitor, that monitor is considered the user program.) Characters are passed on to the user program as soon as they are input, exactly as they are input. The only exceptions are the input control codes SETC (^B), ATTN (^C), XOFF (^S), and XON (^Q) used to set configuration switches, return to monitor level, stop output, and resume output respectively.

Half duplex mode buffers characters as they are input until a full line is typed. A full line consists of either 80 characters or 0 through 79 characters followed by a CR, ESC, or LF. All characters are echoed as they are input. Carriage return echoes as CR-LF and both CR and LF are passed to the user program. Rubout will delete the last character in the buffer (and on the screen) while ^U will cancel the entire line. Once a full line has been entered, no further input will be accepted until the entire line has been read by the user program and the first character on the next line requested. Control characters other than SETC, ATTN, XOFF, XON, RO, Line Cancel, CR, LF, and HT are echoed as ^<char> and have no other effect on the display. They will be passed to the user program when requested exactly as typed, not as ^<char>. *

In this release, RO and ^U may not update the display correctly if tabs are erased or the input line exceeds one display line. Regardless of what appears on the display, RO and ^U always have the correct effect on the input line buffer.

The third input mode is block mode. In this mode, the user can generate an entire block of data using all the editing capabilities of the intelligent terminal system. By inputing the XMIT (End of Text) code (^D), all data on the screen entered since the last XMIT code is sent to the user program. This can be particularly effective in such applications as filling in the entries on a computer generated form. When in block mode, no control characters are passed on to the user program except the implied carriage returns at the end of each line of data, horizontal tabs to indicate a field of protected data, and the EOT to mark the end of the transmission.

* EX.: When Typed
A ^H B CR

When Echoed

B CR LF

^H = Backspace
CR = Carriage Return
LF = Line Feed

Keyboard Commands

Except as noted in Appendix I, all commands can also be executed by the user program through calls to OUTCHR.

The notation ∧<char> is used to indicate the ASCII code generated by holding down the control key while the <char> key is depressed. Some control characters such as ESC (∧[) may require using both the control and shift keys. Many keyboards include separate keys for some of the frequently typed control codes. For example, virtually every keyboard has a CR (or Return) key, which generates the same code as control M. Appendix I is a list of all the commands, their assigned control characters, and equivalent letter codes. In the definitions which follow, only the letter code is given to avoid confusion.

Cursor Controls. The following commands move the cursor about the screen. To retain compatibility with the LSI ADM-3A, vertical tab and form feed require preceding ESC characters. All cursor controls are nondestructive (i.e., they do not affect any of the data on the display).

Backspace (∧H). Each time a backspace is executed, the cursor moves one position to the left. Cursor action when the cursor is already in the leftmost column is determined by OFFLFT.

Horizontal Tab (∧I). Each time a horizontal tab is executed, the cursor moves right to the next tab stop. Tab stops are set at every eighth column. Cursor action when the next tab stop is beyond the right end of the line is determined by OFFRT.

Linefeed (^J). Each time a line feed is executed the cursor moves down to the same position on the next line. If the cursor is already on the bottom line, either the cursor will wrap around to the top line or the entire display will scroll up one line (losing the contents of the top line) as determined by OFFBOT.

Vertical Tab (^[, ^K). Each time a vertical tab is executed the cursor moves down to the next vertical tab stop. These stops are set every eight lines. If the next tab stop is off the bottom of the display, cursor action is determined by OFFBOT. This is a two character command because the VT character is used for the upline command.

Upline (^K). Each time an upline is executed the cursor moves to the same character position in the line immediately above the current one. If the cursor is already on the top line, display action is determined by OFFTOP.

Forespace (^L). Each time a forespace is executed the cursor moves to the next character position. If the cursor is on the last position on a line, the next character position is determined by the OFFRT switch.

Return (^M). This code moves the cursor to the first character position of the present line. When input from the keyboard in half duplex or block mode, a line feed is automatically appended and executed.

Home (^^). The cursor is moved to the upper left display position; line 1, column 1.

Load Cursor (^[, '=', <Y>, <X>). The next two characters following the

∧[, '=' sequence represent the absolute line and column (Y and X) coordinates which are used to position the cursor. The upper left cursor position is line 1, column 1. The characters required are calculated by adding 31 (decimal) to the desired line (or column) number. The Home Command is equivalent to the Load Cursor sequence ∧[, '=', SP, SP.

Editing Commands

The following commands are used to manipulate data on the screen. They may be output by the user program at any time. However, they are executable from the keyboard only when in block input mode. The half duplex input mode editing command Rubout is described in the Input Mode section.

Form Feed (∧[, ∧L). The form feed command sequence clears the screen and moves the cursor to the first position on the top line. This is a two character command because the FF character is used for the Forespace Command.

Clear Screen (∧Z). This deletes all data on the screen. The cursor position is not changed.

Line Insert (∧W). The line containing the cursor and all following lines move down one line. The bottom line on the screen is lost.

Line Delete (∧U). The line containing the cursor is deleted. All lines below the cursor are moved up one line and a blank line is moved into the bottom line. In half duplex modes, the entire line buffer is deleted.

Char Insert (∧V). The character indicated by the cursor and succeeding characters on the same line are shifted right one character. The cursor position is set to a blank. This function will not operate if the last position on the line contains data.

Char Delete (∧X). The character indicated by the cursor is removed. Characters to the right of the cursor on the same line are moved one position to the left.

Insert Mode (∧[, 'I'). This command simplifies insertion of long strings of data. Logically precedes each succeeding character with an insert character command. The Insert Mode is terminated by any control character, which is otherwise ignored.

EX : ∧[I FFFF = ∧VF∧VF∧VF∧VF

Transmit Block (∧D). Transmit all screen data from the last transmit command (from line 1, column 1 if not on screen) up to the current cursor position, to the user program. Trailing blanks on each line (unless explicitly entered by the user) are ignored. Individual lines are separated by carriage return line feed sequences. End of Text character (∧D) is appended to the end of the transmission to signify end of text block. Protected fields are replaced by Horizontal Tabs (∧I). No other control characters are transmitted. *

Set XMIT Start (∧[, ∧D). Changes the cursor position associated with the last Transmit Block command to the current cursor position. This allows the user to select a command from a menu or repeatedly input the same string (as long as it stays on the screen).

*EX:AAAA■■■■ BBBB is transmitted as AAAA∧I∧I∧I∧I BBBB
■ Background display = protected field ∧I is only one character long.

Special Commands

Four special command codes are implemented to maximize system utility. The first one, SETC, is used to change the terminal configuration switches. The other three are normally system monitor functions and can be deleted if the monitor (if any) in use provides the same function.

These switches are always detected and acted upon while the software is in use. Any keyboard input other than these commands while output is being processed is ignored.

Set Configuration (^B, <CMND>, <PARM>). The SETC command allows the user or program complete control over the terminal configuration. Each configuration switch change requires a complete three character sequence. The ^B causes the following two characters to be interpreted as the switch to change and the desired value. The switch must be an upper or lower case letter, the value can be any character other than a control character. When executed from the keyboard, the user will be prompted on the top line. Any response other than 'Y' to the question mark will cause the request to be aborted. The 'Y' should not be included when setting configuration switches from the user program.

The key codes for different setting values are given in Appendix II. Switches which can be set are:

<A> Use the ALT-2480 display at the address indicated.

<B> Set OFFBOT to determine whether to wrap around to the top line or scroll the screen up when the cursor is moved below the bottom line.

<C> Select cursor character. The character input becomes the new cursor character.

<D> Display lower case as lower case (normal).

<E> Select escape character for block mode keyboard input.

<F> Select XOFF character.

<G> Display lower case characters using the greek symbol set.

<H> Display lower case as upper case. (This switch should always be used with the 2480-C option.)

<I> Select ATTN character.

<J> Set display line length. If length is forty or less, the display generated will be compatible with the ALT-2480 low resolution option.

<K> Reserved.

<L> Set OFFLFT to determine whether to back up to the previous line, wrap around on the same line, or remain in the first position on the line when the cursor is moved past the left edge of the display.

<M> Select input mode.

<N> Select XON character.

<O>, <P>, <Q> Reserved

<R> Set OFFRT to determine whether to start a new line, wrap around on the same line, or remain in the last position when the cursor is moved past the last position on a line.

<S> Set or clear "TTY lock." The TTY lock shifts all lower case characters to upper case on input. It does not affect program output.

<T> Set OFFTOP to determine whether to wrap around to the bottom line or scroll the screen down when the cursor is moved above the top line.

<U> through <Z> Unused.

Attention (^C). This command returns control to a user specified address. Normally this would be the monitor restart or breakpoint trap address.

Stop Output (^S). This command stops all output processing until a Resume output command is given. This allows the user to stop the program long enough to read the output and then resume processing. Only special commands may be entered while this command is in effect.

Resume Output (^Q). This nullifies the output freeze caused by a stop output command.

Additional Commands

Six additional commands are provided for additional flexibility.

Auto Answer Back (^E). In response to the ENQ command from the program, the software will respond with a short HERE IS message. This can be convenient for identifying specific versions of the 2480 software which have been specially modified for a given application.

Bell Subroutine (^G). Since the ALT-2480 does not provide an acoustic warning tone, a special routine is provided. This routine can either be modified to ring a user provided bell interface or left as is to flash the screen once.

Keyboard Lock and Unlock (∧O and ∧N). These command codes disable and enable keyboard input respectively. An attempt to input from the keyboard while it is locked will trap to the monitor entry point specified by MONLVL.

Select Foreground Display (∧[, ∧_). Display all following characters in normal video.

Select Background Display (∧[, ∧Y). Display all following characters in inverse video (and/or blink as strapped in hardware). Note that fields in background mode are not input in block mode but are replaced by horizontal tabs.

Software Interfacing Guide

There are only three primary entry points in the MATROX 2480 Software Package. There is one routine call to output to the display, one to read the next available keyboard input, and one to see if any keyboard input is available. The same three routine calls are used regardless of the input mode in use or the style of output desired. A fourth entry point is also provided to allow independent, noninterfering output. Local storage for this routine is totally independent of that used for program output facilitating adaptation of the package to interrupt driven keyboard input.

All four routines obey the following register conventions: 1) All registers except the PSW are preserved. 2) Values are returned in register A with the flags set to match. 3) Output routines expect the argument to be in register C. In accordance with convention 2, this argument is returned in register A as well.

Primary Entry Points

OUTCHR

The character in register C is displayed at the current cursor position and the cursor is advanced to the next character position. Characters with numerical values less than 32 (blank) are assumed to be control characters (parity is ignored). The action taken for any particular control character is determined by the lookup table at the address in CONAT. If a control character is not in the referenced table, it is displayed as ^<char>. Lower case characters may be optionally shifted to their upper case or greek equivalents. The parity bit will be set or cleared to match the current display mode (background or foreground respectively).

INCHRW

The next available input character is returned in register A. If no input data is available (e.g. a line terminator has not yet been typed in half duplex input mode), the cursor character is flashed at the current cursor location to prompt the user. Only one character is returned with each call to this routine. However, once an input line or block is terminated, there is no delay in subsequent calls as long as buffered data is available.

There is no requirement that all available data be input before processing any output, but be careful with the block mode, as any input data shifted off screen before being input will be irrecoverably lost. Half duplex line storage is limited to 80 bytes. If this limit is reached before a line terminator is entered, the entire line buffer will be released to the user program and no further keyboard data entry will be accepted until the entire buffer has been read by the user program. Care must also be exercised when changing input modes to avoid undesired loss of buffered input data.

When using block mode input, keep in mind that no distinction is made between displayed program output and displayed keyboard input. This distinction can be maintained by using foreground mode for keyboard input and background mode for program output.

TSTIN

This routine allows the user program to check if any data is available for input. If calling INCHRW would result in a delay (i.e. a character, line, or block is not available), this routine returns with register A set to zero. If data is currently available, register A is set to FF(hex). Flags are set to match the contents of register A.

ECHOCH

This routine is similar to OUTCHR but is modified to simplify keyboard echo, specially in interrupt driven systems. Its functioning is identical to OUTCHR with the following exceptions.

1) Output is independent of the XOFF command.
2) The control table referenced by ECONAT is used.
3) Parity is not ignored. Characters with the parity bit zero are treated the same as in OUTCHR. However, all characters with parity bit set are considered control characters and searched for in the control table.
4) Escape and other multiple character sequences are maintained independent of any in progress in OUTCHR.

Required User Supplied Subroutines

To interface with the user supplied keyboard, this package requires two user defined routines. These routines may use any registers desired, the only requirement is that they return their value in register A.

INKBS

This routine must return the status of the user keyboard. Register A should be zero if a character is not available. Any other value implies a character is available immediately by calling INKBD. For compatibility with potential user programs, there should not be any response time requirement between a positive response to INKBS and the subsequent call to INKBD. INKBS is called at address STFDX + 3.

INKBD

This routine should return in register A the ASCII character input by the user. It is called only after a positive response to INKBS is received. (Note that more than one positive response to INKBS may be required before a call to INKBD depending on the user program.) The parity bit may be set or clear as desired. It is ignored by the package but is provided to the user in full duplex and half duplex input modes. INKBD is called at address INFDD + 1.

Lookup Tables and Variables

Most of the power and flexibility of this package are due to the extensive use of run time interpretation of critical parameters and control character definitions along with strict segregation of program code (ROMable) and program data (RAM only). By appropriate use of the SETC command, a single copy of this package can independently control multiple MTX -2480 displays.

The use and allowable values of all variables are documented in the source listing. Some of the more powerful or unusual ones are:

CPTRS

To maintain the identity of specific points on the display as characters and lines are added or deleted and as scrolling occurs, these character pointers are updated by all routines which move data about the screen. Each pointer requires two bytes. The low byte is the column and the high byte is the line. The total number of pointers is determined by the compilation switch CPNUM, currently set to three. The first two pointers are used in block input mode to keep track of which character to transmit next and when to stop transmitting. The third pointer is available for other uses.

CURSAT

This pointer is the current cursor position. It is tested before displaying any data to verify that it is on screen. Action taken when off screen is determined by the variables OFFBOT, OFFLFT, OFFRT, and OFFTOP which are interpreted by the routine TSTCUR. Note that TSTCUR modifies only the cursor data supplied in registers H,L and if necessary, the display. It does not modify the contents of CURSAT.

MTXAT

This word contains the base address to use in all references to the MATROX display memory. It can also be used to provide a left margin by increasing the address by the desired value and decreasing the line width accordingly.

BLKEND

This byte defines both the terminate block (XMIT) character and the second character of the set block start command.

MONLVL

This defines the address to call if the ATTN character is detected on input. It is also called if an attempt is made to input a character while the keyboard is locked. Both conditions are ignored if the address is zero.

ECONAT

This word defines the control character lookup table used by the routine ECHOCH. It must contain the address of a valid control character lookup table. CONAT performs the same function for OUTCHR. In this package ECONAT and CONAT are the same. If wanted a new lookup table can be created for routine ECHOCH.

INTRAP

This word is tested before each attempt to get a character from the keyboard. If it is not zero, the address contained is jumped to. A RET instruction will return the value in register A as if it had been input from the keyboard. A JMP to INFDK will proceed with normal acquisition of keyboard input. Useful to control input data, and output data or commands from the user program when in block of half-duplex mode.

MULJMP

This word is tested by OUTCHR after registers B and C have been set up but before any processing is begun. If not zero, the contents are considered the address of a routine and called. If output has been inhibited by an XOFF command, it will not be tested until output is permitted to resume. The routine called should return with the CY flag clear if output processing of the contents of register C is desired. CY flag set squelches further processing. Only the contents of register B must be preserved. IMULJM performs the same function for ECHOCH. The same use as INTRAP but in output controlling.

CONTAB

This is an ECHOCH and OUTCHR control character definition table. The table is built of three byte entries consisting of the value of the control character and the address of the routine to execute it. By convention, a character with the parity bit set is equivalent to the same character preceded by the escape character. If a match is found, the associated routine is called with register B positive if from OUTCHR, negative if from ECHOCH (guaranteed not to change sign if incremented less than 100 times). Register C contains the character matched and registers H and L contain the line and column of the current cursor position respectively. The routine called may use any registers desired, including register B.

Table entries may be for any eight bit value. However, the table is only searched for characters from 0 through 31 and 128 through 255. The entries may be in any order with the exception of the null control character. The last entry in the table must be zero in order to terminate the search. The table is linearly searched and only the first occurrence of a character is detected. This is utilized to redefine the carriage return in block input mode without duplicating the entire table.

