PHILIPS

Electronic components and materials

Technical note 064

Digital cassette interface for a 2650 microprocessor system

The digital cassette recorder offers many attractive facilities to the microcomputer user: its large storage capacity (0,5 Mbyte) and high data I/O rate are far superior to either punched cards or paper tape. Moreover, its sturdy construction and small size make it an excellent medium for data storage and transport. In use, the digital cassette recorder is quiet, reliable and simple to operate.

This publication describes an interface between the Philips Digital Cassette Drive (DCR) and a 2650-based microcomputer system. Data transfer is performed under interrupt program control in order to prevent undue wastage of processing time. The data format on the cassettes complies with the ECMA* standards 34 and 41 for BASIC labelled cassettes.

Table 1 gives brief specifications of the DCR and Table 2 a summary of the functions provided by the interface.

TABLE 1 Brief specification of the DCR

head type	single track, read after write (one head with double gap).
recording technique	phase encoded, bit serial, character serial.
recording density	800 bpi.
tape speed	19 cm/s (7,5 in/s).
data rate	750 eight-bit characters/s.
start time	20 ms max.
stop time	20 ms max.
max. rewind time	45 s.
cassette type	twin hub coplanar.
tape size	86 m long, 3,81 mm wide, 19 μm thick.
data capacity	$1{,}9 \times 10^6$ bits.
number of tracks	one on each cassette side (A or B).
tape side indentifier	asymmetrically positioned cut out in the rear edge of cassette frame.
tape markers	two 0,6 mm holes centrally positioned in the tape at 450 ± 30 mm from the transparent leaders.
file protect	two replaceable write enable plugs in the rear edge of cassette frame.

TABLE 2 Summary of interface functions

block size	2 to 256 data bytes + 2 CRC bytes + 1 byte preamble and 1 byte postamble.
error detection	CRC using the polynomial $X^{16} + X^{15} + X^2 + 1$.

cassette functions	write tape mark(s);
	search tape mark(s);
	write one or more data blocks;
	read one or more data blocks.

* ECMA: European Computer Manufacturers Association.

signetics

Cassette data format

Data is recorded on the cassette in a single track, serial mode in blocks of 8-bit bytes. The least significant bit of a byte is written and read first. By turning the cassette onto its other side, a second track is available, recorded in the reverse direction. The DCR can distinguish between side A and side B by sensing an asymmetrical slot in the cassette.

For recognition and synchronization purposes, the data in a block is preceded by a preamble character and followed by two CRC* bytes and a postamble character. The preamble and postamble characters are each a single byte with the hexadecimal value 'AA' (binary pattern 10101010).

Tape marks are used to define the beginning-of-file, end-of-records and end-of-file. A tape mark consists of a preamble character, two bytes of all zeros and a postamble character.

Tape marks and data blocks are all separated by a gap corresponding to at least 100 ms at the nominal transport speed.

A data block may contain from 2 to 256 data bytes; this interface has been designed to write a variable length block in this range as specified by the user's write command. When requested to write more than 256 data bytes, the interface directs the DCR to write a variable length block followed by a number of blocks each of 256 data bytes. The interface allows from 1 to 256 data blocks to be read or written from one command.

Recording technique

The DCR uses the well-known phase-encoding technique in which a data bit is recorded as a change in the magnetic flux of the tape. A *zero* is recorded as a flux transition from a north pole to a south pole, and a *one* as a flux transition from a south pole to a north pole. This method of encoding allows a high bit-density on the tape.

When two or more consecutive ones or zeros are to be written, additional flux changes are required between the bit flux changes. These phase flux changes occur at the nominal mid-point between the bit flux changes. Figure 1 shows the relationship between the data to be recorded and the bit and phase flux changes.

Interface description

The interface is designed to perform data I/O under interrupt program control: the microcomputer requests action from the DCR and then continues with its main program until informed by an interrupt that the DCR is ready for data transfer. Upon receipt of the interrupt, the microcomputer stops execution of the current program, saves the status and program address and jumps to execute the DCR interrupt program. When the data transfer is complete, the microcomputer restores its previous status and program address and continues execution of the interrupted program. In this manner, the microcomputer does not have to wait while the DCR searches for information or rewinds the cassette etc.

From the foregoing description it will be realized that both interface hardware, between the DCR and the microcomputer data I/O system, and interface software, comprising the interrupt program and the connection routine in the program requiring the cassette action, are required. Figure 2 shows the interface hardware and DCR as part of the microcomputer system.

A program using this interface specifies the cassette action required by defining a control table for the connection routine. Using the information in this table, the interface can perform eight read/write functions on the tape:

write data;
write data from beginning of tape;
read data;
read data from beginning of tape;
write tape mark(s);
write tape mark(s) from beginning of tape;
search for tape mark(s);
search for tape mark(s) from beginning of tape.

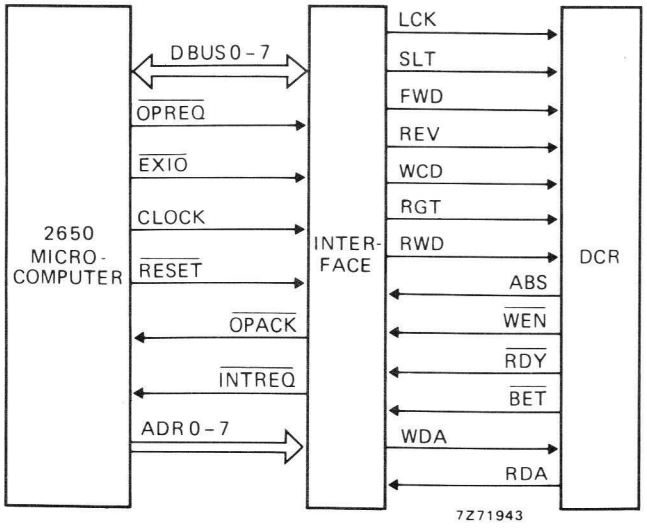


Fig. 2 Block diagram of the microcomputer system with cassette drive.

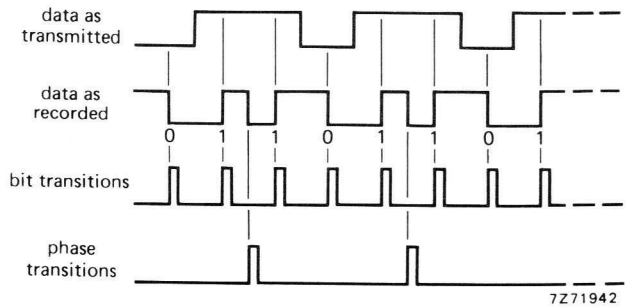


Fig. 1 Phase-encoded data showing the bit and phase flux transitions.

