PHILIPS

Electronic components and materials

Technical note 089

CRT display using a standard TV monitor for 2650-based microcomputers

The CRT has become an important and almost indispensible part of many small computer systems. More often known as a visual display unit (VDU), it can be bought complete with a keyboard and all the necessary interface and control logic. However, a simple TV monitor can be used in place of specialized VDUs, providing the attractive facilities of a VDU at a fraction of the cost. Of course, a certain amount of control logic is required, as is a software program to control this. This Technical Note describes the hardware and software required to use a standard TV monitor as an I/O peripheral for a 2650 microcomputer system.

Display specification

A standard TV monitor is used to provide the display. It is driven by a video signal containing black-and-white character information and line and frame synchronization pulses. The screen is scanned in a 312-line frame, not interlaced. Character exchange can be performed during the line flyback time or in blocks during the display time. Figure 1 shows the display frame and character format.

The character display is 22 rows of 40 characters per row. Each character is formed in a 7 x 5 matrix with five lines below the character to provide a space between rows. A character row thus consists of 12 lines. The display is normally white characters on a black background, although this can be reversed with a wire link.

The cursor can be positioned to any of the 880 character positions on the screen and is displayed by inverting the video signal for that character position. A reset input allows the screen to be erased and the cursor to be returned to the top left-hand position.

Display logic

The display logic is responsible for generating the video signal to the monitor according to the information supplied from the microcomputer via the data exchange interface. Figure 2 shows the block diagram of the display logic, with inputs from the data exchange and an output to the monitor.

Horizontal counter

The horizontal counter determines the position of each character along the line. Because video is displayed for only 46 of the 64 μs line-time, the counter should reach the value 40 (number of characters) in 46 μs. Thus, after 64 μs, the counter will reach the value 55 which will be decoded to produce the line-sync pulse via the flip-flop SYNFF. The rate at which the counter must be incremented (56 counts in 64 μs) determines a system clock of 875 kHz.

signetics

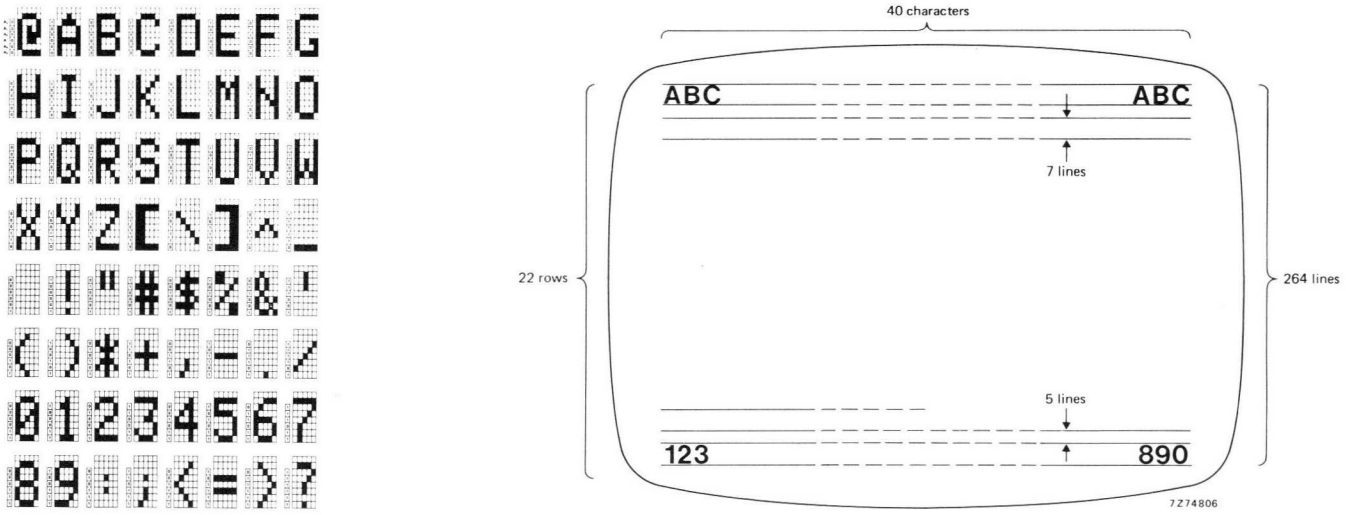


Fig. 1 Display frame and character format on the CRT.

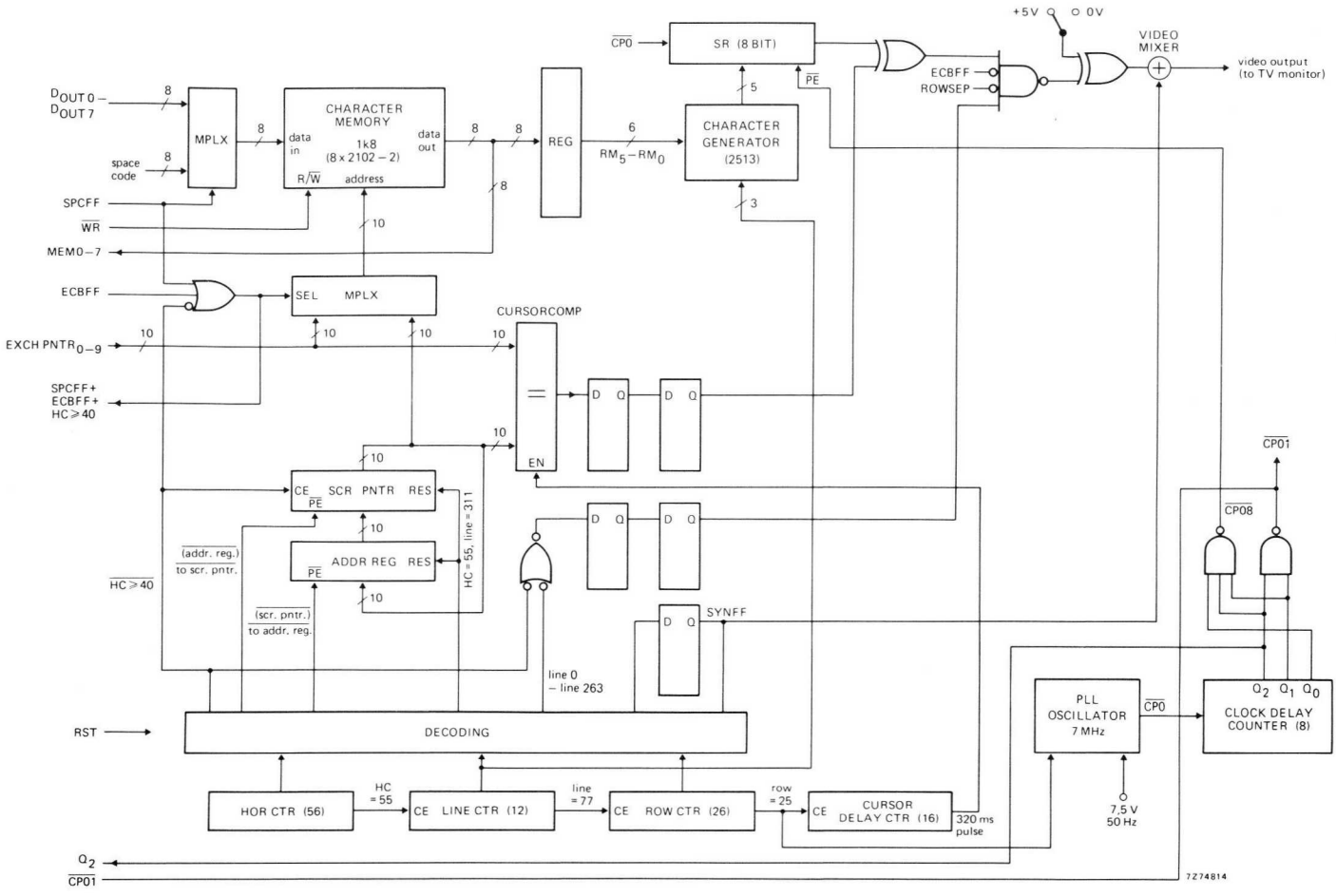


Fig. 2 Block diagram of the display logic.

Vertical timing

Line counter

The line counter is used to count the number of lines per character row. It is incremented each time the horizontal counter reaches its maximum value of 55. The least significant three bits of the counter are used to control the vertical scan of the character generator (7-row matrix). It is reset after a count of 11 (12 lines).

Row counter

The row counter is incremented on each count of 11 of the line counter. The row counter is responsible for maintaining frame synchronization and character positioning.

Character memory

The screen capacity is 22 x 40 = 880 characters. This number of characters can conveniently be stored in a 1k8 RAM. The character memory is loaded with data from the data exchange or with the code for a space from the space logic.

Memory addressing

The character memory is addressed by either the exchange pointer or the screen pointer. The exchange pointer is used during data exchange between the microcomputer and the memory. This may occur when:

- the space flip-flop is set,
- the exchange-by-blocks flip-flop is set, or,
- the horizontal counter value is greater than or equal to 40.

In all other cases, the memory is addressed by the screen pointer. The screen pointer starts at zero, corresponding to the top left-hand screen position, and is incremented at the same rate as the horizontal counter for the first 40 characters. At the end of each of the first 11 lines of a character row, the screen pointer is reloaded with the value in the address register (initially zero) to maintain the correct offset in the character memory for each row. At the end of the twelfth line of a row, the screen pointer has been incremented to the value of the address register + 40. This value is then stored in the address register to provide the offset for the next row of 40 characters. The screen pointer and address register are reset when the line counter is 311 and the horizontal counter is 55.

Character generation

The output of the character memory is used to address a ROM character generator, Signetics type 2513. This can be obtained with 64 standard ASCII characters on a 7 x 5 matrix. Only bits 0 to 5 of the memory output are used to address the character. The seven lines of the character are addressed by the three least significant bits of the line counter.

Each 5-bit line of the 7 x 5 character matrix is parallel loaded into an eight-bit shift register, with the three remaining bits being set to zero. The three zero bits provide spacing between successive characters. The data in the shift register is then output serially using a 7 MHz clock (8 x the frequency of the system clock) which was determined by the horizontal character counter. This serial output is passed, via the display blanking circuit, to the video mixer before being output to the TV monitor.

Cursor control

The cursor indicates the screen position with which data exchange can occur. The cursor is displayed on the screen by inverting the video at that character position three times a second.

The output of the cursor is an approximate 3 Hz signal which alternately enables/disables the cursor position comparator. When the value of the screen pointer is equal to that of the exchange pointer and the comparator is enabled (for 16 frames every 32 frames) the shift register output is inverted for one character-time by the exclusive OR gate.

Display blanking

The data stream to the video mixer must be inhibited during:

- data exchange by blocks (ECBFF = 1),
- the five separation lines in each row (ROWSEP = 1),
- the time that the horizontal counter value is $\geqslant 40$,
- the time that the row counter value is > 22.

This function is achieved using a four-input NAND gate.

The complete display on the screen can be inverted by selection of a wire link to the final exclusive OR gate: the display is then black characters on a white background in place of white characters on a black background.

