BUILD THE GT180 COLOR GRAPHICS BOARD

PART 2: HARDWARE

BY STEVE CIARCIA

A description of the components and the hardware schematic

Last month I introduced my new Professional Graphics Adapter (PGA)–like GT180 graphics adapter for the SB180 and SB180FX computers and discussed the basics of CRT controllers. This month I'll continue with a description of the major VLSI components. I'll also include the hardware schematic of the GT180 itself. While presented as an SB180 XBUS peripheral device, the GT180's combination of high-level integrated circuitry, parallel port interfacing, and Borland International's Graphix Toolbox standard software facilitates its use in many other "ported" graphics applications (see photo 1).

FEW CHIPS ARE USED

The key to the GT180's design is a recently announced CMOS VLSI graphics processor, the Hitachi HD63484 ACRTC (advanced CRT controller), and two companion chips: HD63485 GMIC (graphic memory interface controller) and HD63486 GVAC (graphic video attributes controller). These are supplemented by a highly integrated CMOS palette D/A converter—the Brooktree BT450.

The block diagram in figure 1 shows that these chips, in combination with a 512K-byte frame buffer (sixteen 64K by 4-bit dynamic RAM chips), need little extra support to achieve an entire graphics drawing and display subsystem. The 512K-byte frame buffer can hold three or four screens as well as a library of graphics objects like windows, fonts, and icons. A standard shift register allows input from an IBM PC–compatible keyboard. Finally, a few TTL chips and a PAL (programmable array logic) generate the control signals to interface the ACRTC, keyboard shift register, and palette D/A converter to the SB180's XBUS. The GT180 can directly connect to a number of standard CRTs—both TTL and analog—like the Princeton Graphics SR-12 and SR-12P.

THE HD63484 ACRTC

The ACRTC, an incredibly powerful CMOS chip containing more than 100,000 transistors, needs to perform a number of tasks. First, it must generate the CRT timing signals, including HSYNC and VSYNC. Next, it must perform display-control tasks like scanning the frame buffer while keeping track of cursors, split screens, and windows. Finally, it must assist the application pro-

(continued)

Steve Ciarcia (pronounced "see-ARE-see-ah") is an electronics engineer and computer consultant with experience in process control, digital design, nuclear instrumentation, and product development. The author of several books on electronics, he can be reached at P.O. Box 582, Glastonbury, CT 06033.

grammer by providing built-in drawing algorithms and commands. The ACRTC integrates all these functions on a single device.

Examining the ACRTC block diagram (see figure 2) reveals that it contains three separate microprocessors (16-bit CPUs)—one CPU for each of the primary tasks: CRT timing generation, display control, and drawing. The ACRTC contains more than 200 bytes of command and control registers used in conjunction with 38 high-level drawing commands (see figure 3). The registers and commands will be explained and put to good use in next month's software description.

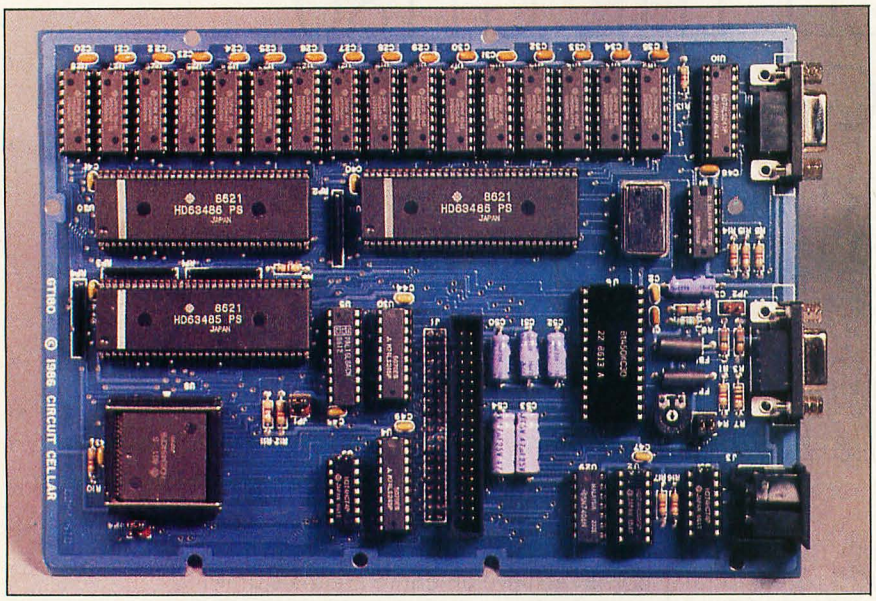


Photo 1: *The* GT180 *graphics adapter. The various connectors along the right-hand side are (from top to bottom): a* DB-9 *connector for analog* RGB *output, another* DB-9 *connector for* RGBI TTL *output, and a* DIN *connector for an* IBM PC*–compatible keyboard.*

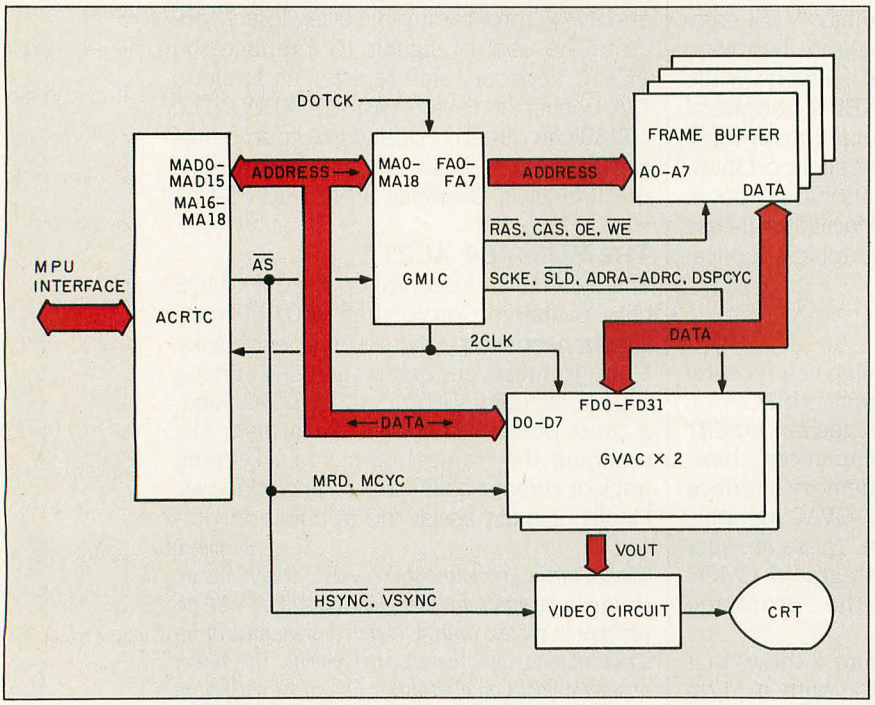


Figure 1: A *block diagram of the* GT180.

GMIC AND GVAC

It is possible to use the ACRTC by itself in a graphics design. However, for a design exploiting all the ACRTC features, the amount of TTL support logic required is large. Furthermore, due to the complexity and high speed required of the support logic, such a circuit would be difficult to design and debug.