* CRC: Cyclic Redundancy Check for detection of bit errors in the recorded data.

Interface hardware

This is comprised of 27 TTL integrated circuits. Figure 3 shows the schematic diagram of the cassette interface hardware. Table 3 lists the signals to and from the microcomputer while Table 4 lists those to and from the DCR.

TABLE 3 Signals between microcomputer and interface

signal	function
$\overline{\text{OPREQ}}$	operation request from microcomputer, active LOW.
$\overline{\text{OPACK}}$	operation acknowledge from interface, active LOW.
$\overline{\text{INTREQ}}$	interrupt request from interface, active LOW.
$\overline{\text{EXIO}}$	extended I/O signal from the microcomputer instructing the interface to compare its address with the data on the address bus. It is an active LOW signal, formed by the combination: $\overline{\overline{M/IO} \cdot E/\overline{NE}}$
DBUS0-7	8-bit bidirectional data bus.
ADR0-7	8-bit command/peripheral number bus.
CLOCK	nominal 1 MHz clock from microcomputer.
$\overline{\text{RESET}}$	general system reset signal, active LOW.

TABLE 4 Signals between DCR and interface

signal	function
control lines	
LCK	locks the cassette retrieval bar in position and lights the LOCK lamp.
SLT	selects the DCR – enables cassette action.
FWD	forward, causes the tape to move forwards at capstan speed.
REV	reverse, causes tape to move in the reverse direction at capstan speed.
WCD	write command, enables the write logic.
RGT	read gate, enables the read logic.
RWD	rewind the tape at hub speed.
status lines	
ABS	A or B side of cassette, LOW when A side in use (on top).
$\overline{\text{WEN}}$	write enabled, active LOW, indicates that the write enable plug is present for the side in use.
$\overline{\text{RDY}}$	ready, active LOW, indicates that the tape has been positioned and the LCK and SLT lines are active.
$\overline{\text{BET}}$	Beginning-or-End-of-Tape, active LOW, indicates that the BOT or EOT marker (hole in tape) or transparent leader has been detected by the photocell.
data lines	
WDA	write data, serial, phase-encoded data to be recorded.
RDA	read data, serial, phase-encoded data read from the tape.

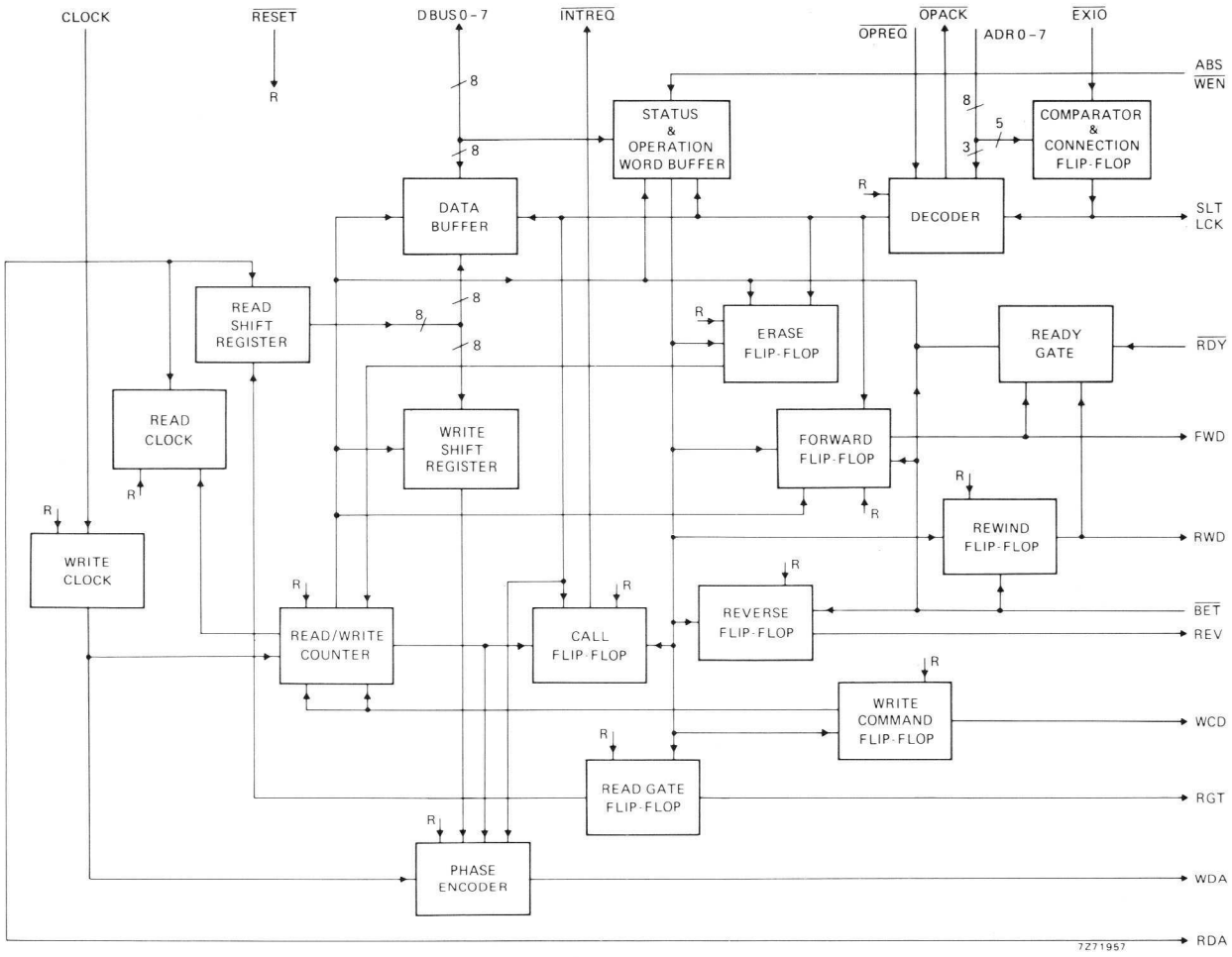


Fig. 3 Block diagram of the interface hardware.