Display logic clocks

A stable clock is necessary if a clear display is required. Thus, the 7 MHz clock is derived from a phase-locked-loop which has a low-voltage reference from the 50 Hz mains supply. Division of this clock by eight provides the system clock, defined by the horizontal counter. Unless otherwise shown in the diagrams, all flip-flops, counters and registers are driven by this system clock.

Data exchange logic

This provides the interface between the microcomputer and the display logic. The block diagram of the data exchange is shown in Fig. 3. It is divided into three subsections dealing with: data signals, display logic control and data transfer control. The circuit diagrams of these subsections are shown in Figs 4, 5 and 6.

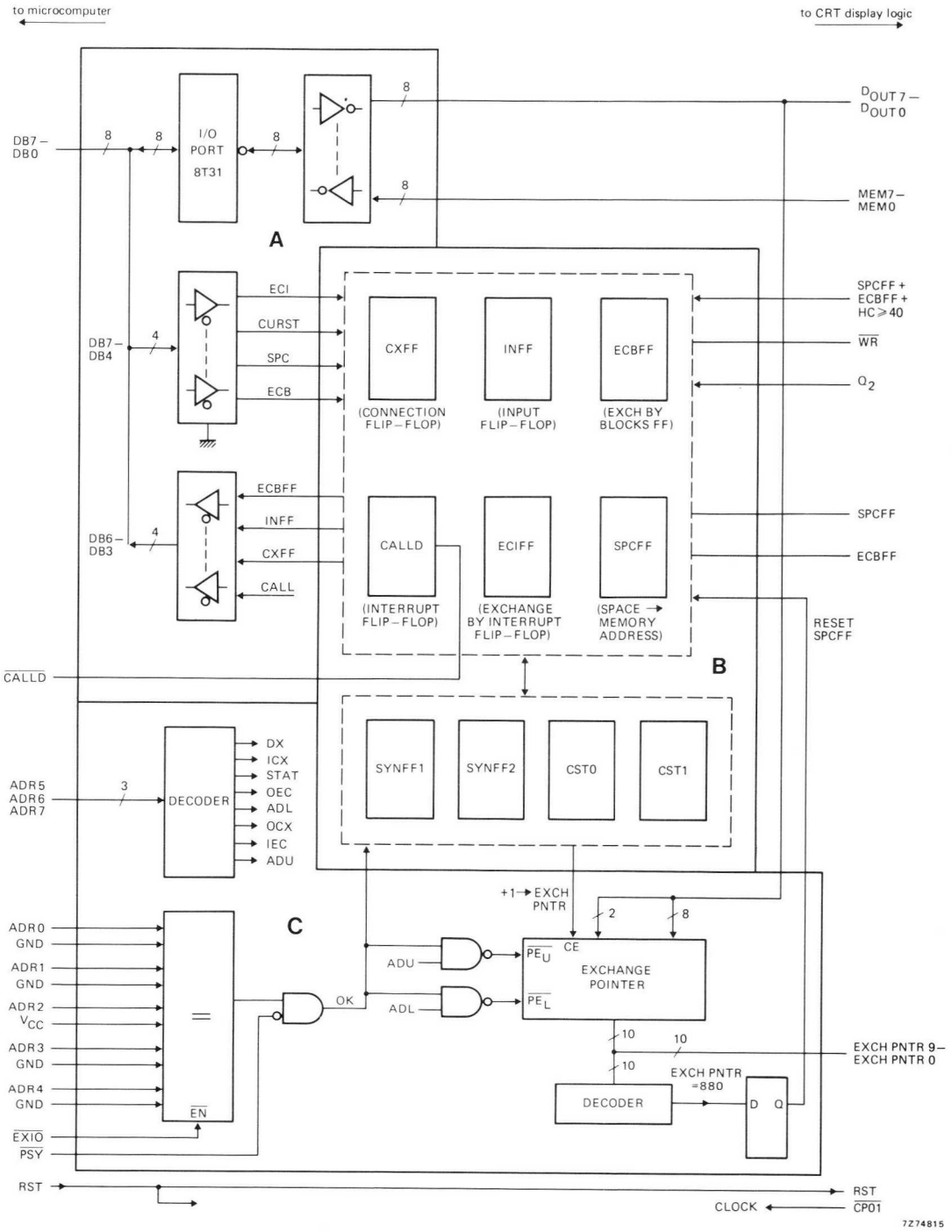


Fig. 3 Block diagram of the data exchange logic.

Data signals

Figure 4 shows the connection of the microcomputer data bus to the character memory via an 8T31 bi-directional I/O port and 8T26 inverting transceivers. Bits 3 to 7 of the data bus are also used to transmit/receive signals for the display logic control subsection. During connection commands (ICX or OCX), the four most significant bits of the control word are passed to the command flip-flops in the display logic control. These bits are:

- ECB Exchange-by-blocks, bit 7; the character memory is continuously addressed when ECB = 1. When ECB = 0, the character memory is addressed during the flyback time.
- SPC Spaces, bit 6; fills the character memory with spaces and resets the cursor to the left-hand position. If SPC = 1, it is only effective if ECI = 1.
- CURST Cursor reset, bit 5; the cursor is reset when CURST = 1.
- ECI Exchange-by-interrupt, bit 4; data exchange is performed under interrupt control when ECI = 1. When ECI = 0, exchange is controlled by the main program.

  During a status command, the state of four of the control flip-flops is transferred to the data bus.

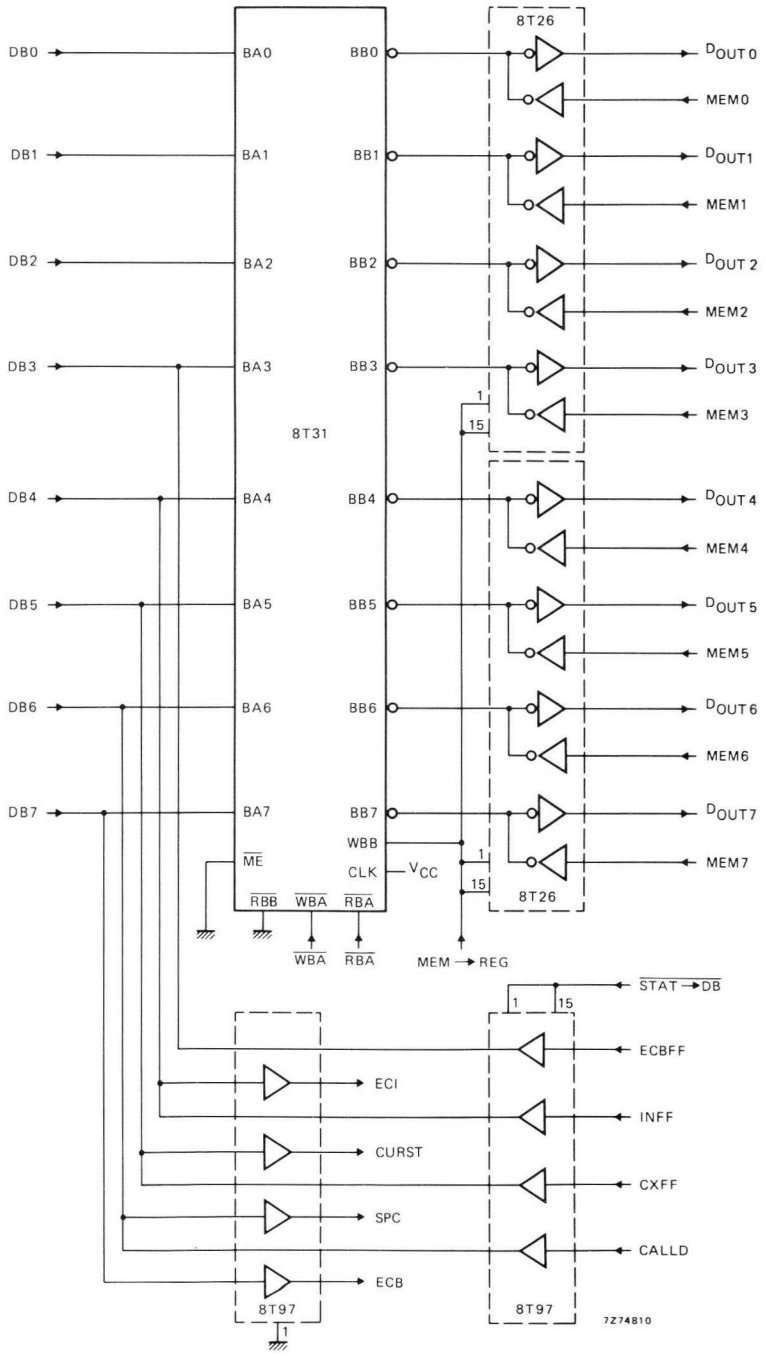


Fig. 4 Circuit diagram of subsection A, data signals, of the data exchange logic.

Display logic control

Figure 5 shows the circuit of the control logic. At each connection command, the state of the ECB, SPC and ECI bits of the control word are stored until the following disconnection command. Flip-flops SYNFF1 and SYNFF2 are used to synchronize commands to the peripheral clock while CST1 and CST0 control the sequence of actions for data transfer.

Command decoding

The logic that performs the command and peripheral number decoding and contains the exchange pointer is shown on Fig. 6. The peripheral number decoder is shown here connected for peripheral number 4. The exchange pointer is a 10-bit counter used to address the character memory. It can be set to any value by the microcomputer using the commands ADU and ADL.

The command and peripheral number are transferred to the peripheral via the eight least significant bits of the microprocessor address bus. Bits 0 to 4 of the address bus are used to transfer the peripheral number, while bits 5 to 7 specify the command. The peripheral responds to a command when the address on bits 0 to 4 of the bus corresponds to the hardwired address. The commands used with the CRT interface are detailed in Table 1.