APPENDIX I

Control Codes

Code		ASCII	Function	Input FDX	Input HDX	Input BLK	Output
^@	00	NUL					
^A	01	SOH					
^B	02	STX	SETC: Set Configuration Switches	x	x	x	x
^C	03	ETX	ATTN: Return to Monitor	x	x	x	
^D	04	EOT	Transmit Block			x	
^E	05	ENQ	Auto Answer Back				x
^F	06	ACK					
^G	07	BEL	Bell Subroutine			x	x
^H	08	BS	Backspace			x	x
^I	09	HT	Horizontal Tab		x	x	x
^J	0A	LF	Line Feed		x	x	x
^K	0B	VT	Upline			x	x
^L	0C	FF	Forespace			x	x
^M	0D	CR	Carriage Return		x	x	x
^N	0E	SO	Unlock Keyboard			x	x
^O	0F	SI	Lock Keyboard			x	x
^P	10	DLE					
^Q	11	DC1	XON - Resume Output	x	x	x	
^R	12	DC2	Reserved				
^S	13	DC3	XOFF - Stop Output	x	x	x	
^T	14	DC4					
^U	15	NAK	Delete Line		x	x	x
^V	16	SYN	Insert Character			x	x
^W	17	ETB	Insert Line			x	x
^X	18	CAN	Delete Character			x	x
^Y	19	EM	Reserved				
^Z	1A	SUB	Clear Screen			x	x
^	1B	ESC	Escape Char for Multi Char Commands			x	x
^	1C	FS					
^	1D	GS					
^^	1E	RS	Home			x	x
^_	1F	US					
Rubout	7F	DEL	Delete Character		x		

Escape Character Sequences

Code	Function	Input FDX	Input HDX	Input BLK	Output
ESC '=' <X> <Y>	Direct Cursor Addressing			x	x
ESC 'I'	Insert Mode			x	x
ESC FF	Form Feed			x	x
ESC VT	Vertical Tab			x	x
ESC EOT	Set Start of XMIT Block			x	
ESC US	Select Foreground Display			x	x
ESC EM	Select Background Display			x	x

APPENDIX II

Configuration Switches

^B, <CMND>, <PARM>

ND>	Function	<PARM>	Set to
A	Set 2480 Base Address	0	0000H
		1	1000H
		9	9000H
		:	A000H
		;	B000H
		<	C000H
		=	D000H
		>	E000H
		?	F000H
B	Set OFFBOT Switch	1	Wrap around to top
		H	Scroll up
C	Select Cursor Char	<char>	Cursor becomes the char
D	Display Lower Case as Lower	0	
E	Select Escape Char	<char>	Escape becomes the char
F	Select XOFF Char	<char>	XOFF becomes the char
G	Display LC as Greek	0	
H	Display Lower Case as Upper	0	
I	Select ATTN Char	<char>	ATTN becomes the char
J	Set Display Line Length	X	40 wide
		x	72 wide
		SP	80 wide
K	Reserved		
L	Set OFFLFT Switch	SP	Back up to previous line
		1	Overwrite first char on line
		0	Wrap around to end of line
M	Set Input Mode	0	Half duplex
		1	Full duplex
		2	Block mode
N	Select XON Char	<char>	XON becomes the char
O P Q	Reserved		
R	Set OFFRT Switch	SP	First char of next line
		1	Wrap around on same line
		0	Overwrite last char
S	Set TTY Upper Case Lock	1	On
		0	Off (normal)
T	Set OFFTOP Switch	1	Scroll down
		H	Wrap around to bottom
U through Z	Undefined		

APPENDIX III

The Demonstration Program 1

To permit evaluation of this software package, a simple demonstration program is included as part of the package. To run the demonstration, load the object paper tape using a standard Intel format hex loader. The program loads starting at address 0100 hex and requires less than 3K bytes of memory. Once loaded, manually patch the address of your INKBS routine into the JMP at location 0103 hex, the address of your INKBD routine into the JMP at location 0106 hex (see Software Interfacing Guide for the definitions of the INKBD and INKBS routines), the address of a routine to read your current console device (value returned in register A) into the JMP at location 0109 hex and the address of a routine to output the character in register C or A on your console device into the JMP at location 010C hex. The console I/O routines are not required if only the first phase of the demonstration is executed. If desired, the location MONLVL (address 0A5A hex) may be patched to the breakpoint or restart address of your monitor.

Display software parameters are initially set to the following values. They may be modified as desired using the Set Configuration Switch command.

-ALT-2480 addressed at E000 hex.

-Line length is 40 characters (low resolution).

-Input mode is full duplex.

-Input upper case shift lock is off.

-Output displays lower case as upper case.

-Cursor character is inverse video underline.

-OFFBOT set to scroll display.

-OFFTOP set to wrap around to bottom line.

-OFFRT set to start a new line.

-OFFLFT set to overwrite the first character on the line.

-Control characters are defined to correspond with the User's Guide.

To run the demonstration, start execution at location 0100 hex. If your ALT-2480 is addressed in the memory block starting at E000 hex, a flashing cursor will appear in the upper left corner of the display. If your ALT-2480 is not addressed at E000 hex, type in the command sequence ^B, A, n, Y where n is the character 0 through 9 or :, ;, <, =, or ?. See Appendix II for the correct value to use. This command sequence will reset the software to use the ALT-2480 at the specified address.

The first phase of the demonstration is a simple loop where a character is read by INCHRW and displayed by OUTCHR. The input mode is initially set to full duplex so that characters are displayed by OUTCHR exactly as typed in. By changing to half duplex input mode, (type ^B, M, Ø, Y) it is possible to see the line at a time editing ability of the package. When a line is terminated by either CR, LF, or ESC, the entire line is redisplayed by OUTCHR. (If the line is terminated by an ESC, the first character provided by the next input from INCHRW will be processed by OUTCHR as the second character of an escape sequence, so use care). Similarly, the full editing power of the display may be tried by selecting block mode (type ^B, M, 2,Y). When using block mode for the very first time, clear the screen first. This will initialize the line fill table and character pointers from the "random" contents left from the loading process. When changing from half duplex or block mode, type the terminator character immediately after executing the mode change to exit the input buffer fill code and permit the mode change to take effect.

The second phase of the demonstration program is an independent test of OUTCHR. This phase is entered by typing the control character FS (^ \) during phase one. Note that this will not change test phases if input mode is block mode, nor will the phase change in half duplex mode until the line is terminated and the FS character is received by the demonstration program. Phase two accepts characters from the console and displays them using OUTCHR. The routines INCHRW and INSTS are not involved. This permits extensive evaluation of display output characteristics without interference from input restrictions. This phase is exited by typing a US (^ _) on the console.

The final phase of the test program is an independent test of the various INCHRW modes. Characters are output to the console exactly as they would have been received by a user program. Typing an RS (^^) will return the demonstration program to the initial phase.

Application Notes

This package is provided to permit the user to experiment with various system capabilities and differing applications with a minimum of programming effort. While this section discusses various applications using the MTX2480 software package, it is important to keep in mind that this software package is not production level software. Efficiency, size, and speed of execution are all deliberately sacrificed to provide a wide range of capabilities and maximum flexibility.

A sophisticated intelligent terminal can be assembled from a minimum number of parts; display, keyboard, CPU, serial port, ROM, and a little RAM. The MTX2480 software package demonstrates many of the capabilities found in the popular Hazeltine 2000 intelligent terminal. In this case, however, many terminal characteristics can be modified by simple keyboard commands. Even production models could be radically modified simply by changing the ROM program,greatly simplifying last minute specification changes or custom variations.

In mini and micro computer based systems, the display can be integrated directly into the system, eliminating the need for extra I/O ports and utilizing idle processor time and memory. This also permits highly interactive, real time control of the display, which is often not practical over typical communication lines.

When used as the operator's console, system parameters can be displayed and updated by the operating system in real time with a minimum of overhead. For example, to display a status message on a PDP-11, an eight word routine is sufficient:

```
      MOV   R1, #MESSAGE          ;Address of message text
      MOV   R2, #DISPLAY          ;Address of display area to use
      MOV   R3, #LENGTH           ;Length of message
LOOP: MOVB  (R2)+, (R1)+          ;Transfer the message
      SOB   R3, LOOP              ;Repeat until done
```

On a Z-80, only ten bytes are required!

```
LD      BC, LENGTH              ;Message length
LD      DE, DISPLAY             ;Display area desired
LD      HL, MESSAGE             ;Message desired
LDIR                            ;Display it.
```

Integrated into a small business system, the display can significantly enhance thruput and accuracy, especially with unsophisticated users. For example, order forms can be filled in by displaying the appropriate blank form and guiding the user through the required entries one step at a time using the line at a time input mode provided by MTX2480. Entries can be checked by the applications software for validity and consistency at the time of entry, allowing immediate interactive correction.

Considering the display can do anything a CRT terminal can do, only far faster, the possibilities are limitless. Except for operations requiring mass data movement (e.g. scrolling), even the MTX2480 software package can display several thousand characters a second. In general, the primary limitation on display update speed is the time required to generate or retrieve the data. This capability to read or write any data on the display almost instantly makes practical applications not even contemplated with conventional terminals.

Demonstration Program 2

This program sets the page mode, clears the screening, sets the cursor at home, and line length in 80 characters. The INTRAP location points to the address of a routine that test the column numbers. If it is 75, the bell is outputted using the OUTCHR routine.

When a block is terminated, it is outputted on the screem without blanks.

ADDRESS	CONTENT		MNEMONIC	COMMENT
ØCØØ	Ø6	BEGIN	MOVI B,Ø8	/Load B with number of
ØCØ1	Ø8			/Codes
ØCØ2	21		LXI, H,L	/Load H,L with first
ØCØ3	ØØ			/Address of codes
ØCØ4	ØD			
ØCØ5	4E	LOOP	MOV C,M	/Get one code and
ØCØ6	CD		CALL	/Output it
ØCØ7	ØC		OUTCHR	
ØCØ8	Ø2			
ØCØ9	23		INX H	/Point to next code
ØCØA	Ø5		DCR B	/Decrement counter and
ØCØB	C2		JNZ LOOP	/Test for all done
ØCØC	Ø5			
ØCØD	ØC			
ØCØE	CD	HERE	CALL	/Get a block or
ØCØF	6B		INCHRW	/Buffer character if block
ØC1Ø	Ø2			/Terminated
ØC11	FE		CPI ' '	/If it is a blank
ØC12	2Ø			/Get next character
ØC13	CA		JZ HERE	/
ØC14	ØE			
ØC15	ØC			
ØC16	CD		CALL	/If not output it
ØC17	ØC		OUTCHR	
ØC18	Ø2			
ØC19	C3		JMP HERE	/Get next character or
ØC1A	ØE			/Block
ØC1B	ØC			

The bell call is made with the following routine. Manually patch the address of this routine (ØC4Ø) in the INTRAP location (ØAB5).

ØC4Ø	2A	LHLD	/Load H,L with
ØC41	32	CURSAT	/Cursor position
ØC42	ØA		
ØC43	7D	MOV A,L	
ØC44	FE	CPI 75D	/Test if it is
ØC45	4B		/Equal to 75
ØC46	C2	JNZ INFOK	/If not, get the
ØC47	86		/Next character
ØC48	Ø5		