Address bus

The least significant eight bits of the microcomputer address bus are used for peripheral selection when the signal M/$\overline{IO}$ indicates I/O action. ADR0 to ADR4 are used as a peripheral address, the peripheral responding when the data on these lines coincides with its own hard-wired address. ADR5 to ADR7 are used to command the interface, the commands are listed in Table 5. Figure 4 shows the peripheral address and command decoding logic.

TABLE 5 Commands to the interface

mnemonic	command	ADR7	ADR6	ADR5
STAT	status request	1	0	1
CX	connection	0	1	0
OEC	output exchange	1	0	0
IEC	input exchange	0	0	1
DX	disconnection	1	1	1

Data bus

Data from the microcomputer, on the data bus, is clocked into the peripheral register during the Output Exchange Command (OEC) and transferred to the write circuit. During an Input Exchange Command (IEC), the data that has been read from the tape is transferred to the data bus.

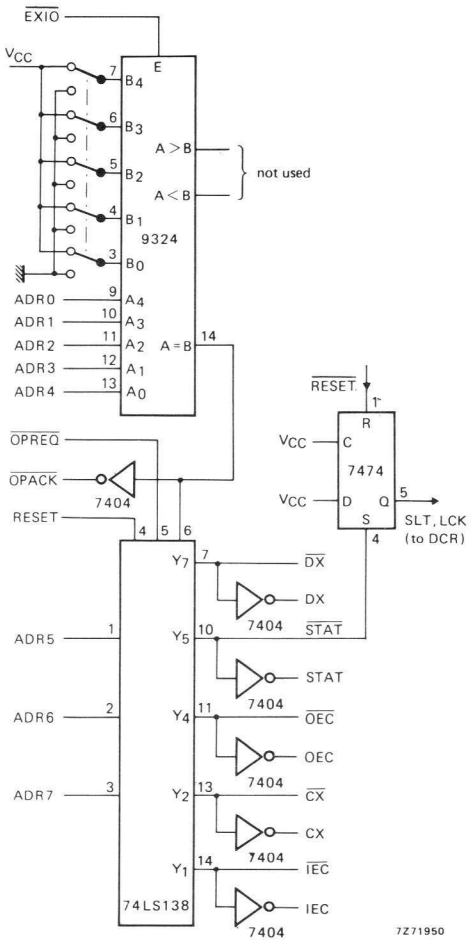


Fig. 4 Address and command decoding logic.

Status and control logic

Before commencing data transfer, the microcomputer must establish that the DCR is ready for such an action. The status request command to the peripheral asks for the status word, the value of which describes the condition of the peripheral. The information conveyed by the status word is given in Table 6.

TABLE 6 Status words

bit	function
0	ABS signal: 0, side A; 1, side B.
1	$\overline{WEN}$, write enable signal, 0: write enabled.
2	not used.
3	not used.
4	RDY, ready signal.
5	not used.
6	GAPS signal, indicates no data on tape, gap between tape marks and/or data blocks.
7	$\overline{BET}$ signal, 0: marker indicating Beginning-or-End-of-Tape sensed.

Once the microcomputer has ascertained the status of the DCR, it issues a Connection Command (CX) and outputs an operation word to the data bus. When the interface recognizes the connection command, the operation word on the data bus is clocked into the operation word buffer. Each bit of the operation word, apart from bit 3 which is not used, corresponds to a particular signal within the interface:

bit 0 A signal, which sets the Read Gate (RGT) flip-flop to enable reading after the start-reading delay (20 ms) is finished. The RGT flip-flop is reset either by the reset signal or by a disconnection command.

bit 1 B signal, which sets the CALL flip-flop. The not-true output of the CALL flip-flop is the signal $\overline{INTREQ}$. The flip-flop is reset either by the reset signal or by a disconnection or exchange command.

bit 2 C signal, which sets the Write Command (WCD) flip-flop to enable writing after the start-writing delay (60 ms) is finished. The flip-flop is reset either by the reset signal or by a disconnection command.

bit 3 Not used.

bit 4 E signal, which sets the ERASE and WCD flip-flops. The not-true output of the ERASE flip-flop is used to set the WDA flip-flop so that the data line does not convey data during the erase function.

bit 5 F signal, which sets the Forward (FWD) flip-flop. It is reset by the reset signal or a disconnection command.

bit 6 G signal, which sets the Reverse (REV) flip-flop. It is reset by the reset signal or a disconnection command.

bit 7 H signal sets the Rewind (RWD) flip-flop. The rewind flip-flop is reset either by the reset signal or by the signal $\overline{BET}$ (not Beginning-or-End-of-Tape).

The bidirectional operation/status word register and the associated control flip-flops are shown in Fig. 5.