For example, during a drawing operation, the ACRTC treats the frame buffer as a simple linear sequence of 16-bit words, just like a regular 16-bit CPU. However, during a display refresh, the frame buffer needs to read 64 bits at one time to keep up with the high speed requirements of a fast CRT. Thus, the frame buffer's address-decode and data-buffer circuits must be dynamically reconfigured for 16- and 64-bit operations. On the display-controller side, ACRTC features (like windows, horizontal smooth scroll, and zooming) require circuits to interact with and control the 64-bit high-speed video shift registers.

To help the designer eliminate much of the TTL support logic and lessen the design complexity, I have used the HD63485 GMIC and HD63486 GVAC support ICs. The GMIC manages the multiplexed addressing of the frame buffer's DRAMs and generates the required control signals like $\overline{RAS}$, $\overline{CAS}$, and $\overline{WE}$. Each GVAC includes two 16-bit shifters, so two GVACs provide a 64-bit shift register. The GVACs also map ACRTC 16-bit drawing accesses to the 64-bit frame-buffer bus. The GMIC and GVAC are manufactured using a special process called Hi-BiCMOS, which combines the speed and drive of bipolar (64 megahertz, IOL=24 milliamperes) with the low power operation of MOS (less than 1 watt).

This subsystem (ACRTC, GMIC, GVAC, and frame buffer) can directly interface to a digital RGBI (red-green-blue-intensity) monitor. The addition of the BT450 palette D/A converter allows the connection of an analog

CRT for finer color selection. In the past, the designer would have to build a palette D/A converter by combining a static RAM palette with three separate video D/A converters—a difficult and expensive task. Now, the BT450 combines the 16- by 12-bit palette RAM and three separate 4-bit 30-MHz to 70-MHz D/A converters. Watch for the availability of new palette D/A converter chips like the BT450 to cause a trend to analog displays, even for low-cost computers.

THE GT180 HARDWARE

The GT180 board consists of a number of distinct sections: the graphics display and control section (the ACRTC, GMIC, and GVACs), monitor interface (the palette D/A converter and TTL buffers), SB180 XBUS interface circuitry, and the IBM PC keyboard adapter. The complete GT180 schematic is shown in figure 4.

The interface between the ACRTC, PC keyboard register, palette D/A con-

(continued)

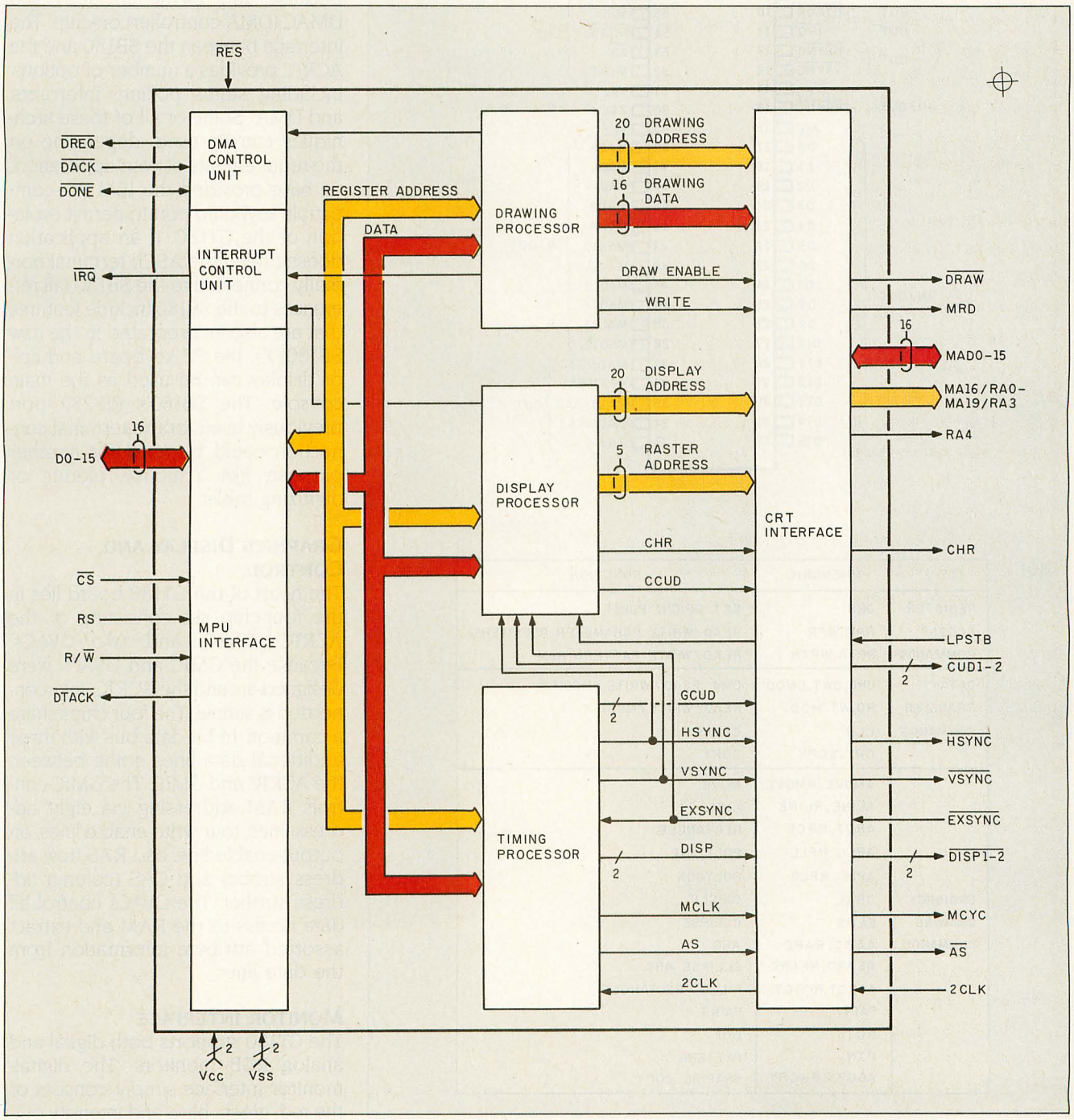
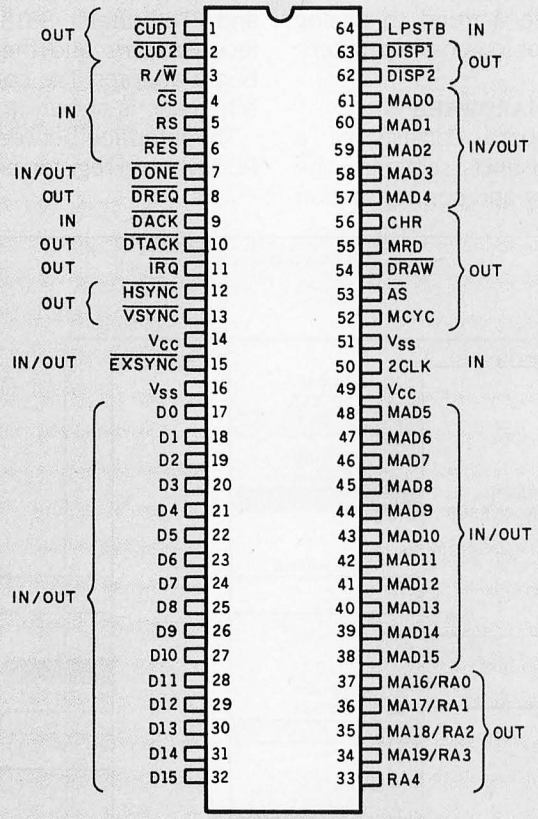


