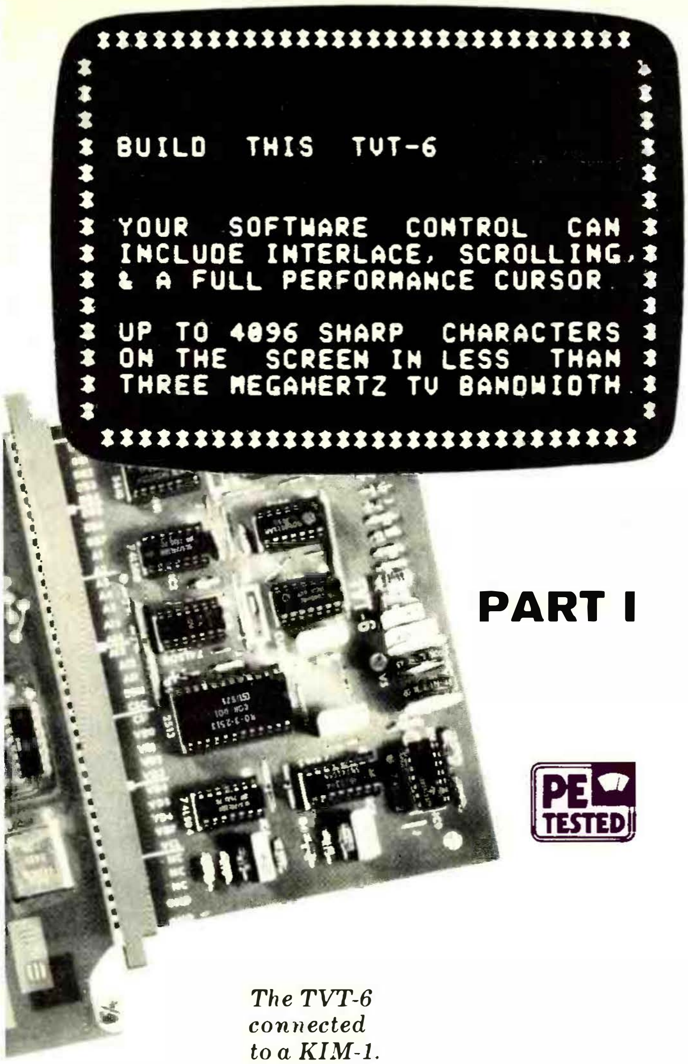


The TVT-6 connected to a KIM-1.

PART I

Build the TVT-6: A Low-Cost DIRECT VIDEO DISPLAY

PE TESTED

$35 microcomputer "add-on" provides:

- *User-selectable line lengths*
- *Scrolling*
- *Up to 4k on-screen characters with only 3-MHz bandwidth*

BY DON LANCASTER

Thanks to some software tricks, a simple and low-cost add-on circuit, and a new way to speed up a microprocessor, you can now build a video interface for your microcomputer for an investment of only $20 to $35. The TVT-6 video system described here permits the choice of virtually any format including 16/32 (16 lines of 32 characters), 16/64, or 32/64. It also features full editing capability and full-performance cursor.

In spite of its simplicity (10 low-cost IC's), the circuit employs a new approach to video processing that permits up to 4000 characters to be displayed on-screen within a 3-MHz bandwidth. Although the TVT-6 was designed for the 6502 microprocessor based KIM-1, software can be used to easily map into the JOLT, EBKA, or Ohio Scientific microcomputers. In addition, the TVT-6 can be adapted to other microprocessors, including the popular 6800, 8080, and Z80. It is easiest to use with 16-address-line systems that operate on a single 5-volt supply and 1-μs cycle time.

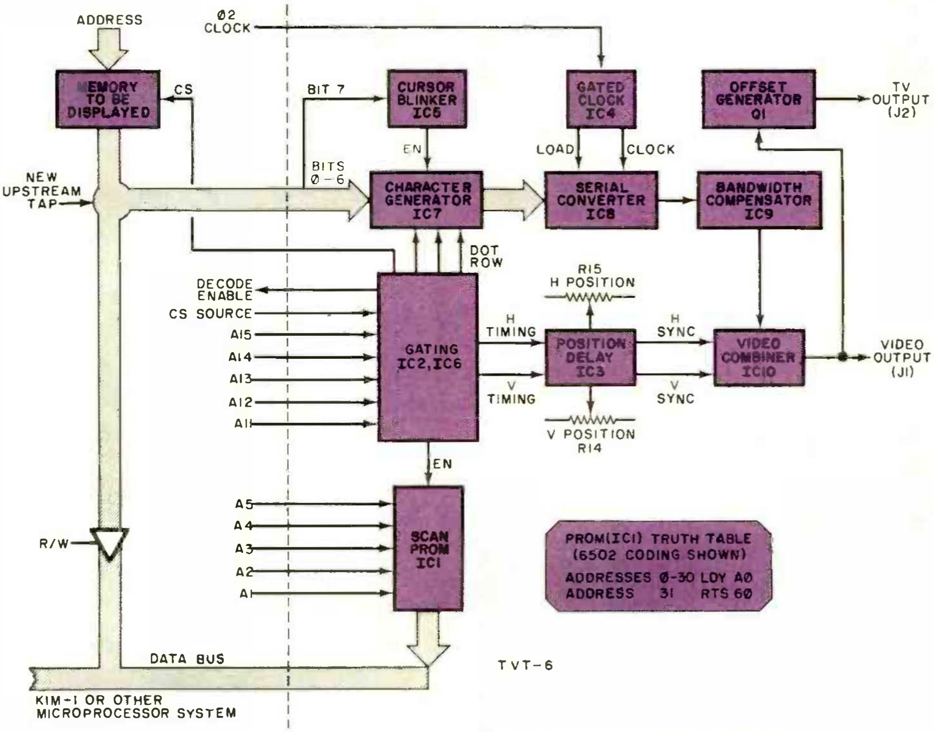


Fig. 1. TVT-6 block diagram and truth table for the PROM.

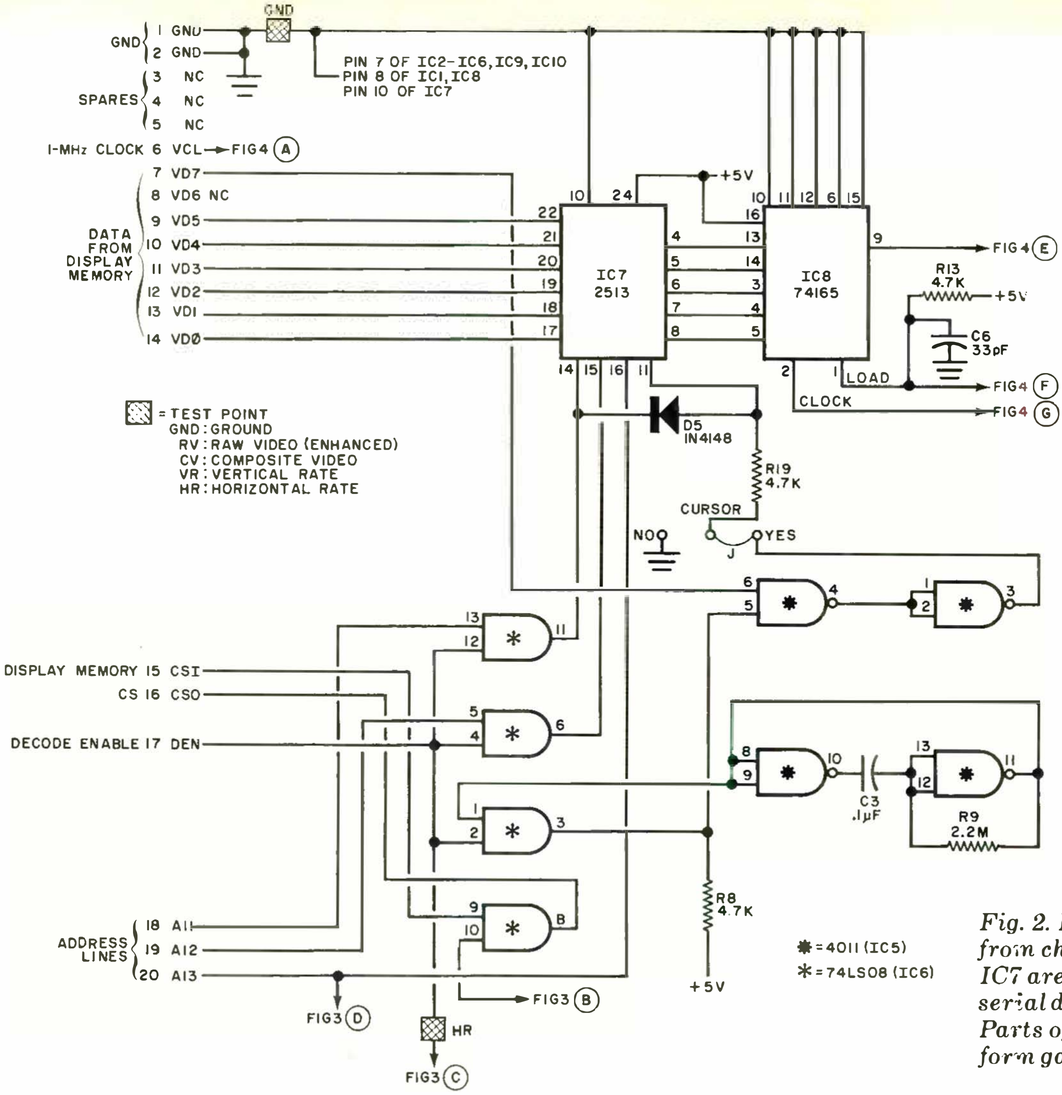


Fig. 2. Parallel outputs from character generator IC7 are converted into serial data in IC8. Parts of IC5 and IC6 form gating circuits.

Other systems will require software and microprogramming translation for their particular machine languages.

In this first of a two-part article, we will cover the hardware and construction details for the TVT-6. Next month, we will cover debugging, some useful software for the system, and provide instructions on how to couple the TVT-6 to other microprocessors.

Circuit Operation. A block diagram of the TVT-6, as used with the KIM-1 system, is shown in Fig. 1. The complete schematic diagram of the video system is shown in Figs. 2 through 4.