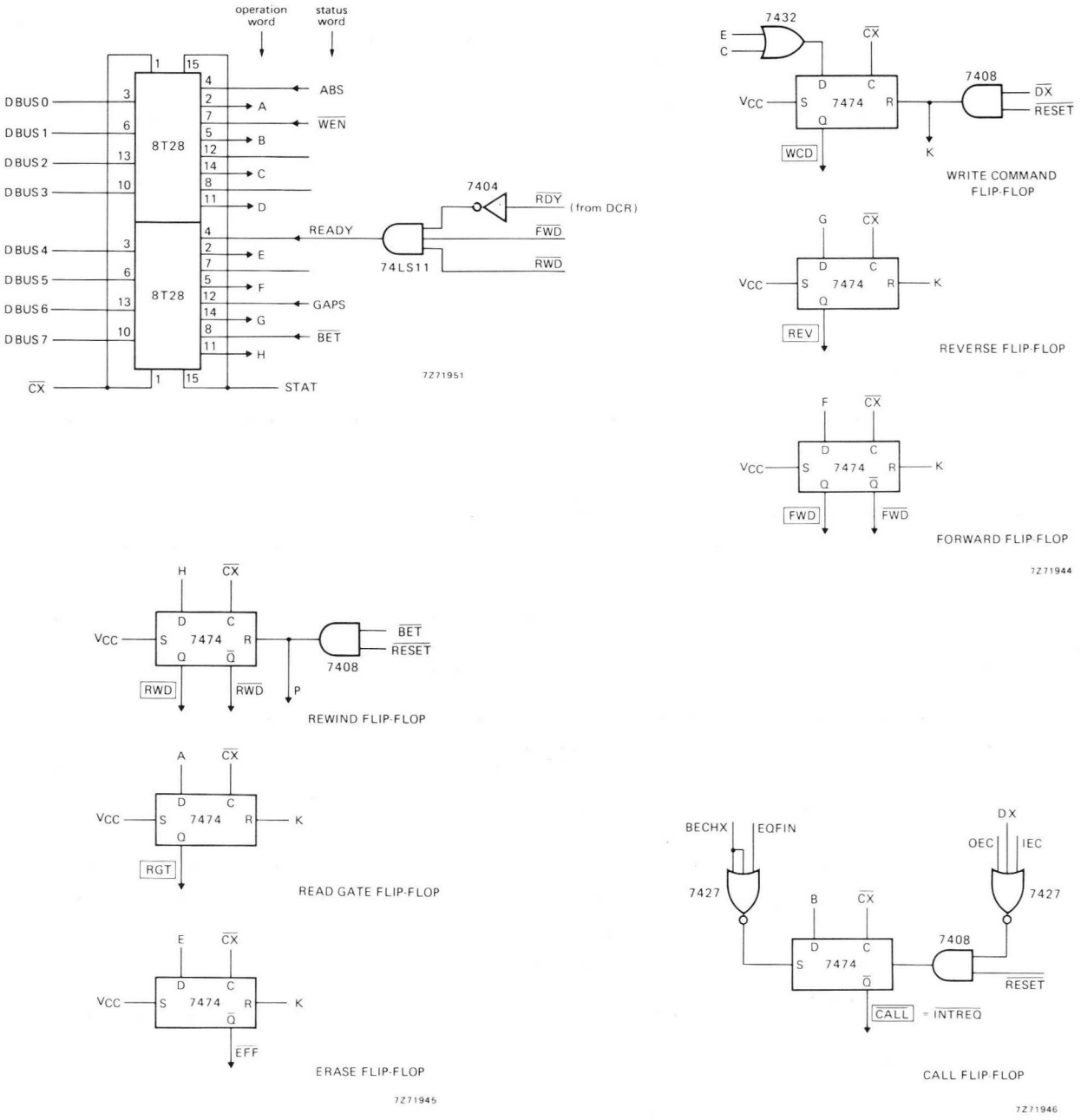


Fig. 5 Operation/status word register and logic.

Write logic

The write logic consists of the write clock, write/read counter, write shift register, phase encoder and WCD flip-flop. The write logic is activated when the microcomputer has issued the appropriate connection command followed by an output exchange command with the data to be written. During the output exchange command, the data is transferred via the data buffer to the write shift register. The data path can be seen in Fig. 3 and the timing sequence is shown in Fig. 6.

The circuit of the write clock is shown in Fig. 7; it comprises two decade counters to derive the 12 kHz square-wave from the 1 MHz clock from the microcomputer.

The write/read counter, Fig. 8, is used to count cycles of the write clock waveform corresponding to the eight bits of a character. The output QA is used to shift each of the data bits serially out of the shift register, the BECHX signal being generated by the carry output at the end of a character.

The write data circuit is shown in Fig. 9. The data is converted from parallel to serial format by the register and then phase encoded by the exclusive-OR gate. The QA signal to the exclusive-OR gate provides the phase flux transitions when these are required. The output of the phase encoder is stored in the WDA (write data) flip-flop for a period determined by the WCP (write clock pulse) signal. The $\overline{EFF}$ signal from the ERASE flip-flop is connected to the set input of the WDA flip-flop so that the data line remains HIGH during erasing.

At the end of each character, the BECHX signal allows the write register to be loaded with the next character from the data buffer. The BECHX signal also causes the CALL flip-flop to be set, in order to receive the next character (command) from the microcomputer.

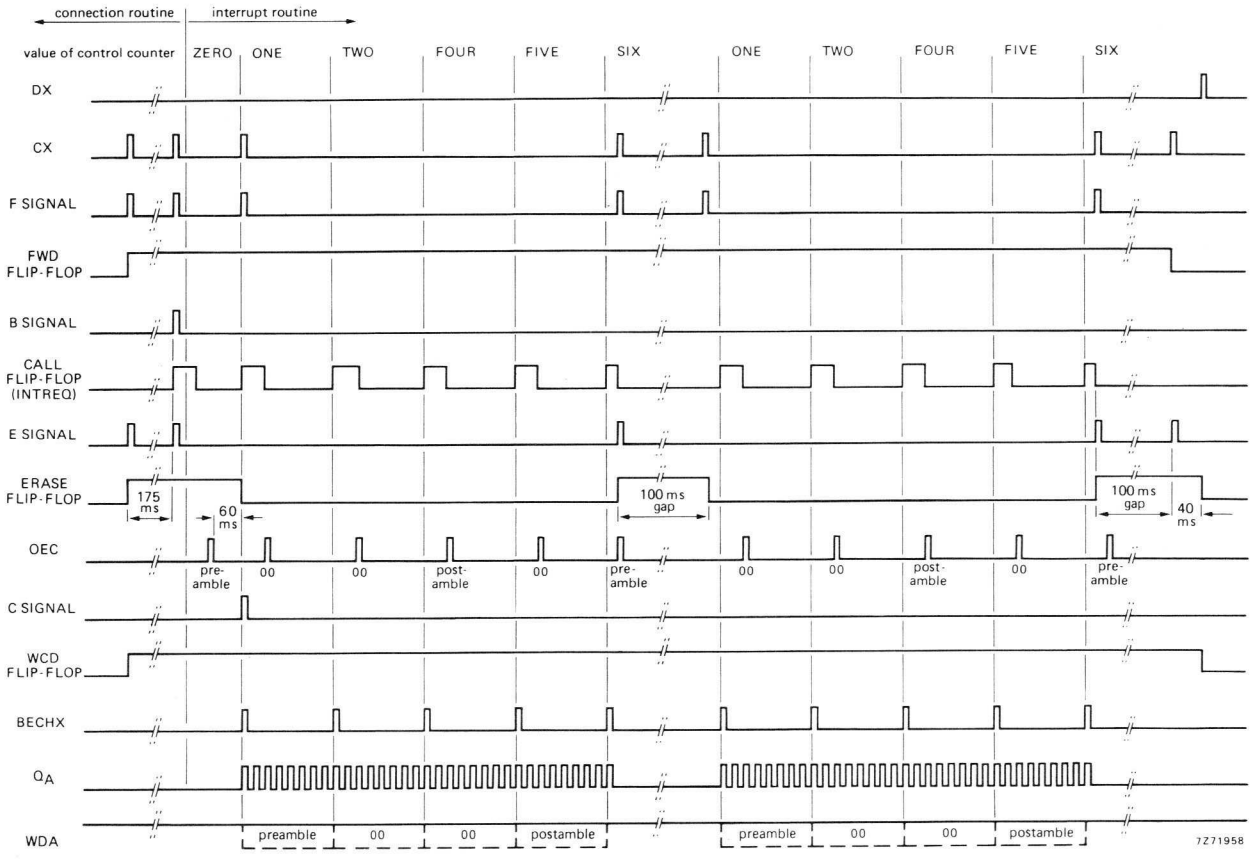


Fig. 6 Timing sequence for writing.