Figure 2: A *block diagram of the* ACRTC *chip.*

(a)



(b)

TYPE	MNEMONIC	FUNCTION
REGISTER-ACCESS COMMANDS	ORG	SET ORIGIN POINT
	RPR,WPR	READ/WRITE PARAMETER REGISTERS
	RPTN,WPTN	READ/WRITE PATTERN RAM
DATA-TRANSFER COMMANDS	DRD,DWT,DMOD	DMA READ/WRITE/MODIFY
	RD,WT,MOD	READ/WRITE/MODIFY
	CLR	CLEAR
	CPY,SCPY	COPY
GRAPHIC-DRAWING COMMANDS	AMOVE,RMOVE	MOVE
	ALINE,RLINE	LINE
	ARCT,RRCT	RECTANGLE
	APLL,RPLL	POLYLINE
	APLG,RPLG	POLYGON
	CRCL	CIRCLE
	ELPS	ELLIPSE
	AARC,RARC	ARC
	AEARC,REARC	ELLIPSE ARC
	AFRCT,RFRCT	FILLED RECTANGLE
	PAINT	PAINT
	DOT	DOT
	PTN	PATTERN
	AGCPY,RGCPY	GRAPHIC COPY

Figure 3: (*a*) A *pin-out diagram of the* ACRTC; (*b*) *the* ACRTC's *command list.*

verter, and the SB180 (or SB180FX) XBUS expansion bus requires some "glue" logic. Fortunately, a PAL is easily used for a major portion of this circuitry. Besides decoding port addresses for the ACRTC, D/A converter, and keyboard port, the PAL inverts the polarity of the R/W signal during direct memory access. This is required because the 64180 includes the DMAC (DMA controller) on-chip. The interface between the SB180 and the ACRTC provides a number of options, including status polling, interrupts, and DMA. Some or all of these techniques can be used, depending on the requirements of your application.

I have provided the IBM PC–compatible keyboard port to permit evolution of the GT180. If an application doesn't need the ASCII terminal normally connected to the SB180 (all references to the SB180 include features that are also incorporated in the new SB180FX), the PC keyboard and color display can be used as the main console. The SB180's RS-232 port previously used for the terminal connection could be used for another purpose, like a mouse, plotter, or digitizing tablet.

Graphics Display and Control

The heart of the GT180 board lies in the four-chip set consisting of the ACRTC, GMIC, and two GVACs. Because the GMIC and GVACs were designed around the ACRTC, interconnection is simple. The four chips share a common 16-bit data bus with three additional data lines going between the ACRTC and GMIC. The GMIC controls RAM addressing via eight address lines, four write-enable lines, an output-enable line, and RAS (row address strobe) and CAS (column address strobe). The GVACs control all data access to the RAM and extract assorted attribute information from the data lines.

Monitor Interface

The GT180 supports both digital and analog RGB monitors. The digital-monitor interface simply consists of the red, green, blue, and intensity outputs of the GVACs plus HSYNC and VSYNC from the ACRTC buffered

through a 74S241 (IC10). Jumper JP3 allows selectable inversion of both the HSYNC and VSYNC signals. If the jumpers are installed, IC1 inverts the signals.

The analog RGB section is a bit more involved. The RGBI signals coming from the GVACs are fed into the palette D/A converter (IC6). These signals are used to reference the color look-up table, which provides the color information to the D/A converters. The RGB outputs of IC6 are capable of driving the monitor directly with 75-ohm termination (R1, R2, and R3). The SYNC signal is made up of HSYNC combined with VSYNC via an exclusive-NOR operation and buffered through IC10. Finally, the MODE signal controls whether the Princeton SR-12P monitor runs in 400-line or 480-line mode. You should remove JP2 for 480-line operation.

Noise is a critical factor to consider in the design of video circuitry. To minimize the noise imposed on the palette D/A converter through the power supply, ferrite beads are used to isolate power for IC6 from the rest of the board. The beads filter out any high-frequency noise created elsewhere on the board that could result in garbage in the final picture.

Interface Circuitry

The GT180 is designed to be plugged into the 40-pin XBUS on either an SB180 or the SB180FX. The SB180 doesn't provide any output buffers, so line loading is an important consideration when producing a peripheral (the 180 XBUS is buffered). The GT180 uses the data, address, and bus-control lines in numerous places on the board, so two 74LS245s (IC4 and IC30) have been added to buffer the majority of interface lines. The direction of data flow through the data-bus buffer (IC4) is controlled by an $\overline{RD}$ line from the SB180 and an enable output from the PAL (IC5).

We need some additional circuitry to decode I/O ports for the devices on the board. The ACRTC needs two ports, the palette D/A converter needs two ports, and the keyboard needs at least one port. It would also be nice to use jumpers to change the location of the GT180 I/O ports within the expansion space to avoid conflicts with other peripherals plugged into the same system. Ordinarily, such decode circuitry would require many TTL gates occupying a good deal of board space. Instead, this is a perfect application for a PAL.

PAL Decreases Interface Complexity

A PAL is a device similar to a PROM in that it can be custom-programmed for an application by the designer. The GT180 uses a 16L8. The "16" means the device has 16 possible inputs, the "8" means it has 8 possible outputs, and the "L" in the middle indicates that the outputs are active-low.

A PAL gains more flexibility from the fact that most of the outputs can also be defined as inputs. However, since there are only 18 pins on the chip for I/O, the sum of inputs plus outputs cannot exceed 18. You can, for example, have 14 inputs and 4 outputs. The GT180's PAL uses 11 inputs and 7 outputs.

Inside the chip is a series of AND and OR gates. The inputs to the chip are directed to the inputs of the AND gates, and the outputs of the AND gates are directed to the inputs of the OR gates. The OR gates' outputs are sent to a buffered inverter and then to the chip's outputs. This arrangement of gates makes implementing standard sum-of-products logic equations easy.

The programmable array comes into play when you decide how the inputs to the chip are to be connected to the inputs of the gates within the chip. An array of fuse links is used for this. The chip's input pins are connected to the columns of the array, and the inputs to the gates are connected to the rows. At each row-column intersection is a fuse connecting the row to the column. On an unprogrammed chip, all the fuses are intact, so all the chip's inputs are connected to all the logic inputs. In the process of programming the chip, all unwanted fuse links are literally blown out by the PAL programmer, leaving just the links that connect the desired chip input to a particular gate input.

Most of the PAL inputs are standard bus signals necessary for I/O port decoding. Two select lines are also defined that let you select one of four base locations for the GT180 I/O ports. JP1 is used to set these select lines.

The $\overline{DMA}$ input is not a standard bus signal. When the ACRTC generates a DMA request, it is latched by IC3 and sent to the $\overline{DMA}$ input on the PAL (as well as to the $\overline{DREQ0}$ input on the SB180). The ACRTC requires that R/W be inverted during a DMA transfer, so this is taken care of within the PAL.