As shown in Fig. 1, bits ϕ through 6 from the "upstream tap" on the KIM display memory drive character generator *IC7* whose blanking and formatting are helped along by the AND gates in *IC6*. The cursor bit (bit 7) is stripped off the upstream tap and routed to cursor blinker *IC5*, which introduces a blinking cursor into the character generator's enable input.

The parallel outputs from *IC7* go to

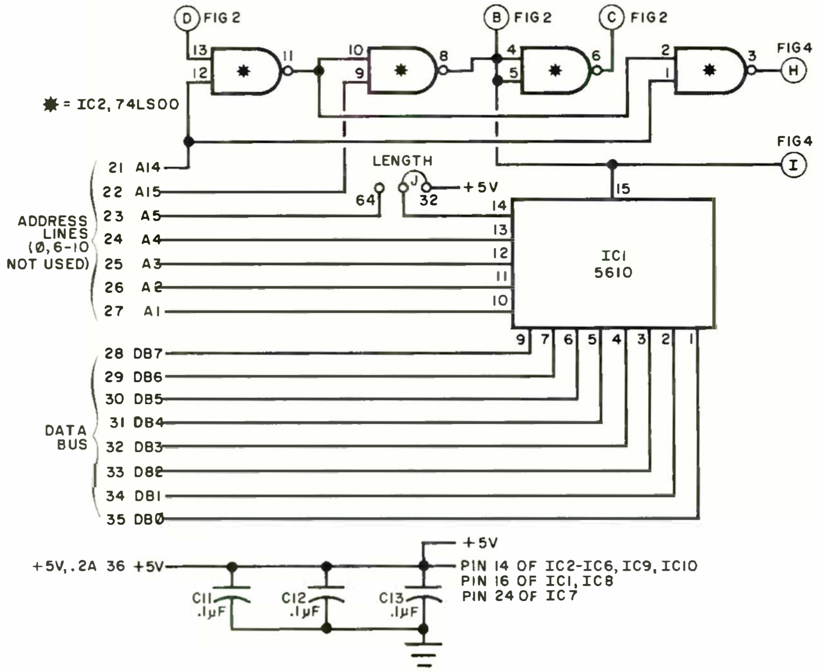


Fig. 3. New SCAN instruction uses PROM IC1, which also has the line length option in its circuit.

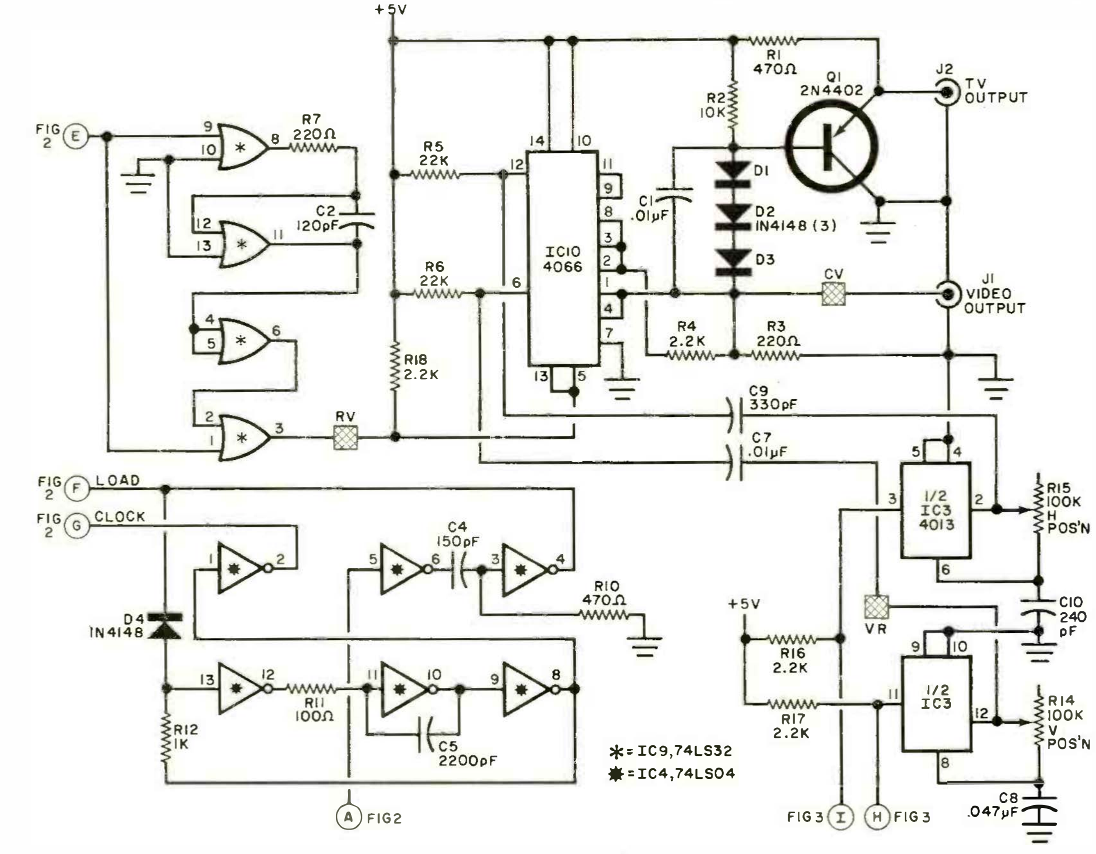


Fig. 4. Video combiner (IC10), offset generator (Q1) and sync delay circuits deliver video to TV. Gated clock (IC4) controls parallel-to-serial converter.

PARTS LIST

C1, C7—0.01-μF Mylar capacitor
C2—120-pF polystyrene capacitor
C3, C11, C12, C13—0.1-μF Mylar capacitor
C4—150-pF polystyrene capacitor
C5—2200-pF polystyrene or Mylar capacitor
C6—33-pF polystyrene capacitor
C8—0.047-μF Mylar capacitor
C9—330-pF polystyrene capacitor
C10—240-pF polystyrene capacitor
D1 through D5—IN4148 silicon diode
IC1—IM5610 32×8 PROM (or similar)
IC2—74LS00 quad tri-state NAND gate IC
IC3—4013 dual-D flip-flop IC
IC4—74LS04 hex inverter IC
IC5—4011 quad NAND gate IC
IC6—74LS08 quad AND gate IC
IC7—2513 character generator (must be single-supply type, such as General Instruments No. RO-3-2513)
IC8—74165 PISO shift register
IC9—74LS32 quad OR gate IC
IC10—4066 quad analog switch IC
J1, J2—Pc-mount phono jack (Molex No. 15-24-2181 or similar)
Q1—2N4402 or MPS6523 (Motorola) transistor
The following resistors are ¼ watt, 10% tolerance:
R1, R10—470 ohms
R2—10,000 ohms
R3,R7—220 ohms
R4,R16,R17,R18—2200 ohms
R5,R6—22,000 ohms
R8,R13,R19—4700 ohms
R9—2.2 megohms
R11—100 ohms
R12—1000 ohms
R14,R15—100,000-ohm pc-type (upright) potentiometer
Misc.—Sockets for IC's (seven 14-pin, two 16-pin, one 24-pin); 36-contact edge connector with 0.156" centers (Amphenol 225 or similar); solid hook-up wire for jumpers; insulated sleeving; test-point terminals (5); solder; etc.

Note: The following items are available from PAIA Electronics, Box 14359, Oklahoma City, OK 73114: No. PVI-1PC printed circuit board for $5.95; complete kit of all parts, No. PVI-1K, for $34.95 (specify blank or KIM-1 programmed *IC1*); KIM-1 coded cassette, with programs, No. PVI-1CC, for $5.00. All prices postpaid.

shift register *IC8*, where they are converted into a serial video signal. The clock and load commands for *IC8* come from gated oscillator *IC4*, which derives its signals from the microcomputer's clock. It is important that the correct clock phase be selected to permit the loading of *IC8* to occur when the output of the character generator is valid and settled. This is phase 2 in the KIM-1. (If you are using a different μP based computer, check this detail.)

The serial video from *IC8* goes to the TV Bandwidth Compensator in *IC9*, which predistorts the video by delaying the video output and OR'ing it against itself. This widens the vertical portions of all characters to generate clean and crisp characters that require minimum bandwidth. The amount of widening is determined by *C2* (Fig. 4). The optimum value of *C2* is obtained when the generated M or W in the video display just barely closes.

The vertical and horizontal timing signals from *IC2* in the gating circuit are delayed by *IC3*. The display positioning can be varied by potentiometers *R14* and *R15*. The vertical and horizontal sync signals are combined with the enhanced video from *IC9* into video combiner *IC10*. The output from *IC10*, available at *J1*, is composite video, with the sync tips at ground, black at 0.4 volt, and white at 1.6 volts. This output can be used to drive conventional video monitors and converted TV receivers. The video output from *IC10* is also fed to *Q1*, which is offset to deliver a +4-volt output for the white level. This output, available at *J2*, can be connected directly to the first video amplifier of most transformer-powered solid-state TV receivers (see box for details) without requiring biasing, coupling, or translation circuits.

Two options are provided with the TVT-6, both of which are jumper selected. The LENGTH option allows a choice of either 32 or 64 characters/line. The CURSOR option gives the choice of either no cursor or allows the cursor to be displayed under software control.

Construction. The actual-size etching and drilling guide for the printed circuit board used in the TVT-6 is shown in Fig. 5, along with the component-installation diagram. Start assembly by installing and soldering into place the 21 jumpers and test points. (Note that insulated sleeving must be used on two of the long jumpers.) Install the IC sockets, resistors, capacitors, diodes, jacks, and position controls *R14* and *R15*. Do not install the IC's at this time. The correct IC installation sequence and the waveforms to be observed will be discussed in Part 2 next month.

Computer Interface. Detailed in Table I are the requirements of each of the edge connector contacts on the TVT-6 and how to use each contact. Table I also contains the KIM-1 interface connection instructions. The interface consists of adding a new connector and making some add-on connections. One circuit board trace is cut on the KIM-1's pc board to permit an optional change-over switch (or jumper) to be added to the microcomputers. This permits KIM-1 to be used with or without the TVT-6.

General Operation. Since most of today's TVT circuits are used with a microprocessor or microcomputer, it is best to do as much of the display control as possible with the microprocessor and some software. What may not be obvious is that almost all of the timing in the system can also be done using the microprocessor. All this takes is a few dozen words of code.