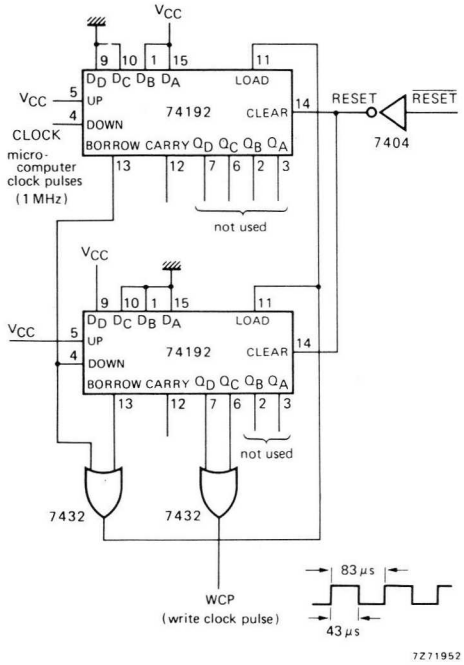


Fig. 7 Write clock circuit.

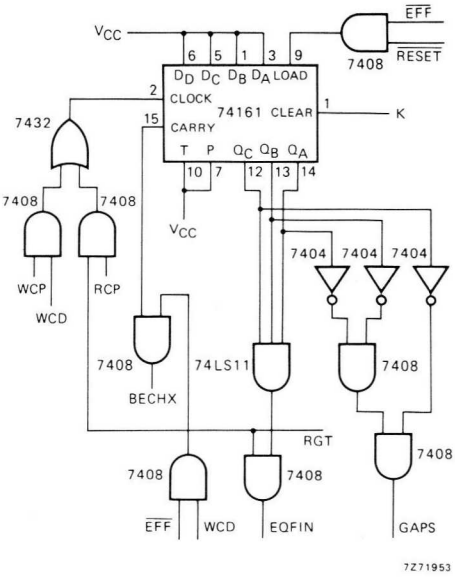


Fig. 8 Write/read counter.

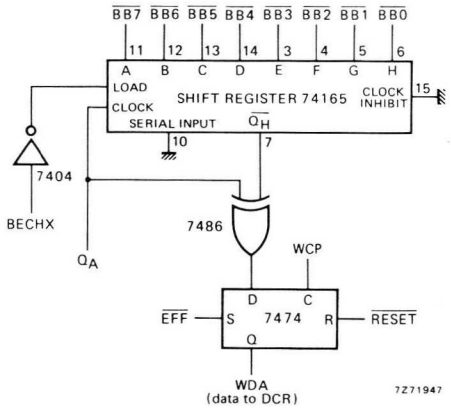


Fig. 9 Write data circuit.

Read logic

The read logic comprises the read clock, read shift register, write/read counter and read gate flip-flop.

The read clock acts as a discriminator between the bit flux transitions and the phase flux transitions in the data signal from the DCR. Both preamble and postamble characters possess only bit flux transitions, so that the read clock, which is a monostable, can be synchronized to these and then mask out the phase flux transitions in the data signal. The phase flux transitions occur midway between bit flux transitions and the monostable remains set for 130 μs in a period of 166 μs. The circuit of the read clock is shown in Fig. 10.

In the read mode, the write/read counter counts the bit flux transition pulses and sets the EQFIN signal at the end of a byte (eight pulses). The EQFIN signal enables the three-state outputs of the read register and sets the CALL flip-flop to inform the microcomputer that the data is ready for transfer.

The read data circuit is shown in Fig. 11. The phase-encoded output from the DCR is shifted through the register, clocked by the read clock to ensure that only data is accepted. When the byte is complete, the EQFIN signal enables the three-state data buffer to accept the data from the parallel outputs of the shift register. The timing sequence for reading is shown in Fig. 12.

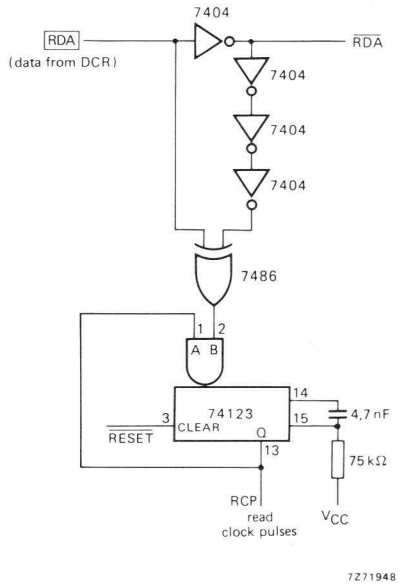


Fig. 10 Read clock circuit.

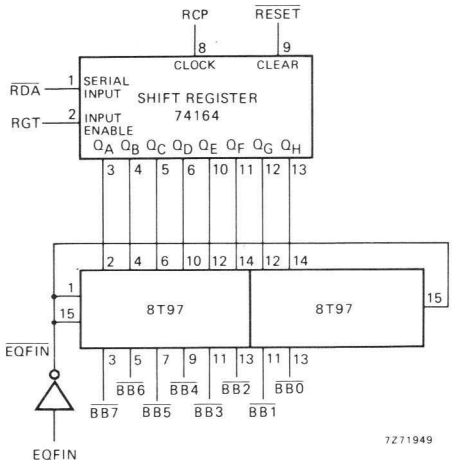


Fig. 11 Read data circuit.