ADDRESS	CONTENT	MNEMONIC	COMMENT
ØC49	ØE	MOVI CØ7H	/If yes
ØC4A	Ø7		
ØC4B	CD	CALL	/Output the
ØC4C	ØC	OUTCHR	/Bell command
ØC4D	Ø2		
ØC4E	C3	JMP INFOK	/Then get the
ØC4F	86		/Next character
ØC5Ø	Ø5		

```
02AF C9                 RET


                ;***********************************************
                ;
                ;       OUTPUT SUBROUTINES
                ;
                ;
                ;THESE ROUTINES ALL MUST PRESERVE REGISTERS B AND C
                ;
                ;***********************************************

                ;ROUTINE CONTRL (C=CHAR, B=LEVEL)
                ;       CONTROL CHARACTER PROCESSING ROUTINE.
                ;       SCANS TABLE CONTAB OR ECONTAB AS DETERMINED BY
                ;          LEVEL FOR THE CHARACTER.
                ;       IF A MATCH IS FOUND, THE INDICATED ROUTINE IS
                ;          CALLED WITH B=LEVEL, C=CHAR, H=LINE AND
                ;          L=COLUMN (OF CURRENT CURSOR POSITION).
                ;       CALLED ROUTINES MAY UTILIZE ANY REGISTERS.
                ;       CONTROL CHARACTER TABLES (ADDRESS IN CONAT OR EC
                ;          MUST CONCLUDE WITH THE NULL CHARACTER (OOHEX)
                ;       REGISTERS A, D, E, FLAGS, H AND L MODIFIED.
                ;       CY FLAG IS SET IF TNE CHARACTER IS NOT FOUND.

02B0 C5         CONTRL: PUSH    B       ;SAVE VITALS
02B1 2AC70A             LHLD    CONAT   ;ASSUME OUTPUT MODE
02B4 04                 INR     B       ;  IS IT?
02B5 F2BB02             JP      CNTRO   ;     IT IS
02B8 2AB30A             LHLD    ECONAT  ;GET ECHO CONTROL TABLE
02BB 7E         CNTRO:  MOV     A,M     ;GET TABLE ENTRY
02BC 23                 INX     H       ;ON TO ADDRESS
02BD B9                 CMP     C       ;DESIRED CHARACTER?
02BE CACA02             JZ      CNTR1   ;  YES.  DO IT
02C1 23                 INX     H       ;STEP TO NEXT ENTRY
02C2 23                 INX     H
02C3 B7                 ORA     A       ;BUT CHECK FOR END OF TABLE
02C4 C2BB02             JNZ     CNTRO   ;  BEFORE CONTINUING
02C7 37                 STC             ;FLAG AS FAILURE TO FIND
02C8 C1                 POP     B       ;RESTORE VITAE
02C9 C9                 RET

                    ;EXECUTE THE DESIRED CONTROL FUNCTION

02CA 5E         CNTR1:  MOV     E,M     ;LOW BYTE OF ADDRESS
02CB 23                 INX     H
02CC 56                 MOV     D,M     ;AND HI BYTE
02CD 21D602             LXI     H,CNTRB ;FAKE A CALL
02D0 E5                 PUSH    H
02D1 D5                 PUSH    D       ;CALL ADDRESS
02D2 2A320A             LHLD    CURSAT  ;CURSOR POSITION
02D5 C9                 RET             ;WOULD YOU BELIEVE 'CALL'?
02D6 C1         CNTRB:  POP     B       ;RESTORE VITALS
02D7 AF                 XRA     A       ;CLEAR CARRY
```

```
02D8 C9                  RET               ;  AND RETURN SUCCESSFULLY

               ;SUBROUTINE TSTCUR (H=LINE, L=COLUMN)
               ;       ADJUST H,L TO THE NEAREST ON SCREEN POINT
               ;       H AND L ARE TREATED AS SIGNED 8 BIT INTEGERS
               ;       ACTION TAKEN ON OFF SCREEN POINTS IS DETERMINED
               ;        BY THE SWITCHES OFFLFT, OFFRT, OFFTOP AND
               ;        OFFBOT.
               ;       IF ORIGINAL POINT IS ON SCREEN IT IS NOT MODIFIE
               ;       A, D, E, FLAGS AND HL MODIFIED.

02D9 AF        TSTCUR: XRA     A       ;TEST FOR OFF LEFT FIRST
02DA B5                ORA     L
02DB F2EB02            JP      TST10   ;OK SO FAR, TEST RIGHT SIDE
02DE 3AC50A            LDA     OFFLFT  ;OFF THE LEFT, WHAT TO DO?
02E1 3D                DCR     A
02E2 F2E702            JP      TST05   ;USE A, LINE # IS OK
02E5 25                DCR     H       ;UP ONE LINE
02E6 2F                CMA             ;  AND CORRECT COL NUMBER
02E7 6F        TST05:  MOV     L,A     ;SET NEW COLUMN
02E8 C3FD02            JMP     TST20   ;AND TEST LINE #

02EB 3ABC0A    TST10:  LDA     WIDTH   ;TEST FOR OFF RIGHT SIDE
02EE 3D                DCR     A       ;MAX LEGAL IS WIDTH-1
02EF BD                CMP     L       ;STILL ON?
02F0 D2FD02            JNC     TST20   ;YES.  CHECK LINE
02F3 3AC60A            LDA     OFFRT   ;OFF THE RIGHT, NOW WHAT?
02F6 3D                DCR     A
02F7 F2FC02            JP      TST15   ;ADJUST COL ONLY
02FA 24                INR     H       ;DOWN ONE LINE
02FB AF                XRA     A       ;  AND 1ST COLUMN
02FC 6F        TST15:  MOV     L,A     ;SET CORRECT COLUMN

                  ;COLUMN IS NOW OK.  CHECK THE LINE.

02FD AF        TST20:  XRA     A       ;TEST FOR OFF TOP
02FE B4                ORA     H
02FF F20D03            JP      TST30   ;TOP OK, CHECK BOTTOM
0302 3AC30A            LDA     OFFTOP  ;OFF TOP.  NOW WHAT?
0305 3D                DCR     A
0306 67                MOV     H,A     ;NEW LINE NUMBER
0307 E5                PUSH    H       ;SAVE CURSOR
0308 CC8F03            CZ      SCRLDN  ;SCROLL IF REQUIRED
030B E1                POP     H       ;RETRIEVE CURSOR
030C C9                RET             ;ALL DONE

030D FE18      TST30:  CPI     24D     ;TEST FOR OFF BOTTOM
030F D8                RC              ;A-OK.
0310 3AC40A            LDA     OFFBOT  ;DOWN TOO FAR, SO FIX
0313 3D                DCR     A
0314 67                MOV     H,A     ;NEW LINE NUMBER
0315 E5                PUSH    H       ;SAVE CURSOR
0316 C44303            CNZ     SCRLUP  ;SCROLLING AS REQUIRED
0319 E1                POP     H
031A C9                RET

               ;SUBROUTINE PUTUP (C=CHAR, H=LINE, L=COLUMN)
```

```
        Z

                     ;        MATROX 2480 SUBROUTINE PACKAGE

                     ;        VERSION 2.05    JAN 21, 1978

                     ;              COPYRIGHT (C) 1978
                     ;              DR VINCENT C JONES
                     ;              258 NORTH MAGNOLIA
                     ;              SATELLITE BCH, FLA
                     ;                    32937


                     ;
                     ;        COMPILATION SWITCHES
                     ;
0000                          FALSE   EQU     0
FFFF                          TRUE    EQU     NOT FALSE
0000                          SALONE  EQU     FALSE   ;STAND ALONE VERSION
FFFF                          DEMO    EQU     TRUE    ;COMPILE AS DEMONSTRATIO
0003                          CPNUM   EQU     3       ;CURSOR POINTERS
0050                          LINSIZ  EQU     80D     ;LINE BUFFER SIZE

                                  IF DEMO

                         ;STAND ALONE DEMONSTRATION PROGRAM  1
0100                          ORG     100H    ;WORK UNDER CP/M
0100 C30F01                   JMP     BEGIN
0103 C312F0          INKBS:   JMP     CSTS    ;KEYBOARD STATUS
0106 C303F0          INKBD:   JMP     CI      ;KEYBOARD DATA
0109 C303F0          CILOC:   JMP     CI      ;READ CONSOLE
010C C309F0          COLOC:   JMP     CO      ;WRITE CONSOLE
010F 310002          BEGIN:   LXI     SP,STACK
                         ;SELF CONTAINED TEST
0112 CD6B02          LOOP0:   CALL    INCHRW  ;GET A CHAR
0115 4F                       MOV     C,A
0116 CD0C02                   CALL    OUTCHR  ;DISPLAY IT
0119 FE1C                     CPI     FS      ;TIME FOR NEXT TEST?
011B C21201                   JNZ     LOOP0   ;NOT YET
                         ;OUTPUT TEST
011E CD0901          LOOP:    CALL    CILOC
0121 4F                       MOV     C,A
0122 CD0C02                   CALL    OUTCHR
0125 FE1F                     CPI     US      ;SHIFT TIME?
0127 C21E01                   JNZ     LOOP
                         ;INPUT TEST
012A CD6B02          LOOP2:   CALL    INCHRW
012D F5                       PUSH    PSW
012E 4F                       MOV     C,A
012F CD0C01                   CALL    COLOC
0132 F1                       POP     PSW
0133 FE1E                     CPI     RS
0135 C22A01                   JNZ     LOOP2
0138 C31E01                   JMP     LOOP

0200                          ORG     200H

F003                          CI      EQU     0F003H  ;DEFINE FOR ZAPPLE
```

```
F009                    CO      EQU     0F009H
F012                    CSTS    EQU     0F012H

0200                    STACK   EQU     $

                           ENDIF

                ;
                ;       ****    TOP LEVEL ROUTINES      ****
                ;
                ;EXCEPT AS NOTED ALL REGISTERS ARE PRESERVED.
                ;
                ;
                ;ROUTINE OUTCHR (C=CHAR)
                ;     DISPLAY THE ASCII CHARACTER IN C AT THE
                ;CURRENT CURSOR POSITION AND ADVANCE THE CURSOR
                ;TO THE NEXT CHARACTER POSITION.
                ;
                ;     CHARACTERS WITH NUMERICAL VALUES LESS THAN
                ;32 (SPACE) ARE ASSUMED TO BE CONTROL CHARACTERS.
                ;
                ;
                ;ROUTINE INCHRW
                ;     RETURNS THE NEXT AVAILABLE INPUT CHARACTER
                ;IN REGISTER A (FLAGS ARE SET TO MATCH).
                ;IF NO CHARACTER IS AVAILABLE, THIS ROUTNE WILL WAIT
                ;UNTIL ONE IS.
                ;     THIS ROUTINE IS USED FOR ALL INPUT MODES.
                ;IF IN A BUFFERED MODE (HALF DUPLEX OR BLOCK)
                ;NO CHARACTERS WILL BE RETURNED UNTIL A COMPLETED
                ;BUFFER IS AVAILABLE.  ONCE THE BUFFER IS RELEASED
                ;BY THE KEYBOARD, EACH SUCCESSIVE CALL TO INCHRW
                ;WILL RETURN THE NEXT CHARACTER IN THE BUFFER.
                ;
                ;
                ;ROUTINE INSTS
                ;     RETURNS THE ACCUMULATOR SET TO TRUE (FF HEX)
                ;IF A CHARACTER IS AVAILABLE FOR INPUT FROM INCHR.
                ;OTHERWISE, A IS CLEARED TO FALSE (00 HEX).
                ;     FLAGS ARE SET TO MATCH.
                ;
                ;
                ;ROUTINE ECHOCH (C=CHAR)
                ;     SAME AS OUTCHR EXCEPT THAT MULTIPLE CHARACTER SEQUE
                ;ARE MAINTAINED INDEPENDENTLY TO ALLOW NONCONFLICTING
                ;ECHOING CONCURRENTLY WITH PROGRAM OUTPUT.
                ;


                ;**** INTERRUPT LEVEL (ECHO) ENTRY POINT

0200 E5         ECHOCH: PUSH    H       ;SAVE THE WORLD
0201 D5                 PUSH    D
0202 C5                 PUSH    B
0203 F5                 PUSH    PSW
0204 0680               MVI     B,80H   ;SET INTERRUPT LEVEL FLAG
```

```
0206 2AC10A           LHLD    IMULJM   ;CHECK FOR MULTICHAR
0209 C31F02           JMP     OUTCO    ;REST OF PROCESSING IS
                                       ;  COMMON WITH OUTCHR

               ;***** NORMAL ENTRY POINT FOR PROGRAM OUTPUT

020C E5       OUTCHR: PUSH    H        ;SAVE THE WORLD
020D D5               PUSH    D
020E C5               PUSH    B
020F F5               PUSH    PSW
0210 79               MOV     A,C      ;CLEAR PARITY
0211 E67F             ANI     7FH
0213 4F               MOV     C,A
0214 0600             MVI     B,00H    ;SET NORMAL OUTPUT FLAG
0216 CD3A08   OUTCE:  CALL    XOFFED   ;OUTPUT THROTTLED?
0219 C21602           JNZ     OUTCE    ;YES,  KEEP TRYING
021C 2ABF0A           LHLD    MULJMP   ;MULTI CHARACTER SEQUENCE?
021F 7C       OUTCO:  MOV     A,H      ;CHECK IF ZERO
0220 B5               ORA     L
0221 CA2C02           JZ      OUTC2    ;NOTHING TO IT
0224 112902           LXI     D,OUTC1  ;FAKE A CALL TO IT
0227 D5               PUSH    D        ;RETURN ADDRESS
0228 E9               PCHL             ;'CALL' ROUTINE
0229 DA3902   OUTC1:  JC      OUTC9    ;CY SET MEANS ALL DONE
022C 79       OUTC2:  MOV     A,C      ;PROCESS THE CHARACTER
022D FE20             CPI     ' '      ;CONTROL CHARACTER?
022F DA3E02           JC      OUTCC    ;  YES
0232 B7               ORA     A        ;PARITY BIT SET?
0233 FA3E02           JM      OUTCC    ;  YES,  TREAT AS CONTROL
0236 CD5D02           CALL    OUTCX    ;NORMAL PRINTING CHAR,
                                       ;  DISPLAY IT
0239 F1       OUTC9:  POP     PSW      ;ALL DONE, RESTORE
023A C1               POP     B        ;  REGISTERS
023B D1               POP     D        ;  AND RETURN
023C E1               POP     H
023D C9               RET

               ; CONTROL CHARACTER PROCESSING

023E CDB002   OUTCC:  CALL    CONTRL   ;SEE IF LEGITIMATE CONTROL
0241 D23902           JNC     OUTC9    ;YES, ALL DONE
0244 79               MOV     A,C      ;UNRECOGNIZED CONTROL CHAR
0245 FEA0             CPI     ' ' OR 80H ;PRINTING CHAR?
0247 D25702           JNC     OUTC3    ;  YES, PRINT IT
024A C640             ADI     '@'      ;SHIFT TO UC ALPHA
024C 4F               MOV     C,A
024D C5               PUSH    B        ;  AND SAVE FOR LATER
024E E680             ANI     PARON    ;SAVE FOR/BACK BIT
0250 C65E             ADI     '^'      ;AND INDICATE CONTROL CHAR
0252 4F               MOV     C,A
0253 CD5D02           CALL    OUTCX    ;  BY LEADING UP-ARROW
0256 C1               POP     B        ;GET THE CHARACTER BACK
0257 CD5D02   OUTC3:  CALL    OUTCX    ;DISPLAY CHARACTER
025A C33902           JMP     OUTC9    ;  AND RETURN

              ;  INTERNAL SUBROUTINE OUTCX (C = CHAR)
              ;     ADJUST CURSOR TO LIE ON THE SCREEN.
```

```
                ;       DISPLAY THE CHARACTER AT THE ADJUSTED
                ;               CURSOR POSITION.
                ;       MOVE THE CURSOR TO THE NEXT COLUMN
                ;               (MAY BE OFF SCREEN).

025D 2A320A     OUTCX:  LHLD    CURSAT  ;GET CURRENT CURSOR
0260 CDD902             CALL    TSTCUR  ;CHECK AND ADJUST
0263 CD1B03             CALL    PUTUP   ;DISPLAY IT
0266 2C                 INR     L       ;NEXT COLUMN
0267 22320A             SHLD    CURSAT  ;SAVE NEW CURSOR
026A C9                 RET

                ;
                ;****** NORMAL ENTRY POINT FOR PROGRAM INPUT
                ;

026B E5         INCHR:  PUSH    H       ;SAVE THE WORLD
026C D5                 PUSH    D
026D C5                 PUSH    B
026E 3A540A             LDA     FDUX    ;WHAT INPUT MODE?
0271 3C                 INR     A
0272 CA8402             JZ      INCHF   ;  FULL DUPLEX.
0275 F27E02             JP      INCHH   ;  HALF DUPLEX
0278 CDA806             CALL    INBLK   ;BLOCK MODE
027B C38A02             JMP     INCHX   ; AND RETURN
027E CDF005     INCHH:  CALL    INHDX   ;GET A LINE BUFFERED CHAR
0281 C38A02             JMP     INCHX
0284 CD7D05     INCHF:  CALL    INFDX   ;GET NEXT KEYSTROKE
0287 DA8402             JC      INCHF   ;CY SET SO TRY AGAIN
028A B7         INCHX:  ORA     A       ;SET FLAGS
028B C1                 POP     B
028C D1                 POP     D
028D E1                 POP     H