There are also seven outputs defined on the PAL. The $\overline{BDSEL}$ line goes active anytime the board is referenced and is used to enable the data-bus buffer (IC4). $\overline{ACRCS}$ provides chip select to the ACRTC, and ACR/$\overline{W}$ provides the R/W signal to the ACRTC. $\overline{ACDACK}$ is the DMA acknowledge signal to the ACRTC. $\overline{DACCS}$ is the chip select for the palette D/A converter, while $\overline{KEYCS}$ is the chip select for the keyboard adapter.

The last PAL output is labeled $\overline{EXSEL}$. A little history is necessary to explain this signal. When I designed the COMM180 board (see the December 1985 Circuit Cellar), I hadn't considered that I would be designing something like the GT180. Consequently, I made the assumption that only one expansion board would ever be plugged into the SB180 at any one time. Therefore, the data-bus buffer is enabled whenever $\overline{EXSEL2}$ from the SB180 goes active, even though there are gaps in the expansion space that the board uses. When I sat down to look at doing a graphics board, I realized that if someone wanted to use the GT180 with a hard disk on an SB180, I had to make some allowance for plugging both a COMM180 and a GT180 into an SB180. (The SB180FX incorporates both the floppy disk and SCSI controllers on the same board and therefore does not require the COMM180 to add a hard disk. Refer to the What's New section in this issue for further details.)

The GT180 uses eight I/O ports. The base address for the ports is jumper-selectable and can be set for either E0, E8, F0, or F8 hexadecimal. (For the remainder of the article, all port

(continued)

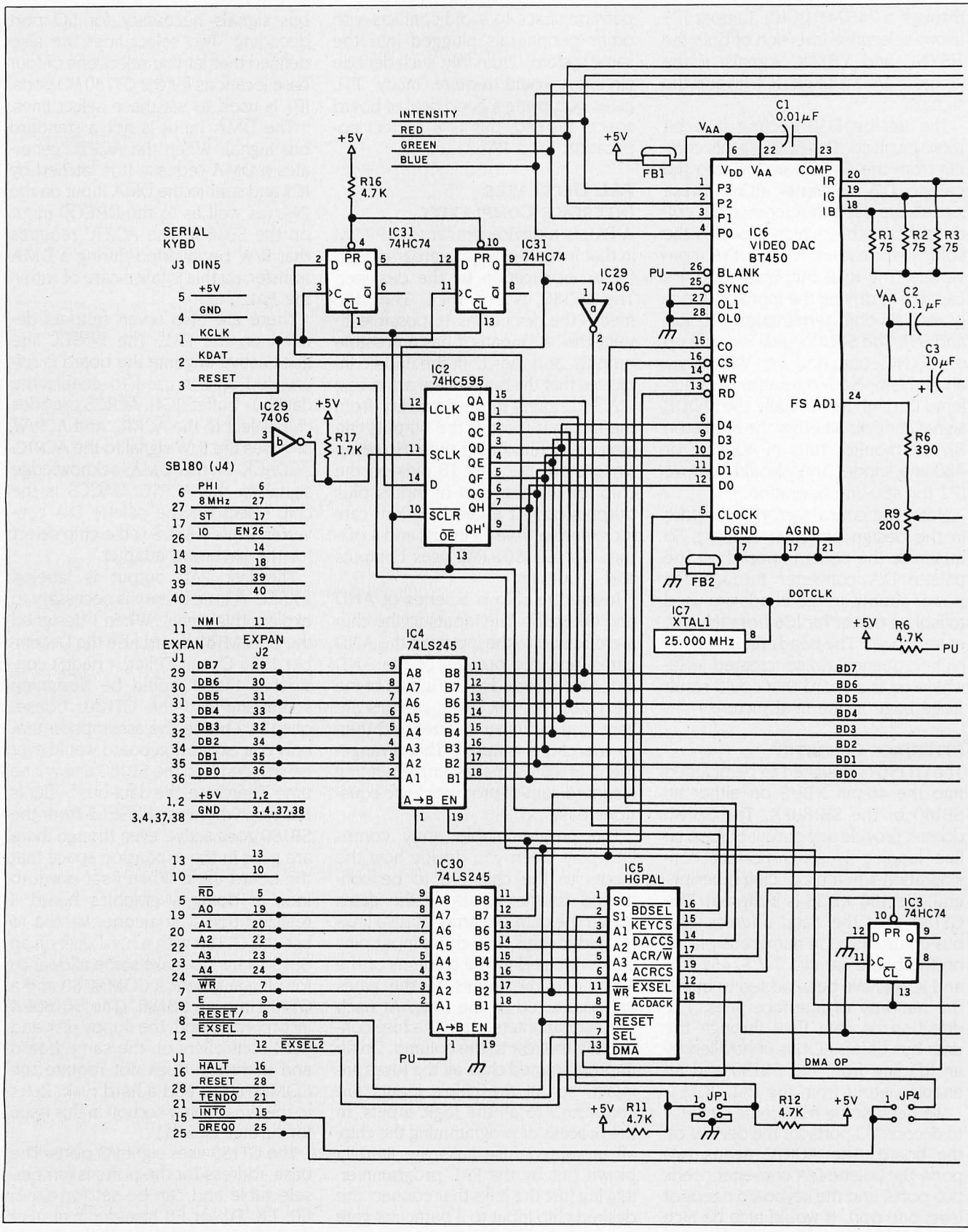
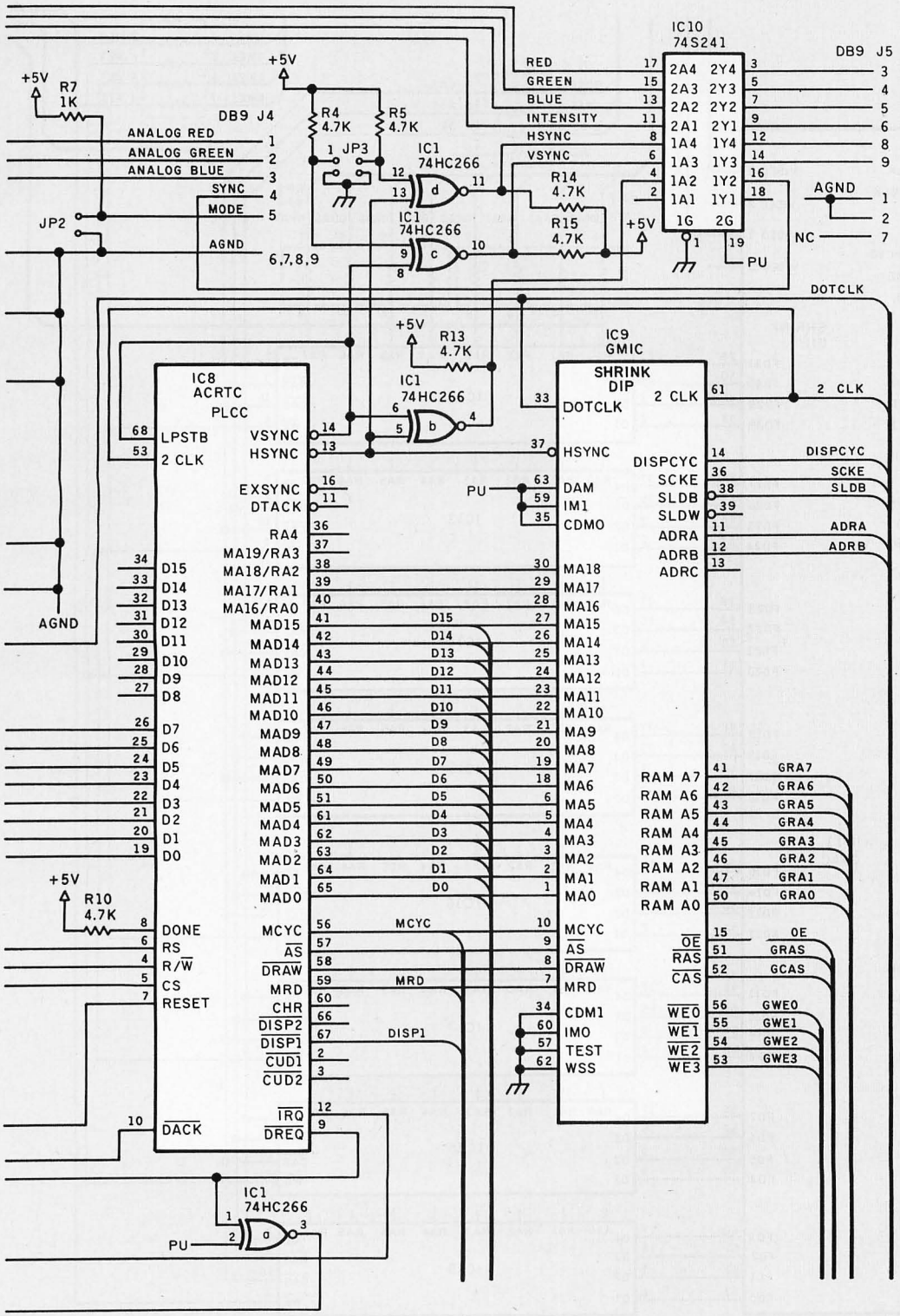