The four key secrets of operation for the TVT-6 are:

1. Carefully choose how the address lines are defined for TVT operation.

2. Add a new instruction, which we call SCAN, to rapidly address 32 or 64 sequential memory locations.

3. Permanently connect an upstream

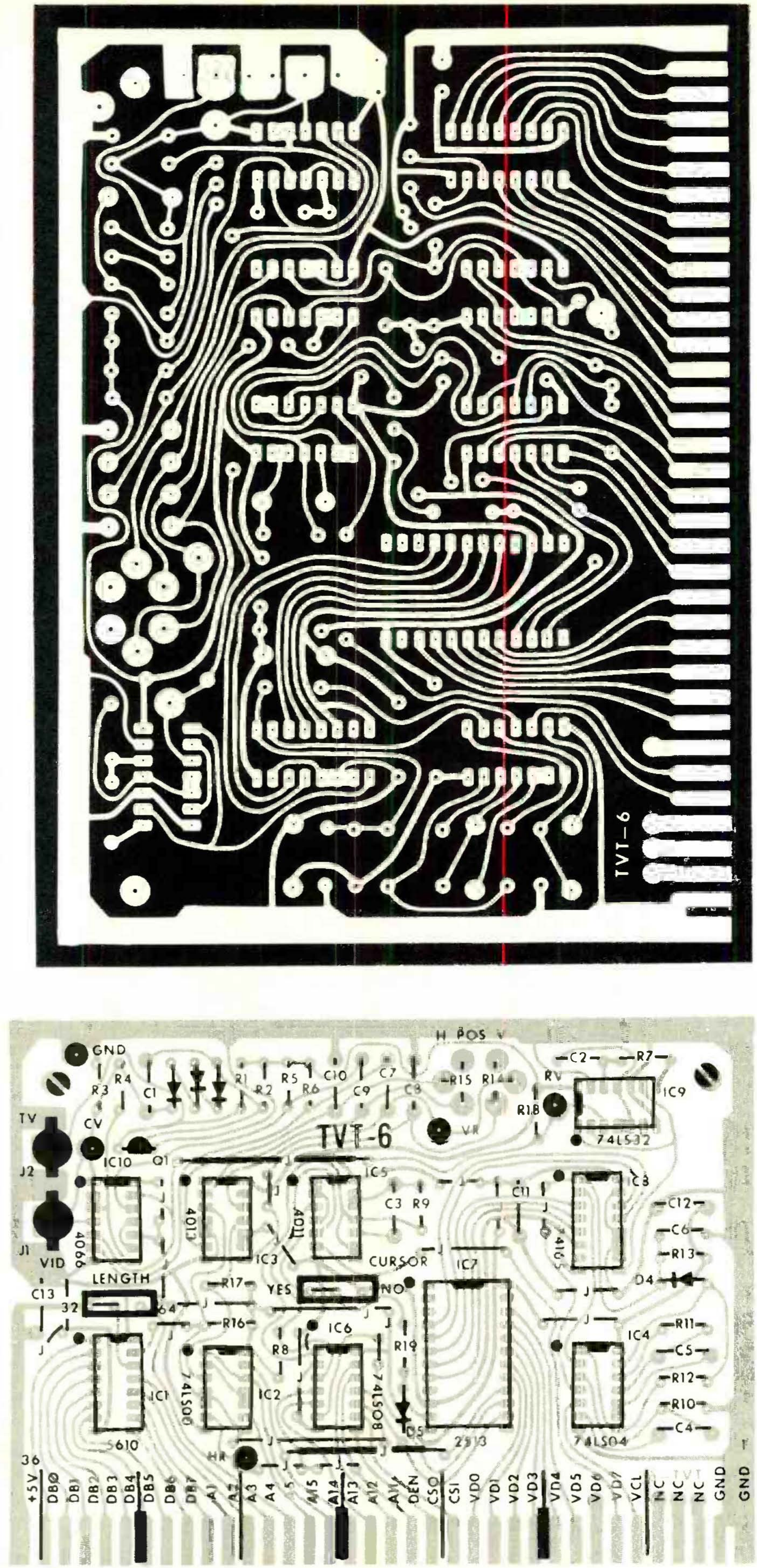


Fig. 5. Actual-size foil pattern (top) and component installation (below). Use sockets for all IC's. Edge connectors go to KIM-1.

TABLE I
TVT-6 PINOUT AND KIM-1 INTERFACE

TVT-6 CONTACT	NAME	REMARKS
1,2	GND	Heavy wire to expansion contact 22 or similar point in KIM-1
3, 4, 5	NC	Spares
6	VCL	1-MHz clock from expansion contact U(ϕ2). (In other systems clock phase must be selected so that load pulse arrives when CG is valid.)
7,8,9,10, 11,12,13, 14	VD7, VD6, VD5, VD4, VD3, VD2, VD1, VDΦ	Data output from memory display; drives character generator. For KIM-1 to display any part of pages 00 through 03, connections must be made as follows: TVT-6 contact: to pin 12 of KIM-1 IC: 7 U5 8 U6 9 U7 10 U8 11 U9 12 U10 13 U11 14 U12
15	CSI	Display memory chip select from μP; negative logic OR combined with TVT-6 chip select. From pin 1 of U4 on KIM-1.
16	CSO	Display memory chip select source; enables display memory when either TVT-6 is active or contact 15 is low. Goes to pin 13 of U5 through U12 in KIM-1 when displaying any part of pages 00 through 03. Existing Kϕ connection in KIM-1 must be broken.
17	DEN	Decode enable; goes low when μP is operated in normal mode, high when TVT-6 is doing an active scan. Goes to KIM-1 Applications contact K. Any external ground on applications contact K should be removed.
18,19,20, 21,22,23, 24, 25, 26, 27	A11, A12, A13, A14, A15, A5, A4, A3, A2, A1	Address inputs from μC, positive true. Addresses A'Φ, A6 through A10 not sent to TVT-6. Connections to KIM-1 expansion: KIM-1 contact: to TVT-6 contact: N (A11) 18 P (A12) 19 R (A13) 20 S (A14) 21 T (A15) 22 F (A5) 23 E (A4) 24 D (A3) 25 C (A2) 26 B (A1) 27
28, 29, 30, 31,32,33, 34, 35	DB7, DB6, DB5, DB4, DB3, DB2, DB1, DBΦ	μP data bus; tri-state active high from IC1 during active scan, not used at other times. Connections to KIM-1 expansion: KIM-1 contact: to TVT-6 contact: 8 (BD7) 28 9 (DB6) 29 10 (DB5) 30 11 (BD4) 31 12 (DB3) 32 13 DB2) 33 14 (DB1) 34 15 (DBΦ) 35
36	+5V	Regulated +5-volt (200-mA) power bus; should be heavy wire. From KIM-1 expansion contact 21 or similar point to contact 36 in TVT-6.

Note: KIM-1 conversion consists of breaking one foil trace and adding a new 36-pin socket (Amphenol 127 or similar). Connection to be broken originates as KΦ(pin 1 of U4). Routing of KΦthat goes to memory chip select pin 13 of U5 through U12 should be broken. Other KΦ connections, such as that to pin 1 of U16 should remain intact. Any external ground connections to Application connector contact K (decode-enable) must be removed. All wiring should be made with a wiring pencil.

When KIM-1 is used *without* displaying video, it will behave normally and transparently as long as TVT-6 is plugged in and addresses 8000 through DFFF are not used. To restore KIM-1 operation with TVT-6 out of socket, or to use available addresses for other programs, jumper pin 15 to pin 16 and separately jumper pin 1 to pin 17 in the KIM-1. Note that this jumpering is to be done only when TVT-6 is out of its connector. I you wish, a dpdt changeover switch can be added to perform the jumpering. Switch positions should be changed only when power is off.

memory tap to the character generator and display circuit.

4. Create special software that will allow TVT-6 scanning.

All 16 address lines are used, assigned as shown in Fig. 6A for a 32-character/line system or as shown in Fig. 6B for a 64-character/line system. Address A15 is the horizontal sync pulse and the key to jumping to the new SCAN instruction. This pulse is followed in descending address order by the vertical sync (A14) and three lines (L4, L2, L1) that produce the "what row of dots do we want?" information for the character generator. The lower address lines are used to select a page of display memory and to select the character that goes into any particular horizontal and vertical location on the display.

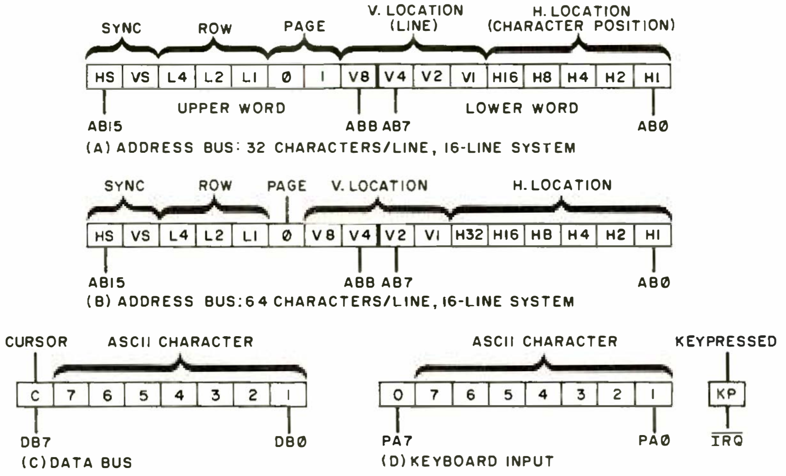
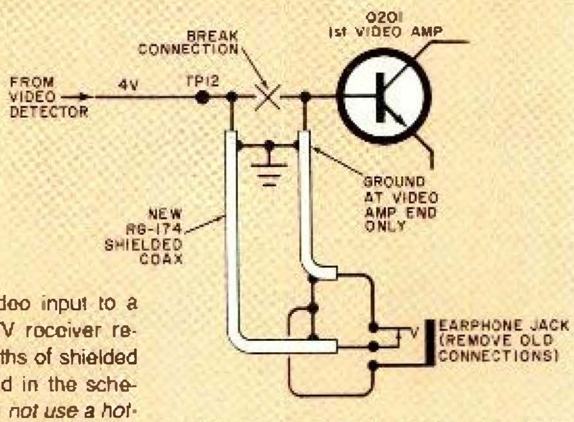


Fig. 6. Bus definitions as used with the TVT-6. All 16 address lines are used as described in text.