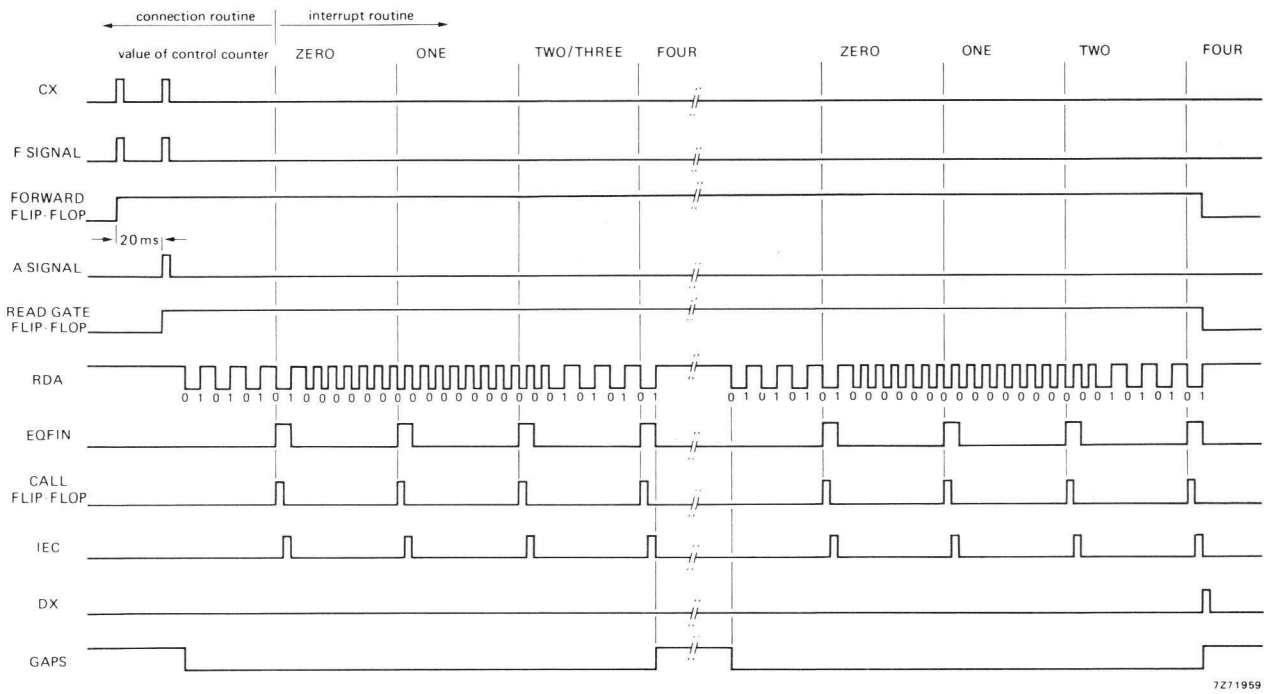


Fig. 12 Timing sequence for reading.

Data buffer

The data buffer is a bidirectional three-state buffer: there are two buses which operate independently, one in each direction. In Fig. 13 bus A is the microcomputer data bus and bus B is the internal bus to the read and write registers. In the writing mode, the buffer accepts data from the microcomputer data bus when the output exchange command is active. The data is then held until the BECHX signal allows transfer to the write shift register.

In the reading mode, the buffer accepts data from the read register when the EQFIN signal is active. The data is held until an input exchange command is received, when it is transferred to the microcomputer data bus.

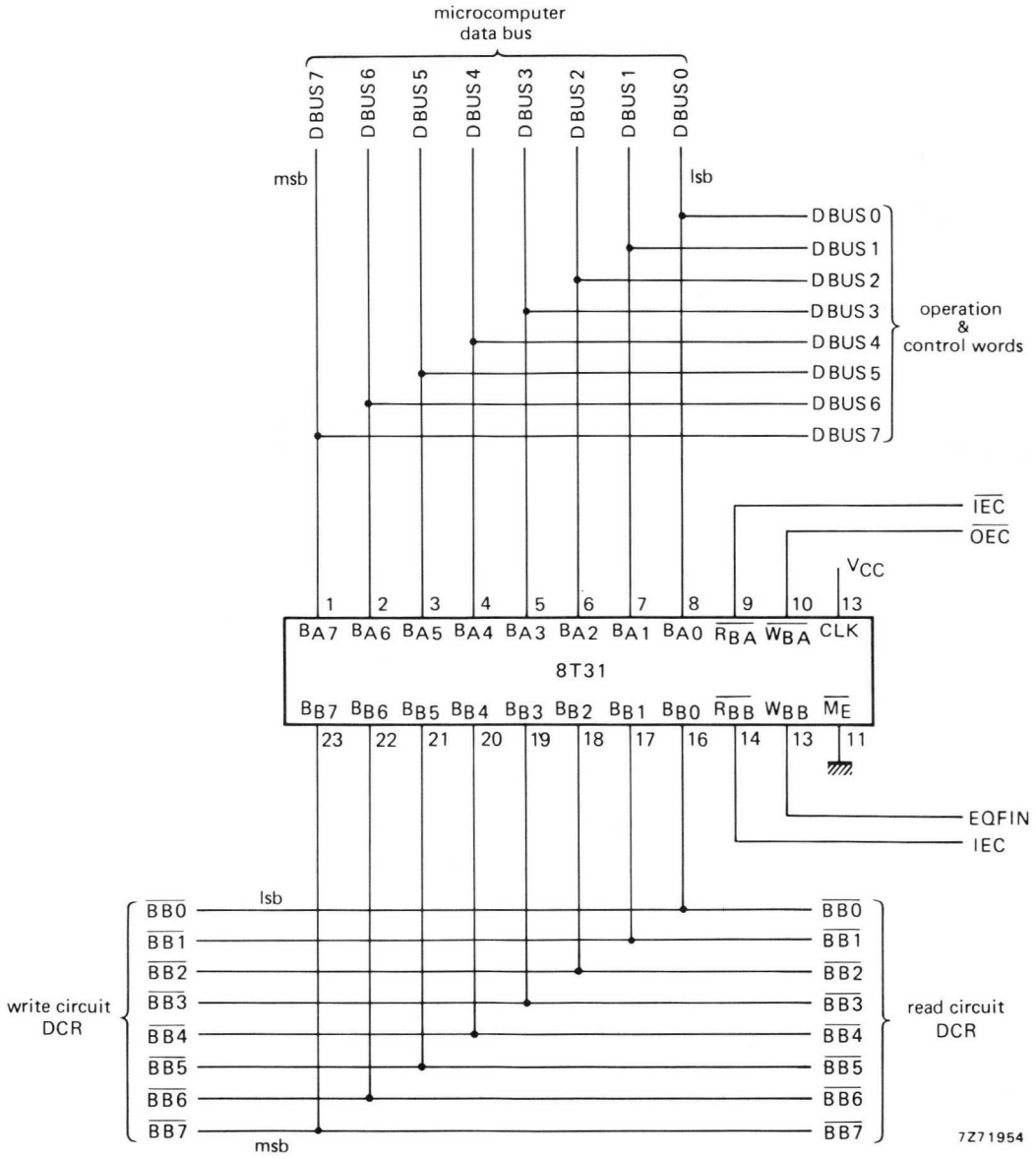


Fig. 13 Bidirectional data buffer.