Figure 4: *The schematic of the* GT180.


IC10
74S241
DB9 J5
RED
GREEN
BLUE
INTENSITY
HSYNC
VSYNC
2A4 2Y4
2A3 2Y3
2A2 2Y2
2A1 2Y1
1A4 1Y4
1A3 1Y3
1A2 1Y2
1A1 1Y1
1G 2G
AGND
NC
PU
+5V
R7
1K
DB9 J4
ANALOG RED
ANALOG GREEN
ANALOG BLUE
SYNC
MODE
AGND
JP2
R4
4.7K
R5
4.7K
JP3
IC1
74HC266
R14
4.7K
R15
4.7K
6,7,8,9
DOTCLK
R13
4.7K
IC8
ACRTC
PLCC
IC9
GMIC
SHRINK
DIP
LPSTB
2 CLK
VSYNC
HSYNC
EXSYNC
DTACK
RA4
MA19/RA3
MA18/RA2
MA17/RA1
MA16/RA0
MAD15
MAD14
MAD13
MAD12
MAD11
MAD10
MAD9
MAD8
MAD7
MAD6
MAD5
MAD4
MAD3
MAD2
MAD1
MAD0
D15
D14
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
DOTCLK
HSYNC
DAM
IM1
CDMO
MA18
MA17
MA16
MA15
MA14
MA13
MA12
MA11
MA10
MA9
MA8
MA7
MA6
MA5
MA4
MA3
MA2
MA1
MA0
2 CLK
DISPCYC
SCKE
SLDB
SLDW
ADRA
ADRB
ADRC
RAM A7
RAM A6
RAM A5
RAM A4
RAM A3
RAM A2
RAM A1
RAM A0
GRA7
GRA6
GRA5
GRA4
GRA3
GRA2
GRA1
GRA0
OE
RAS
CAS
GRAS
GCAS
WE0
WE1
WE2
WE3
GWE0
GWE1
GWE2
GWE3
CDM1
IMO
TEST
WSS
R10
4.7K
DONE
RS
R/W
CS
RESET
MCYC
AS
DRAW
MRD
CHR
DISP2
DISP1
CUD1
CUD2
IRQ
DREQ
DACK
MCYC
MRD
DISP1
PU