TABLE 1 Commands to the CRT interface

mnemonic	command	ADR7	ADR6	ADR5
ADU	preset exchange pointer bits 8 and 9	0	0	0
IEC	input exchange	0	0	1
OCX	output connection	0	1	0
ADL	preset exchange pointer bits 0 to 7	0	1	1
OEC	output exchange	1	0	0
STAT	status request	1	0	1
ICX	input connection	1	1	0
DX	disconnection	1	1	1

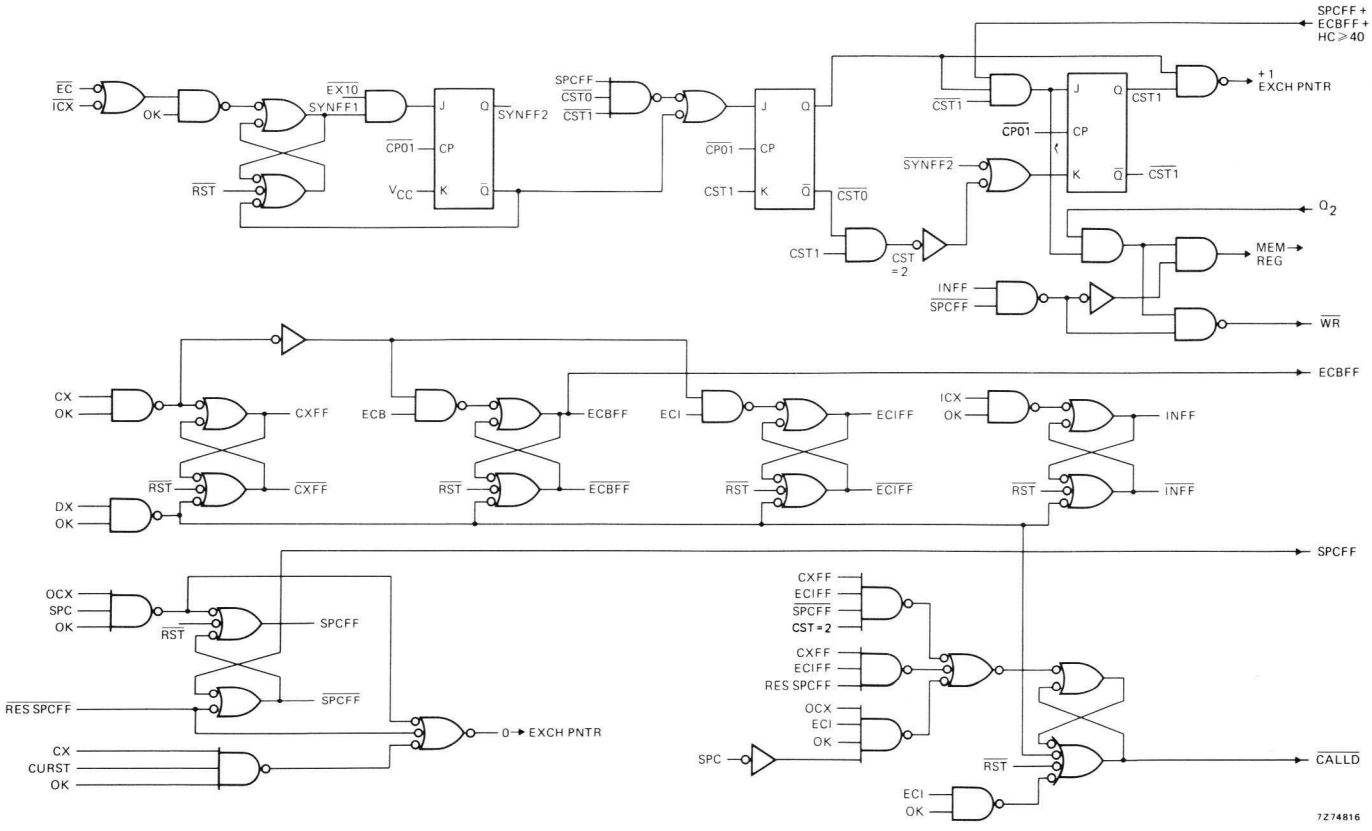


Fig. 5 Circuit diagram of subsection B, display logic control, of the data exchange logic.

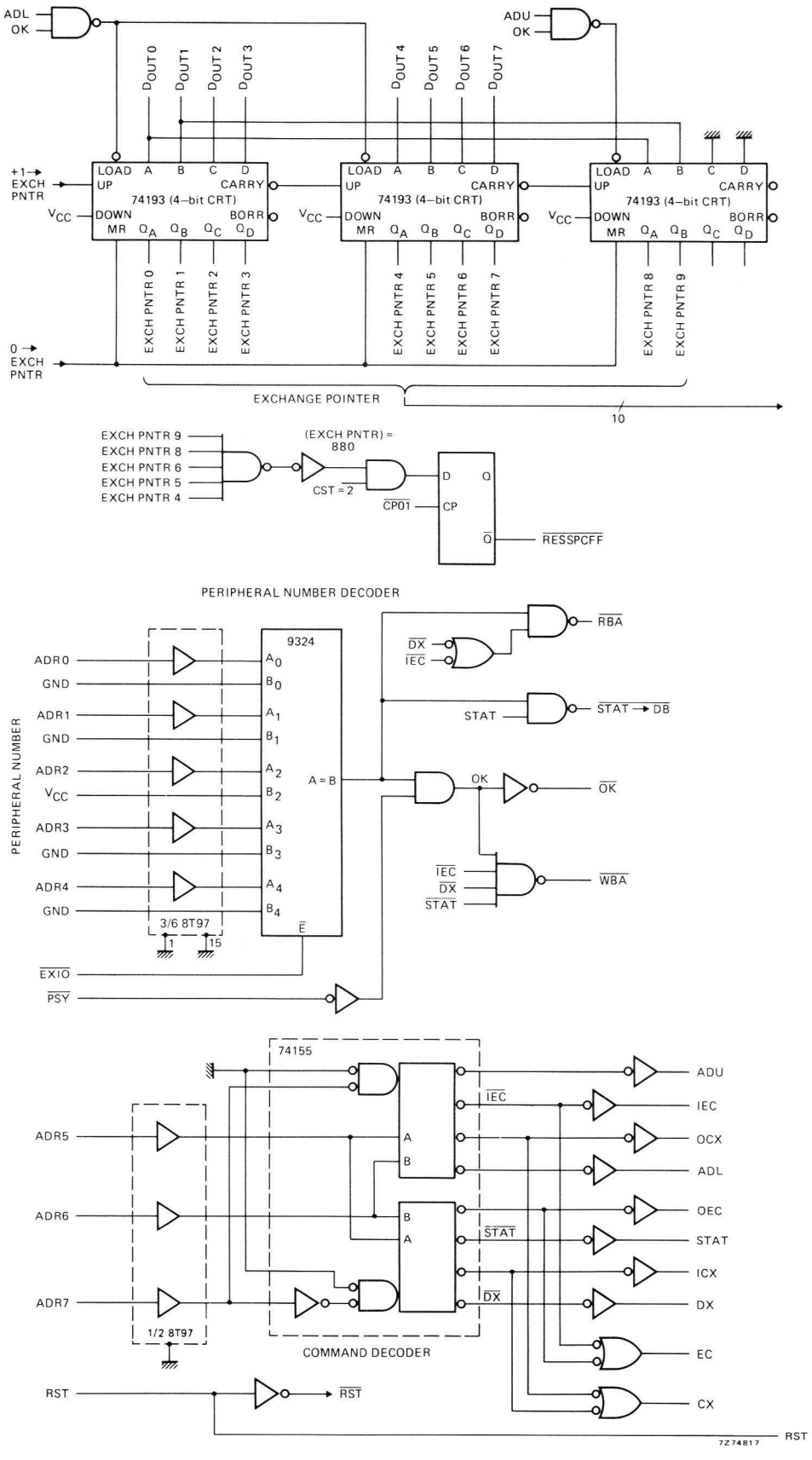


Fig. 6 Circuit diagram of subsection C, data transfer control, of the data exchange logic.

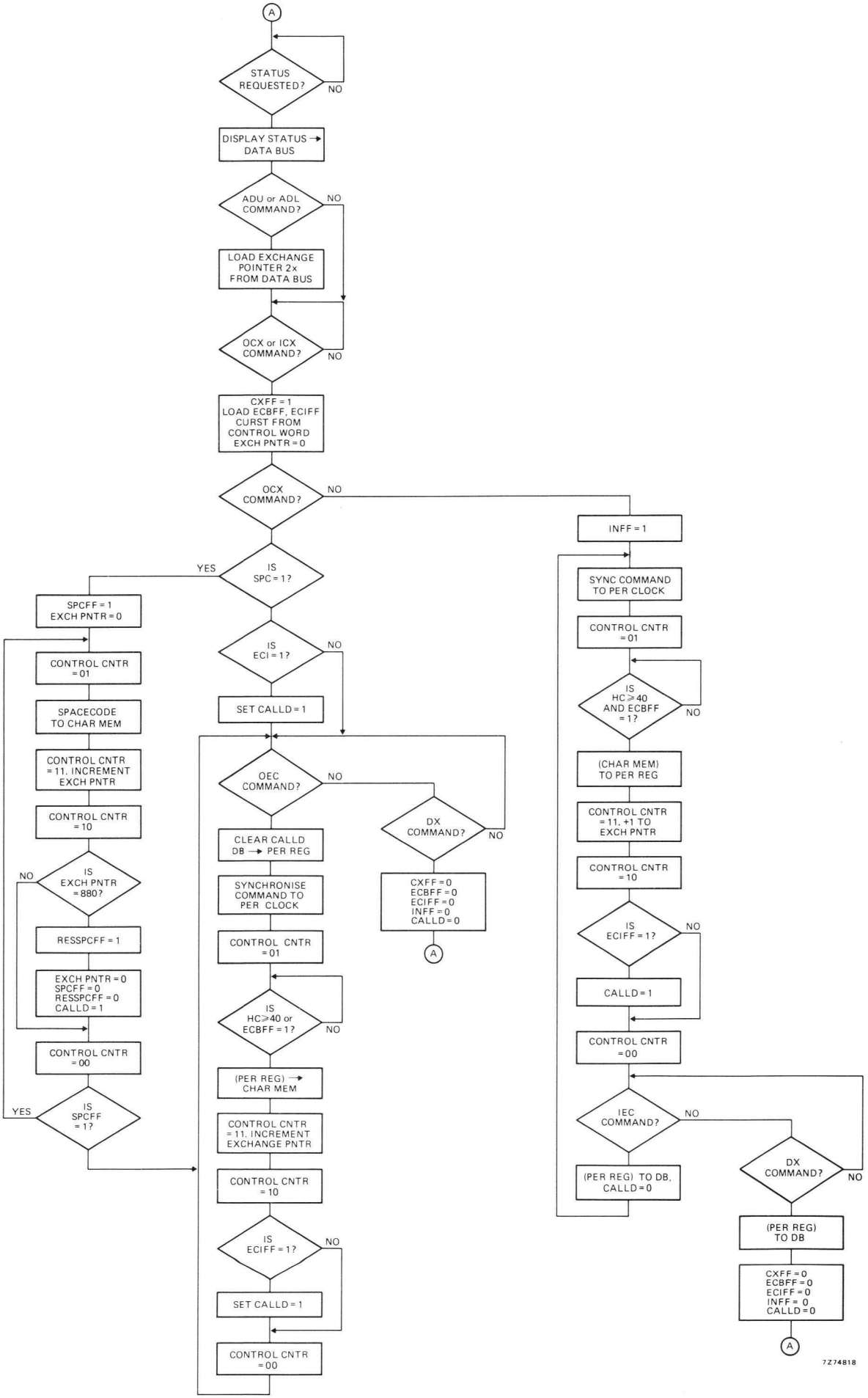


Fig. 7 Hardware operation sequence during data transfer.

Operation sequence

The data exchange must operate in one of three modes:
- input: data from character memory to data bus;
- output: data from data bus to character memory;
- space: space code to character memory.

The input and output modes are selected by the relevant connection command (ICX or OCX). The space mode is selected when the SPC bit in the control word is set.

Figure 7 shows the operation sequence in the form of a flow chart. Each data transfer sequence starts with a status request by the microcomputer, followed by ADU and ADL commands to set the exchange pointer, if required. When an OCX or IXC command is received, the control word bits are stored to determine the operation mode.

Space mode (SPCFF = 1, INFF = 0)

The character memory is filled with the code for spaces and the cursor is returned to the top left-hand position.