DIRECT-VIDEO INPUT CONVERSION



Adding a TVT-6 direct-video input to a small-screen solid-state TV receiver requires only two short lengths of shielded coaxial cable, as illustrated in the schematic. (Important Note: *Do not use a hot-chassis TV receiver!* Make absolutely certain that the TV receiver you use is transformer powered from the ac line.) The conversion circuit shown here is for the Sears No. 562-50260500 (Sams Photofact No. 1565-1). Other TV receivers can be modified in a similar manner.

The earphone jack in the circuit provides automatic changeover from normal receiver performance to video access. Correct bias is provided by TV output of the TVT-6. As an option, you can defeat the sound trap in the Sears TV receiver by lifting one end of capacitor *C201*.

The data within the machine (see Fig. 6C) uses the lowest seven bits as ASCII character storage. This is arranged by putting the least-significant ASCII character bit in the least-significant data slot, and so on up through the more significant bits. The eighth data bit (DB7) is reserved for a cursor. If DB7 is a zero, a character is displayed, while if it is a one, a cursor box is optionally displayed.

The existing KIM-1 keypad can be used as an ASCII keyboard for many applications, particularly for setup and debugging. If you wish to add an external ASCII keyboard and encoder, connect it to the KIM-1's parallel interface A, following the assignments shown in Fig. 6D. The seven ASCII bits go to the seven low-order data lines, while PA7 is hard wired for a zero. The keypress, or strobe, signal from the keyboard must pull the IRQ (interrupt request line) to ground for 10 μs to enter a character or machine command.

The truth table for PROM *IC1* is shown in Fig. 1. This truth table stores the SCAN instruction, activated by addresses 8000 through DFFF. When *IC1* is enabled, it causes the microprocessor's program counter to appear on the address lines for 32 or 64 consecutive scans that advance one count per microsecond. This automatically and sequentially addresses the display memory and produces exactly the data needed for a horizontal scan of TVT characters. The scan instruction runs at least twice as fast as the microprocessor normally moves, which is the key to TVT timing with a microprocessor.

To use the SCAN instruction, jump to a subroutine whose starting address is within the 8000 to DFFF range. For example, if you call JRS 8200, the SCAN instruction will deliver a horizontal sync pulse and initiate operation on the top row of characters, starting with the first character on page 2. After a selected 32

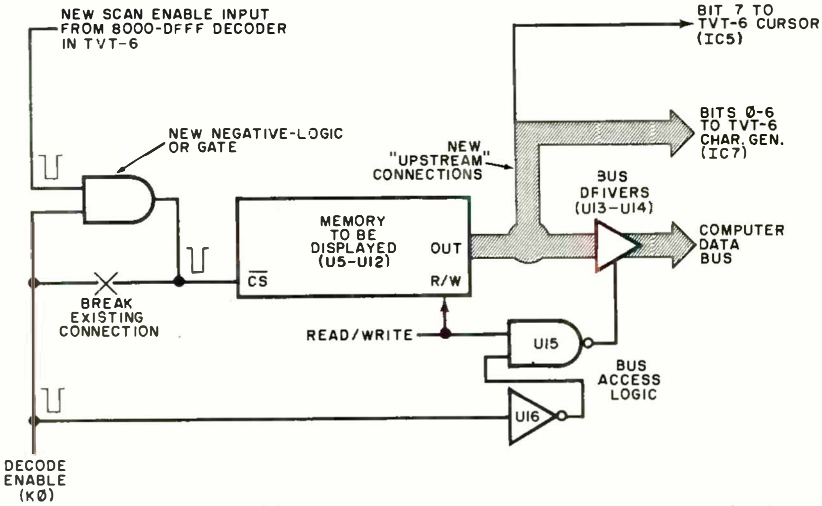


Fig. 7. Adding the upstream tap to the memory to be displayed.

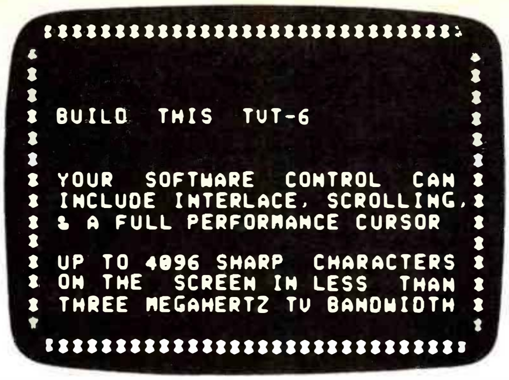
or 64 characters, the SCAN instruction automatically jumps back to the main program.

The SCAN instruction can be viewed as a "portable subroutine" because it readily moves around to automatically output the correct page and character generator's row information, starting with an easily computed JSR address. Addresses above DFFF will not activate the SCAN instruction. This includes the interrupt and reset vectors on the KIM-1 so that the operating system will work compatibly and properly with the new SCAN instruction.

There are many possible codings for the SCAN program with the limitation that the last address is a return-to-subroutine (RTS) instruction. The obvious choice of NOP or EA runs at only half speed and can't be used. Of the three dozen instructions that operate at full speed, the choice of LDY is the one that does not disturb the accumulator or its flags. This adds flexibility to other programs. The Y register can be viewed as a write-only memory in the SCAN software and we can think of the whole SCAN instruction as a group of double-speed fetch-but-don't-execute instructions. Theoretically, a 64-word PROM would be required for a 64-character line, but this can be overcome by ignoring address AØ and changing the PROM's address every second cycle of the machine.

Upstream Tap. The SCAN instruction will sequentially address 32 or 64 memory slots per horizontal scan line at a rate of one-per-clock cycle (1 μs). These addresses are presented to the entire memory in the computer, including the memory to be displayed. However, during the display times, the SCAN instruction and its PROM have control of the data bus so that the display memory (or anything else) cannot output information to the data bus.

The upstream tap is added as shown in Fig. 7. This tap is always outputting information to the character generator in the TVT-6. The output information is present even (and especially) when the display memory data bus drivers have been inactive. ◇



BUILD THE TVT-6 Part II

System debugging, software, and how to interface to other processors.

BY DON LANCASTER

LAST MONTH, we discussed construction of the TVT-6 TV typewriter and explained how it works and how it is connected to a KIM-1 microcomputer. We also started a discussion of the operating secrets of the TVT-6. Here, we complete the "secrets" discussion and go on to system debugging, some useful programs, and tell you how to interface the TVT-6 with other microprocessors.

Software. Four examples of tested, annotated, and workable KIM-1 software are given in the tables in this article. Table II contains a 16 × 32 scan program with full interlace. It automatically generates almost all the timing required by the TVT-6 and its companion TV monitor for this display format. The program is run by jumping to memory location 17Ad. The display is stopped by interrupting with the operating system, the cursor, or other program.

Table III is an optional full-performance cursor for the 16 × 32 system and includes scrolling, full cursor motion, and erase-to-end-of-screen capabilities. It is run by allowing the keypress signal from the keyboard to interrupt the scan program (any of the three Tables) via the $\overline{\text{IRQ}}$ interrupt line. Note that the cursor program is totally independent of the SCAN program. The only things the two programs share in common are the same pages of display memory. The screen-read-to-cassette can be performed using the existing KIM-1 operating system programs. You can also load from cassette to display, using the automatic search firmware.

Table IV is a 16-line/64-character scan program that requires only 64 words to be written into memory for the entire program. This program can be used to display the entire 1k of minimum KIM-1 memory for use as a super front-panel display if desired. For display-only applications, 1k of contiguous memory

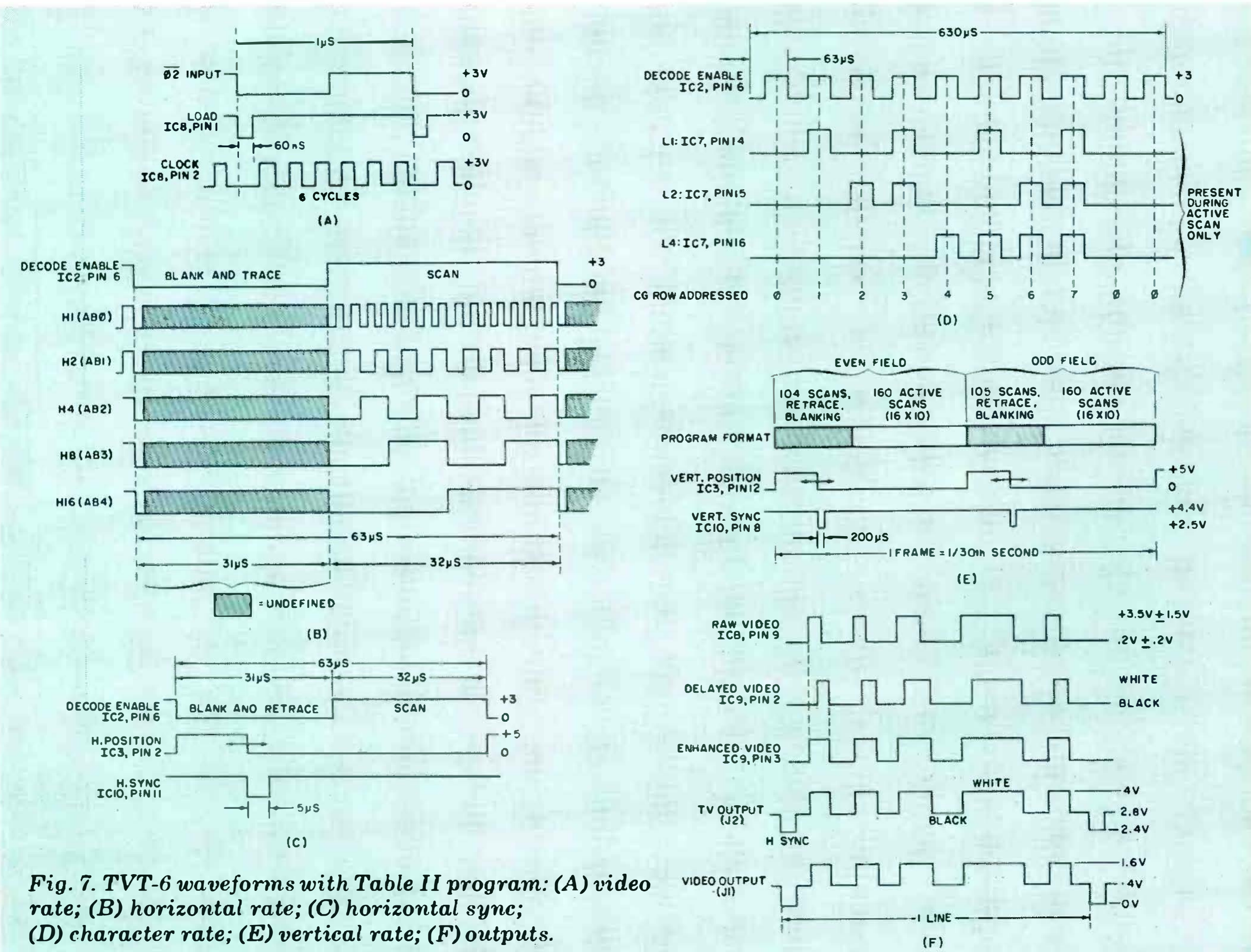


Fig. 7. TVT-6 waveforms with Table II program: (A) video rate; (B) horizontal rate; (C) horizontal sync; (D) character rate; (E) vertical rate; (F) outputs.