028E C9                 RET

                ;
                ;****** NORMAL ENTRY POINT FOR INPUT STATUS
                ;

028F E5         INSTS:  PUSH    H       ;SAVE THE WORLD
0290 D5                 PUSH    D
0291 C5                 PUSH    B
0292 3A540A             LDA     FDUX    ;WHAT INPUT MODE?
0295 3C                 INR     A
0296 CAA802             JZ      STCHF   ;FULL DUPLEX.
0299 F2A202             JP      STCHH   ;  HALF DUPLEX
029C CDB707             CALL    STBLK   ;BLOCK MODE
029F C3AB02             JMP     STCHX   ; AND RETURN
02A2 CDB107     STCHH:  CALL    STHDX   ;GET A LINE BUFFERED CHAR
02A5 C3AB02             JMP     STCHX
02A8 CDA507     STCHF:  CALL    STFDX   ;GET NEXT KEYSTROKE
02AB B7         STCHX:  ORA     A       ;SET FLAGS
02AC C1                 POP     B
02AD D1                 POP     D
02AE E1                 POP     H
```

```
                ;       BARE MINIMUM ALT-2480 DRIVER
                ;       VERSION 1.00  <>  OCT 20,1977
                ;
                ;           COPYRIGHT (C) 1977
                ;           DR VINCENT C. JONES
                ;           11017 BENNINGTON AVE
                ;           KANSAS CITY MO 64134
                ;
                ;     BARE MINIMUM DRIVER ROUTINE FOR MATROX ALT-2480
                ;   DISPLAY.  EMULATES A SIMPLE SCROLLING TERMINAL.
                ;   THE ONLY CONTROL CHARACTERS RECOGNIZED ARE LINE
                ;   FEED AND CARRIAGE RETURN.
                ;     THE USING PROGRAM MUST DEFINE THE BASE ADDRESS
                ;   OF THE 2480 IN USE (MTXAD) AND A ONE BYTE LOCATION
                ;   IN RAM (CURSOR).
                ;     CHARACTER TO BE DISPLAYED MUST BE IN REGISTER C.
                ;   A AND FLAGS ARE MODIFIED.
                ;
                ;
0000 =                  FALSE   EQU     0
FFFF =                  TRUE    EQU     NOT FALSE
FFFF =                  DEMO    EQU     TRUE    ;SUBROUTINE OR DEMO?
FFFF =                  W40     EQU     TRUE    ;40 WIDE OR 80?
000D =                  CR      EQU     0DH     ;DEFINE CARRIAGE RET
000A =                  LF      EQU     0AH     ;  AND LINE FEED
                ;
                ;
                                IF DEMO

                ;       DEMONSTRATION DRIVER

0100                            ORG 100H

0100 310002     BEGIN:  LXI     SP,STACK
0103 DB00       KBIN:   IN      0       ;WAIT FOR INPUT
0105 E601               ANI     1
0107 CA0301             JZ      KBIN
010A DB01               IN      1       ;GET THE CHARACTER
010C E67F               ANI     7FH     ;DELETE PARITY
010E 4F                 MOV     C,A     ;MOVE INTO POSITION
010F CD1501             CALL    MTXOUT  ;AND ECHO IT
0112 C30301             JMP     KBIN    ;AND KEEP ON DOING IT

0200 =                  STACK   EQU     200H    ;DEFINE SOME STACK
0200 =                  CURSOR  EQU     200H    ;AND A BYTE OF RAM

E000 =                  MTXAD   EQU     0E000H  ;MATROX IS HERE

                ;       END OF DEMONSTRATION DRIVER

                                ENDIF
                ;
                ;       START OF ACTUAL 2480 ROUTINE
                ;
0115 79         MTXOUT: MOV     A,C     ;CHECK IF CONTROL
0116 FE0D               CPI     CR
0118 CA3E01             JZ      OUTCR   ;CARRIAGE RETURN
```

```
011B FE0A              CPI     LF
011D CA4301            JZ      OUTLF    ;LINEFEED
0120 3A0002            LDA     CURSOR   ;DISPLAY AT NEXT LOCATION
                       IF W40           ;40 WIDE
0123 FE28              CPI     40D
                       ENDIF
                       IF NOT W40
                       CPI     80D      ;80 WIDE
                       ENDIF
0125 DA2C01            JC      OUT20    ;OK AS IS
0128 CD4301            CALL    OUTLF    ;SCROLL
012B AF                XRA     A        ;CARRIAGE RETURN
012C E5        OUT20:  PUSH    H        ;SAVE WORK REGS
012D D5                PUSH    D
012E 6F                MOV     L,A      ;DESIRED COLUMN
012F 3C                INR     A        ;ADVANCE CURSOR
0130 320002            STA     CURSOR   ;  FOR NEXT TIME
0133 2600              MVI     H,0      ;CALCULATE ADDRESS
0135 1180EB            LXI     D,MTXAD+128D*23D ;1ST ON LAST LINE
                       IF W40
0138 29                DAD     H        ;40 WIDE IS EVEN ONLY
                       ENDIF
0139 19                DAD     D        ;ADDRESS OF CHAR
013A 71                MOV     M,C      ;DISPLAY IT
013B D1                POP     D
013C E1                POP     H
013D C9                RET
               ;
               ;LOCAL ROUTINES FOR MTXOUT
               ;
013E AF        OUTCR:  XRA     A        ;BEGINNING OF LINE
013F 320002            STA     CURSOR
0142 C9                RET
               ;
0143 E5        OUTLF:  PUSH    H        ;SCROLL UP ONE LINE
0144 D5                PUSH    D
0145 C5                PUSH    B
0146 2180E0            LXI     H,MTXAD+128D ;SOURCE
0149 1100E0            LXI     D,MTXAD ;DESTINATION
014C 01500B            LXI     B,22D*128D+80D ;BYTE COUNT
014F 7E        OUT80:  MOV     A,M      ;Z-80 LDIR
0150 12                STAX    D
0151 23                INX     H
0152 13                INX     D
0153 0B                DCX     B
0154 78                MOV     A,B
0155 B1                ORA     C
0156 C24F01            JNZ     OUT80
0159 012050            LXI     B,80D*256D+' ' ;ZAP LAST LINE
015C 71        OUT85:  MOV     M,C
015D 2B                DCX     H
015E 05                DCR     B
015F C25C01            JNZ     OUT85
0162 C1                POP     B
0163 D1                POP     D
0164 E1                POP     H
0165 C9                RET
```

```
                ;       DISPLAY THE CHARACTER IN C AT THE SCREEN
                ;          POSITION INDICATED.
                ;       ADJUST THE CHARACTER TO CORRESPOND WITH
                ;          OUTPUT SWITCHES GREEK AND FORBAK.
                ;       H AND L MUST CONTAIN A VALID, ON SCREEN POINT.
                ;       REGISERS A, D, E AND FLAGS MODIFIED

031B E5         PUTUP:  PUSH    H       ;SAVE H,L FOR LATER
031C 5C                 MOV     E,H
031D 1600               MVI     D,0
031F 7D                 MOV     A,L     ;COLUMN WITH NEW CONTENTS
0320 21340A             LXI     H,LINFIL
0323 19                 DAD     D       ;ADDRESS OF PREVIOUS MAX
0324 BE                 CMP     M       ;HAVE MORE NOW?
0325 DA2B03             JC      PUTU0   ;  NO
0328 3C                 INR     A       ;FILL IS COL + 1
0329 77                 MOV     M,A     ;NEW MAXIMUM
032A 3D                 DCR     A       ;BACK TO COLUMN
032B 63         PUTU0:  MOV     H,E     ;RETRIEVE POSITION
032C 6F                 MOV     L,A
032D CDFA03             CALL    MTXAD   ;CALCULATE ADDRESS
0330 EB                 XCHG            ;PUT IN D,E
0331 21BD0A             LXI     H,GREEK ;NOTE:  GREEK AND FORBAK
0334 79                 MOV     A,C     ;  MUST BE CONSECUTIVE
0335 E660               ANI     60H     ;LOWER CASE?
0337 FE60               CPI     60H
0339 79                 MOV     A,C     ;GET FRESH COPY
033A C23E03             JNZ     PUTU1   ;NOT LC SO OK AS IS
033D A6                 ANA     M       ;CONVERT TO GREEK OR UC AS REQ
033E 23         PUTU1:  INX     H       ;SAME FOR FORGROUND/BACK
033F B6                 ORA     M       ;SET INVERT/BLINK AS REQ
0340 12                 STAX    D       ;DISPLAY IT
0341 E1                 POP     H       ;RESTORE HL
0342 C9                 RET

                ;SUBROUTINE DELINE (H=LINE)
                ;       SCROLL THE LINE INDICATED AND ALL LINES BENEATH
                ;          IT UP ONE LINE.
                ;       THE LINE INDICATED BY H IS LOST.
                ;       H MUST CONTAIN A VALID LINE NUMBER BETWEEN
                ;          0 AND 23 INCLUSIVE (NOT CHECKED).
                ;       THE TOP LINE IS LOST.
                ;       CP*L AND LINFIL ARE UPDATED AS REQUIRED.
                ;       A, D, E, H, L AND FLAGS MODIFIED.


0343 2600       SCRLUP: MVI     H,0     ;DO THE WHOLE SCREEN
0345 C5         DELINE: PUSH    B       ;SAVE THE SACRED
0346 2E17               MVI     L,23D
0348 E5                 PUSH    H       ;AND CURSOR FOR LATER
0349 7D                 MOV     A,L     ;FIRST CORRECT LINFIL TABLE
034A 94                 SUB     H
034B CA8803             JZ      SCRLST
034E 214B0A             LXI     H,LINFIL+23D
0351 1E00               MVI     E,0     ;BOTTOM LINE GETS 0
0353 56         SLUP0:  MOV     D,M     ;GET CURRENT CONTENTS
0354 73                 MOV     M,E     ;SET TO NEW
```

```
0355 5A                 MOV     E,D       ;SET NEW TO CURRENT
0356 2B                 DCX     H         ;NEXT ENTRY
0357 3D                 DCR     A
0358 C25303             JNZ     SLUP0

               ; UPDATE CP*L POINTERS

035B C1                 POP     B         ;RETRIEVE CURSOR
035C 212D0A             LXI     H,CP1L    ;LINE POINTER
035F 0E03               MVI     C,CPNUM   ;POINTER COUNT
0361 78                 MOV     A,B       ;LINE MOVED
0362 BE         SLUP1:  CMP     M         ;WAS POINTER MOVED
0363 D26703             JNC     SLUP2     ;NO
0366 35                 DCR     M         ;MOVE UP ONE LINE
0367 23         SLUP2:  INX     H         ;ON TO NEXT POINTER
0368 23                 INX     H
0369 0D                 DCR     C         ;IF ANY
036A C26203             JNZ     SLUP1
               ;FINALLY DO THE ACTUAL SCROLL
036D 60         SLUP4:  MOV     H,B       ;FIND ADDRESS
036E 2E00               MVI     L,0
0370 CDFA03             CALL    MTXAD
0373 3E17               MVI     A,23D     ;HOW MANY LINES?
0375 90                 SUB     B
0376 47                 MOV     B,A
0377 EB                 XCHG              ;DESTINATION IN DE
0378 05         SLUP5:  DCR     B         ;DONE YET?
0379 FDE03            JM      SLDN4     ;REST IS COMMON
037C 218000             LXI     H,128D    ;OFFSET TO SOURCE
037F 19                 DAD     D
0380 E5                 PUSH    H         ;SAVE FOR NEXT TIME
0381 CDED03             CALL    MOV80     ;COPY LINE UP
0384 D1                 POP     D         ;NEW DESTINATION
0385 C37803             JMP     SLUP5
0388 C1         SCRLST: POP     B         ;CLEAN STACK
0389 324B0A             STA     LINFIL+23D ;BOTTOM LINE IS EMPTY
038C C36D03             JMP     SLUP4

                ;SUBROUTINE SCRLDN (H = LINE)
                ;       SCROLL INDICATED LINE AND ALL LINES BENEATH IT
                ;         DOWN ONE LINE.
                ;       H MUST CONTAIN A VALID LNE NUMBER (0 - 23).
                ;       CP*L AND LINFIL ARE UPDATED AS REQUIRED.
                ;       A, FLAGS, D, E, H & L MODIFIED.

038F C5         SCRLDN: PUSH    B         ;SAVE THE SACRED
0390 2E17               MVI     L, 23D
0392 E5                 PUSH    H         ;SAVE ARGUMENTS FOR LATER
0393 7D                 MOV     A,L       ;HOW MANY LINES?
0394 94                 SUB     H
0395 CA8803             JZ      SCRLST    ;BOTTOM LINE IS SPECIAL
              ;UPDATE LINFIL TABLE
0398 214B0A             LXI     H,LINFIL+23D
039B 2B         SLDN0:  DCX     H         ;PICK UP NEW VALUE
039C 56                 MOV     D,M
039D 23                 INX     H         ; AND PUT IT WHERE
039E 72                 MOV     M,D       ; IT BELONGS
```

```
039F 2B                 DCX     H         ;NEXT ENTRY
03A0 3D                 DCR     A         ;ANY LEFT?
03A1 C29B03             JNZ     SLDN0     ;  YES.
03A4 3600               MVI     M,0       ;CLEAR LAST ENTRY
                ;UPDATE CHARACTER POINTERS
03A6 D1                 POP     D         ;RETRIEVE PARAMETERS
03A7 D5                 PUSH    D
03A8 212D0A             LXI     H,CP1L    ;LINE POINTER
03AB 0E03               MVI     C,CPNUM   ;LINE POINTER COUNT
03AD 7E         SLDN1:  MOV     A,M       ;GET POINTER
03AE BA                 CMP     D         ;ABOVE TOP LINE?
03AF DAB703             JC      SLDN2     ;YES, NO CHANGE REQ
03B2 BB                 CMP     E         ;ON OR BELOW BOTTOM LINE?
03B3 D2B703             JNC     SLDN2     ;  YES, NO CHANGE REQ
03B6 34                 INR     M         ;MOVE POINTER DOWN 1 LINE
03B7 23         SLDN2:  INX     H         ;NEXT POINTER
03B8 23                 INX     H
03B9 0D                 DCR     C         ;ANY LEFT?
03BA C2AD03             JNZ     SLDN1     ;YES

                ;DO THE ACTUAL SCROLL
03BD C1                 POP     B         ;RETRIEVE PARAMETERS
03BE 79                 MOV     A,C
03BF 90                 SUB     B
03C0 47                 MOV     B,A       ;LINE COUNT IN B
03C1 AF                 XRA     A         ;MOV A,C W/ CY CLEAR
03C2 B1                 ORA     C
03C3 1F                 RAR               ;MULTIPLY LINE BY 128
03C4 57                 MOV     D,A
03C5 3E00               MVI     A,0
03C7 1F                 RAR
03C8 5F                 MOV     E,A       ;FINAL LINE OFFSET
03C9 2A4C0A             LHLD    MTXAT     ;BASE ADDRESS
03CC 19                 DAD     D         ;DESTINATION ADDRESS
03CD EB                 XCHG              ;GOES IN DE
03CE 05         SLDN3:  DCR     B         ;DONE YET?
03CF FADE03             JM      SLDN4     ;YES
03D2 2180FF             LXI     H,-128D   ;UP A LINE
03D5 19                 DAD     D
03D6 E5                 PUSH    H         ;SAVE FOR NEXT
03D7 CDED03             CALL    MOV80     ;COPY A LINE
03DA D1                 POP     D         ;RETRIEVE LAST SOURCE
03DB C3CE03             JMP     SLDN3     ;AND TRY AGAIN

                ;BLANK THE LINE
03DE 214F00     SLDN4:  LXI     H,79D     ;ALSO USED BY SCRLUP
03E1 19                 DAD     D
03E2 112050             LXI     D,80D*100H+' '
03E5 73         SLDN5:  MOV     M,E       ;BLAANK IT
03E6 2B                 DCX     H         ;NEXT
03E7 15                 DCR     D         ;IF ANY
03E8 C2E503             JNZ     SLDN5
03EB C1                 POP     B         ;RESTORE SACRED
03EC C9                 RET

                ;SUBROUTINE MOV80 (DE=DESTINATION, HL=SOURCE)
                ;       MOVE 80 CHARACTERS FROM ADDRESS IN HL TO THE ADD
```

```
                ;         IN DE.
                ;         A, D, E, FLAGS, H AND L MODIFIED.

03ED C5         MOV80:  PUSH    B
03EE 0650               MVI     B,80D
03F0 7E         MOV8L:  MOV     A,M
03F1 12                 STAX    D       ;COPY TO GOAL
03F2 23                 INX     H       ;NEXT SOURCE
03F3 13                 INX     D       ; AND DEST
03F4 05                 DCR     B       ;ANY MORE?
03F5 C2F003             JNZ     MOV8L   ;  YES