Interface software

The software can communicate with the peripheral using the two 2-byte I/O instructions READ EXTENDED (REDE) and WRITE EXTENDED (WRTE). The second byte of these instructions is presented on the address bus as the peripheral address and the command. Table 5 shows the five peripheral commands available. The software is comprised of two parts: the connection routine and the interrupt program.

The connection routine is part of the main program in order to conserve space in the return address stack. Its function is to transfer the control table, specifying the cassette action that is required, to the cassette memory locations CML to CML+5, check the DCR status and issue a connection command if the status is correct; Fig. 14.

The interrupt program is entered each time the microcomputer receives an interrupt request from the CALL flip-flop. This program uses the control table to define its action, in conjunction with the control counter, which is incremented after each phase of the required operation. It is this program that controls the input/output of data, as specified by the control block. Each entry into the interrupt program will result in a transfer of one byte to or from the microcomputer.

The flow chart of the interrupt program is shown in Fig. 15.

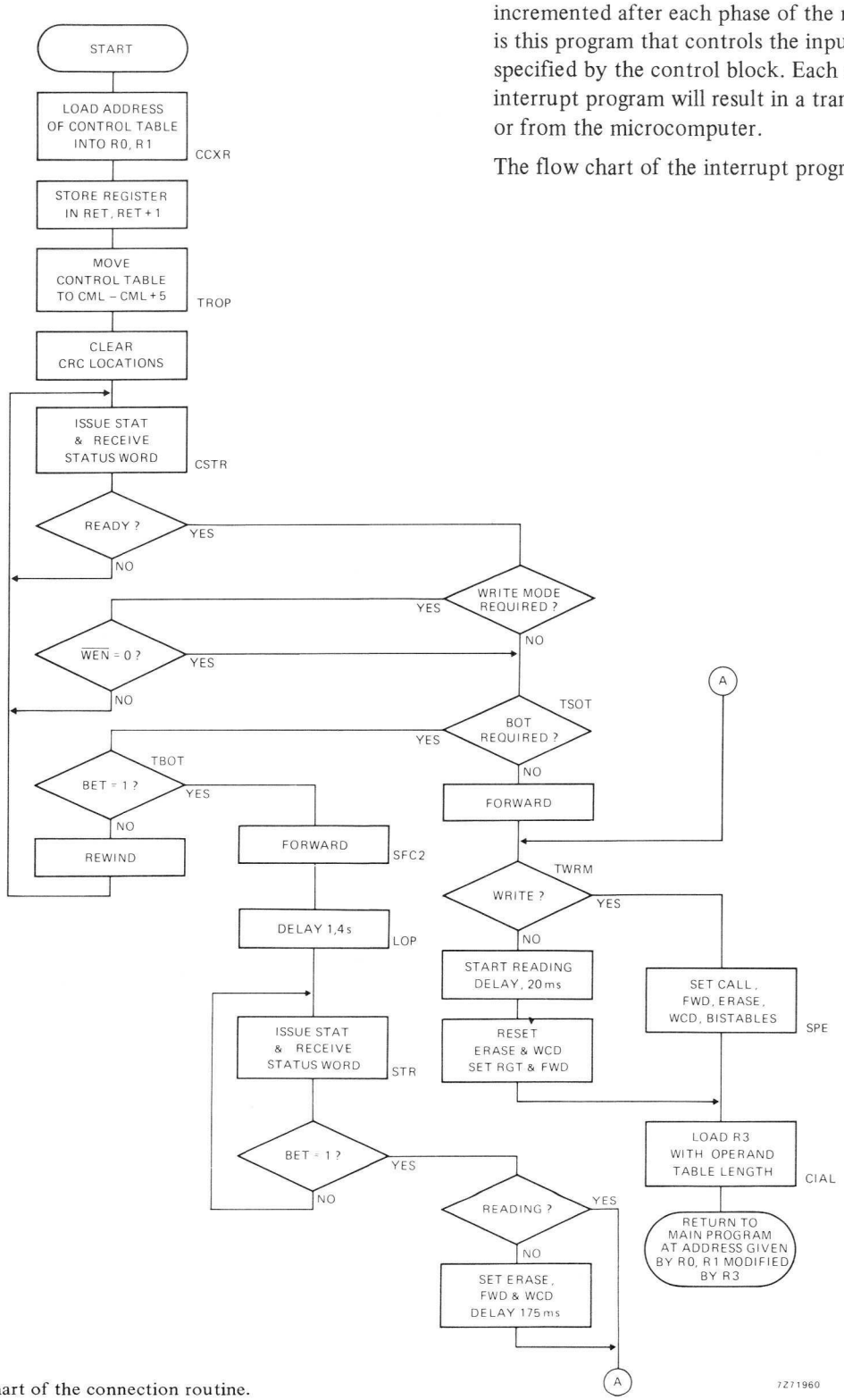


Fig. 14 Flow chart of the connection routine.