TABLE II

16 line X 32 character per line Interlaced
TVT6 Raster Scan:

```
µP - 6502          Start - JMP 17Ad      Displayed 0200-03FF
System - KIM-1     End - Interrupt       Program Space 1780-17E2

[HS][VS][L4][L2][L1][0][1][V8]      [V4][V2][V1][H16][H8][H4][H2][H1]
        Upper Address                        Lower Address

         1780  NOP  EA                 Equalize 2 cycles
         1781  STA  8d (8A)  (17)      Store upper address
         1784  PHA  48                 Equalize 10 cycles
         1785  PLA  68                 Continued

         1786  BNE  d0  00             Continued
         1788  JSR  20  00    80       ////Character Scans 1-8////
         178b  ADC  69  08             Increment Character Scan Counter
         178d  CMP  C9  C0             Is VS = 1?

         178F  BCC  90  F0*            No, do next character scan
         1791  JSR  20 (E0)  (17)      Equalize 15 cycles via sub
         1794  JSR  20  00    80       ////Character Scan 9////
         1797  TAX  AA                 Save Upper Address

         1798  LDA  Ad (89)  (17)      Get Lower Address
         179b  ADC  69  1F             Increment L; Set C on V4 overflow
         179d  STA  8d (89)  (17)      Restore lower word; save carry
         17A0  TXA  8A                 Get Lower Word

         17A1  BNE  d0  00             Equalize 5 cycles
         17A3  NOP  EA                 continued
         17A4  ADC  69  C0             Add carry; Reset VS
         17A6  JSR  20  00    80       ////Character Scan 10////

         17A9  CMP  C9  84             Is it line "17"?
         17Ab  BCC  90  d3*            No, continue character scans
-START-> 17Ad  LDA  Ad (dF)  (17)      Get Interlace word
         17b0  EOR  49  80             Change field

         17b2  BMI  30  05*            Jump if even field
         17b4  STA  8d (dF)  (57)      Odd Field V Sync; Restore Interlace word
         17b7  LDX  A2  66             Load short number of VB scans
         17b9  JSR  20 (E0)  (17)      Equalize 15 cycles via sub

         17bC  JSR  20 (E0)  (17)      Equalize 15 cycles via sub again
         17bF  BPL  10  05*            Jump if odd field
         17C1  STA  8d (dF)  (57)      Even Field V Sync; restore interlace
         17C4  LDX  A2  67             Load long number of V Blank scans

         17C6  JSR  20  1E    80       ////1st V blanking scan ////
         17C9  CLD  d8                 Equalize 9 cycles
         17CA  PHA  48                 Continued
         17Cb  PLA  68                 Continued

         17CC  LDA  A9  00             Initialize lower address
         17CE  STA  8d (89)  (17)      Continued
         17d1  LDA  A9  82             Initialize upper address
         17d3  STA  8d (8A)  (17)      Continued

         17d6  JSR  20  00    80       ///Remaining V Blanking scans////
         17d9  CLC  18                 Initialize carry
         17dA  DEX  CA                 One less scan

         17db  BMI  30  A4*            Start Character scan
         17dd  BPL  10  Ed*            Repeat Vertical blanking scan
         17dF       80                 Interlace word storage
         17E0  BCS  b0  00             ///Equalize 15 SUBROUTINE ////

         17E2  RTS  60                 Continued.
```

NOTES: TVT6 must be connected and scan microprogram PROM (IC1) must be in circuit for program to run.

Both 17b4 and 17C1 require that page 17 be enabled when page 57 is addressed. This is done automatically with KIM-1 circuitry.

Step 1788 goes to where the upper address stored in 178A and the lower address stored in 1789 tells it to. Values in these slots continuously change throughout the program.

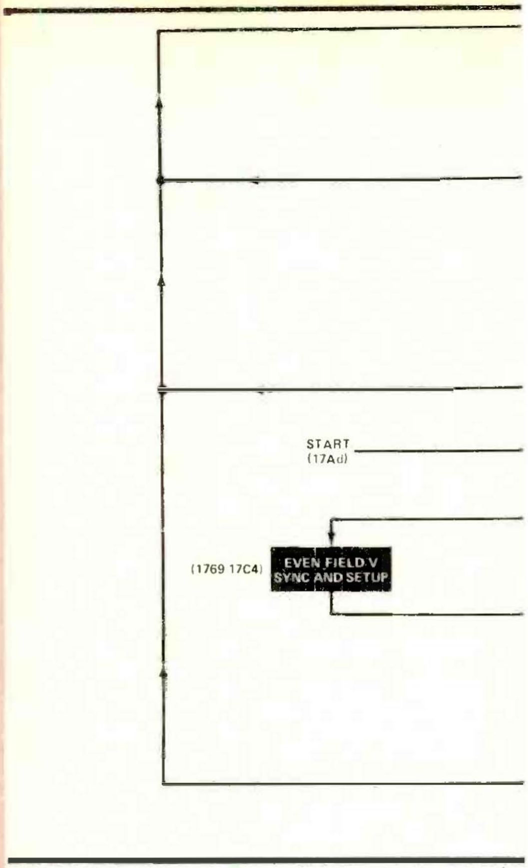
For a 525-line system, use 17b8 64 and 17C5 65 and a KIM-1 crystal of 992.250 kHz. This is only needed for video superposition and titling applications.

Normal program horizontal frequency 15,873.015 Hz; Vertical frequency 60.0114 Hz. 63 us per line; 264.5 lines.

* Denotes a relative branch that is program length sensitive.

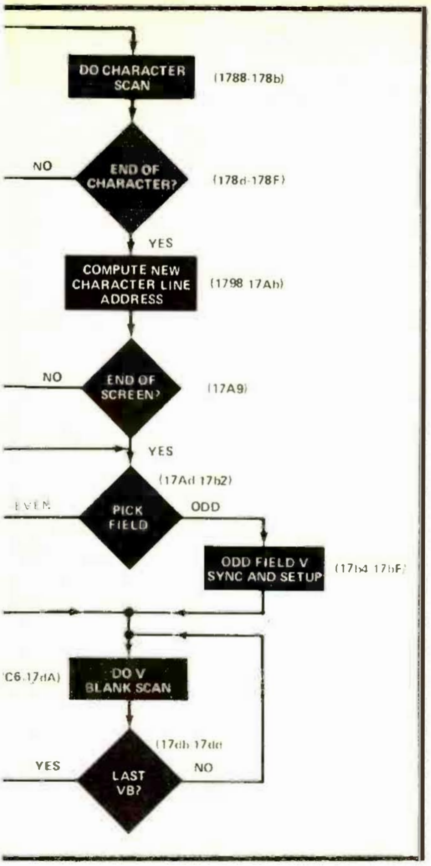
( ) Denotes an absolute address that is program location sensitive.

TVT6 length jumper must be in "32" position.



is required. Keep in mind that the KIM-1 has some operating system slots in the top of page zero and the stack at the top of page one. Unless you actually want to display the stack and operating system parameters, do not use these slots.

The 64-character line makes the TV receiver's horizontal frequency run considerably lower than normal. This will require a readjustment of the horizontal-hold control or some extra capacitance across the existing horizontal-hold capacitor. The width of the raster may also have to be reduced; this is most easily accomplished by adding a low-value inductor in series with the yoke. These changes are best made in a small-screen, *transformer-powered* monochrome TV receiver. The tradeoff of a lowered horizontal frequency produces a long character line but still allows 1 µs/character. This will not tax the bandwidth restrictions of TV receivers or r-f modulators. (Editor's Note: The small-screen Sears TV receiver we used required adjustment of horizontal size and linearity, a 0.033-µF Mylar capacitor in parallel with the 0.068-µF capacitor used for *C408* in the receiver, and an inductor consisting of 60 turns of No. 24 enameled wire on a ½" Nylon form in series with the red yoke lead in the receiver. In addition, it was necessary to disconnect one side of *C201* in the receiver



to defeat the sound trap. *Never attempt to modify a TV receiver that is powered directly from the ac line without an isolating transformer.*)

Table V contains a program that we call "Cruncher the Bear." This program produces 64 fully interlaced characters in each of 32 rows, for a total of 2048 sharp ASCII characters on-screen at one time within the 3-MHz bandwidth. You can add a hex-to-ASCII converter that slowly sequences high- and low-order machine code characters in the same slot and end up with 4096 hex characters displayed in only 3 MHz of bandwidth.

Table V requires a contiguous 2k of memory with a common upstream tap and separate chip enables. However, it is easily incorporated if you really want or need to display as many characters as the program allows.