The space mode is entered after an OCX command is received with the SPC and ECI bits set in the control word. The control logic memorizes the mode by setting flip-flops SPCFF = 1 and INFF = 0. The exchange pointer is reset to zero (top left-hand position of screen), and the control state counter (CST1, CST0) is set to 01 while the space code is written into the character memory. The control state counter then changes to 11 and the exchange pointer is incremented. The control counter then changes to 10 and if the exchange pointer is less than 880, the sequence is repeated. Figure 8 shows the timing of the space mode.

When the exchange pointer reaches 880, all the character memory contains the space code; the exchange pointer and SPCFF flip-flop are reset. The CALLD flip-flop is then set to indicate that the screen-erase action is complete. The setting of the CALLD flip-flop causes an interrupt, thus requiring the ECI bit to have been set in the control word.

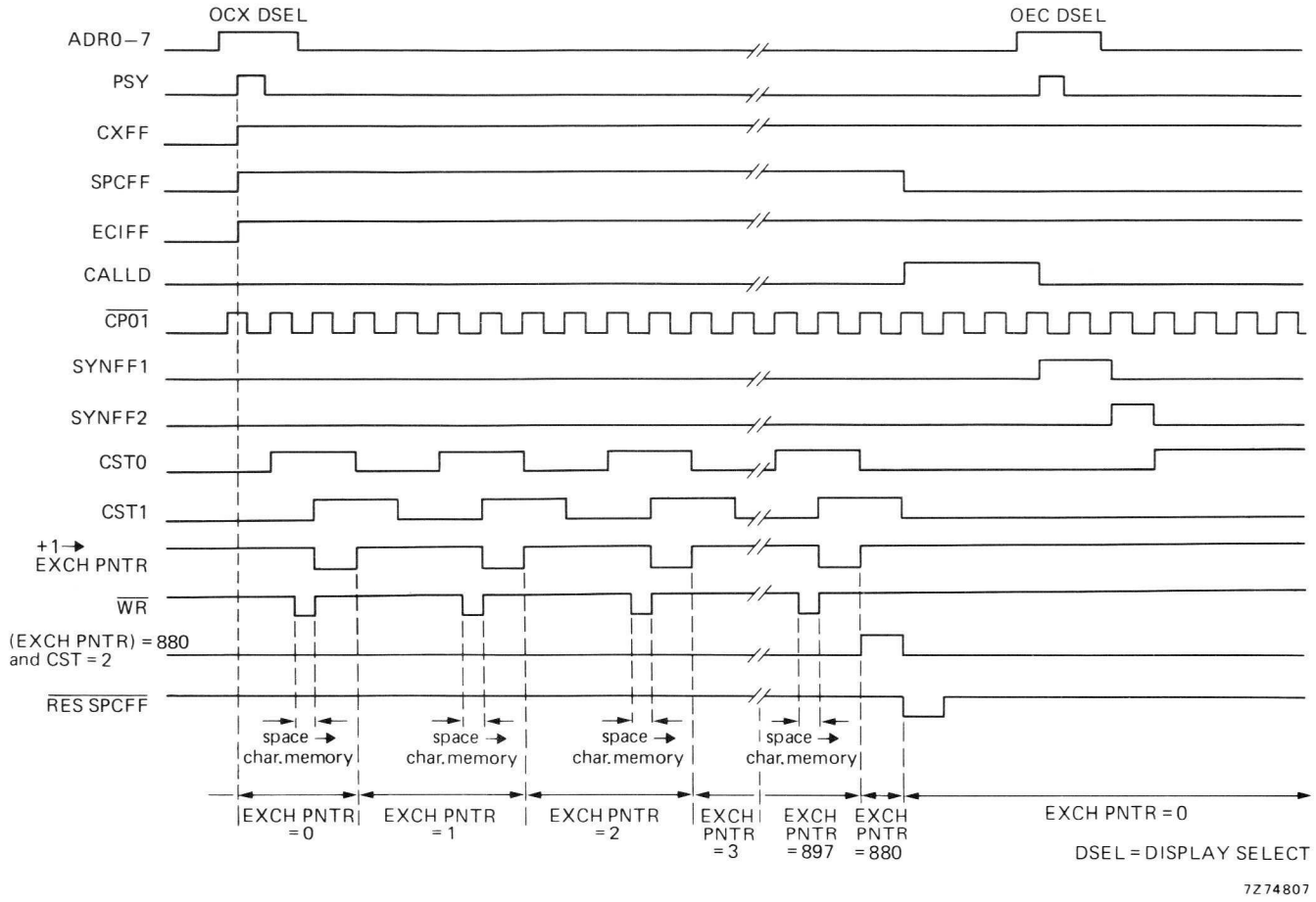


Fig. 8 Timing for the space mode.

Output mode (SPCFF = 0, INFF = 0)

Output can be controlled from the main program or from an interrupt routine, as determined by the ECI bit in the control word for the connection command. If the ECI bit is set, the CALLD flip-flop is set, resulting in an interrupt.

When it receives the OEC command, the control logic sets the control state counter to 01 and waits until the line flyback time starts. At this moment, the contents of the 8T31 peripheral register are transferred to the character memory, see Fig. 9. However, if the ECB bit was set in the control word, the logic does not wait for the flyback time and the character memory is updated immediately. At the next value of the control counter (11) the exchange pointer is incremented. The following control state (10) sets the CALLD flip-flop if more data is to be transferred. The logic then returns to the 00 state, waiting for an OEC or DX command.

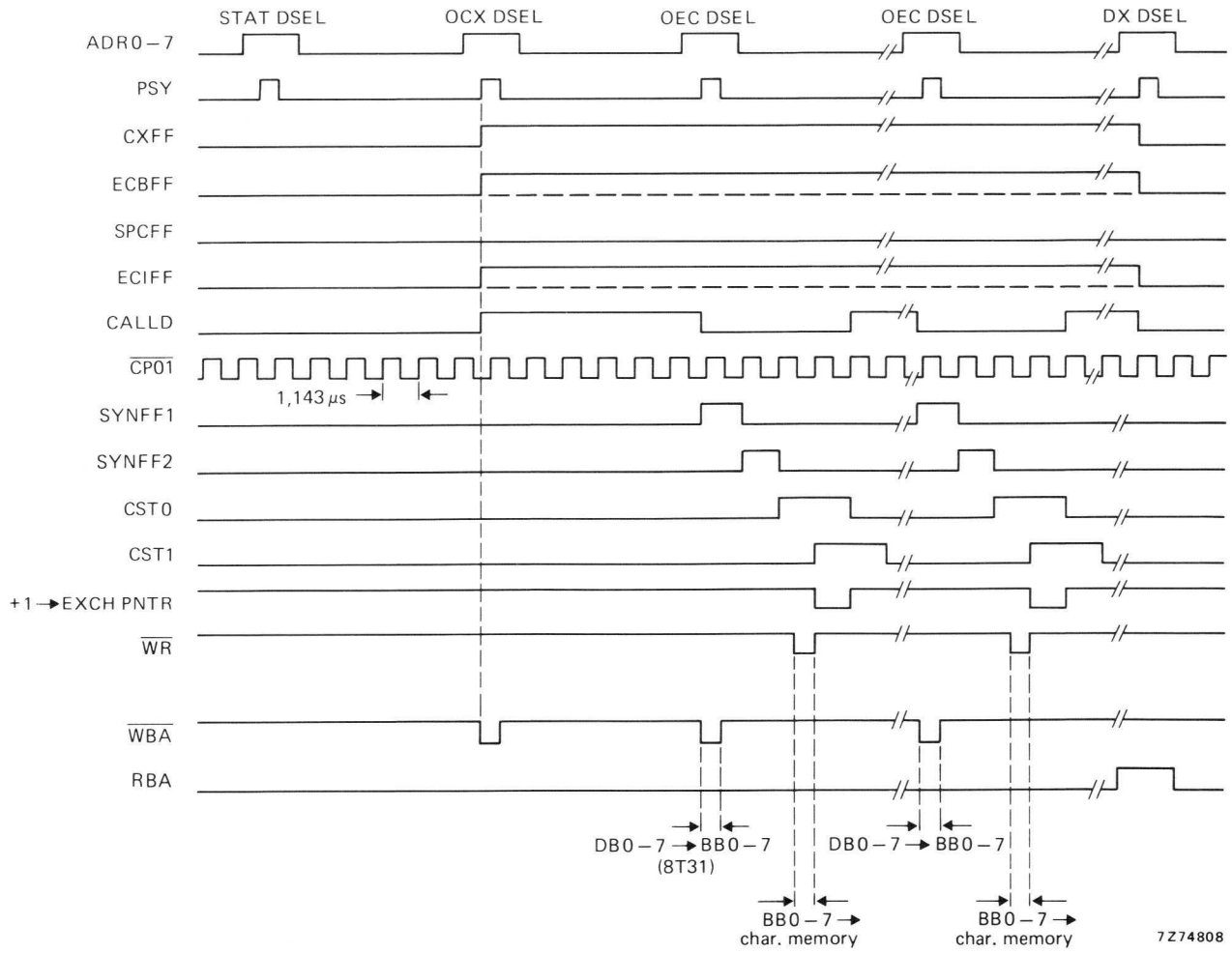


Fig. 9 Timing for the output mode.

Input mode (SPCFF = 0, INFF = 1)

Data input can be controlled by either main or interrupt program, depending upon the ECI in the control word.

An ICX command sets the control counter to 01 and the control logic then waits until the line flyback time before transferring the data byte, addressed by the exchange pointer, to the peripheral register. The exchange pointer is then incremented and the CALLD flip-flop is set if ECI is active. The control counter is then reset to 00 and the system waits for an IEC command to transfer the data from the peripheral register to the microcomputer. This procedure is repeated until a DX command is received from the microcomputer. The timing for the input mode is shown in Fig. 10.