03F8 C1                 POP     B
03F9 C9                 RET

                ;SUBROUTINE MTXAD (H=LINE, L=COLUMN)
                ;       CALCULATE ADDRESS FOR LINE/COLUMN IN H,L.
                ;       ALL REGISTERS EXCEPT B & C MODIFIED.
03FA 3ABC0A     MTXAD:  LDA     WIDTH   ;40 OR 80 WIDE?
03FD FE29               CPI     41D     ;IS IT 40 OR LESS?
03FF D20504             JNC     MTXA0   ;  NO.
0402 7D                 MOV     A,L     ;MULTIPLY COLUMN BY 2
0403 87                ADD     A
0404 6F                 MOV     L,A
0405 7C         MTXA0:  MOV     A,H     ;TAKE LINE NUMBR
0406 29                 DAD     H       ;COLUMN * 2
0407 B7                 ORA     A       ;CLEAR CY SO CAN GET
0408 1F                 RAR             ;  (LINE+COL*2)/2
0409 67                 MOV     H,A
040A 7D                 MOV     A,L
040B 1F                 RAR
040C 6F                 MOV     L,A
040D EB                 XCHG            ;DE = LINE*128D + COLUMN
040E 2A4C0A             LHLD    MTXAT   ;ADR OF LINE 0, COL 0
0411 19                 DAD     D       ;DESIRED ADDRESS
0412 C9                 RET

                ;SUBROUTINE PUTIN (H=LINE, L=COLUMN)
                ;       MAKE ROOM ON A LINE FOR A NEW CHARACTER
                ;       IF LAST CHARACTER POSITION ON THE LINE IS
                ;         NOT A SPACE, HAS NO EFFECT.
                ;       LINFIL AND CP*C ARE UPDATED AS REQUIRED.
                ;       A, D, E, FLAGS, H & L MODIFIED.

0413 0E20       PUTSPC: MVI     C,' '   ;INSERT  A SPACE
0415 C5         PUTIN:  PUSH    B
0416 E5                 PUSH    H
0417 E5                 PUSH    H       ;SAVE EXTRA FPOR LATER
0418 11340A             LXI     D,LINFIL        ;CHECK IF ROOM FOR ANOTH
041B 6C                 MOV     L,H
041C 2600               MVI     H,0
041E 19                 DAD     D       ;LOOK UP CURRENT FILL
041F D1                 POP     D       ;GET LINE AND COL
0420 3ABC0A             LDA     WIDTH   ;FULL LINE SIZE
0423 BE                 CMP     M       ;IS THERE ROOM?
0424 CA3504             JZ      PUTI0   ;  NO
0427 34                 INR     M       ;UPDATE LINFIL
```

```
0428 7E              MOV     A,M      ;INSERT AFTER LINE END?
0429 BB              CMP     E
042A D23804          JNC     PUTIA    ;  NO
042D 73              MOV     M,E      ;THIS CHAR IS LAST
042E E1              POP     H        ;WHERE DOES IT GO?
042F CDFA03          CALL    MTXAD
0432 71              MOV     M,C      ;STUFF IT
0433 C1              POP     B
0434 C9              RET
                 ;CAN'T BE DONE
0435 C1      PUTIO:  POP     B        ;CLEAN UP STACK
0436 C1              POP     B
0437 C9              RET
                 ;SHIFT THE LINE OVER ONE COLUMN
0438 93      PUTIA:  SUB     E        ;NOW MANY COL NEED SHIFTING
0439 47              MOV     B,A      ;SAVE COUNTER IN B
043A EB              XCHG             ;GET IN POSITION
043B CDFA03          CALL    MTXAD    ;PHYSICAL ADDRESS OF INSERT
043E 7E      PUTID:  MOV     A,M      ;GET OLD
043F 71              MOV     M,C      ;STUFF WITH NEW
0440 4F              MOV     C,A      ;MAKE OLD NEW FOR NEXT
0441 23              INX     H        ;NEXT COLUMN
0442 3ABC0A          LDA     WIDTH    ;LOW RESOLUTION MODE?
0445 FE29            CPI     41D
0447 D24B04          JNC     PUTIC    ;NO, OK AS IS
044A 23              INX     H        ;EVERY OTHER
044B 05      PUTIC:  DCR     B        ;ANY LEFT?
044C F23E04          JP      PUTID    ;YES
                 ;FIX CURSOR POINTERS AFFECTED
044F C1              POP     B        ;GET CURSOR
0450 212D0A          LXI     H,CP1L   ;CHECK IF ON LINE
0453 1603            MVI     D,CPNUM  ;INIT COUNTER
0455 7E      PUTIE:  MOV     A,M      ;WHICH LINE?
0456 B8              CMP     B        ;THE ONE MOVED?
0457 CA6204          JZ      PUTIF    ;YES, CHECK IT OUT
045A 23      PUTIH:  INX     H        ;MMOVE TO NEXT
045B 23              INX     H
045C 15              DCR     D        ;ANY LEFT TO CHECK?
045D C25504          JNZ     PUTIE    ;  YES
0460 C1              POP     B        ;RESTORE SACRED
0461 C9              RET
0462 2B      PUTIF:  DCX     H        ;BACK UP TO COL
0463 7E              MOV     A,M
0464 B9              CMP     C        ;TO LEFT OF INSERT?
0465 D27204          JNC     PUTIG    ;  YES, NO CORRECTION REQ
0468 34              INR     M        ;MOVE OVER ONE
0469 3ABC0A          LDA     WIDTH    ;CHECK FOR IN RANGE
046C BE              CMP     M
046D C27204          JNZ     PUTIG    ;OK
0470 36FF            MVI     M,0FFH   ;FLAG AS OFF (SHOULD NEVER HAPPE
0472 23      PUTIG:  INX     H        ;BACK TO LINE
0473 C35A04          JMP     PUTIH    ;DO NEXT


             ;SUBROUTINE SETCON (B=COMMAND, C=PARAMETER)
             ;       PROCESSOR FOR CONFIGURATION SWITCHES.
             ;       USES TABLE SETTAB FOR DEFINITIONS.
```

```
                ;       CALLS SETTAB ROUTINES WITH
                ;          B=COMMAND        C=PARAMETER
                ;          D=PARAMETER-'0'
                ;          E=PARAMETER-'0'+ 60 HEX IF D NEGATIVE
                ;       ALL REGISTERS MODIFIED

0476 78         SETCON: MOV     A,B     ;VERIFY VALID COMMAND
0477 E65F               ANI     5FH     ;INSURE UC
0479 D641               SUI     'A'
047B F8                 RM              ;TOO SMALL
047C FE14               CPI     (SETEND-SETTAB)/2
047E D0                 RNC             ;TOO LARGE
047F 5F                 MOV     E,A     ;CALCULATE ENTRY
0480 1600               MVI     D,0
0482 215505             LXI     H,SETTAB
0485 19                 DAD     D
0486 19                 DAD     D       ;2 BYTES/ENTRY
0487 5E                 MOV     E,M     ;PICK UP ADDRESS
0488 23                 INX     H
0489 56                 MOV     D,M
048A EB                 XCHG
048B 79                 MOV     A,C     ;CALCULATE PARAMETER
048C E67F               ANI     7FH     ;  VARIANTS
048E D630               SUI     '0'
0490 57                 MOV     D,A
0491 F29604             JP      SETCO
0494 C660               ADI     60H
0496 5F         SETCO:  MOV     E,A
0497 E9                 PCHL            ;GO TO IT
0498 C9         SETNOT: RET             ;UNDEFINED, IGNORE

                ;
                ;       SETCON CALL ROUTINES
                ;

                   ;SET ALT-2480 BASE ADDRESS
0499 7A         SETADR: MOV     A,D
049A FE10               CPI     10H     ;MAKE SURE VALID
049C D0                 RNC             ;IT ISN'T
049D 07                 RLC             ;TIMES 16
049E 07                 RLC
049F 07                 RLC
04A0 07                 RLC
04A1 324D0A             STA     MTXAT+1 ;NEW HIGH ADDRESS BYTE
04A4 C9                 RET

                   ;SET OFFBOT SWITCH
04A5 7A         SETBOT: MOV     A,D
04A6 FE19               CPI     25D
04A8 D0                 RNC             ;ILLEGAL
04A9 32C40A             STA     OFFBOT
04AC C9                 RET

                   ;SET OFFTOP SWITCH
04AD 7A         SETTOP: MOV     A,D
04AE FE19               CPI     25D
04B0 D0                 RNC
```

```
04B1 32C30A            STA     OFFTOP
04B4 C9                RET

                ;SET OFFRT SWITCH
                ;    1 SETS TO 1, 0 SETS TO WIDTH
                ;    2 + SETS TO 1-WIDTH
04B5 7A        SETRT:  MOV     A,D
04B6 FE01              CPI     1
04B8 CAC404            JZ      SETRO
04BB 3ABC0A            LDA     WIDTH
04BE DAC404            JC      SETRO
04C1 2F                CMA
04C2 3C                INR     A
04C3 3C                INR     A         ;1-WIDTH
04C4 32C60A    SETRO:  STA     OFFRT
04C7 C9                RET

                ;SET OFFLFT SWITCH
                ;    SAVE ACTION AS SETRT
04C8 7A        SETLFT: MOV     A,D
04C9 FE01              CPI     1
04CB CAD704            JZ      SETLO
04CE 3ABC0A            LDA     WIDTH
04D1 DAD704            JC      SETLO
04D4 2F                CMA
04D5 3C                INR     A
04D6 3C                INR     A
04D7 32C50A    SETLO:  STA     OFFLFT
04DA C9                RET

                ;SET GREEK FOR GREEK TRANSLATION
04DB 3E9F      SETGRK: MVI     A,9FH
04DD 32BD0A            STA     GREEK
04E0 C9                RET

                ;SET GREEK FOR UPPER CASE ONLY
04E1 3EDF      SETUC:  MVI     A,0DFH
04E3 32BD0A            STA     GREEK
04E6 C9                RET

                ;SET GREEK FOR NORMAL LOWER CASE
04E7 3EFF      SETLC:  MVI     A,0FFH
04E9 32BD0A            STA     GREEK
04EC C9                RET

                ;SET DISPLAY WIDTH (LINE LENGTH)
04ED 7B        SETWDH: MOV     A,E
04EE 32BC0A            STA     WIDTH
04F1 C9                RET

                ;DEFINE A NEW CURSOR CHARACTER
04F2 79        SETCUR: MOV     A,C
04F3 32550A            STA     CURSOR
04F6 C9                RET

                ;DEFINE A NEW ESCAPE, ATTN, XON OR XOFF
04F7 21570A    SETESC: LXI     H,ESCAPE
```

```
04FA C30F05             JMP     SETCHR
04FD 21500A    SETATT:  LXI     H,ATTN
0500 C30F05             JMP     SETCHR
0503 21510A    SETXFF:  LXI     H,XOFF
0506 C30F05             JMP     SETCHR
0509 21520A    SETXN:   LXI     H,XON
050C C30F05             JMP     SETCHR

050F 79        SETCHR:  MOV     A,C       ;NEW CHARACTER
0510 E67F               ANI     7FH       ;NO PARITY ALLOWED
0512 77                 MOV     M,A
0513 C9                 RET

                    ;SET INPUT TTY LOCK
                    ;    OFF IF 0,  OTHERWISE ON.
0514 7A        SETTTY:  MOV     A,D
0515 B7                 ORA     A
0516 2F                 CMA               ;ASSUME OFF
0517 CA1C05             JZ      SETLK
051A 3EDF               MVI     A,0DFH    ;TURN ON
051C 32530A    SETLK:   STA     UCLOCK
051F C9                 RET

                    ;SET INPUT MODE
                    ;    0 = FULL DUPLEX, 1= HALF DUPLEX
                    ;    2 = BLOCK MODE
0520 7A        SETMODE:MOV      A,D
0521 B7                 ORA     A
0522 F8                 RM
0523 FE03               CPI     3
0525 D0                 RNC
0526 2F                 CMA
0527 3C                 INR     A
0528 32540A             STA     FDUX      ;SET FLAG
052B 3C                 INR     A         ;SET ECONTAB TO MATCH
052C 21D10A             LXI     H,CONTAB ;ASSUME FDUX
052F CA3B05             JZ      SETMF
0532 21CE0A             LXI     H,CONBLK ;MAYBE BLOCK?
0535 FA3B05             JM      SETMF
0538 211C0B             LXI     H,HDCON ;MUST BE HALF DUPLEX
053B 22B30A    SETMF:   SHLD    ECONAT
053E 2A320A             LHLD    CURSAT    ;XMIT POINTER
0541 222C0A             SHLD    CP1C
0544 21610A             LXI     H,LINBUF ;LINE BUFFER POINTER
0547 225E0A             SHLD    LBPTR
054A AF                 XRA     A         ;LINE BUFFER FILL COUNT
054B 32600A             STA     LBCNT
054E 325C0A             STA     LDONE     ;NO LINE AVAIL
0551 325D0A             STA     BDONE     ;NO BLOCK AVAIL
0554 C9                 RET


                    ;LOOKUP TABLE SETTAB
               ;        CONFIGURATION SWITCH SETTING DEFINITIONS
               ;        FORMAT IS
               ;            ADDRESS       ;FOR SWITCH 'A'
               ;            ADDRESS       ;FOR SWITCH  'B'
```

```
                ;          ETC.
                ;          SETEND MUST BE DEFINED TO SET TABLE LENGTH.

0555 9904       SETTAB: DW         SETADR   ;A=SET 2480 ADDRESS
0557 A504               DW         SETBOT   ;B=SET OFFBOT SWITCH
0559 F204               DW         SETCUR   ;C=SET CURSOR CHARACTER
055B E704               DW         SETLC    ;D=DISPLAY LOWER CASE AS LC
055D F704               DW         SETESC   ;E=DEFINE ESCAPE CHARACTER
055F 0305               DW         SETXFF   ;F=DEFINE XOFF CHARACTER
0561 DB04               DW         SETGRK   ;G=DISPLAY LOWER CASE AS GREEK
0563 E104               DW         SETUC    ;H=DISPLAY LOWER CASE AS UPPER
0565 FD04               DW         SETATT   ;I=DIFINE ATTN CHARACTER
0567 ED04               DW         SETWDH   ;J=SET DISPLAY WIDTH
0569 9804               DW         SETNOT   ;K=
056B C804               DW         SETLFT   ;L=SET OFFLFT SWITCH
056D 2005               DW         SETMODE  ;M=SET INPUT MODE
056F 0905               DW         SETXN    ;N=DEFINE XON CHARACTER
0571 9804               DW         SETNOT   ;O=
0573 9804               DW         SETNOT   ;P=
0575 9804               DW         SETNOT   ;Q=
0577 B504               DW         SETRT    ;R=SET OFFRT SWITCH
0579 1405               DW         SETTTY   ;S=SET/RESET TTY LOCK
057B AD04               DW         SETTOP   ;T=SET OFFTOP SWITCH
057D            SETEND: DS         0        ;END OF TABLE

                        ;********************

                ;
                ;          INPUT SUBROUTINES
                ;


                ;SUBROUTINE INFDX
                ;          BASIC KEYBOARD READ ROUTINE
                ;          RETURNS NEXT USER KEYSTROKE IN A
                ;          FLASHES CURSOR TO PROMPT USER.
                ;          CHECKS FOR SPECIAL CHARACTERS
                ;             SETC-SET CONFIGURATION SWITCHES
                ;             ATTN-RETURN TO MONITOR
                ;             XOFF-HALT OUTPUT.
                ;             XON -RESUME OUTPUT
                ;          A, D, E, FLAGS, H AND L MODIFIED.

057D 2AB50A     INFDX:  LHLD     INTRAP   ;CHECK FOR SPECIALS
0580 7C                 MOV      A,H
0581 B5                 ORA      L
0582 CA8605             JZ       INFDK     ;PROCEED
0585 E9                 PCHL               ;CHECK IT OUT
0586 2A320A     INFDK:  LHLD     CURSAT   ;GET CURSOR
0589 CDD902             CALL     TSTCUR   ;MAKE SURE ON SCREEN
058C CDFA03             CALL     MTXAD    ;CONVERT TO ADDRESS
058F 4E                 MOV      C,M      ;SAVE CURRENT CONTENTS
0590 3A550A             LDA      CURSOR   ;  IN C, CURSOR CHAR
0593 47                 MOV      B,A      ;  IN B
                    ;WAIT FOR USER INPUT
                        ;REGISTERS ARE SET UP AS FOLLOWS:
                        ;     B=CURSOR CHARACTER
```

```
                        ;    C=ORIGINAL SCREEN CONTENTS AT CURSOR POS
                        ;    HL=ADDRESS OF SCREEN CHAR CORRESPONDING
                        ;       TO CURRENT CURSOR POSITION.
0594 CDA507    INFDO:   CALL    STFDX    ;ANYTHING AVAILABLE?
0597 C2B405             JNZ     INFDD    ;FINALLY
059A 3E80               MVI     A,80H    ;KILL SOME TIME
059C 3D        INFD3:   DCR     A
059D C29C05             JNZ     INFD3
05A0 3A590A             LDA     FLASH    ;BUMP FLASH COUNTER
05A3 3C                 INR     A