Other software is easily written and developed for the TVT-6. For example, you may wish to have a 32 × 44 or a 32 × 48 character display and still use normal, or nearly normal, horizontal scanning rates. This allows for video titling and superimposition, oversize characters, color graphics, lower-case characters, and game displays. There is no lower limit to the number of character rows or characters per line you can use. If you have limited memory available,

Table III

16 X 32 Full-performance Cursor:

```
µP -- 6502            Start -- IRQ    Displayed 0200-03FF
System -- KIM-1       End  --- RTI    Program Space 0100-01dF

Input to Parallel Word A  | Ø | A7 | A6 | A5 | A4 | A3 | A2 | A1 |  IRQ
                                                            10 µs

Clear - CAN (18)                    Cursor Home - SOH (ØA)
Carriage Return - CR (Ød)           Scroll Up - DC1 (11)
Cursor Up - VT (Øb)                 Erase to End - DC2 (12)
Cursor Down - LF (ØA)               Spare Hook - DC3 (13)

Cursor Left - BS (Ø8)               Enter -- All characters
Cursor Right - HT (Ø9)              Ignore -- All other CTRL

Enter via
  IRQ
        0100  PHA  48                  Save A
        0101  LDY  A0  00              Reset Y Index
        0103  LDA  A5  (EE)            Get Cursor and test for range
        0105  CMP  C9  03              Is cursor on page 3?

        0107  BEQ  F0  04*             Yes, OK to continue
        0109  CMP  C9  02              Is cursor on page 2?
        010b  BNE  d0  3A*             No, Home cursor
        010d  LDA  b1  (Ed)            Get old cursored character
0147
        010F  AND  29  7F              Erase old cursor
        0111  STA  91  (Ed)            Replace character without cursor
        0113  LDA  Ad  00   17         Get new character from A parallel Int.
        0116  CMP  C9  20              Is it a character to be entered?

0142 <--0118  BCS  b0  28*             Yes, go and enter character
        011A  CMP  C9  18              Clear Screen?
015E <--011C  BEQ  F0  40*             Yes, clear screen
        011E  CMP  C9  0d              Return Carriage?

0152 <--0120  BEQ  F0  30*             Yes, Return carriage
        0122  CMP  C9  0b              Cursor Up?
0194 <--0124  BEQ  F0  6E*             Yes, Up Cursor
        0126  CMP  C9  0A              Cursor Down?

0166 <--0128  BEQ  F0  3C*             Yes, Down Cursor
        012A  CMP  C9  09              Cursor Right?
0158 <--012C  BEQ  F0  2A*             Yes, Right Cursor
        012E  CMP  C9  08              Cursor Left?

01A7 <--0130  BEQ  F0  75*             Yes, Left Cursor
        0132  CMP  C9  01              Cursor Home?
0147 <--0134  BEQ  F0  11*             Yes, Home Cursor
        0136  CMP  C9  11              Scroll Up?

0175 <--0138  BEQ  F0  3b*             Yes, Scroll Up
        013A  CMP  C9  12              Spare Hook?
014A <--013C  BEQ  F0  0C*             Ignore--Restore Cursor
        013E  CMP  C9  13              Erase to EOS?

01b1 <--0140  BEQ  F0  6F*             Yes, Erase to EOS
        0142  JSR  20  (D3)  (01)      /////Enter Character/////
        0145  BNE  d0  03*             End of Screen?
        0147  JSR  20  (C2)  (01)      Yes, Home Cursor

        014A  LDA  b1  (Ed)            ////Restore Cursor/////
        014C  ORA  09  80              Add cursor to cursored character
        014E  STA  91  (Ed)            Replace cursored character
        0150  PLA  68                  Get A

<--Out--0151  RTI  40                  Return to Scan
        0152  LDA  A5  (Ed)            ////Carriage Return/////
        0154  ORA  09  1F              Move Cursor to Right End
        0156  STA  85  (Ed)            Restore Cursor

        0158  JSR  20  (d5)  (01)      Increment Cursor
        015b  JMP  4C  (45)  (01)      Finish
        015E  JSR  20  (C2)  (01)      ////Clear////Home Cursor
        0161  JSR  20  (Cb)  (01)      Clear Screen

0147 <--0164  BEQ  F0  E1*             Finish
        0166  LDA  A5  (Ed)            ////Cursor Down//// Get Cursor
        0168  CLC  18                  Clear Carry
        0169  ADC  69  20              Move Cursor Down

        016b  STA  85  (Ed)            Restore Cursor
        016d  BCC  90  03*             Overflow of page?
        016F  JSR  20  (d9)  (01)      Yes, increment upper page
        0172  JMP  4C  (45)  (01)      Finish

        0175  JSR  20  (C2)  (01)      ////Scroll Up//// Home Cursor
        0178  LDY  A0  20              Add Offset to Index
        017A  LDA  b1  (Ed)            Get Offset Indexed Character
        017C  LDY  A0  00              Remove Offset from Index

        017E  JSR  20  (d3)  (01)      Enter Moved Character and Increment
        0181  BNE  d0  F5*             Repeat?
        0183  CLC  18                  Clear Carry
        0184  LDA  A9  01              Set A to page 3

        0186  STA  85  (EE)            Set Cursor to Page 3
        0188  LDA  A9  E0              Set A to start of last line
        018A  STA  85  (Ed)            Set Cursor to Start of last line
014A <--018C  BCS  b0  bC*             Finish if carry set
```

(Continued on next page.)

Table III (Continued)

```
          018E  JSR  20 (Cb)  (01)  Clear last line
          0191  SEC  38             Set Carry
          0192  BCS  b0  F0*        Restore Cursor to start of last line
          0194  LDA  A5 (Ed)        ////Cursor Up/////Get Cursor

          0196  SEC  38             Set Carry
          0197  SBC  E9  20         Move up one line
          0199  STA  85 (Ed)        Restore Cursor
014A <----019b  BCS  b0  Ad*        Underflow of page?

          019d  DEC  C6 (EE)        Yes, decrement page
          019F  LDA  A9  01         Set A to Page 1
          01A1  CMP  C5 (EE)        Did screen underflow?
014A <----01A3  BNE  d0  A5*        No, Finish

0147 <----01A5  BEQ  F0  A0*        Yes, Home Cursor
          01A7  DEC  C6 (Ed)        ///Cursor Left////Decrement Cursor
          01A9  LDA  A9  FF         Set A to page underflow
          01Ab  CMP  C5 (Ed)        Test for page underflow

019d <----01Ad  BEQ  F0  EE*        Change Page if off Page
014A <----01Af  BNE  d0  99*        Finish if on page
          01b1  LDA  A5 (EE)        ////Erase to EOS///Get Cursor
          01b3  PHA  48             Save Upper cursor location

          01b4  LDA  A5 (Ed)        Get lower cursor location
          01b6  PHA  48             Save lower cursor location
          01b7  JSR  20 (Cb)  (01)  Clear to End of Screen
          01bA  PLA  68             Get lower cursor location

          01bb  STA  85 (Ed)        Restore lower cursor
          01bd  PLA  68             Get upper cursor location
          01bE  STA  85 (EE)        Restore upper cursor
014A <----01C0  BNE  d0  88*        Finish

          01C2  LDA  A9  00         ///SUB//Home Cursor///
          01C4  STA  85 (Ed)        Set lower cursor to zero
          01C6  LDA  A9  02         Put page 2 in A
          01C8  STA  85 (EE)        Set upper cursor to 0200

          01CA  RTS  60             Return to main program
       +->01Cb  LDA  A9  20         ///SUB//Enter Space///
       |  01Cd  JSR  20 (d3)  (01)  Enter space via Sub
       +--01d0  BNE  d0  F9*        Repeat if not to end

          01d2  RTS  60             Return to main program
          01d3  STA  91 (Ed)        ////SUB//Enter,Increment// store
          01d5  INC  E6 (Ed)        Increment Cursor
          01d7  BNE  d0  06*        Overflow?

       +->01d9  INC  E6 (EE)        Yes, Increment cursor page to 03
       |  01db  LDA  A9  04         Load A with page 4
       |  01dd  CMP  C5 (EE)        Test for Overflow
       +->01df  RTS  60             Return to main program
```

NOTES: $\overline{IRQ}$ vector must be stored in 17FE 00 and 17FF 01.

Total available stack length is 32 words. Approximately 16 are used by operating system, cursor, and scan program. Stack must be initialized to 01FF as is done in KIM-1 operating system. For 30 additional stack locations, relocate subroutines starting at 01C2 elsewhere.

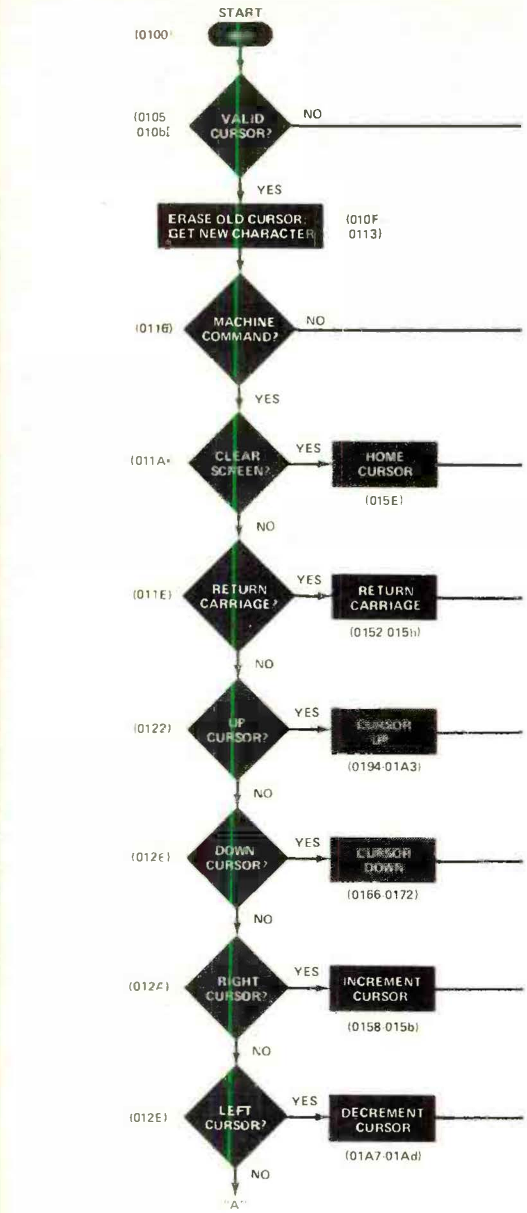
To protect page, load 00F3 04. To enable entry load 00F3 00.