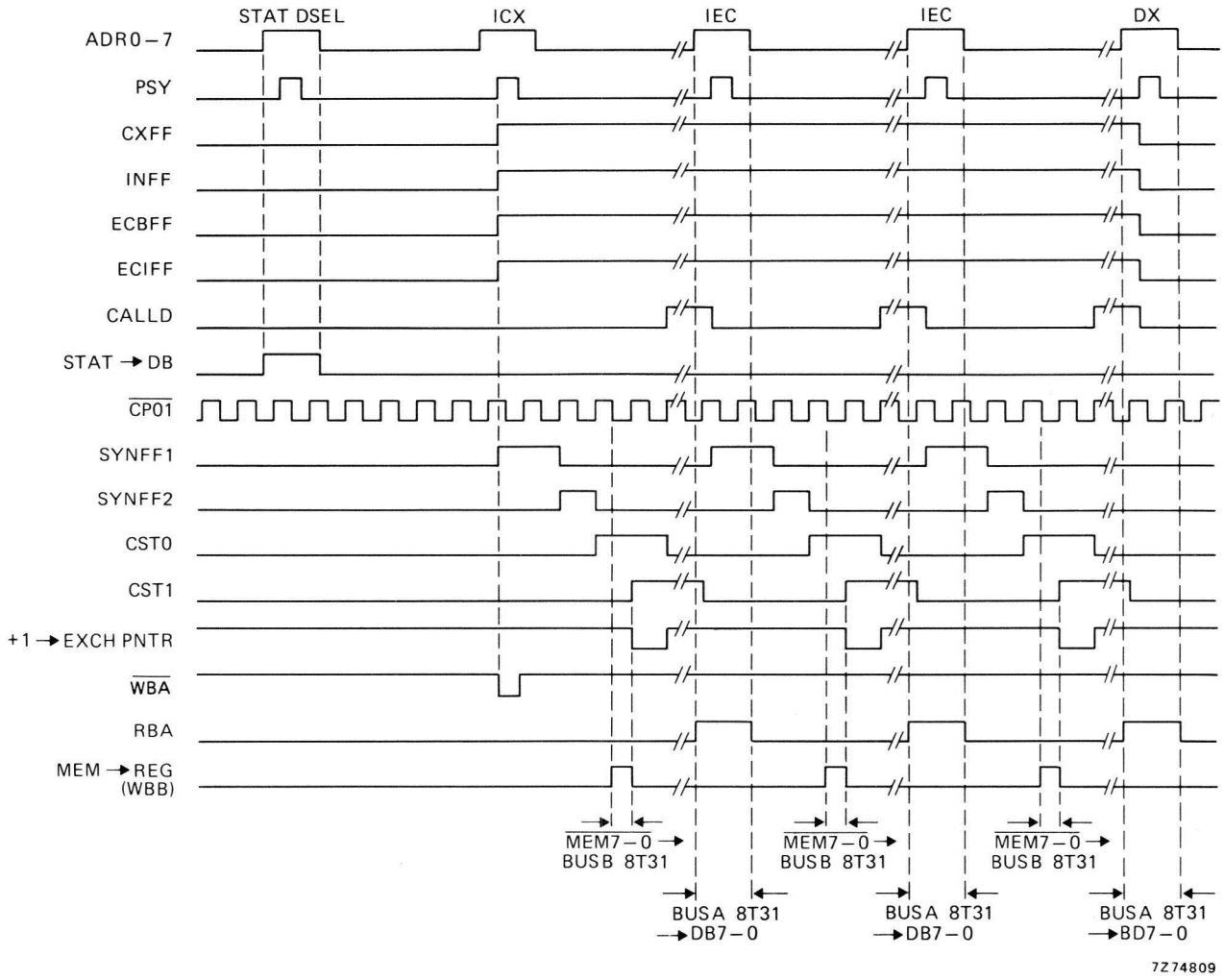


Fig. 10 Timing for the input mode.

Data exchange software

The software consists of two parts: the connection routine and the data transfer routine. The data transfer routine can be designed either as an interrupt routine or as part of the main program. Listings of the software can be found in the Appendix.

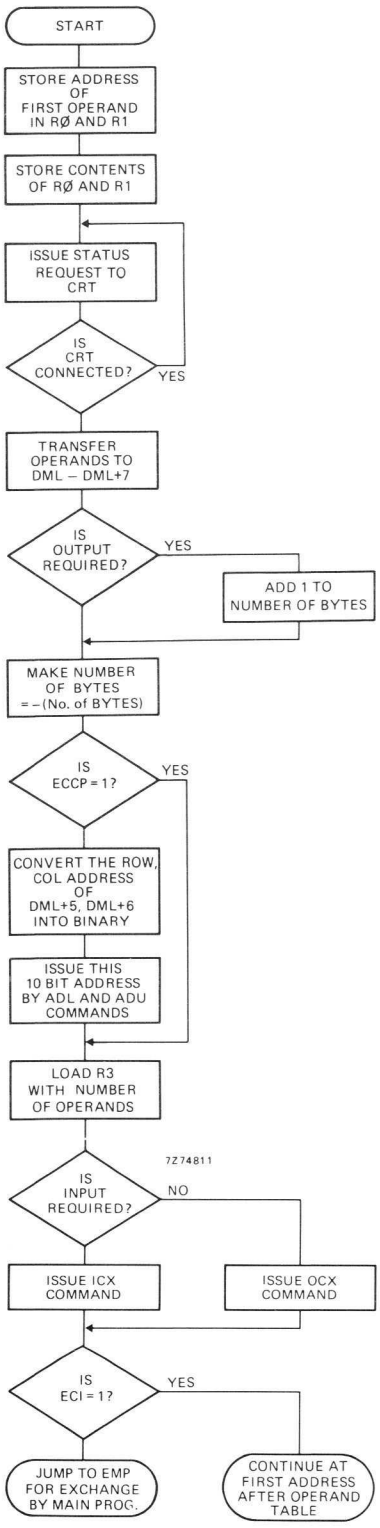


Fig. 11 Flow chart of the connection routine.

Connection routine

The connection is part of the main program: it takes care of the preparations for data transfer. Before connecting the peripheral it issues a status request to check that the peripheral is available and prepares the control word. If the peripheral is available, the appropriate command (ICX or OCX) is issued and the routine continues or branches to the data transfer part of the main program as required by the ECI bit.

A flow chart of the connection routine is shown in Fig. 11. If the status of the display is found to be correct, the routine stores the operand table in the display memory locations DML to DML+7. Table 2 gives details of the operands.

TABLE 2 Operand table for the display

location	number of bytes	operand
DML	1	control word
DML+1	2	number of bytes to be transferred
DML+3	2	address of first byte in computer memory
DML+5	2	address of first byte in character memory
DML+7	1	end character

The use of the four most significant bits of the control word has already been explained in the description of the data exchange logic. The four least significant bits of the control word are not used by the data exchange logic but are used for communication between the software routines. These bits are:

SOER: (bit 0) stop on end character required. If SOER = 1, the data transfer is halted when the end character specified in the operand table is encountered. If SOER = 0, the data transfer is halted after the number of bytes specified in the operand table.

I/$\overline{O}$: (bit 1) I/$\overline{O}$ = 1, the display is input device. I/$\overline{O}$ = 0, the display is output device.

ECCP: (bit 2) exchange at current cursor position if ECCP = 1. If ECCP = 0, exchange at screen position determined by the contents of DML+5 and DML+6.

Bit 3 of the control word is unused.

Once the operand table is loaded into the DML locations, a connection command is given (ICX or OCX) and the control word is output on the data bus. Only the four most significant bits are accepted by the data exchange logic. If the ECI bit is set in the control word, the program returns to the first address following the operand table and continues with the main program until an interrupt occurs. If the ECI bit is not set, the program jumps to label EMP in the main program, to perform the data transfer.

Data transfer routines

The data transfer routine can be a part of the main program or it can be a separate interrupt program. If the exchange-by-blocks and screen-erase functions are required, the data transfer must be under interrupt control.

Interrupt program

The flow chart of the interrupt program is shown in Fig. 12. The first task of the interrupt program is to save the microcomputer status word and the contents of any registers that are used in the interrupt program. After incrementing the byte counter and checking that there are more bytes to be transferred, an IEC or OEC command is issued. If there are no more bytes to be transferred, a disconnect command (DX) is given and the saved registers restored before returning to the main program. If, in the case of input from the display, the stop-on-end-character-required bit is set, a disconnect command is issued and the character received during the disconnect command is compared with the SOER character. If the characters are not identical, a connection command is issued, and control returns to the main program.

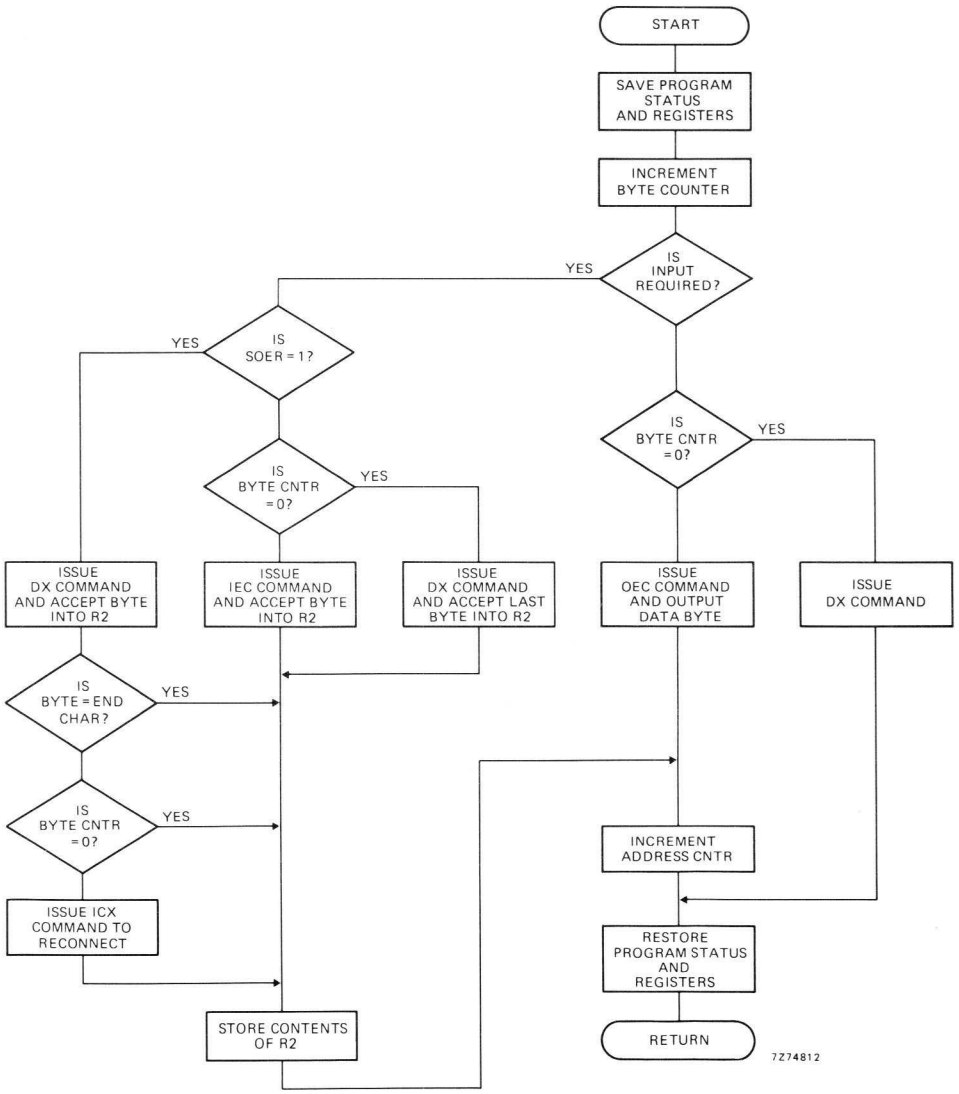


Fig. 12 Flow chart of the interrupt program.

Data transfer by main program

This routine is similar to the interrupt program except for the save and restore operations required by the interrupt software. In this routine, the number of operands (in R3) must be saved. This is done by reserving R3 for the exclusive use of the data transfer routine. The flow chart of the data transfer routine is shown in Fig. 13.