05A4 32590A             STA     FLASH
05A7 C2AB05             JNZ     INFD1    ;TIME FOR CURSOR?
05AA 70                 MOV     M,B      ;YES
05AB FE80      INFD1:   CPI     80H      ;TIME FOR CURRENT?
05AD C2B105             JNZ     INFD2    ;  NO
05B0 71                 MOV     M,C
05B1 C39405    INFD2:   JMP     INFDO    ;KEEP TRYING

                    ;GET A CHARACTER AND CHECK IT OUT
05B4 71        INFDD:   MOV     M,C      ;RESTORE DISPLAY
05B5 CD0601             CALL    INKBD    ;GET USER INPUT
                                         ;  ALL REGISTERS EXPENDABLE
05B8 4F                 MOV     C,A      ;SAVE USER INPUT
05B9 214F0A             LXI     H,SETC   ;CHECK IF SPECIAL
05BC E67F               ANI     7FH      ;REMOVE PARITY BEFORE CHECKING
05BE BE                 CMP     M        ;SETC?
05BF CAD207             JZ      KSETC    ;  YES
05C2 23                 INX     H
05C3 BE                 CMP     M        ;ATTN?
05C4 CAC007             JZ      BREAK    ;  YES
05C7 23                 INX     H
05C8 BE                 CMP     M        ;XOFF?
05C9 C2D105             JNZ     INFDE    ;  NO
05CC 3EFF               MVI     A,OFFH   ;  YES
05CE C3D705             JMP     INFDF
05D1 23        INFDE:   INX     H
05D2 BE                 CMP     M        ;XON?
05D3 C2DD05             JNZ     INFDZ    ;  NO
05D6 AF                 XRA     A        ;CLEAR FLAG
05D7 324E0A    INFDF:   STA     XOFFD    ;SET XOFFED AS REQ.
05DA AF        INFDG:   XRA     A        ;RETURN NULL
05DB 37                 STC              ;  BUT WITH CY SET
05DC C9                 RET

05DD FE7F      INFDZ:   CPI     DEL      ;SPECIAL CASE
05DF CAEE05             JZ      INFDU    ;  NOT REALLY LC BUT IS.
05E2 E660               ANI     60H      ;LOWER CASE?
05E4 EE60               XRI     60H      ;CLEAR CY REGARDLESS
05E6 C2EE05             JNZ     INFDU    ;OK AS IS
05E9 3A530A             LDA     UCLOCK   ;FIX UP AS REQUIRED
05EC A1                 ANA     C
05ED 4F                 MOV     C,A
05EE 79        INFDU:   MOV     A,C      ;SET UP FOR RETURN
05EF C9                 RET              ;NOTE: CY MUST BE CLEAR

               ;
```

```
                ;HALF DUPLEX INPUT PROCESSING
                ;

05F0 3A5C0A     INHDX:  LDA     LDONE     ;GOT A LINE YET?
05F3 B7                 ORA     A
05F4 CC1006             CZ      INHDY     ;GO READ ONE
05F7 2A5E0A             LHLD    LBPTR     ;BUFFER POINTER
05FA 7E                 MOV     A,M       ;GET NEXT CHARACTER
05FB 23                 INX     H         ;UPDATE POINTER
05FC 225E0A             SHLD    LBPTR     ;NEW POINTER
05FF 21600A             LXI     H,LBCNT   ;CHARACTER COUNT
0602 35                 DCR     M         ;  IS ONE LESS
0603 C0                 RNZ               ;DONE IF NOT LAST
0604 215C0A             LXI     H,LDONE   ;RESET TO EMPTY
0607 3600               MVI     M,0
0609 21610A             LXI     H,LINBUF
060C 225E0A             SHLD    LBPTR
060F C9                 RET

                ;FILL UP THE LINE BUFFER

0610 CD7D05     INHDY:  CALL    INFDX     ;GET A CHAR
0613 47         INHD0:  MOV     B,A       ;SAVE ORIGINAL
0614 E67F               ANI     7FH       ;CLEAR PARITY
0616 4F                 MOV     C,A       ; AND SAVE A COPY
0617 FE7F               CPI     DEL       ;RUBOUT?
0619 CA5D06             JZ      RUBOUT    ;  YES
061C FE15               CPI     NAK       ;LINE CANCEL?
061E CA8206             JZ      CANCEL    ;  YES
                ;ADD CHAR TO BUFFER AND ECHO IT
0621 CD0002             CALL    ECHOCH    ;ECHO IT
0624 2A5E0A             LHLD    LBPTR     ;BUFFER POINTER
0627 70                 MOV     M,B       ;PUT IN BUFFER
0628 23                 INX     H         ;NEXT
0629 225E0A             SHLD    LBPTR     ;NEW POINTER
062C 21600A             LXI     H,LBCNT   ;CHAR COUNTER
062F 7E                 MOV     A,M
0630 34                 INR     M         ;BUMP IT
0631 FE4F               CPI     LINSIZ-1  ;FULL LINE?
0633 CA5206             JZ      INHD1     ;YES
0636 79                 MOV     A,C       ;GET COPY WITHOUT PARITY
0637 FE0D               CPI     CR        ;CARRIAGE RETURN?
0639 CA5806             JZ      INHDC     ;  APPEND A LINEFEED
063C FE1B               CPI     ESC       ;ESCAPE?
063E CA4606             JZ      INHDZ     ;  END THE LINE
0641 FE0A               CPI     LF        ;LINE FEED?
0643 C21006             JNZ     INHDY     ;  NO, GET ANOTHER CHARACTER
0646 3EFF       INHDZ:  MVI     A,0FFH    ;SET LINE COMPLETE FLAG
0648 325C0A             STA     LDONE
064B 21610A             LXI     H,LINBUF          ;RESET BUFFER POINTER
064E 225E0A             SHLD    LBPTR     ;  TO SCAN BUFFER
0651 C9                 RET
0652 79         INHD1:  MOV     A,C       ;TEST FOR CR
0653 FE0D               CPI     CR        ;  WHIH IS A SPECIAL
0655 C24606             JNZ     INHDZ     ;  CASE IN LINE OVERFLOW
0658 3E0A       INHDC:  MVI     A,LF      ;APPEND A LINEFEED
065A C31306             JMP     INHD0
```

```
                    ;RUBOUT LAST TYPED CHARACTER

065D 21600A     RUBOUT: LXI     H,LBCNT ;FIX CHAR COUNT
0660 7E                 MOV     A,M
0661 B7                 ORA     A       ;ANYTHING TO DELETE?
0662 CA1006             JZ      INHDY   ;NO
0665 35                 DCR     M       ;ONE LESS CHAR IN BUFFER
0666 2A5E0A             LHLD    LBPTR   ;FIX POINTER
0669 7E                 MOV     A,M     ;CHECK WHAT IS GETTING ZAPPED
066A 2B                 DCX     H
066B 225E0A             SHLD    LBPTR
066E 21320A     INHDR:  LXI     H,CURSC ;BACK UP ONE SPACE
0671 35                 DCR     M       ;NOTE: THIS ALGORITHM IS NOT
0672 0E20               MVI     C,' '   ;  GOOD WITH TABS
0674 CD0002             CALL    ECHOCH
0677 35                 DCR     M
0678 E660               ANI     60H     ;WAS IT A CONTROL CHAR?
067A 3EFF               MVI     A,0FFH  ;  IF SO, REPEAT TO DELETE
067C CA6E06             JZ      INHDR   ;  THE PRECEEDING UP ARROW
067F C31006             JMP     INHDY   ;PROCESS NEXT

                    ;CANCEL THE ENTIRE LINE TYPED
                    ;    NOTE:  IF LINE HAS OVERFLOWED ONTO NEXT LINE,
                    ;       THIS ALGORITHM WILL NOT CLEAN UP PREVIOUS PHYS

0682 21320A     CANCEL: LXI     H,CURSC ;CURRENT COLUMN
0685 0E20               MVI     C,' '   ;FILL WITH BLANKS
0687 35         INHDD:  DCR     M       ;BACK UP
0688 FA9206             JM      INHDE   ;ALL DONE
068B CD0002             CALL    ECHOCH  ;BLANK IT
068E 35                 DCR     M
068F C38706             JMP     INHDD   ;REPEAT TO COL 0
                    ;RESET POINTERS AND COUNTERS
0692 34         INHDE:  INR     M       ;FIX UP CURSOR
0693 AF                 XRA     A
0694 32600A             STA     LBCNT   ;NO BUFFER CONTENTS
0697 23                 INX     H       ;WHAT LINE DID WE ZAP?
0698 4E                 MOV     C,M
0699 47                 MOV     B,A     ;SET APPROPRIATE ENTRY
069A 21340A             LXI     H,LINFIL ;  IN LINFIL TABLE TO
069D 09                 DAD     B       ;   ZERO
069E 77                 MOV     M,A
069F 21610A             LXI     H,LINBUF
06A2 225E0A             SHLD    LBPTR   ;RESET POINTER
06A5 C31006             JMP     INHDY   ;NEXT CHARACTER


                ;
                ;BLOCK MODE INPUT ROUTINE
                ;

06A8 3A5D0A     INBLK:  LDA     BDONE   ;GOT A BLOCK YET?
06AB B7                 ORA     A
06AC CC4B07             CZ      INBLN   ;  NO, GET ONE
06AF 2A2C0A             LHLD    CP1C    ;GET 'AT' POINTER
06B2 7C                 MOV     A,H     ;CHECK IF ON SCREEN
```

```
06B3 FE18             CPI    24D      ;OFF BOTTOM OR TOP?
06B5 D2CB06           JNC    INBLZ    ;YES.  FATAL ERROR
06B8 B5               ORA    L        ;IS COLUMN POSITIVE?
06B9 F2C206           JP     INBL0    ;SEEMS TO BE
06BC 210000           LXI    H,00     ;START AT COL 0 LINE 0
06BF 222C0A           SHLD   CP1C
06C2 EB      INBL0:   XCHG            ;PUT AT IN DE
06C3 2A2E0A           LHLD   CP2C     ;GET LAST POINTER
06C6 7C               MOV    A,H      ;IS IT OFF SCREEN?
06C7 B5               ORA    L
06C8 F2E106           JP     INBL1    ;STILL ON SCREEN
06CB 2A2E0A  INBLZ:   LHLD   CP2C     ;COPY END POINTER TO
06CE 7C               MOV    A,H      ;  AT POINTER UNLESS
06CF B5               ORA    L        ;  WOULD BE OFF SCREEN,
06D0 3C               INR    A        ;  IN WHICH CASE GO BACK
06D1 C2D706           JNZ    INBLU    ;  TO LINE 0 COL 0
06D4 210000           LXI    H,0000H
06D7 222C0A  INBLU:   SHLD   CP1C
06DA 215D0A           LXI    H,BDONE  ;RESET BDONE
06DD 7E               MOV    A,M      ;WHILE PICKING
06DE 3600             MVI    M,0      ;  UP TERMINATOR.
06E0 C9               RET
06E1 7A      INBL1:   MOV    A,D      ;TEST FOR END OF TEXT
06E2 BC               CMP    H        ;HOW DO LINES COMPARE
06E3 DAEE06           JC     INBL2    ;NOT DOWN TO LAST YET
06E6 C2CB06           JNZ    INBLZ    ;WENT TOO FAR!
06E9 7B               MOV    A,E      ;COLUMN?
06EA BD               CMP    L
06EB D2CB06           JNC    INBLZ    ;THAT'S ALL FOLKS
06EE 4A      INBL2:   MOV    C,D      ;CURRENT LINE
06EF 0600             MVI    B,0      ;ANY DATA LEFT ON IT?
06F1 21340A           LXI    H,LINFIL
06F4 09               DAD    B
06F5 7E               MOV    A,M      ;CHARACTRS ON LINE
06F6 EB               XCHG            ;MEANTIME...
06F7 2C               INR    L        ;   SET POINTERS FOR
06F8 222C0A           SHLD   CP1C     ;   NEXT ITERATION
06FB 2D               DCR    L        ;BACK TO PRESENT
06FC BD               CMP    L        ;PAST END OF LINE?
06FD CA0C07           JZ     INBL3    ;SEND CR
0700 DA0F07           JC     INBL4    ;SEND LF
0703 CDFA03           CALL   MTXAD    ;SEND NEXT CHARACTER
0706 7E               MOV    A,M
0707 B7               ORA    A        ;BACKGROUND?
0708 FA1807           JM     INBLF    ;YES.
070B C9               RET
070C 3E0D    INBL3:   MVI    A,CR     ;CARRIAGE RETURN
070E C9               RET
070F 24      INBL4:   INR    H        ;DOWN 1 LINE
0710 2E00             MVI    L,0      ;FIRST COLUMN
0712 222C0A           SHLD   CP1C     ;NEW AT POINTER
0715 3E0A             MVI    A,LF     ;RETURN LF
0717 C9               RET
0718 212C0A  INBLF:   LXI    H,CP1C   ;BACK UP TO CHECKED
071B 35               DCR    M
071C 2A2C0A  INBL5:   LHLD   CP1C     ;SKIP TO NEXT FOREGROUND
071F 2C               INR    L        ;  FIELD
```

```
0720 222C0A            SHLD    CP1C      ;TRY NEXT CHARACTER
0723 3E17              MVI     A,23D     ;CHECK FOR OFF BOTTOM
0725 BC                CMP     H
0726 3E09              MVI     A,HT      ;ASSUME OFF
0728 D8                RC
0729 4C                MOV     C,H       ;HOW LONG IS CURRENT LINE?
072A 0600              MVI     B,0
072C EB                XCHG
072D 21340A            LXI     H,LINFIL
0730 09                DAD     B
0731 7B                MOV     A,E       ;LINE LENGTH
0732 BE                CMP     M         ;TRIED THEM ALL?
0733 EB                XCHG
0734 D24207            JNC     INBL6     ;YES.  NEXT LINE
0737 CDFA03            CALL    MTXAD     ;CONVERT TO ADDRESS
073A 7E                MOV     A,M       ;FORGROUND?
073B B7                ORA     A
073C FA1C07            JM      INBL5     ;NO, TRY NEXT
073F 3E09              MVI     A,HT
0741 C9                RET
                ;MOVE DOWN TO NEXT LINE
0742 24        INBL6:  INR     H
0743 2E00              MVI     L,0       ;RESET COL
0745 222C0A            SHLD    CP1C
0748 C31807            JMP     INBLF

                ;ACCEPT A BLOCK FROM THE KEYBOARD

074B 2A2C0A    INBLN:  LHLD    CP1C      ;IF START POINTER IS
074E 7C                MOV     A,H       ;  OFF SCREEN, RESET TO HOME
074F B5                ORA     L         ;  BEFORE PROCEEDING
0750 3C                INR     A
0751 C25A07            JNZ     INBLL     ;OK AS IS
0754 210000            LXI     H,0000H   ;RESET REQUIRED
0757 222C0A            SHLD    CP1C
075A CD7D05    INBLL:  CALL    INFDX     ;GET A CHARACTER
075D E67F              ANI     7FH       ;CLEAR PARITY
075F 4F                MOV     C,A       ;SAVE A COPY
0760 3A560A            LDA     FIXUP     ;WAS PREVIOUS AN ESCAPE?
0763 B7                ORA     A
0764 3E00              MVI     A,0       ;CLEAR FLAG REGARDLESS
0766 32560A            STA     FIXUP
0769 CA7007            JZ      INBLM     ;NO FIXUP REQUIRED
076C 79                MOV     A,C       ;PATCH IT UP
076D F680              ORI     80H
076F 4F                MOV     C,A
0770 79        INBLM:  MOV     A,C       ;GET  A COPY
0771 21570A            LXI     H,ESCAPE  ;GOT AN ESCAPE?
0774 BE                CMP     M
0775 CA8907            JZ      INBLE     ;YES
0778 23                INX     H         ;GOT AN END OF TEXT?
0779 BE                CMP     M
077A CA9A07            JZ      INBLT     ;  YES
077D EE80              XRI     80H       ;GOT SET START?
077F BE                CMP     M         ;  =ESC EOT
0780 CA9107            JZ      INBLV     ;YES
0783 CD0002            CALL    ECHOCH    ;ECHO IT
```

```
0786 C35A07            JMP     INBLL    ;AND GET ANOTHER

0789 3EFF      INBLE:  MVI     A,OFFH   ;SET FIXUP FLAG
078B 32560A            STA     FIXUP
078E C35A07            JMP     INBLL    ;GET ANOTHER CHARACTER

0791 2A320A    INBLV:  LHLD    CURSAT   ;SET START MARKER
0794 222C0A            SHLD    CP1C
0797 C35A07            JMP     INBLL    ;GET THE NEXT
079A 79        INBLT:  MOV     A,C      ;SET BUFFER READY FLAG
079B 325D0A            STA     BDONE
079E 2A320A            LHLD    CURSAT   ;SET END MARKER
07A1 222E0A            SHLD    CP2C
07A4 C9                RET


               ;INPUT STATUS ROUTINES
               ;       RETURN A=0 (Z=1) IF NO CHAR AVAILABLE
               ;       RETURN A=FF(HEX) (Z=0) IF A CHAR IS AVAILABLE
               ;       A AND FLAGS MODIFIED.

                   ;FULL DUPLES INPUT MODE
07A5 C5        STFDX:  PUSH    B        ;EXTERNAL INTERFACES ARE
07A6 D5                PUSH    D        ;  INHERENTLY UNTRUSTWORTHY
07A7 E5                PUSH    H
07A8 CD0301            CALL    INKBS    ;KEYBOARD READY?
07AB E1                POP     H
07AC D1                POP     D
07AD C1                POP     B
07AE C3BA07            JMP     STHDB    ;TEST FLAGS

                   ;HALF DUPLEX INPUT MODE
07B1 3A5C0A    STHDX:  LDA     LDONE    ;SEEN TERMINATOR YET?
07B4 C3BA07            JMP     STHDB    ;SET FLAGS ACCORDINGLY

                   ;BLOCK INPUT MODE
07B7 3A5D0A    STBLK:  LDA     BDONE    ;'END OF TEXT' ENTERED?