Cursor address is stored at 00Ed low and 00EE high on page zero.

To display cursor load 014d 80. To not display cursor load 014d 00.

* Denotes a relative branch that is program length sensitive.

( ) Denotes an absolute address that is program location sensitive.



you can run 8 × 32, 4 × 64, 1 × 64, or even 1 × 8 character formats. All this takes is software changes, and the circuitry of the TVT-6 remains the same.

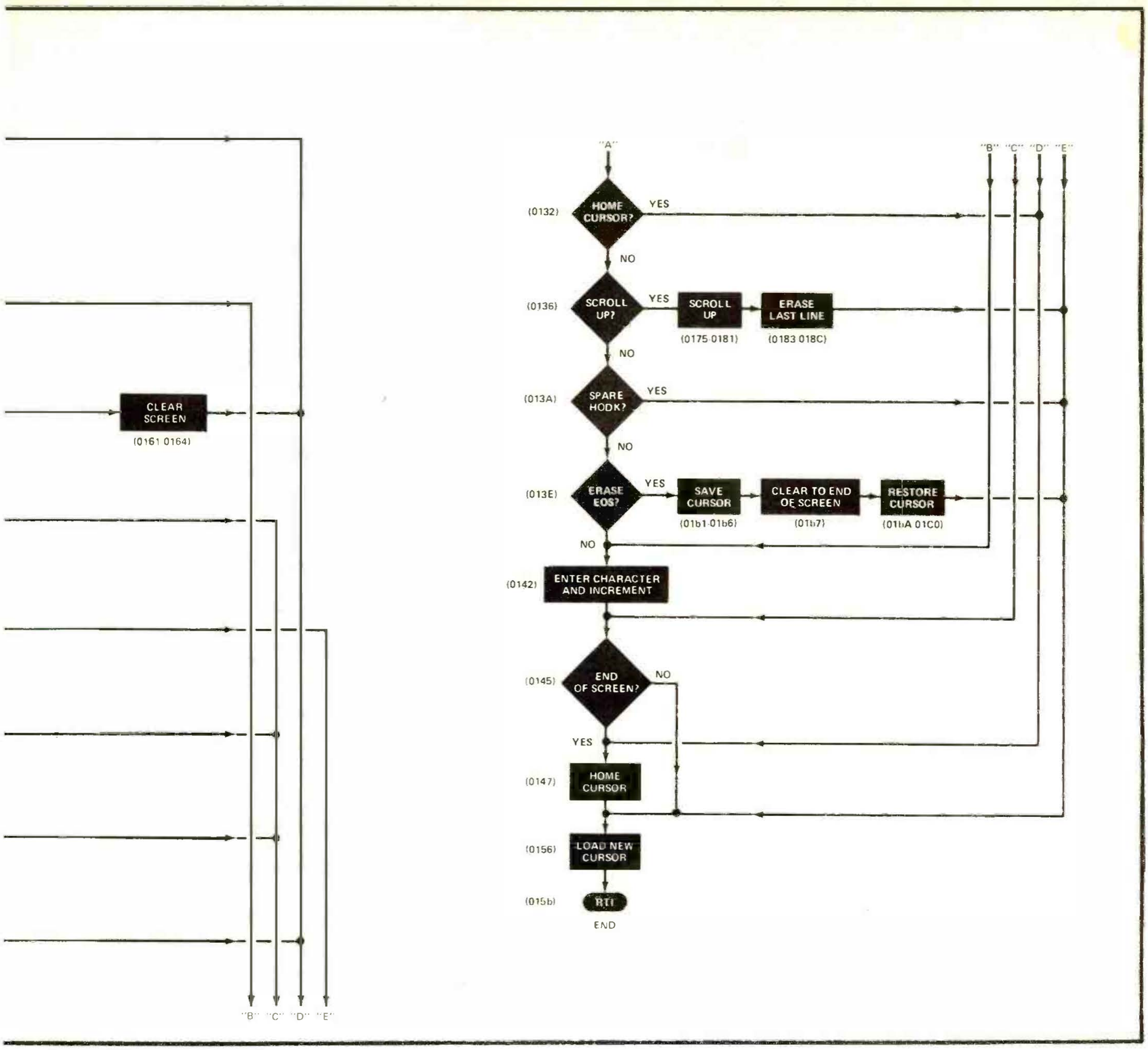
Initial Debugging. At this point, there should be no IC's in the sockets of the TVT-6 board assembly. Start by connecting the LENGTH jumper to 32 and the CURSOR jumper to YES on the TVT-6 board. (Note: These points are pads located at the center of the circuit board, not the edge-connector contacts.) Temporarily insert a jumper wire between pins 3 and 14 on the *IC5* socket. Center the two position control potentiometers and install *IC1*, *IC2*, and *IC6* in their respective sockets.

Connect your video monitor to the TVT-6 board and power up the system. Check for the presence of the SCAN instructions (see PROM Truth Table in Fig. 1 of Part 1) at hex locations 8000 through 8020. Write a simple program that jumps to a subroutine at location 8000 and then loops. Single-step through this program to verify proper operation of the SCAN instruction. Do not

USING THE TVT-6 WITH OTHER POPULAR MICROPROCESSORS

Both parts of this article have used the TVT-6 with the 6502 microprocessor-based KIM-1 microcomputer. Here is how to use the TVT-6 in μC's that use other popular microprocessors.

6800. The 6800 μP is very similar to the 6502 and, therefore, is easiest to convert. The SCAN microprogram can be LDAB(C6) for words 0 through 30 and RTS(39) for word 31. A literal translation of the tightest part of the SCAN program (1D;1782 through 178C) is: STA(B7); JSR(BD); ADDA(8B); CMPA(81); BCC(24). This routine requires 25 μs to cycle through as compared to the 21 μs required for the 6502.



8080. A stock 8080 μP can normally change its program counter once every 2 μs, but it can be "tricked" into doubling its speed *during a SCAN microprogram* by driving the usual address line A9 of the display memory from SYNC. The SCAN microprogram is then NOP(00) for words 0 through 30 and RET(A9) for word 31. A tighter than literal translation of the SCAN program (1D;1782 through 178C) is: STAXB(02); CALL(AD); ADD(82); CMP(BB); JNC(DB), which requires 24 μs to cycle through. Here, the TVT-6 address lines A5 through A1 must be relabelled A4 through AØ, respectively.

Z80. The Z80 μP can use 8080-developed software with speed-doubling scans, or it can simply be run faster, allowing the program counter to change once every microsecond. Use a literal translation of the program for the 6502.

12 Address Line μP's. The four upper address lines of 12 address line μP's can be decoded to allow normal operation, 8 to 12 lines of scan, a vertical sync pulse, an operating return system, and an optional "page-change" command. This leaves a 256-character page on the bottom eight bits, and the "page-change" command can be latched to change to any number of additional pages, as required.

General Hints. Horizontal scan should last at least 62, 63.5, or 64 μs for conventional ho izontal-frequency operation. The microprogram scan must *end* exactly this number of microseconds later for each horizontal line in the total scan program. The total number of lines must produce a vertical frequency between 59.9 and 60.1 Hz per field. Note that a portion of the RTS time will be spent during the active (microprogram) scan time. Horizontal scans that last longer than 85 μs may make it difficult to obtain TV interface.

You can shorten a *blank* microprogram active scan by an *even* number simply by jumping ahead when you call your subroutine. For example, a JSR 8000 may produce a 32-character scan, while a JSR 8002 can produce a 30-character scan. This approach can come in handy when there is a need for equalizing scan lengths between character rows and during vertical retrace.

TABLE IV

16 line X 64 character per line TVT6 Raster Scan:

```
µP - 6502          Start - JMP 17AA       Displayed 0000 - 03FF
System - KIM-1     End - Interrupt        Program Space 1780-17bE

[HS|VS|L4|L2|L1|0|V8|V4]    [V2|V1|H32|H16|H8|H4|H2|H1]

     Upper Address                  Lower Address

        1780  LDA  A9  80              Initialize Upper Address
        1782  STA  8d  (87)  (17)      Store Upper Address
        1785  JSR  20  00     80       ////Character Scans 1-8////
        1788  ADC  69  08              Increment character scan counter

        178A  CMP  C9  C0              Is VS = 1?
        178C  BCC  90  F4*             No, Do next character scan
        178E  TAX  AA                  Save Upper Address
        178F  LDA  Ad  (86)  (17)      Get lower address

        1792  BCS  b0  00              Equalize 3 cycles
        1794  JSR  20  04     80       ////Character Scan 9////
        1797  BCS  b0  00              Equalize 3 cycles
        1799  ADC  69  3F              Increment Lower; Set C on V2 overflow

        179b  STA  8d  (86)  (17)      Restore Lower Address; save carry
        179E  TXA  8A                  Get upper addreas
        179F  JSR  20  00     80       ////Character Scan 10////
        17A2  ADC  69  C0              Add Carry; Reset VS

        17A4  CMP  C9  84              It is "Line 17"?
        17A6  BCC  90  dA*             No, continue character scans
        17A8  BCS  b0  00              Yes, Go to vertical blanking scans
- START-17AA  CLD  d8                  Equalize 2 cycles

        17Ab  JSR  20  00     C0       ////Vertical Sync Scan////
        17AE  LDX  A2  22              Load #V Blank Scans -2
        17b0  LDA  A9  00              Initialize Lower Address
        17b2  STA  8d  (86)  (17)      Continued

        17b5  CLC  18                  Equalize 2 cycles
        17b6  BCS  b0  00              Equalize 2 cycles again
        17b8  JSR  20  00     80       ////Vertical Blanking Scans////
        17bb  DEX  CA                  One less scan

        17bC  BMI  30  C2*             Start Character Scan
        17bE  BPL  10  F5*             Repeat Vertical Blanking scans
```

NOTES: TVT6 must be connected and scan microprogram PROM (IC1) must be in circuit for program to run

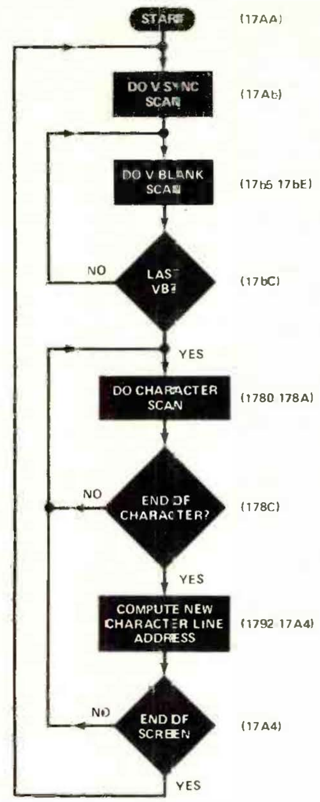
Step 1785 goes to where the upper address stored in 1787 and the lower address stored in 1786 tells it to. Values in these slots continuously change throughout the program.