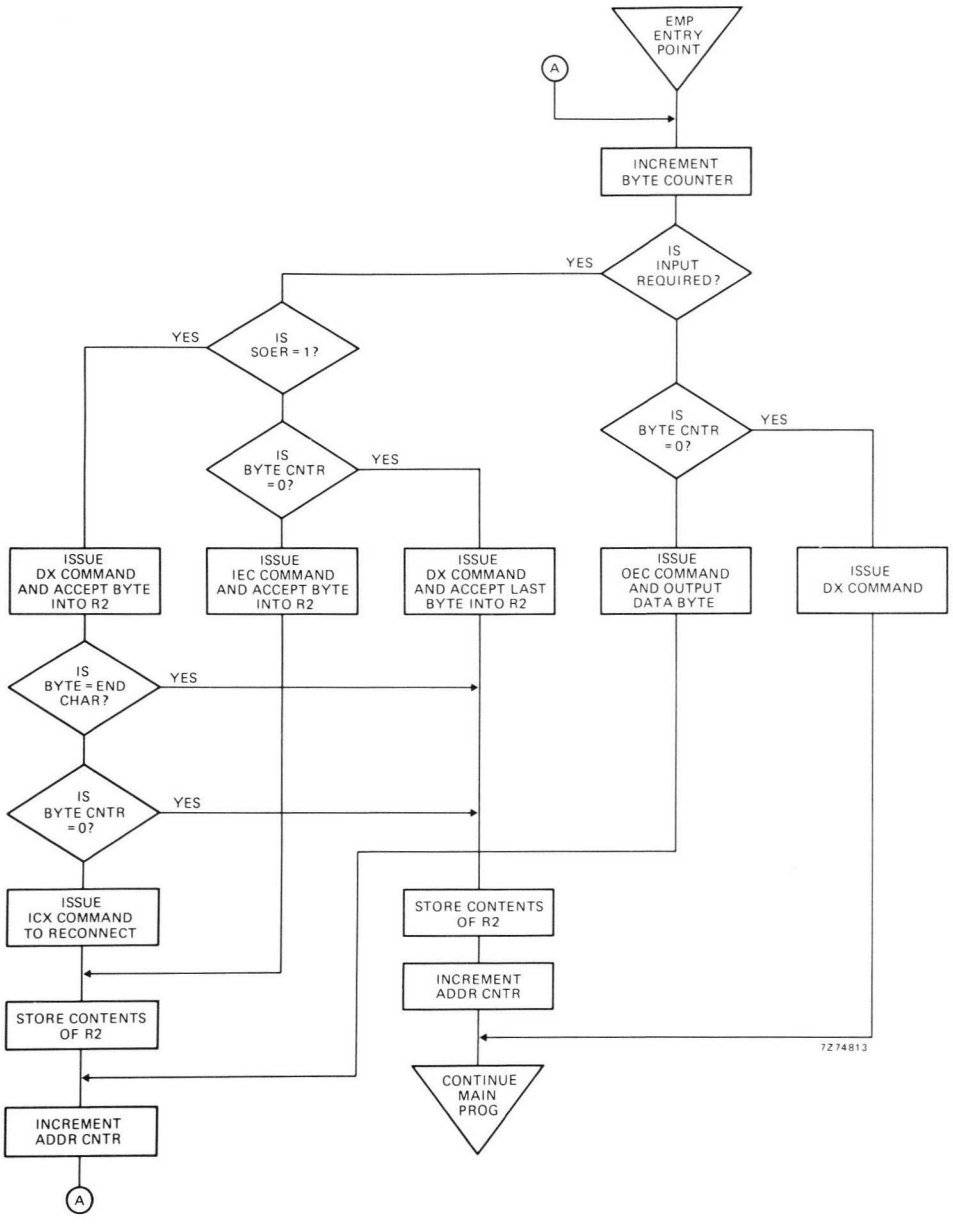


Fig. 13 Flow chart of the routine to transfer data during the main program.