07BA B7        STHDB:  ORA     A        ;RETURN Z=1, A=0 IF A=0
07BB C8                RZ               ;       Z=0, A=-1 IF A NOT 0
07BC 3EFF              MVI     A,OFFH
07BE B7                ORA     A
07BF C9                RET

               ;SUBROUTINE BREAK
               ;       JUMP TO MONITOR ENTRY POINT
               ;       WILL NOT JUMP IF ADDRESS IS FFFF HEX.

07C0 21DA05    BREAK:  LXI     H,INFDG
07C3 E5                PUSH    H        ;RETURN ADDRESS
07C4 2A5A0A            LHLD    MONLVL   ;BREAK ADDRESS
07C7 7C                MOV     A,H      ;CHECK IF SPECIFIED
07C8 A5                ANA     L
07C9 3C                INR     A
07CA CACE07            JZ      BRK00    ;NONE SPECIFIED
07CD E9                PCHL             ;JMP TO IT
07CE 3A500A    BRK00:  LDA     ATTN     ;RESTORE A
07D1 C9                RET              ;AND GO TO INFDG
```

```
                ;SUBROUTINE KSETC
                ;       TRAP NEXT TWO KEYSTROKES TO SET CONFIGURATION
                ;       PROMPTS USER TO AVOID CONFUSION.

07D2 2A320A     KSETC:  LHLD    CURSAT  ;SAVE CURSOR
07D5 E5                 PUSH    H
07D6 2A4C0A             LHLD    MTXAT   ;DISPLAY ADDRESS
07D9 1610               MVI     D,10H   ;SAVE TOP LINE
07DB 46         KSET0:  MOV     B,M
07DC 3620               MVI     M,' '   ;CLEAR SOME ECOJHO SPACE
07DE 23                 INX     H
07DF 4E                 MOV     C,M
07E0 3620               MVI     M,' '
07E2 23                 INX     H
07E3 C5                 PUSH    B
07E4 15                 DCR     D       ;ANY MORE LEFT?
07E5 C2DB07             JNZ     KSET0   ;  YES
07E8 E5                 PUSH    H       ;SAVE LAST ADDRESS
07E9 210000             LXI     H,0     ;SET CURSOR
07EC 22320A             SHLD    CURSAT
07EF 213608             LXI     H,KSETM ;MESSAGE
07F2 0604               MVI     B,KSETN-KSETM ;  AND LENGTH
07F4 4E         KSET1:  MOV     C,M
07F5 CD0002             CALL    ECHOCH
07F8 23                 INX     H
07F9 05                 DCR     B
07FA C2F407             JNZ     KSET1
                     ;ACCEPT NEW PARAMETERS
07FD CD7D05             CALL    INFDX   ;COMMAND
0800 F5                 PUSH    PSW     ;SAVE FOR LATER
0801 4F                 MOV     C,A
0802 CD0002             CALL    ECHOCH  ;ECHO IT
0805 0E20               MVI     C,' '   ;SPAC OVER
0807 CD0002             CALL    ECHOCH
080A CD7D05             CALL    INFDX   ;GET VALUE
080D 4F                 MOV     C,A
080E CD0002             CALL    ECHOCH  ;ECHO IT
0811 F1                 POP     PSW     ;RETRIEVE COMMAND
0812 47                 MOV     B,A
0813 C5                 PUSH    B       ;SAVE ENTIRITY
0814 0E3F               MVI     C,'?'   ;VERIFY CORRECT
0816 CD0002             CALL    ECHOCH
0819 CD7D05             CALL    INFDX
081C E65F               ANI     5FH     ;TAKE CARE OF LC
081E FE59               CPI     'Y'     ;YES???
0820 C1                 POP     B       ;RETRIEVE ARGUMENTS
0821 CC7604             CZ      SETCON  ;DO IT IF OK
                     ;CLEAN UP THE RESULTS
0824 E1                 POP     H       ;RESTORE ADDRESS
0825 1610               MVI     D,10H   ;COUNT
0827 C1         KSET9:  POP     B       ;GET 2 CHARS
0828 2B                 DCX     H
0829 71                 MOV     M,C     ;AND RESTORE THEM
082A 2B                 DCX     H
```

```
082B 70                 MOV     M,B
082C 15                 DCR     D         ;ANY LEFT?
082D C22708             JNZ     KSET9     ;KEEP TRUCKING
0830 E1                 POP     H         ;RESTORE CURSOR
0831 22320A             SHLD    CURSAT
0834 37                 STC               ;FLAG AS IGNORABLE
0835 C9                 RET               ;AND DONE
0836 5345542D KSETM:    DB      'SET-'    ;PROMPT
083A          KSETN:    DS      0         ;END OF PROMPT


083A C5        XOFFED:  PUSH    B         ;SAVE SACRED
083B CD0301             CALL    INKBS     ;ANY USER ACTION?
083E B7                 ORA     A
083F C47D05             CNZ     INFDX     ;CHECK IT OUT
0842 3A4E0A             LDA     XOFFD     ;CHECK FLAG
0845 B7                 ORA     A
0846 C1                 POP     B         ;RESTORE SACRED
0847 C9                 RET

               ;        ********************************


               ;
               ;        CONTROL CHARACTER  PROCESSING ROUTINES
               ;

               ;ESCAPE CHARACTER PROCESSING
               ;        TRAP TO ADD PARITY BIT TO NEXT CHARACTER
0848 215708    ESCCHR:  LXI     H,ESCTRP ;SET UP TRAP
084B 04        TRPSET:  INR     B         ;WHICH ONE?
084C F25308             JP      ESCAO     ;OUTPUT
084F 22C10A             SHLD    IMULJM    ;ECHO
0852 C9                 RET
0853 22BF0A    ESCAO:   SHLD    MULJMP
0856 C9                 RET
                   ;ESCAPE TRAP
0857 79        ESCTRP:  MOV     A,C       ;TURN ON PARITY BIT
0858 F680               ORI     80H       ; AND CLEAR CY
085A 4F                 MOV     C,A
085B 210000    TRPCLR:  LXI     H,0       ;RESET TRAP VECTOR
085E C34B08             JMP     TRPSET

               ;LOCK KEYBOARD

0861 2A5A0A    LCKKB:   LHLD    MONLVL    ;TRAP ATTEMPT TO ACCESS
0864 22B50A             SHLD    INTRAP
0867 C9                 RET

               ;UNLOCK KEYBOARD

0898                    UNLKKB  SET       CLRTRP

               ;RETURN HEREIS MESSAGE
```

```
0868 217A08   HEREIS: LXI     H,HRISM ;MESSAGE ADDRESS
086B 22B90A           SHLD    HRISP   ;GOES INPOINTER
086E 3E11             MVI     A,HRISN-HRISM ;COUNTER
0870 32BB0A           STA     HRISC
0873 218B08           LXI     H,HERETO ;HEREIS TRAP
0876 22B50A           SHLD    INTRAP
0879 C9               RET

087A 4D617472 HRISM:  DB      'MATROX ALT-2480',CR,LF
087E 6F782041
0882 4C542D32
0886 3438300D
088A 0A
088B          HRISN:  DS      0

088B 2AB90A   HERETO: LHLD    HRISP   ;MESSAGE POINTER
088E 7E               MOV     A,M     ;GET A CHAR
088F 23               INX     H       ;SET FOR NEXT
0890 22B90A           SHLD    HRISP
0893 21BB0A           LXI     H,HRISC ;COUNTER
0896 35               DCR     M       ;ANY LEFT?
0897 C0               RNZ             ;YES
0898 210000   CLRTRP: LXI     H,0     ;COMMON CODE SEGMENT
089B 22B50A           SHLD    INTRAP
089E C9               RET

              ;BACKSPACE

089F 2D       BACKSP: DCR     L       ;BACK ONE
08A0 CDD902   TESTIT: CALL    TSTCUR  ;STAY ON PAGE
08A3 22320A           SHLD    CURSAT
08A6 C9               RET

              ;HORIZONTAL TAB

08A7 7D       HORTAB: MOV     A,L     ;NEXT COL MOD 8
08A8 C608             ADI     08H     ;OVER 8
08AA E6F8             ANI     0F8H    ;AND BACK TO LAST MOD 8
08AC 6F               MOV     L,A
08AD C3A008           JMP     TESTIT

              ;CARRIAGE RETURN AND LINE FEED
              ;WARNING***  LINE FEED MUST FOLLOW IMMEDIATELY

08B0 2E00     CRLF:   MVI     L,0     ;RESET COLUMN

              ;LINE FEED

08B2 24       LNFEED: INR     H       ;DOWN ONE LINE
08B3 C3A008           JMP     TESTIT

              ;VERTICAL TAB

08B6 7C       VERTAB: MOV     A,H     ;NEXT LINE MOD 8
08B7 C608             ADI     08H
08B9 E6F8             ANI     0F8H
08BB 67               MOV     H,A
```

```
08BC C3A008             JMP     TESTIT

                ;FORM FEED

08BF 210000     FORMFD: LXI     H,0000  ;UPPER LEFT
08C2 222C0A             SHLD    CP1C
08C5 22320A             SHLD    CURSAT
08C8 C3F009             JMP     CLEAR   ;CLEAR THE SCREEN TOO

                ;CARRIAGE RETURN

08CB 2E00       CARRET: MVI     L,00    ;COLUMN ZERO
08CD 22320A             SHLD    CURSAT
08D0 C9                 RET

                ;UPLINE

08D1 25         UPLINE: DCR     H       ;UP ONE LINE
08D2 C3A008             JMP     TESTIT

                ;FORESPACE

08D5 2C         FORSPC: INR     L       ;NEXT COLUMN
08D6 C3A008             JMP     TESTIT
                ;DIRECT CURSOR ADDRESSING

08D9 21DF08     DCACOM: LXI     H,DCAY
08DC C34B08             JMP     TRPSET  ;SET TRAP VECTOR
                    ;READ LINE DESIRED
08DF 21FE08     DCAY:   LXI     H,DCAX
08E2 CD4B08             CALL    TRPSET
08E5 79                 MOV     A,C
08E6 D620               SUI     ' '
08E8 FE18               CPI     24D
08EA DAEF08             JC      STATMP  ;OK AS IS
08ED 3E00               MVI     A,0
08EF E5         STATMP: PUSH    H       ;SAVE FOR GP ROUTINE
08F0 21C90A             LXI     H,TEMP  ;ASSUME OUTPUT MODE
08F3 04                 INR     B
08F4 F2FA08             JP      DCAY2
08F7 21B70A             LXI     H,TEMPE ;WRONG
08FA 77         DCAY2:  MOV     M,A     ;STORE IT
08FB E1                 POP     H
08FC 37                 STC             ;INHIBIT FURTHER PROCESSING
08FD C9                 RET
                    ;READ COLUMN AND SET CURSOR
08FE 210000     DCAX:   LXI     H,0
0901 CD4B08             CALL    TRPSET
0904 79                 MOV     A,C
0905 D620               SUI     ' '
0907 21BC0A             LXI     H,WIDTH
090A BE                 CMP     M
090B DA0F09             JC      DCAX1   ;OK AS IS
090E 7E                 MOV     A,M
090F 6F         DCAX1:  MOV     L,A
0910 CD1909             CALL    LDATMP  ;GET STORED ARG
0913 67                 MOV     H,A
```

```
0914 22320A          SHLD    CURSAT
0917 37              STC
0918 C9              RET

               ;ROUTINES LDATMP AND STATMP TO LOAD AND
               ;    STORE THE TEMPORARY VARIABLE AS PER ECHO OR OUTPUT.

0919 E5        LDATMP: PUSH    H          ;SAVE H
091A 21C90A            LXI     H,TEMP     ;ASSUME UTPUT MODE
091D 04                INR     B
091E F22409            JP      LDATP      ;GOOD ASSUMPTION
0921 21B70A            LXI     H,TEMPE
0924 7E        LDATP:  MOV     A,M        ;FETCH IT
0925 E1                POP     H
0926 C9                RET

               ;INSERT A STRING OF CHARACTERS

0927 212D09    PUTSTR: LXI     H,PUTSO ;SET UP TRAP
092A C34B08            JMP     TRPSET

092D 3E1F      PUTSO:  MVI     A,' '-1 ;VALID CHARACTER?
092F B9                CMP     C          ;QUIT IF CONTROL
0930 3F                CMC                ;RETURN FLAG VALUE
0931 DA5B08            JC      TRPCLR     ;END OF INSERT
0934 2A320A            LHLD    CURSAT     ;GOES HERE
0937 C5                PUSH    B          ;SAVE VITAE
0938 CD1304            CALL    PUTSPC     ;MAKE ROOM
093B C1                POP     B
093C AF                XRA     A          ;ALLOW FURTHER PROCESSING
093D C9                RET

               ;DELETE THE CHARACTER SPECIFIED BY H,L

093E EB        DECHAR: XCHG               ;MAKE ROOM IN HL
093F 4A                MOV     C,D        ;CHECK LINE FILL
0940 0600              MVI     B,0
0942 21340A            LXI     H,LINFIL
0945 09                DAD     B
0946 7E                MOV     A,M        ;NUMBER OF CHARS ON LINE
0947 0C                INR     C          ;  PLUS ONE
0948 B9                CMP     C          ;CHAR TO BE DELETED
0949 D8                RC                 ;NOTHING THERE
094A 35                DCR     M          ;ONE LESS NOW
                   ;FIX UP CURSOR POINTERS
094B 0603              MVI     B,CPNUM ;COUNTER
094D 212D0A            LXI     H,CP1L     ;LINE POINTER
0950 7A        DECH1:  MOV     A,D        ;CHECK LINE
0951 BE                CMP     M          ;SAME ONE?
0952 C25D09            JNZ     DECH2      ;NO, CAN IGNORE
0955 2B                DCX     H          ;CHECK COLUMN
0956 7B                MOV     A,E
0957 BE                CMP     M          ;TO THE RIGHT?
0958 D25C09            JNC     DECH3      ;  NO, NOT AFFECTED
095B 35                DCR     M          ;FIX IT UP
095C 23        DECH3:  INX     H          ;BACK TO LINE POINTER
095D 23        DECH2:  INX     H          ;ON TO NEXT
```

```
095E 23                 INX     H
095F 05                 DCR     B         ;ANY LEFT?
0960 C25009             JNZ     DECH1     ;  DO IT
                ;FINALLY, DELETE THE CHARACTER
0963 EB                 XCHG              ;SET UP FOR MTXAD
0964 E5                 PUSH    H         ;SAVE FOR LATER
0965 CDFA03             CALL    MTXAD     ;CONVERT TO ADDRESS
0968 E3                 XTHL              ;SAVE AND RETRIEVE
0969 3ABC0A             LDA     WIDTH     ;LAST ON LINE
096C 0E00               MVI     C,0       ;HIGH OR LOW RESOLUTION?
096E FE29               CPI     41D       ;ASSUME HIGH
0970 D27509             JNC     DECH6     ;  IT IS
0973 0E80               MVI     C,80H     ;LOW RESOLUTION FLAG
0975 6F         DECH6:  MOV     L,A       ;ADDRESS OF LAST POSITION
0976 2D                 DCR     L         ;CONVERT TO COL NUM
0977 CDFA03             CALL    MTXAD     ; AND THEN TO ADDRESS
097A D1                 POP     D         ;STARTING AT
097B EB                 XCHG
097C 7B                 MOV     A,E       ;FINAL ADDRESS
097D BD         DECH4:  CMP     L         ;DONE YET?
097E CA9809             JZ      DECH5     ;YES
0981 23                 INX     H         ;MOVE A CHAR
0982 0C                 INR     C         ;CHECK FOR LOW RES
0983 F28709             JP      DECH7
0986 23                 INX     H         ;EVERY OTHER ADDRESS
0987 46         DECH7:  MOV     B,M       ;GET A CHAR
0988 2B                 DCX     H         ;BACK UP ONE SPACE
0989 0C                 INR     C         ;CHECK RESOLUTION
098A F28E09             JP      DECH8
098D 2B                 DCX     H         ;LOW
098E 70         DECH8:  MOV     M,B       ;STUFF IT
098F 23                 INX     H         ;SET UP FOR NEXT
0990 0D                 DCR     C         ;CHECK RESOLUTION
0991 F27D09             JP      DECH4     ;HIGH
0994 23                 INX     H
0995 C37D09             JMP     DECH4
0998 3620       DECH5:  MVI     M,' '     ;CLEAR LAST COLUMN
099A C9                 RET

                ;SET CONFIGURATION SWITCHES

099B 04         CONSET: INR     B         ;ONLY VALID ON OUTPUT
099C F8                 RM
099D 21A409             LXI     H,CONS1   ;SET UP TRAP
09A0 22BF0A             SHLD    MULJMP
09A3 C9                 RET
09A4 21AE09     CONS1:  LXI     H,CONS2   ;SET NEW TRAP
09A7 22BF0A             SHLD    MULJMP
09AA 79                 MOV     A,C       ;SAVE COMMAND
09AB C3EF08             JMP     STATMP    ;SAVE AND RETURN
09AE 210000     CONS2:  LXI     H,0       ;RESET TRAP
09B1 22BF0A             SHLD    MULJMP
09B4 CD1909             CALL    LDATMP    ;RETRIEVE COMMAND
09B7 47                 MOV     B,A
09B8 CD7604             CALL    SETCON
09BB 37                 STC               ;ALL DONE
09BC C9                 RET
```

```
                 ;BELL (FLASH SCREEN) ROUTINE

09BD 2A4C0A      BELL:   LHLD    MTXAT