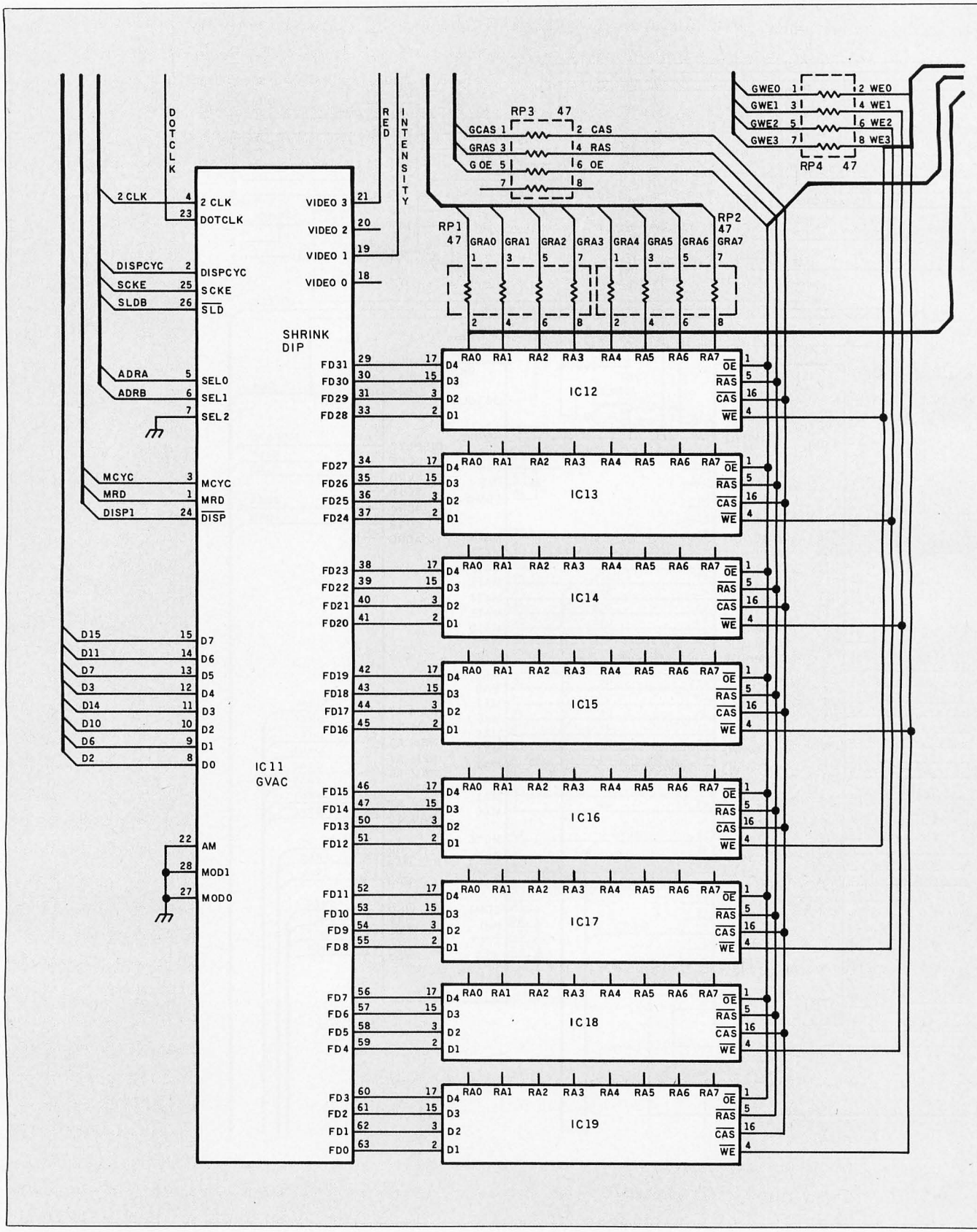
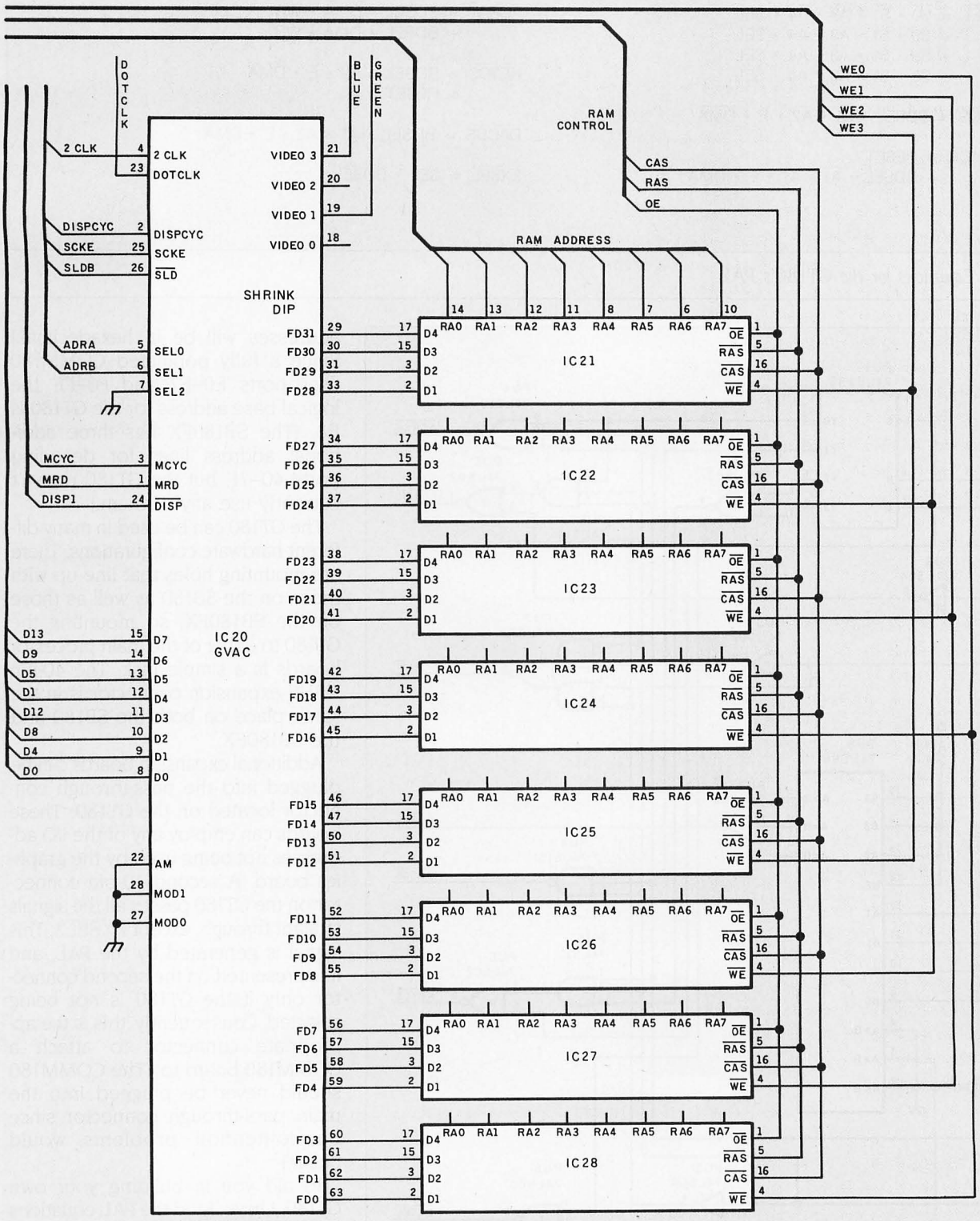


Figure 4 *continued.*


DOTCLK
BLUE
GREEN
RAM CONTROL
WE0
WE1
WE2
WE3
CAS
RAS
OE
RAM ADDRESS
2 CLK
DOTCLK
VIDEO 3
VIDEO 2
VIDEO 1
VIDEO 0
DISPCYC
SCKE
SLD
SHRINK DIP
ADRA
ADRB
SEL0
SEL1
SEL2
MCYC
MRD
DISP
D13
D9
D5
D1
D12
D8
D4
D0
D7
D6
D5
D4
D3
D2
D1
D0
IC20 GVAC
AM
MOD1
MOD0
FD31
FD30
FD29
FD28
FD27
FD26
FD25
FD24
FD23
FD22
FD21
FD20
FD19
FD18
FD17
FD16
FD15
FD14
FD13
FD12
FD11
FD10
FD9
FD8
FD7
FD6
FD5
FD4
FD3
FD2
FD1
FD0
IC21
IC22
IC23
IC24
IC25
IC26
IC27
IC28
RA0 RA1 RA2 RA3 RA4 RA5 RA6 RA7
D4 D3 D2 D1
OE RAS CAS WE

$$\overline{BDSEL} = \overline{S0} * \overline{S1} * \overline{A3} * \overline{A4} * SEL + \overline{S0} * \overline{S1} * A3 * \overline{A4} * SEL + \overline{S0} * S1 * \overline{A3} * A4 * SEL + S0 * S1 * A3 * A4 * SEL$$

$$\overline{ACRCS} = BDSEL * \overline{A1} * \overline{A2} * E * \overline{DMA}$$

$$\overline{ACDACK} = RESET + BDSEL * \overline{A1} * \overline{A1} * E * DMA$$

$$ACR/\overline{W} = BDSEL * \overline{DMA} * \overline{WR} + BDSEL * DMA * \overline{WR}$$

$$\overline{KEYCS} = BDSEL * A2 * E * \overline{DMA} + RESET$$

$$\overline{DACCS} = BDSEL * A1 * \overline{A2} * E * \overline{DMA}$$

$$\overline{EXSEL} = SEL * \overline{BDSEL}$$

Figure 5: *Equations for the* GT180's PAL.