Appendix

Assembler listings of the software routines for the CRT interface.

```
TWIN ASSEMBLER VER 2.0                                          PAGE 0001

LINE ADDR  OBJECT  E SOURCE

0001                *JMB 760915-15.00
0002                *THE FOLLOWING ROUTINES ARE USED TO ACCOMPLISH A
0003                *DATATRANSFER BETWEEN THE CRT DISPLAY AND A 2650
0004                *BASED SYSTEM.
0005                *IT CONSISTS OF A CONNECTION ROUTINE (EXECUTED DURING
0006                *MAIN PROGRAM), AN INTERRUPT ROUTINE AND AN
0007                *EXCHANGE ROUTINE EXECUTED BY THE MAIN PROGRAM.
0008                ************************************************
0009                *
0010                * DEFINITIONS OF SYMBOLS:
0011                *
0012 0000          R0   EQU    0          PROCESSOR REGISTERS
0013 0001          R1   EQU    1
0014 0002          R2   EQU    2
0015 0003          R3   EQU    3
0016 0080          S    EQU    H'80'      PSU: SENSE
0017 0040          F    EQU    H'40'           FLAG
0018 0020          II   EQU    H'20'           INTERRUPT INHIBIT
0019 0007          SP   EQU    H'07'           STACKPOINTER
0020 00C0          CC   EQU    H'C0'      PSL: CONDITION CODE
0021 0020          IDC  EQU    H'20'           INTER DIGIT CARRY
0022 0010          RS   EQU    H'10'           REGISTER BANK SELECT
0023 0008          WC   EQU    H'08'           1=WITH, 0=NO CARRY
0024 0004          OVF  EQU    H'04'           OVERFLOW
0025 0002          COM  EQU    H'02'           1=LOG., 0=ARITH. COMP.
0026 0001          C    EQU    H'01'           CARRY/NO BORROW
0027 0000          Z    EQU    0          BRANCH COND: ZERO
0028 0001          P    EQU    1                       POSITIVE
0029 0002          N    EQU    2                       NEGATIVE
0030 0000          EQ   EQU    0                       EQUAL
0031 0001          GT   EQU    1                       GREATER THAN
0032 0002          LT   EQU    2                       LESS THAN
0033 0003          UN   EQU    3                       UNCONDITIONAL
0034 0000          A1   EQU    0                       ALL BITS ARE 1
0035 0002          N1   EQU    2                  NOT ALL BITS ARE 1
0036                *
0037 0004          CRT  EQU    4          CRT DISPLAY
0038 00A0          STAT EQU    H'A0'      STATUS COMMAND
0039 0040          OCX  EQU    H'40'      OUTPUT CONNECTION COMMAND
0040 0080          OEC  EQU    H'80'      OUTPUT EXCHANGE COMMAND
0041 00C0          ICX  EQU    H'C0'      INPUT CONNECTION COMMAND
0042 0020          IEC  EQU    H'20'      INPUT EXCHANGE COMMAND
0043 00E0          DX   EQU    H'E0'      DISCONNECTION COMMAND
0044 0000          ADU  EQU    H'00'      PRESET CHAR. MEM. ADDR. U COMMAND
0045 0060          ADL  EQU    H'60'      PRESET CHAR. MEM. ADDR. L COMMAND
0046                *
0047                *VARIABLES
0048 0000               ORG    H'300'
0049 0300          DML  RES    8          DISPLAY MEMORYLOCATIONS
0050 0308          RET  RES    2          KEEPS ADDRESS FIRST OPERAND
0051 030A          LOC  RES    2          SAVE LOCATIONS OF R0 AND PSL
```

```
LINE ADDR  OBJECT  E SOURCE

0053                *CONNECTION ROUTINE FOR CRT DISPLAY
0054                *
0055                *DURING MAIN PROGRAM
0056 030C                ORG     H'80'
0057 0080 7519           CPSL    RS+WC+C
0058 0082 0400           LODI,R0 <$+7        FETCH ADDR. H OF FIRST OPERAND
0059 0084 0589           LODI,R1 >$+5        FETCH ADDR. L OF FIRST OPERAND
0060 0086 1F0100         BCTA,UN DCXR        CALL CRT CONNECTION ROUTINE
0061 0089 A2             DATA    H'A2'       CONTROLWORD (EXAMPLE)
0062 008A 0014           DATA    D'0,20'     NUMBER OF BYTES (EXAMPLE)
0063 008C 0400           DATA    H'4,0'      ADDRESS OF FIRST BYTE (EXAMPLE)
0064 008E 0C1E           DATA    D'12,30'    COORDINATES  OF FIRST BYTE IN CHAR. MEM.
0065                *                          ROW = 12, COLUMN = 30 (EXAMPLE)
0066 0090 58             DATA    A'X'        ENDCHARACTER (EXAMPLE)
0067                *
0068 0091                ORG     H'100'
0069                *
0070 0100 CC0308    DCXR STRA,R0 RET         SAVE ADDR. FIRST OPER.
0071 0103 CD0309         STRA,R1 RET+1
0072 0106 54A4      DSTT REDE,R0 STAT+CRT    READ CRT STATUS
0073 0108 4420           ANDI,R0 H'20'       CLEAR BITS EXCEPT BIT 5
0074 010A 987A           BCFR,Z  DSTT        BR. IF CRT CONNECTED
0075                *
0076 010C 0708           LODI,R3 8           LOAD R3 WITH NUMBER OF OPER.
0077 010E 0FC308    DOPT LODA,R0 *RET,R3,-   TRANSFER OPERANDS
0078 0111 CF6300         STRA,R0 DML,R3        INTO DISPLAY MEM. LOCATIONS
0079 0114 5B78           BRNR,R3 DOPT          DML---DML+7
0080                *
0081 0116 0D0300         LODA,R1 DML         FETCH CONTROLWORD
0082 0119 F501           TMI,R1  H'01'       TEST BIT 0 (I/O)
0083 011B 9802           BCFR,A1 ONBC        GO TO ONBC IF OUTPUT
0084                *
0085 011D 7709           PPSL    WC+C        ADD WITH CARRY
0086 011F 0702      ONBC LODI,R3 2
0087 0121 0F4301    DNBC LODA,R0 DML+1,R3,-  CHANGE NUMB. OF BYTES INTO
0088 0124 24FF           EORI,R0 H'FF'         -(NUMB. OF BYTES) FOR INPUT OR
0089 0126 8400           ADDI,R0 0             -(NUMB. OF BYTES+1) FOR OUTPUT
0090 0128 CF6301         STRA,R0 DML+1,R3
0091 012B 5B74           BRNR,R3 DNBC
0092                *
0093 012D F504           TMI,R1  H'04'       TEST BIT 2 (ECCP)
0094 012F 9816           BCFR,A1 CONV        BRANCH IF ECCP=0
0095 0131 0708      CTWS LODI,R3 8           NUMBER OF OPERANDS INTO R3
0096 0133 F501           TMI,R1  H'01'       TEST BIT 0 (I/O)
0097 0135 980C           BCFR,A1 OTP         BRANCH IF OUTPUT
0098                *
0099 0137 D5C4           WRTE,R1 ICX+CRT     CONNECT CRT IN INPUTMODE
0100                *
```

```
LINE ADDR  OBJECT  E SOURCE

0102 0139 F510      TECI TMI,R1  H'10'     TEST BIT 4 (ECI)
0103 013B 1803           BCTR,A1 LOCB      BR. IF EXCH. BY INTERR. REQ.
0104 013D 1F01DB         BCTA,UN EMP       BR. TO MAIN PROGRAM EXCH.
0105 0140 9F8308    LOCB BXA     *RET      RETURN TO MAIN PROGRAM
0106                *
0107 0143 D544      OTP  WRTE,R1 OCX+CRT   CONNECT CRT IN OUTPUTMODE
0108 0145 1B72           BCTR,UN TECI
0109                *
0110 0147 0C0305    CONV LODA,R0 DML+5     FETCH ROW OF DISPLAY
0111 014A C3             STRZ    R3          INTO R0 AND R3
0112 014B 7509           CPSL    WC+C
0113 014D D0             RRL,R0            R0 2 X ENLARGED
0114 014E D0             RRL,R0            R0 4 X ENLARGED
0115 014F 83             ADDZ    R3        R0 5 X ENLARGED
0116 0150 0600           LODI,R2 0
0117 0152 7708           PPSL    WC        ROTATE WITH CARRY
0118 0154 D0             RRL,R0
0119 0155 D2             RRL,R2
0120 0156 D0             RRL,R0
0121 0157 D2             RRL,R2
0122 0158 D0             RRL,R0
0123 0159 D2             RRL,R2            R2,R0 CONTAIN 40 X ROW
0124 015A 8C0306         ADDA,R0 DML+5
0125 015D 8600           ADDI,R2 0         R2,R0 CONTAIN COL + 40 X ROW
0126 015F D604           WRTE,R2 ADU+CRT   OUTPUT 2 HIGHER BITS OF CHAR. ADDR.
0127 0161 D464           WRTE,R0 ADL+CRT   OUTPUT 2 LOWER BITS OF CHAR. ADDR.
0128 0163 1B4C           BCTR,UN CTWS
```

```
LINE ADDR  OBJECT  E SOURCE

0130                *INTERRUPT ROUTINE FOR CRT
0131                *
0132 0165 3F01C4          BSTA,UN SAVE      SAVE REGISTERS, STATUS
0133 0168 0501            LODI,R1 1
0134 016A 0C0302          LODA,R0 DML+2     FETCH BYTECTR. L
0135 016D D808            BIRR,R0 RST1      INCREMENT, IF ZERO
0136 016F 0D0301          LODA,R1 DML+1     FETCH BYTECTR. H
0137 0172 D900            BIRR,R1 RST2      INCREMENT BYTECTR. H
0138 0174 CD0301    RST2  STRA,R1 DML+1     STORE BYTECTR. H
0139 0177 CC0302    RST1  STRA,R0 DML+2     STORE BYTECTR. L
0140                *
0141 017A 0C0300          LODA,R0 DML       FETCH CONTR. WORD
0142 017D F401            TMI,R0  H'01'     TEST BIT 0 (I/O)
0143 017F 9823            BCFR,A1 OPD       BR. IF CRT IS OUTPUTDEVICE
0144 0181 F402            TMI,R0  H'02'     TEST BIT 1 (SOER)
0145 0183 9814            BCFR,A1 SOBP      BR. IF STOP ON NUMB. OF BYTES
0146                *
0147 0185 56E4            REDE,R2 DX+CRT    INPUT BYTE TO R2, DISCONN.
0148 0187 EE0307          COMA,R2 DML+7     COMPARE BYTE AND ENDCHAR.
0149 018A 1808            BCTR,EQ STOM      BRANCH IF EQUAL
0150 018C 01              LODZ    R1        FETCH BYTECTR. H
0151 018D 1805            BCTR,Z  STOM      BR. IF BYTECTR.=0
0152 018F 0C0300          LODA,R0 DML       FETCH CONTR. WORD
0153 0192 D4C4            WRTE,R0 ICX+CRT   CONNECT CRT, WRITE CONTR.WORD
0154 0194 CE8303    STOM  STRA,R2 *DML+3    (R2) TO MEM. POINTED BY ADDR. CTR.
0155 0197 1B13            BCTR,UN INAD
0156                *
0157 0199 01        SOBP  LODZ    R1        FETCH BYTECTR. H
0158 019A 1804            BCTR,Z  TRE1      BR. IF BYTECTR.=0
0159 019C 5624            REDE,R2 IEC+CRT   INPUT BYTE
0160 019E 1B74            BCTR,UN STOM
0161 01A0 56E4      TRE1  REDE,R2 DX+CRT    INPUT BYTE,DISCONNECT CRT
0162 01A2 1B70            BCTR,UN STOM
0163                *
0164 01A4 01        OPD   LODZ    R1        FETCH BYTECTR. H
0165 01A5 1819            BCTR,Z  TRE2      BR. IF BYTECTR.=0
0166 01A7 0C8303          LODA,R0 *DML+3    FETCH BYTE
0167 01AA D484            WRTE,R0 OEC+CRT   OUTPUT BYTE
0168                *
0169 01AC 0C0304    INAD  LODA,R0 DML+4     DOUBLE LENGTH INCR.
0170 01AF D808            BIRR,R0 RST4        OF ADDR.CTR.
0171 01B1 0D0303          LODA,R1 DML+3
0172 01B4 D900            BIRR,R1 RST3
0173 01B6 CD0303    RST3  STRA,R1 DML+3
0174 01B9 CC0304    RST4  STRA,R0 DML+4
0175                *
0176 01BC 3F01CE    RST5  BSTA,UN REST      RESTORE REGISTERS, STATUS
0177 01BF 37              RETE,UN
0178 01C0 54E4      TRE2  REDE,R0 DX+CRT    DISCONNECT CRT
0179 01C2 1B78            BCTR,UN RST5
0180                *
```

```
LINE ADDR  OBJECT  E SOURCE

0182                *SUBROUTINES
0183 01C4 CC030A    SAVE STRA,R0 LOC       SAVE R0 IN LOC
0184 01C7 7710           PPSL    RS        SELECT REG. BANK #1
0185 01C9 13             SPSL              (PSL) INTO R0
0186 01CA CC030B         STRA,R0 LOC+1     SAVE (PSL) IN LOC+1
0187 01CD 17             RETC,UN           RETURN
0188                *
0189                *
0190 01CE 0C030B    REST LODA,R0 LOC+1     FETCH PSL IN R0
0191 01D1 C1             STRZ    R1        (R0) TO R1
0192 01D2 93             LPSL              (R0) TO PSL
0193 01D3 0C030A         LODA,R0 LOC       FETCH OLD VALUE R0
0194 01D6 45C0           ANDI,R1 H'C0'     RECONSTR. CC IN PSL
0195 01D8 7510           CPSL    RS        SELECT REG. BANK #0
0196 01DA 17             RETC,UN           RETURN
0197                *
```

```
LINE ADDR  OBJECT  E SOURCE

0199               *ROUTINE FOR EXCHANGE BY MAIN PROGRAM
0200               *
0201 01DB 0501     EMP  LODI,R1 1
0202 01DD 0C0302        LODA,R0 DML+2      FETCH BYTECTR. L
0203 01E0 D808          BIRR,R0 RST6       INCREMENT, IF ZERO
0204 01E2 0D0301        LODA,R1 DML+1      FETCH BYTECTR. H
0205 01E5 D900          BIRR,R1 RST7       INCREMENT BYTECTR. H
0206 01E7 CD0301   RST7 STRA,R1 DML+1      STORE BYTECTR. H
0207 01EA CC0302   RST6 STRA,R0 DML+2      STORE BYTECTR. L
0208               *
0209 01ED 0C0300        LODA,R0 DML        FETCH CONTR. WORD
0210 01F0 F401          TMI,R0  H'01'      TEST BIT 0 (I/O)
0211 01F2 9824          BCFR,A1 OPDE       BR. IF CRT IS OUTPUTDEVICE
0212 01F4 F402          TMI,R0  H'02'      TEST BIT 1 (SOER)
0213 01F6 9813          BCFR,A1 SBP        BR. IF STOP ON NUMB. OF BYTES
0214               *
0215 01F8 56E4          REDE,R2 DX+CRT     INPUT BYTE TO R2, DISCONN.
0216 01FA EE0307        COMA,R2 DML+7      COMPARE BYTE AND ENDCHAR.
0217 01FD 1815          BCTR,EQ CSUB       BRANCH TO CSUB IF EQUAL
0218 01FF 01            LODZ    R1         FETCH BYTECTR. H
0219 0200 1812          BCTR,Z  CSUB       BR. TO CSUB IF BYTECTR.=0
0220 0202 0C0300        LODA,R0 DML        FETCH CONTR. WORD
0221 0205 D4C4          WRTE,R0 ICX+CRT    CONNECT CRT, WRITE CONTR.WORD
0222 0207 3B21     LOC1 BSTR,UN STIN       BRANCH TO SUBR. STIN
0223 0209 1B50          BCTR,UN EMP
0224               *
0225 020B 01       SBP  LODZ    R1         FETCH BYTECTR. H
0226 020C 1804          BCTR,Z  TRE3       BR. IF BYTECTR.=0
0227 020E 5624          REDE,R2 IEC+CRT    INPUT BYTE
0228 0210 1B75          BCTR,UN LOC1
0229 0212 56E4     TRE3 REDE,R2 DX+CRT     INPUT BYTE,DISCONNECT CRT
0230 0214 3B14     CSUB BSTR,UN STIN       BRANCH TO SUBR. STIN
0231 0216 1B0F          BCTR,UN RTN
0232               *
0233 0218 01       OPDE LODZ    R1         FETCH BYTECTR. H
0234 0219 180A          BCTR,Z  TRE4       BR. IF BYTECTR.=0
0235 021B 0C8303        LODA,R0 *DML+3     FETCH BYTE
0236 021E D484          WRTE,R0 OEC+CRT    OUTPUT BYTE
0237 0220 3B0B          BSTR,UN ICAD       BRANCH TO SUBR. ICAD
0238 0222 1F01DB        BCTA,UN EMP        BRANCH TO START
0239               *
0240 0225 54E4     TRE4 REDE,R0 DX+CRT     DISCONNECT CRT
0241 0227 9F8308   RTN  BXA     *RET       RETURN
```

```
TWIN ASSEMBLER VER 2.0                                          PAGE 0007

LINE ADDR  OBJECT  E SOURCE

0243               *SUBROUTINE
0244               *
0245 022A CE8303   STIN STRA,R2 *DWL+3     (R2) TO MEM. POINTED BY ADDR.CTR.
0246 022D 0C0304   ICRD LODA,R0 DWL+4      DOUBLE LENGTH INCR.
0247 0230 D808          BIRR,R0 RST9         OF ADDR.CTR.
0248 0232 0D0303        LODA,R1 DWL+3
0249 0235 D900          BIRR,R1 RST8
0250 0237 CD0303   RST8 STRA,R1 DWL+3
0251 023A CC0304   RST9 STRA,R0 DWL+4
0252 023D 17            RETC,UN
0253               *
0254 0080               END     H'80'

 TOTAL ASSEMBLY ERRORS = 0000
```