09C0 E5                  PUSH    H
09C1 CDCF09              CALL    BELLO
09C4 1B          BELLK:  DCX     D          ;KILL SOME TIME
09C5 7A                  MOV     A,D
09C6 B3                  ORA     E
09C7 C2C409              JNZ     BELLK
09CA E1                  POP     H
09CB CDCF09              CALL    BELLO
09CE C9                  RET

09CF 11000C      BELLO:  LXI     D,128D*24D      ;TOTAL OF CHARS
09D2 7E          BELL1:  MOV     A,M
09D3 EE80                XRI     80H
09D5 77                  MOV     M,A
09D6 1B                  DCX     D
09D7 23                  INX     H
09D8 7A                  MOV     A,D
09D9 B3                  ORA     E
09DA C2D209              JNZ     BELL1
09DD C9                  RET

                 ;FOREGROUND FOLLOWS

09DE AF          OUTFOR: XRA     A
09DF 32BE0A              STA     FORBAK
09E2 C9                  RET

                 ;BACKGROUND FOLLOWS

09E3 3E80        OUTBAK: MVI     A,80H
09E5 32BE0A              STA     FORBAK
09E8 C9                  RET

                 ;HOME CURSOR

09E9 210000      HOMEIT: LXI     H,0000
09EC 22320A              SHLD    CURSAT
09EF C9                  RET

                 ;SUBROUTINE CLEAR
                 ;       SET ALL DISPLAY POSITIONS TO BLANK
                 ;       RESET LINFIL TABLE TO ZERO
                 ;       ALL REGISTERS EXCEPT B & C ARE MODIFIED.

09F0 C5          CLEAR:  PUSH    B
09F1 21340A              LXI     H,LINFIL        ;RESET LINFIL 1ST
09F4 111800              LXI     D,24D    ;BYTE COUNT
09F7 0E00                MVI     C,00     ;STUFF WITH ZEROS
09F9 CD090A              CALL    FILLUP   ;SET THEM ALL
09FC 2A4C0A              LHLD    MTXAT    ;DISPLAY ADDRESS
09FF 11010C              LXI     D,24D*128D+1 ;ERASE BETWEEN LINES TOO
0A02 0E20                MVI     C,' '    ;FILL WITH BLANKS
```

```
0A04 CD090A            CALL    FILLUP
0A07 C1                POP     B
0A08 C9                RET

               ;SUBROUTINE FILLUP (C=VALUE, DE=BYTE COUNT, HL=ADDR
               ;       SET DE POINTS TO C STARTING AT H
               ;       RETURNS WITH A=0, DE=0, HL=NEXT ADDRES
               ;       WARNING***  DE = 0 DOES 64K
               ;       A, D, E, FLAGS, H & L MODIFIED

0A09 71        FILLUP: MOV     M,C     ;STUFF ONE
0A0A 23                INX     H       ;NEXT ADR
0A0B 1B                DCX     D       ;ONE LESS TO DO
0A0C 7A                MOV     A,D
0A0D B3                ORA     E
0A0E C2090A            JNZ     FILLUP  ;ANY LEFT?
0A11 C9                RET

0A12 04        ANULL:  INR     B       ;ECHO OR PRINT '^O'
0A13 FA210A            JM      ANULE   ;ECHO IT
0A16 0E5E              MVI     C,'^'
0A18 CD0C02            CALL    OUTCHR
0A1B 0E30              MVI     C,'O'
0A1D CD0C02            CALL    OUTCHR
0A20 C9                RET
0A21 0E5E      ANULE:  MVI     C,'^'
0A23 CD0002            CALL    ECHOCH
0A26 0E30              MVI     C,'O'
0A28 CD0002            CALL    ECHOCH
0A2B C9                RET


               ;***********************************************
               ;
               ;       VARIABLES AND SWITCHES
               ;
               ;***********************************************

               ;GLOBALS


               ;CHARACTER POINTERS
               ;       THESE POINTERS ARE MAINTAINED TO ALWAYS POINT
               ;         TO THE SAME CHARACTER.  IF THE CHARACTER
               ;         AT THAT POSITION IS DELETED OR OVERWRITTTEN
               ;         THEY POINT TO ITS REPLACEMENT.  IF THE CHARACT
               ;         IS MOVED OFF THE SCREEN, ONE OF ITS COORDINATE
               ;         WILL BE SET TO -1.

0A2C FFFF      CPTRS:  DW      0FFFFH          ;CP1 IS XMIT POINTER
0A2E FFFF              DW      0FFFFH          ;CP2 IS END POINTER
0A2C                   CP1C    EQU     CPTRS ;INIT TO OFF SCREEN
0A2D                   CP1L    EQU     CP1C+1
0A2E                   CP2C    EQU     CPTRS+2
0A2F                   CP2L    EQU     CP2C+1
0A30                   DS      CPNUM*2 - 4 ;EXTRAS
```

```
                ;THE CURRENT CURSOR POSITION
                ;       NOTE: MAY DRIFT PAST EDGES

0A32 0000       CURSAT: DW      0000H   ;START AT UPPER LEFT

0A33                    CURSL   EQU     CURSAT+1  ;LINE NUMBER
0A32                    CURSC   EQU     CURSAT+0  ;COLUMN NUMBER

                ;TABLE OF RIGHTMOST COLUMN IN EACH LINE CONTAINING EXPLI
                ;       (I.E. NOT BLANK FROM CLEAR TYPE FUNCTION)
                ;       INDEXED BY LINE NUMBER

0A34            LINFIL: DS      24D

                ;BASE ADDRESS OF THE 4K MEMORY SPACE USED BY THE 2480

0A4C 00E0       MTXAT:  DW      0E000H


                ;               **********************

                ;INPUT CONTROL SWITCHES

0A4E 00         XOFFD:  DB      0       ;INHIBIT OUTCHR OUTPUT IF -1

0A4F 02         SETC:   DB      STX     ;CONFIGURATION CHANGE CHARACTER

0A50 03         ATTN:   DB      ETX     ;BREAK CHARACTER

0A51 13         XOFF:   DB      DC3     ;INPUT CHAR TO STOP OUTPUT

0A52 11         XON:    DB      DC1     ;INPUT CHAR TO RESUME OUTPUT

0A53 FF         UCLOCK: DB      0FFH    ;-1: NORMAL
                                        ;DFHEX: CONVERT LC TO UC ON INPU

0A54 FF         FDUX:   DB      0FFH    ;00: HALF DUPLEX
                                        ;-1: FULL DUPLEX
                                        ;-2: BLOCK MODE

0A55 DF         CURSOR: DB      '_'+PARON ;CHAR TO USE FOR CURSOR

0A56 00         FIXUP:  DB      0       ;-1 IF PREVIOUS CHAR WAS ESCPE C

0A57 1B         ESCAPE: DB      ESC     ;INPUT ESCAPE CHARATER

0A58 04         BLKEND: DB      EOT     ;INPUT BLOCK TERMINATE CHARACTER

0A59 00         FLASH:  DB      0       ;COUNTER FOR TIMING CURSOR FLASH

0A5A 0000       MONLVL: DW      0000H   ;MONITOR TRAP ADDRESS
                                        ;  DISABLED IF 0000

0A5C 00         LDONE:  DB      0       ;0 = NOT YET
```

```
                                             ;-1=LINE AVAILABLE

0A5D 00        BDONE:  DB       0         ;0 = BLOCK BEING FILLED
                                          ;NON-ZERO = BLOCK COMPLETE

0A5E 610A      LBPTR:  DW       LINBUF    ;LINE BUFFER POINTER

0A60 00        LBCNT:  DB       0         ;LINE BUFFER FILL COUNT

0A61           LINBUF: DS       LINSIZ+2 ;LINE BUFFER

0AB3 D10A      ECONAT: DW       CONTAB    ;ECHO CONTROL TABLE TO USE

0AB5 0000      INTRAP: DW       0000H     ;INPUT TRAP VECTOR

0AB7           TEMPE:  DS       5         ;TEMPORARY VARIABLE SPACE

0ABB                   HRISC    EQU       TEMPE+4
0AB9                   HRISP    EQU       TEMPE+2

               ;****************************************

               ;                OUTPUT SWITCHES

0ABC 28        WIDTH:  DB       40D       ;COLUMNS PER LINE

0ABD 5F        GREEK:  DB       5FH       ;FF: NORMAL
                                          ;5F: LC DISPLAYS AS UPPER CASE
                                          ;1F: LC DISPLAYED AS GREEK

0ABE 00        FORBAK: DB       0         ;00: NORMAL
                                          ;80: INVERSE VIDIO

0ABF 0000      MULJMP: DW       0         ;0000: NORMAL
                                          ;ADDR: ADDRESS OF ROUTINE TO CAL
                                          ;      BEFORE OUTPUTING.

0AC1 0000      IMULJM: DW       0000H     ;SAME AS MULJMP EXCEPT FOR ECHO

                   ;THE FOLLOWING 4 PARAMETERS DETERMINE DISPLAY ACTION
                   ;  WHEN THE CURSOR IS MOVED OFF SCREEN

0AC3 18        OFFTOP: DB       24D       ; 1: SCROLL DOWN
                                          ;24: WRAPAROUND TO BOTTOM

0AC4 18        OFFBOT: DB       24D       ; 1: WRAPAROUND TO TOP LINE
                                          ;24: SCROLL UP

0AC5 01        OFFLFT: DB       1         ; 1: OVERWRITE 1ST CHAR ON LINE
                                          ;WIDTH: WRAPAROUND TO END OF SAM
                                          ;1-WIDTH: BACK UP TO PREVIOUS LI

0AC6 00        OFFRT:  DB       0         ; 0: START NEXT LINE
                                          ; 1: WRAP AROUND TO SAME LINE
                                          ;WIDTH: OVERWRITE LAST CHAR ON L
```

```
0AC7 D10A       CONAT:  DW      CONTAB  ;OUTPUT CONTROL CHAR TABLE

0AC9            TEMP:   DS      5       ;TEMPORARY VARIABLE STORAGE


                ;TABLE CONTAB
                ;       CONTROL CHARACTER FUNCTIONS
                ;       FORMAT IS SETS OF THREE BYTES
                ;         BYTE N   = CHARACTER TO RECOGNIZE
                ;         BYTE N+1 = LOW BYTE OF ADDRESS
                ;         BYTE N+2 = HIGH BYTE OF ADDRESS
                ;       ADDRESS IS OF THE ROUTINE TO CALL TO EXECUTE
                ;         THE REQUIRED FUNCTION.
                ;       LAST ENTRY MUST BE NULL (00HEX).

0ACE 0D         CONBLK: DB      CR              ;SPECIAL ECHO FOR BLOCK
0ACF B008               DW      CRLF            ;  MODE

0AD1 0D         CONTAB: DB      CR              ;CARRIAGE RETURN
0AD2 CB08               DW      CARRET
0AD4 0A                 DB      LF              ;LINE FEED
0AD5 B208               DW      LNFEED
0AD7 09                 DB      HT              ;HORIZONTAL TAB
0AD8 A708               DW      HORTAB
0ADA 8C                 DB      FF+PARON        ;FORM FEED
0ADB BF08               DW      FORMFD
0ADD 08                 DB      BS              ;BACK SPACE
0ADE 9F08               DW      BACKSP
0AE0 8B                 DB      VT+PARON        ;VERTICAL TAB
0AE1 B608               DW      VERTAB
0AE3 1B                 DB      ESC             ;ESCAPE
0AE4 4808               DW      ESCCHR
0AE6 07                 DB      BEL             ;BELL DING
0AE7 BD09               DW      BELL            ;  FLASHER
0AE9 02                 DB      STX             ;SET CONFIG SWITCHES
0AEA 9B09               DW      CONSET
0AEC 0B                 DB      VT              ;UPLINE
0AED D108               DW      UPLINE
0AEF 0C                 DB      FF              ;FORESPACE
0AF0 D508               DW      FORSPC
0AF2 BD                 DB      '='+PARON       ;DIRECT CURSOR ADDRESSIN
0AF3 D908               DW      DCACOM
0AF5 9F                 DB      US+PARON        ;FORGROUND FOLLOWS
0AF6 DE09               DW      OUTFOR
0AF8 99                 DB      EM+PARON        ;BACKGROUND FOLLOWS
0AF9 E309               DW      OUTBAK
0AFB 1A                 DB      XUB             ;CLEAR SCREEN
0AFC F009               DW      CLEAR
0AFE 1E                 DB      RS              ;HOME CURSOR
0AFF E909               DW      HOMEIT
0B01 05                 DB      ENQ             ;HERE IS MESAGE
0B02 6808               DW      HEREIS
0B04 16                 DB      SYN             ;INSERT CHARACTER
0B05 1304               DW      PUTSPC
0B07 18                 DB      CAN             ;DELETE CHARACTER
0B08 3E09               DW      DECHAR
0B0A 17                 DB      ETB             ;INSERT LINE
```

```
0B0B 8F03                        DW      SCRLDN
0B0D 15                 DB       NAK              ;DELETE LINE
0B0E 4503                        DW      DELINE
0B10 0F                 DB       SI               ;LOCK KEYBOARD
0B11 6108                        DW      LCKKB
0B13 0E                 DB       SO               ;UNLOCK KEYBOARD
0B14 9808                        DW      UNLKKB
0B16 C9                 DB       'I'+PARON        ;STRING INSERT
0B17 2709                        DW      PUTSTR
                  ;END OF LIST MUST BE NUL <<<<<<<<<<<
0B19 00                 DB       NUL              ;NULL
0B1A 120A                        DW      ANULL


               ;
               ;CONTAB FOR HALF DUPLEX ECHOING
               ;

0B1C 0D        HDCON:   DB       CR      ;CARRIAGE RETURN
0B1D CB08                        DW      CARRET
0B1F 0A                 DB       LF               ;LINE FEED
0B20 B208                        DW      LNFEED
0B22 09                 DB       HT               ;HOR TAB
0B23 A708                        DW      HORTAB
0B25 00                 DB       NUL
0B26 120A                        DW      ANULL


               ;****************************************************
               ;
               ;        CONTROL CHARACTER DEFINITIONS
               ;
               ;****************************************************

0000                    NUL      EQU     00H      ;NULL
0001                    SOH      EQU     01H      ; ^A
0002                    STX      EQU     02H      ;
0003                    ETX      EQU     03H      ;
0004                    EOT      EQU     04H      ;
0005                    ENQ      EQU     05H      ;
0006                    ACK      EQU     06H      ;
0007                    BEL      EQU     07H      ;
0008                    BS       EQU     08H      ;BACK SPACE
0009                    HT       EQU     09H      ;HORIZONTAL TAB
000A                    LF       EQU     0AH      ;LINE FEED
000B                    VT       EQU     0BH      ;VERTICAL TAB
000C                    FF       EQU     0CH      ;FORM FEED
000D                    CR       EQU     0DH      ;CARRIAGE RE
000E                    SO       EQU     0EH
000F                    SI       EQU     0FH
0010                    DLE      EQU     10H
0011                    DC1      EQU     11H
0012                    DC2      EQU     12H
0013                    DC3      EQU     13H
0014                    DC4      EQU     14H
0015                    NAK      EQU     15H
0016                    SYN      EQU     16H
```

```
0017                    ETB     EQU     17H
0018                    CAN     EQU     18H
0019                    EM      EQU     19H
001A                    XUB     EQU     1AH     ;^X (SUB)
001B                    ESC     EQU     1BH
001C                    FS      EQU     1CH
001D                    GS      EQU     1DH
001E                    RS      EQU     1EH
001F                    US      EQU     1FH
007F                    DEL     EQU     7FH

                ;***************************************
                ;
                ;       BIT DEFINITIONS
                ;
                ;***************************************

0080                    PARON   EQU     80H     ;PARITY BIT
0040                    ALPHA   EQU     40H     ;ALPHA CHARACTER BIT
0020                    LC      EQU     20H     ;LOWERCASE = LC + ALPHA
001F                    CTRL    EQU     1FH     ;'X' AND CTRL = ^X

0000                    END
```