Normal program horizontal frequency is 11,764.705 Hz. Vertical Frequency is 60,024 Hz. 85 us per line; 196 lines. Character time 1 us. 160 active lines, 36 retrace. Needs TV set adjustment and possible modification (hold and width).

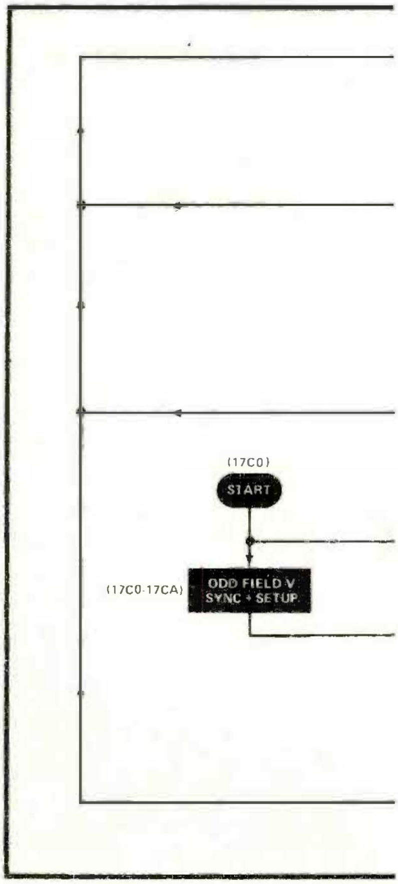
* Denotes a relative branch that is program length sensitive.

( ) Denotes an absolute address that is program location sensitive.

TVT6 length jumper must be in "64" position.



16 x 64 RASTER SCAN



proceed beyond this point until you are certain that the SCAN subroutine is operating properly. (Critical waveforms to be observed with an oscilloscope are illustrated in Fig. 7 using the program listed in Table II.)

Insert *IC3* into its socket and load the program given in Table II. (*Never* install an IC in a powered circuit; always turn off the power, install the IC, and power up again.) Set the address to 17Ad and depress GO. Using an oscilloscope, check at test point VR for the presence of a 60-Hz pulse. Switch the scope to line-sync and observe that the pulse remains fixed or drifts very slowly across the screen. Again, do not proceed until you are certain that the SCAN program is operating properly.

Install all remaining IC's, except *IC5*, in their respective sockets on the TVT-6 board. At this point, the screen should be filled with a stable display of 512 cursor boxes. Viewed up close, the boxes should appear to be "hiding" characters. Do not proceed until you have the indicated display.

Checking with Fig. 7, particularly with respect to the LOAD and CLOCK on *IC8* (Fig. 7A) verify whether or not the appropriate waveforms are present. If they are, remove the jumper wire from the *IC5* socket and install *IC5*. Now, the screen of the monitor should have displayed on it a full array of characters with about half of them winking cursor blocks. Load the following hex numbers into memory, starting at location 0200:

20, 20, 20, 50, 4F, 50, 55, 4C, 41, 52, 20, 20, 45, 4C, 45, 43, 54, 52, 4F, 4E, 49, 43, 53, 20, 20, 54, 56, 54, 2D, 36, 20, 20. Return to address 17Ad and depress GO. The top display line should now read "POPULAR ELECTRONICS TVT-6" and be indented three spaces. If all is well to this point, you can begin feeding in your cursor programs, add external keyboard and/or cassette loads and dumps, etc.

Should you encounter problems with your TVT-6, always begin debugging by using the 16 × 32 format on a KIM-1, even if you plan on using longer line lengths or plan to translate the code into another coding system. Note that the translation *must* be at the machine-language level because the SCAN program must provide the exact number of machine cycles as well as the proper sequencing. The 64-character lines will require some adjustments to be made in the monitor TV receiver's horizontal circuit as detailed earlier.

Closing Remarks. We have presented here full construction and operating details for a very versatile and inexpensive TV typewriter for use with the KIM-1 microcomputer. If you have a computer that uses a microprocessor other than the 6502 used in the KIM-1, we refer you to the box for use details. ◇

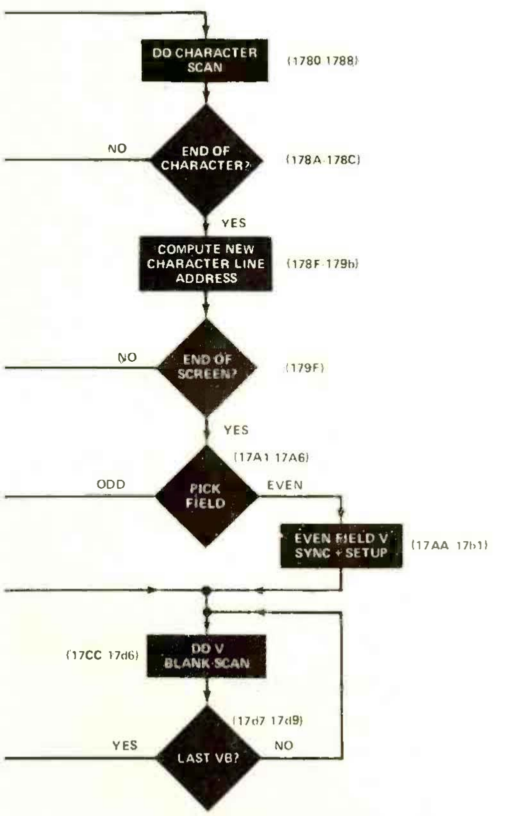


TABLE V

CRUNCHER THE BEAR Program for a 32 line X 64 character per line TVT6 raster scan:

```
µP - 6502          Start - JMP 17C0        Displayed 0000-07FF
System - KIM-1     End - Interrupt         Program Space 1780-17dA

[HS][VS][L4][L2][L1][V16][V8][V4]    [V2][V1][H32][H16][H8][H4][H2][H1]
        Upper Address                         Lower Address

       1780  LDA  A9  80             Initialize Upper Address
       1782  STA  8d  (87)  (17)     Store Upper Address
       1785  JSR  20  00     80      ////Character Scans 0-7////
       1788  ADC  69  10             Increment Character Gen by 2

       178A  CMP  C9  C0             Is VS = 1?
       178C  BCC  90  F4*            No, Do next character scan
       178E  PHA  48                 Save Upper Address
       178F  LDA  Ad  (86)  (17)     Get Lower address

       1792  ADC  69  3F             Increment L; Set Carry on V2 overflow
       1794  STA  8d  (86)  (17)     Restore L; Save carry
       1797  PLA  68                 Get Upper Word
       1798  JSR  20  0C     80      ////Character Scans 8,9 ////

       179b  ADC  69  C0             Add Carry; Reset Upper Address
       179d  CMP  C9  88             Is it "Line 33"?
       179F  BCC  90  E1*            No, repeat Character Scans
       17A1  LDA  Ad  (81)  (17)     Get Interlace word

       17A4  ADC  69  78             Set Carry if Odd Field finished
       17A6  BCC  90  0C*            Start Even Field if Carry Clear
       17A8  LDX  A2  22             Load Even number of V Scans -2
       17AA  LDA  A9  80             Load Even Field Upper Start

       17AC  STA  8d  (81)  (57)     Even Field V Sync + Restore Interlace
       17AF  LDA  A9  88             Even Field Line 33 CMP Value
       17b1  STA  8d  (9E)  (17)     Store Even 33 CMP Value
       17b4  LDA  A9  00             Clear Accumulator

       17b6  STA  8d  (86)  (17)     Initialize Lower Address
       17b9  LDY  A0  06             Equalize 31 cycles
       17bb  DEY  88                 continued
       17bC  BPL  10  Fd             continued

       17bE  BCS  b0  0C*            Jump if even field
-START-17C0  LDA  A9  88             Load Odd Field Upper Start
       17C2  STA  8d  (81)  (57)     Odd Field V Sync + Restore Interlace
       17C5  LDA  A9  90             Odd Field line 33 CMP Value

       17C7  STA  8d  (9E)  (17)     Store Odd 33 CMP Value
       17CA  LDX  A2  23             Load Odd number of V Scans
       17CC  JSR  20  3F     80      ////  1st V Blanking Scan ////
       17Cf  PHA  48                 Equalize 7

       17d0  PLA  68                 continued
       17d1  CLD  d8                 Equalize 4
       17d2  CLC  18                 continued
       17d3  JSR  20  00     80      //// Other V Blanking Scans ////

       17d6  DEX  CA                 One Less scan
       17d7  BMI  30  A7*            Start Character Scans
       17d9  BPL  10  F6*            Repeat V Blanking Scan
```

NOTES: TVT6 must be connected and scan microprogram PROM IC1 must be in circuit for program to run. TVT6 length jumper must be in "64" position.

Step 1785 goes to where the upper address stored in 1787 and the lower address stored in 1789 tells it to. Values in these slots continuously change throughout the program.

Step 1781 is 80 for even fields and 88 for odd fields. Step 179E is 88 for even fields and 90 for odd fields.

Both 17AC and 17C2 require that page 17 be enabled when page 57 is addressed. This is done automatically with KIM-1 circuitry.

Note that 2K worth of contiguous memory from 0000 to 07FF is needed. This takes a KIM-1 modification. Both sets of 1k words must share a common upstream tap but be separately enabled.

Normal program horizontal frequency is 11,764.705 Hz. Vertical Frequency is 59.8712 Hz. For 60 Hz vertical use 1.002150 MHz crystal. 85 us per line; 196.5 interlaced lines per field; two fields per frame. One us character time, 160 active lines per field. Needs TV set adjustment and possible modification (hold and width).

* Denotes a relative branch that is program length sensitive.

( ) Denotes an absolute address that is program location sensitive.