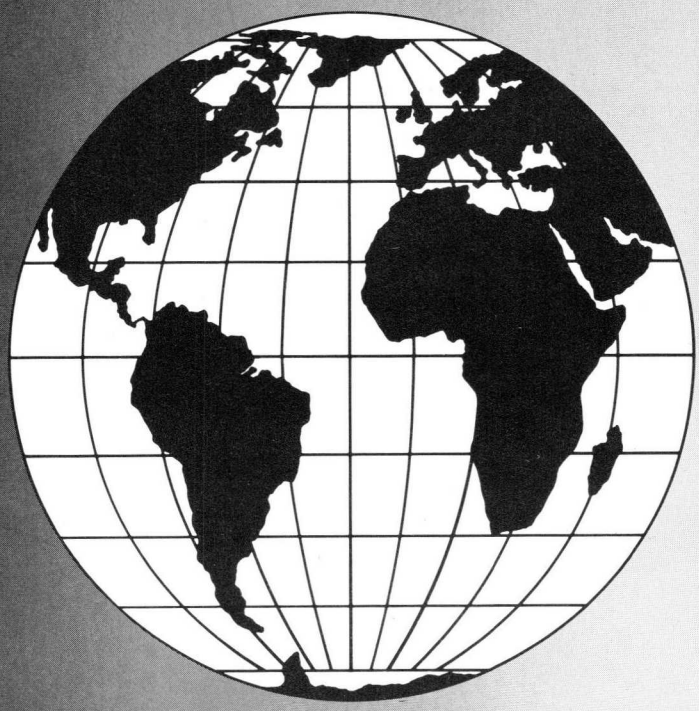
Related 2650 publications

no.	title	summary
AS50	Serial Input/Output	Using the Sense/Flag capability of the 2650 for serial I/O interfaces.
AS51	Bit & Byte Testing Procedures	Several methods of testing the contents of the internal registers in the 2650.
AS52	General Delay Routines	Several time delay routines for the 2650, including formulas for calculating the delay time.
AS52	Binary Arithmetic Routines	Examples for processing binary arithmetic addition, subtraction, multiplication, and division with the 2650.
AS54	Conversion Routines	• Eight-bit unsigned binary to BCD • Sixteen-bit signed binary to BCD • Signed BCD to binary • Signed BCD to ASCH • ASCII to BCD • Hexadecimal to ASCII • ASCII to Hexadecimal
AS55	Fixed Point Decimal Arithmetic	Methods of performing addition, subtraction, multiplication and division of BCD numbers with the 2650.
SP50	2650 Evaluation Printed Circuit Board (PC1001)	Detailed description of the PC1001, an evaluation and design tool for the 2650.
SP51	2650 Demo System	Detailed description of the Demo System, a hardware base for use with the 2650 CPU prototyping board (PC1001 or PC1500).
SP52	Support Software for use with the NCSS Timesharing System	Step-by-step procedures for generating, editing, assembling, punching, and simulating Signetics 2650 programs using the NCSS timesharing service.
SP53	Simulator, Version 1.2	Features and characteristics of version 1.2 of the 2650 simulator.
SP54	Support Software for use with the General Electric Mark III Timesharing System	Step-by-step procedures for generating, editing, assembling, simulating, and punching Signetics 2650 programs using General Electric's Mark III timesharing system.
SP55	The ABC 1500 Adaptable Board Computer	Describes the components and applications of the ABC 1500 system development card.
SS50	PIPBUG	Detailed description of PIPBUG, a monitor program designed for use with the 2650.
SS51	Absolute Object Format	Describes the absolute object code format for the 2650.
MP51	Initialization	Procedures for initializing the 2650 microprocessor, memory, and I/O devices to their required initial states.
MP52	Low-Cost Clock Generator Circuits	Several clock generator circuits, based on 7400 series TTL, that may be used with the 2650. They include RC, LC and crystal oscillator types.
MP53	Address and Data Bus Interfacing Techniques	Examples of interfacing the 2650 address and data busses with ROMs and RAMs, such as the 2608, 2606 and 2602.
MP54	2650 Input/Output Structures and Interfaces	Examines the use of the 2650's versatile set of I/O instructions and the interface between the 2650 and I/O ports. A number of application examples for both serial and parallel I/O are given.
TN 064	Digital cassette interface for a 2650 microprocessor system	Interface hardware and software for the Philips DCR digital cassette drive.
TN 069	2650 Microprocessor keyboard interfaces	Simple interfaces for low-cost keyboard systems.
TN 072	Introducing the Signetics 2651 PCI Terminology and operation modes	Description of the 2651 Programmable Communications Interface IC.
TN 083	Using the Signetics 2651 PCI with popular microprocessors	Simple hardware interfaces to use the 2651 Programmable Communications Interface with various microprocessors.
TN 084	Using seven-segment LED display with the 2650 microprocessor	Interfaces for single and multi-digit LED displays.
TN 085	Cyclic redundancy check by software	A short routine to encode and decode CRC check characters for the 2650.
TN 086	Introducing the Signetics 2655 PPI	Description of the 2655 Programmable Peripheral Interface.
TN 087	Audio cassette recorder interface for the 2650 microprocessor	Economical alternatives to the digital cassette recorder.

Electronic components and materials

for professional, industrial and consumer uses

from the world-wide Philips Group of Companies



Argentina: FAPESA I.y.C., Av. Crovara 2550, Tablada, Prov. de BUENOS AIRES, Tel. 652-7438/7478.
Australia: PHILIPS INDUSTRIES HOLDINGS LTD., Elcoma Division, 67 Mars Road, LANE COVE, 2066, N.S.W., Tel. 427 08 88.
Austria: ÖSTERREICHISCHE PHILIPS BAUELEMENTE Industrie G.m.b.H., Triester Str. 64, A-1101 WIEN, Tel. 62 91 11.
Belgium: M.B.L.E., 80, rue des Deux Gares, B-1070 BRUXELLES, Tel 523 00 00.
Brazil: IBRAPE, Caixa Postal 7383, Av. Paulista 2073-S/Loja, SAO PAULO, SP, Tel. 284-4511.
Canada: PHILIPS ELECTRONICS LTD., Electron Devices Div., 601 Milner Ave., SCARBOROUGH, Ontario, M1B 1M8, Tel. 292-5161.
Chile: PHILIPS CHILENA S.A., Av. Santa Maria 0760, SANTIAGO, Tel. 39-40 01.
Colombia: SADAPE S.A., P.O. Box 9805, Calle 13, No. 51 + 39, BOGOTA D.E. 1., Tel. 600 600.
Denmark: MINIWATT A/S, Emdrupvej 115A, DK-2400 KØBENHAVN NV., Tel. (01) 69 16 22.
Finland: OY PHILIPS AB, Elcoma Division, Kaivokatu 8, SF-00100 HELSINKI 10, Tel. 1 72 71.
France: R.T.C. LA RADIOTECHNIQUE-COMPELEC, 130 Avenue Ledru Rollin, F-75540 PARIS 11, Tel. 355-44-99.
Germany: VALVO, UB Bauelemente der Philips G.m.b.H., Valvo Haus, Burchardstrasse 19, D-2 HAMBURG 1, Tel. (040) 3296-1.
Greece: PHILIPS S.A. HELLENIQUE, Elcoma Division, 52, Av. Syngrou, ATHENS, Tel. 915 311.
Hong Kong: PHILIPS HONG KONG LTD., Comp. Dept., Philips Ind. Bldg., Kung Yip St., K.C.T.L. 289, KWAI CHUNG, N.T. Tel. 12-24 51 21.
India: PHILIPS INDIA LTD., Elcoma Div., Band Box House, 254-D, Dr. Annie Besant Rd., Prabhadevi, BOMBAY-25-DD, Tel. 457 311-5.
Indonesia: P.T. PHILIPS-RALIN ELECTRONICS, Elcoma Division, 'Timah' Building, Jl. Jen. Gatot Subroto, JAKARTA, Tel. 44 163.
Ireland: PHILIPS ELECTRICAL (IRELAND) LTD., Newstead, Clonskeagh, DUBLIN 14, Tel. 69 33 55.
Italy: PHILIPS S.p.A., Sezione Elcoma, Piazza IV Novembre 3, I-20124 MILANO, Tel. 2-6994.
Japan: NIHON PHILIPS CORP., Shuwa Shinagawa Bldg., 26-33 Takanawa 3-chome, Minato-ku, TOKYO (108), Tel. 448-5611.
(IC Products) SIGNETICS JAPAN, LTD., TOKYO, Tel. (03) 230-1521.
Korea: PHILIPS ELECTRONICS (KOREA) LTD., Elcoma Division, Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL, Tel. 794-4202.
Mexico: ELECTRONICA S.A. de C.V., Varsovia No. 36, MEXICO 6, D.F., Tel. 533-11-80.
Netherlands: PHILIPS NEDERLAND B.V., Afd. Elonco, Boschdijk 525, NL 5600 PD EINDHOVEN, Tel. (040) 79 33 33.
New Zealand: PHILIPS Electrical Ind. Ltd., Elcoma Division, 2 Wagener Place, St. Lukes, AUCKLAND, Tel. 867 119.
Norway: NORSK A/S PHILIPS, Electronica Dept., Vitaminveien 11, Grefsen, OSLO 4, Tel. (02) 15 05 90.
Peru: CADESA, Rocca de Vergallo 247, LIMA 17, Tel. 62 85 99.
Philippines: ELDAC, Philips Industrial Dev. Inc., 2246 Pasong Tamo, MAKATI-RIZAL, Tel. 86-89-51 to 59.
Portugal PHILIPS PORTUGESA S.A.R.L., Av. Eng. Duharte Pacheco 6, LISBOA 1, Tel. 68 31 21.
Singapore: PHILIPS SINGAPORE PTE LTD., Elcoma Div., P.O.B. 340, Toa Payoh CPO, Lorong 1, Toa Payoh, SINGAPORE 12, Tel. 53 88 11.
South Africa: EDAC (Pty.) Ltd., South Park Lane, New Doornfontein, JOHANNESBURG 2001, Tel. 24/6701.
Spain: COPRESA S.A., Balmes 22, BARCELONA 7, Tel. 301 63 12.
Sweden: A.B. ELCOMA, Lidingövägen 50, S-10 250 STOCKHOLM 27, Tel. 08/67 97 80.
Switzerland: PHILIPS A.G., Elcoma Dept., Edenstrasse 20, CH-8027 ZÜRICH, Tel. 01/44 22 11.
Taiwan: PHILIPS TAIWAN LTD., 3rd Fl., San Min Building, 57-1, Chung Shan N. Rd, Section 2, P.O. Box 22978, TAIPEI, Tel. 5513101-5.
Turkey: TÜRK PHILIPS TICARET A.S., EMET Department, Inonu Cad. No. 78-80, ISTANBUL, Tel. 43 59 10.
United Kingdom: MULLARD LTD., Mullard House, Torrington Place, LONDON WC1E 7HD, Tel. 01-580 6633.
United States: (Active devices & Materials) AMPEREX SALES CORP., Providence Pike, SLATERSVILLE, R.I. 02876, Tel. (401) 762-9000.
(Passive devices) MEPCO/ELECTRA INC., Columbia Rd., MORRISTOWN, N.J. 07960, Tel. (201) 539-2000.
(IC Products) SIGNETICS CORPORATION, 811 East Arques Avenue, SUNNYVALE, California 94086, Tel. (408) 739-7700.
Uruguay: LUZILECTRON S.A., Rondeau 1567, piso 5, MONTEVIDEO, Tel. 9 43 21.
Venezuela: IND. VENEZOLANAS PHILIPS S.A., Elcoma Dept., A. Ppal de los Ruices, Edif. Centro Colgate, CARACAS, Tel. 36 05 11.

Printed in The Netherlands Date of release: 27 June 1978 T.D. 9399 500 08901