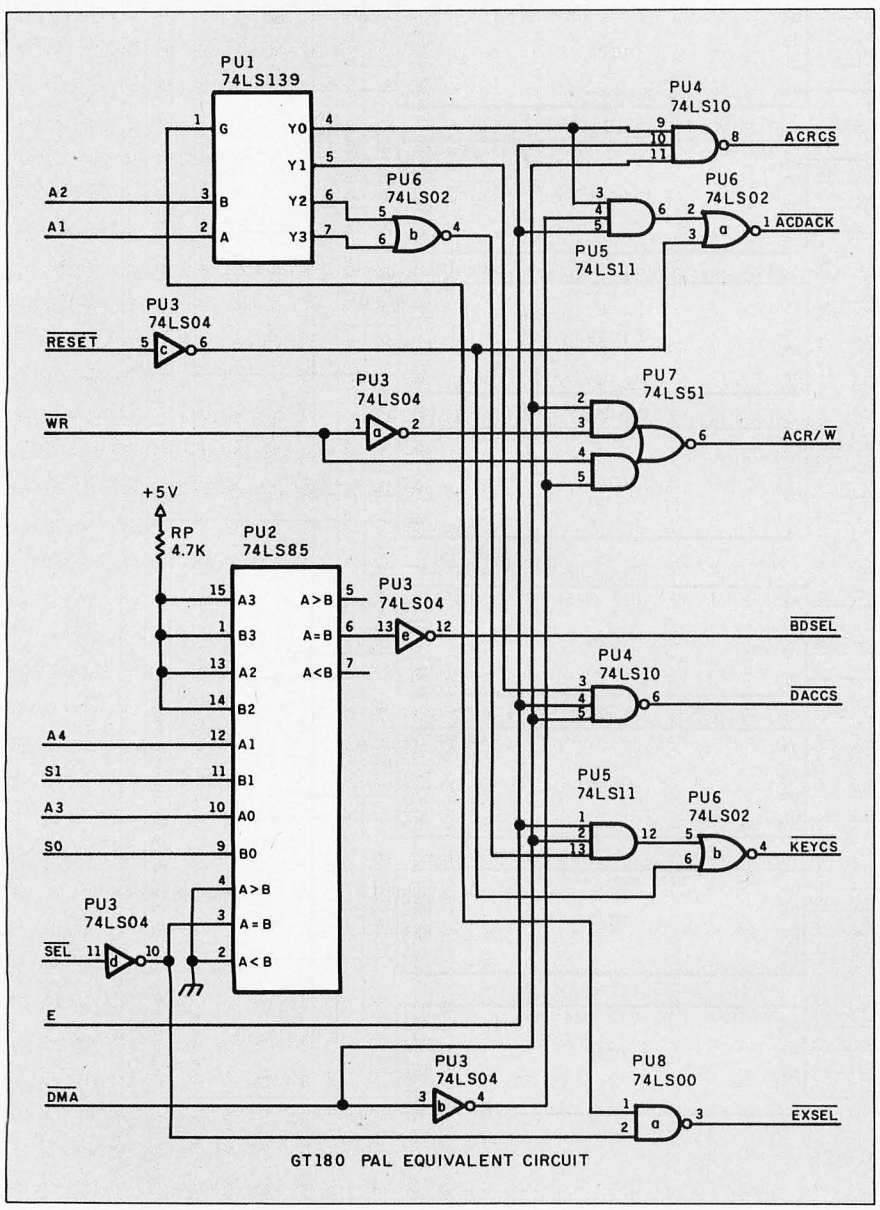


Figure 6: *The schematic of a discrete logic equivalent to the* GT180's PAL.

addresses will be in hexadecimal.) Since a fully populated COMM180 uses ports E0–E7 and F0–FF, the logical base address for the GT180 is E8. (The SB180FX has three additional address lines for decoding ports 40–7F, but the GT180 doesn't currently use any of them.)

The GT180 can be used in many different hardware configurations. There are mounting holes that line up with those on the SB180 as well as those on the SB180FX, so mounting the GT180 to either of the main processor boards is a simple task. The 40-pin XBUS expansion connector is in the same place on both the SB180 and the SB180FX.

Additional expansion boards can be plugged into the pass-through connector located on the GT180. These boards can employ any of the I/O addresses not being used by the graphics board. A second 40-pin connector on the GT180 passes all the signals straight through, except $\overline{EXSEL2}$. This signal is generated by the PAL, and it is presented on the second connector only if the GT180 is not being selected. Consequently, this is the appropriate connector to attach a COMM180 board to. (The COMM180 should never be plugged into the main pass-through connector since bus-contention problems would occur.)

To aid you in building your own GT180, I have listed the PAL equations for IC5 in figure 5. Also, figure 6 shows the schematic of a discrete component equivalent to the PAL if you want to build the GT180 but do not have the facility to program a PAL. When

you are done, you are welcome to download utilities and other graphics software that I will post on BYTEnet, BIX, and the Circuit Cellar BBS to aid in checking out your handiwork.

The Keyboard Adapter

The IBM PC keyboard uses a synchronous serial interface to talk to the host computer. While the typical IBM connection allows data flow to and from the keyboard, the circuit used on the GT180 only reads from the keyboard. This greatly simplifies the circuit.

Five lines connect the keyboard to the computer. Two are for +5 volts and ground, one is a power-on reset signal that initializes the keyboard, and the last two carry the data signal and clock signal. Data is sent from the keyboard using a 9-bit serial stream. You can think of the first bit as a start bit. The remaining 8 bits contain the code for the key being pressed (or released), with the low-order bit arriving first. The falling edge of the clock signal is used to indicate valid data is present on the data line. (Note: IBM PC keyboards are not ASCII-compatible.)

The core of the keyboard adapter is a 74HC595 (IC2) 8-bit shift register with a built-in three-state latch. It also has an unbuffered output, QH′, that reflects the state of the last bit in the shift register. This is normally used to cascade multiple registers. Data is fed into the D input directly from the keyboard. The clock is inverted since the chip assumes valid data on the rising edge of the clock pulse.

When the register is read or a reset is performed, IC2 and IC31 are cleared. Data arriving from the keyboard is shifted into IC2, and the data presently in the register is shifted up one bit (A to B, B to C, etc.). When all the data has been received from the keyboard, the shift register will contain the 8-bit keycode. The problem now is how to latch the data into the register's output buffers and tell the computer that a key was pressed.

Remember that an extra start bit was sent by the keyboard. This high bit will appear on the QH′ output as the seventh data bit being clocked into the shift register. Data from the QH′ output is clocked into IC31a one-half clock period after it appears on QH′. Half a clock period later, the last bit of the keycode is shifted into the shift register. Half a clock period after that, the data from IC31a is clocked into IC31b. The output of IC31b immediately strobes LCLK on IC2, causing the current state of the shift register to be latched into the output buffers. The output of IC31b is also connected to an open-collector inverter (IC29) and can be used as an interrupt to the SB180.

A subsequent read to the keyboard port will place the byte in the output buffers onto the data bus and will clear the shift register and IC31 so they are ready for the next keystroke. The data in the output buffers is not cleared and is available until the next byte is received from the keyboard.

Selecting a Color Monitor

Whether or not a particular monitor is suitable for use with the GT180 depends on two factors: the monitor's bandwidth and its physical interface.

The analog RGB interface has three analog signal lines—one each for red, green, and blue—with a peak voltage of +0.6 V for maximum brightness. There is also a negative composite sync at TTL levels that consists of the HSYNC and VSYNC signals combined. This type of interface is compatible with the SR-12P monitor mentioned before.

The digital TTL interface includes four TTL-level lines for the red, green, blue, and intensity lines. There are also two more TTL-level lines for separate HSYNC and VSYNC signals. This type of interface is compatible with most monitors designed for IBM's CGA board. Unfortunately, some monitors on the market don't fit the format of either of the two interfaces described above. You will probably have problems with such monitors, so it's best to avoid them altogether.

The GT180 is designed and presented as a 640- by 480-pixel resolution display controller. However, you can configure it by software to resolutions of 640 by 400, 640 by 350, 640 by 200, 320 by 200, etc. (all with 16 colors). Ultimately, the frequency of the crystal oscillator you use on your GT180 board controls the resolution you can expect and the monitor you can use. A 25-MHz oscillator will give you a PGA-like 640 by 480 or 640 by 400 resolution but requires a high-

(continued)

Photo 2: *This screen picture was produced using the* GT180 *and a* Princeton SR-12P *monitor.*

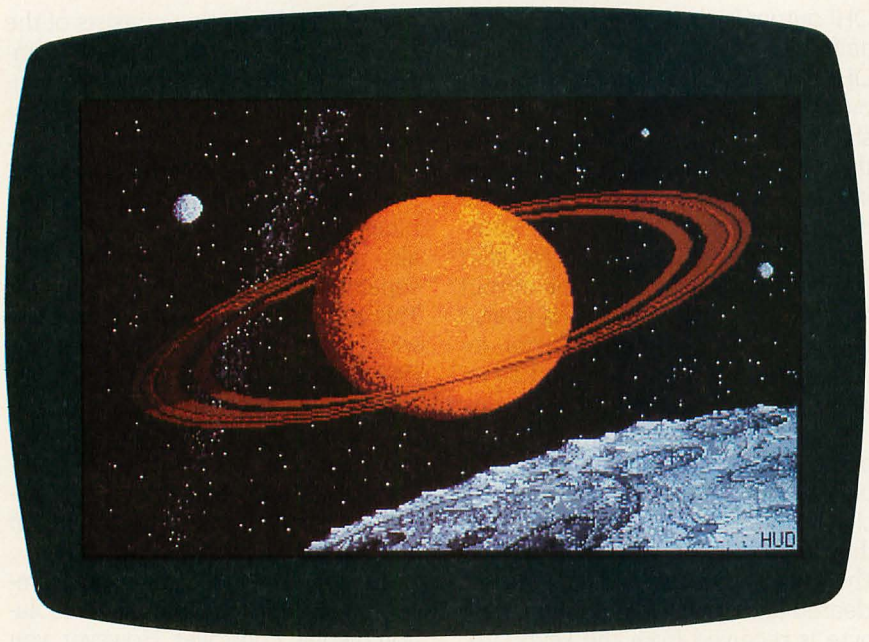


Photo 3: *Another screen picture using the* GT180 *and* SR-12P.

quality 30-MHz monitor like the SR-12P (see photos 2 and 3).

In order to use a less expensive EGA- or CGA-compatible color monitor intended for the IBM PC, you must use a lower oscillator frequency and give up some resolution. A CGA monitor, for example, has a 15-MHz bandwidth and will support only a resolution on the order of 640 by 200. The newer EGA monitors have a higher bandwidth and a resolution of 640 by 350 (an EGA monitor is RGBI and not analog). You will have to slow the GT180 down by a proportionate amount to use a slower scan rate. Use a crystal oscillator in the range of 12 to 13 MHz for a 640 by 200 monitor. (Note: Software utilities included with the GT180 allow use of the lower-resolution monitors.)

The flexibility offered by the interchangeable crystal oscillators coupled with the ability to set up the ACRTC's internal timing registers from software allows you to use most monitors on the market as long as you keep the bandwidth limitations in mind.

One tidy and relatively inexpensive way to get around most of the problems listed above is to use a monitor that can support a range of scan rates and either of the interfaces mentioned. No such animal? Well, NEC makes a MultiSync monitor that will adapt to whatever scan rate you are using and accepts either analog or digital RGB inputs.

CIRCUIT CELLAR FEEDBACK

This month's feedback begins on page 58.

NEXT MONTH

Part 3 looks at the GT180 Graphix Toolbox software. ■

Special thanks to Tom Cantrell, Ken Davidson, and Mike Weisert for their contributions to this project.

All screen pictures presented in this article were produced using the GT180 *with a Princeton* SR-12P *monitor. The bit-mapped pictures were originally composed on an* Atari 520ST *using* DEGAS *by Tom Hudson. They are reproduced and used here by permission.*

The following items are available from

The Micromint Inc.
25 Terrace Dr.
Vernon, CT 06066
(800) 635-3355
(203) 871-6170
Telex: 643331

1. GT180 graphics board: RGBI version less palette D/A converter. Comes with demo disk and user's manual.
board alone $395
board with Modula-2 and GT180 Graphix Toolbox $449

2. GT180 graphics board: RGBI and analog version with palette D/A converter. Comes with demo disk and user's manual.
board alone $449
board with Modula-2 and GT180 Graphix Toolbox $499

3. Borland International's Turbo Modula-2 GT180 Graphix Toolbox software for the SB180 and SB180FX computers, optimized for the 64180 processor. Supplied on 5¼-inch DS/DD SB180 format disks with 300-page manual.
SB180 Modula-2 alone $69
SB180 Modula-2 with Graphix Toolbox alone $89

4. SB180FX 5.75- by 8-inch single-board computer, accommodates 512K bytes of memory, two serial ports, three parallel ports, parallel printer port, floppy disk controller, SCSI controller, ROM monitor, 6-MHz 64180. Comes with ZRDOS, ZCPR3, hard disk BIOS, and user's manuals. Populated with 256K-byte memory, less 53C80 SCSI controller chip.
SB180FX board alone $409
SB180FX board with software $499
SB180FX board fully populated with 512K bytes, SCSI chip, and software $599
9.216-MHz 64180 processor upgrade (SB180FX only) $50

GMIC, GVAC, ACRTC, and palette D/A converter chip sets are available for experimenters who wish to hand-assemble the GT180. Call for price and availability information. Borland's Turbo Modula-2 is also available for most CP/M Z80 machines. Contact Echelon Inc., 885 North San Antonio Rd., Los Altos, CA 94022, (415) 948-3820. The SB180FX is hardware- and software-compatible with the SB180.

Surface delivery (U.S. and Canada only): add $5 for U.S., $10 for Canada. For delivery to Europe via U.S. airmail, add $20. Three-day air freight devlivery: add $8 for U.S. (UPS Blue), $25 for Canada (Purolator overnight), $45 for Europe (Federal Express), or $60 (Federal Express) for Asia and elsewhere in the world. Connecticut residents please add 7.5 percent sales tax.

There is an on-line Circuit Cellar bulletin board system that supports past and present projects. You are invited to call and exchange ideas and comments with other Circuit Cellar supporters. The 300/1200/2400-bps BBS is on-line 24 hours a day at (203) 871-1988.

To be included on the Circuit Cellar mailing list and receive periodic project updates and support materials, please circle 100 on the Reader Service inquiry card at the back of the magazine.