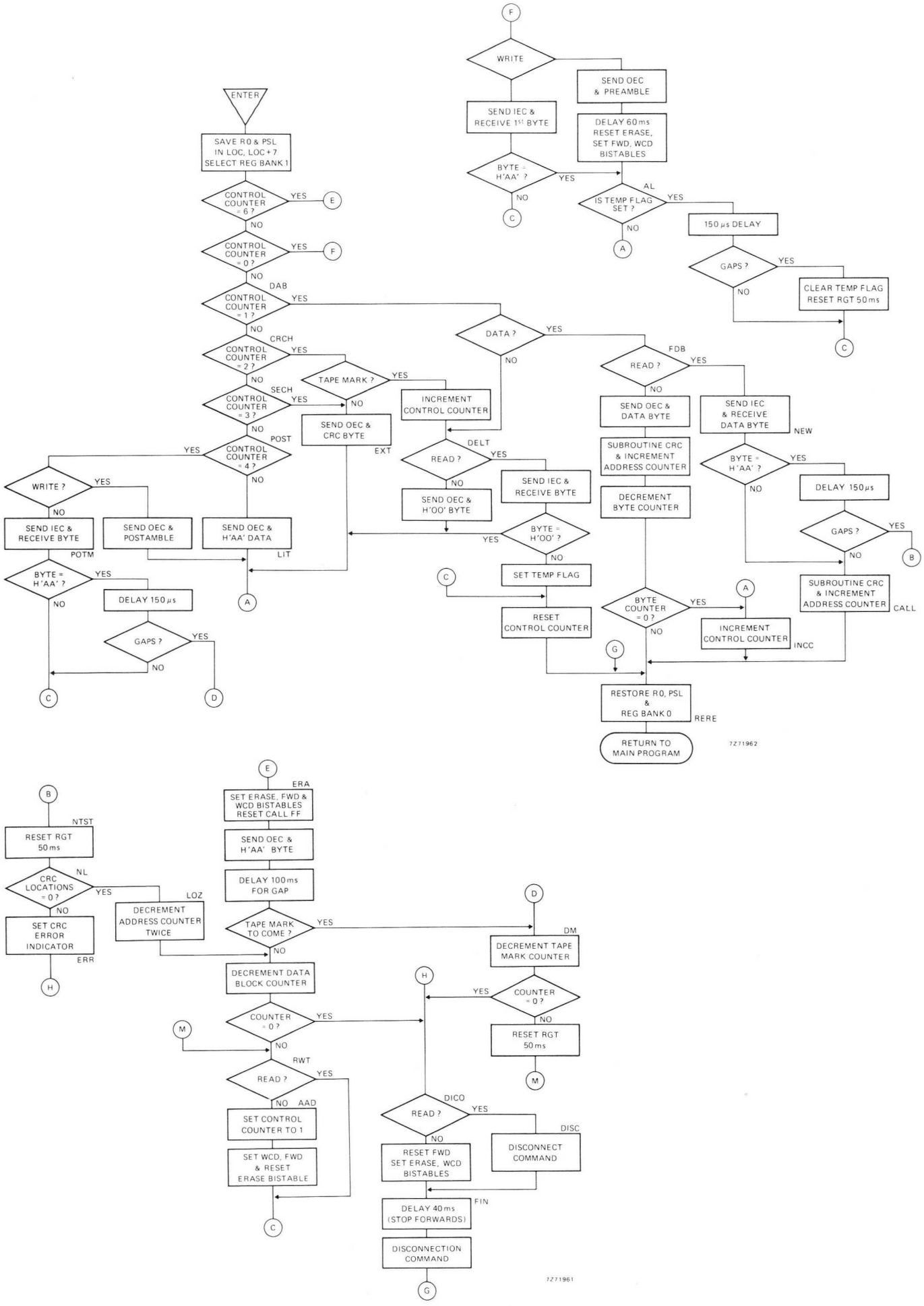


Fig. 15 Flow chart of the interrupt program.

Cassette operation sequence

Data exchange with the DCR is initiated by entry into the connection routine: this stores the continuation address of the main program and transfers the control table into the locations CML to CML+5 for use by the interrupt routine. Figure 14 shows the flow chart of the connection routine.

The information in the control table is shown in Table 7, with the operands that must be defined for a particular function in Table 8.

TABLE 7 Contents of the control table

operand	byte	function
1	1	control word: bit 0, bit 1, bit 2 } three-bit control counter used by interrupt routine. Must initially be zeros.
		bit 3 not used.
		bit 4 1 indicates read/write from BOT.
		bit 5 CRC error flag. 1 if error detected.
		bit 6 data/tape mark indicator, 1 indicates tape mark.
		bit 7 read/write select, 1 indicates write.
2	2	number of bytes in the first block to be written.
3	3	number of tape marks to be read/written.
4	4	number of data blocks to be read/written.
5	5, 6	address of the first byte to be read/written.

TABLE 8 Specification of control word and operands

control word	function	operands to be specified
H'80'	write data	2,4,5
H'90'	write data from beginning of tape	2,4,5
H'00'	read data	4,5
H'10'	read data from beginning of tape	4,5
H'CO'	write tape marks	3
H'DO'	write tape marks from beginning of tape	3
H'40'	read tape marks	3
H'50'	read tape marks from beginning of tape	3

After requesting the DCR status and, if necessary, waiting until this is correct, the connection routine issues a connection command accompanied by an operation word to start the required cassette action, e.g. rewind to beginning of tape. When the tape has been positioned as required, the connection routine takes into account any delays, e.g. start-reading delay, and issues another connection command. The operation word will be set to enable reading or writing and in the case of writing, the CALL flip-flop will be set, enabling the interface to ask for the first byte of write data. In the reading mode, the CALL flip-flop will be set by the end-of-character signal EQFIN, informing the microcomputer that the requested data is available. Setting the CALL flip-flop results in the signal $\overline{\text{INTREQ}}$, causing entry to the interrupt program to perform the data transfer.

After performing the above actions, the connection routine forms a branch indexed absolute instruction to the address following the control table, to continue execution of the main program.

The microcomputer will enter the interrupt program after receiving an interrupt request from the interface hardware. In this, the first action is to save the contents of R0 and the PSL in locations LOC and LOC+1.

Further actions of the interrupt program depend on the value of the control counter, held in the least significant three bits of the control word. The control counter is initially zero.

Counter = 0

Writing mode: the microcomputer sends an OEC and a preamble character. It then goes to the start-writing delay subroutine (60 ms) before resetting the ERASE flip-flop and setting the WCD and FWD flip-flops. The control counter is then incremented and the saved registers restored before control reverts to the main program.

Reading mode: the microcomputer sends an IEC to receive a byte. If this is a preamble, the control counter is incremented, otherwise it remains unchanged.

Counter = 1

Writing tape marks: the microcomputer sends an OEC accompanied by an '00' byte and increments the control counter.

Reading tape marks: an IEC is sent to receive a byte. If the byte is '00', the control counter is incremented, otherwise the initial conditions are restored and the interface again looks for a tape mark.

Writing data: an OEC and data byte are sent. The address counter is incremented and the byte counter decremented. If the byte counter is zero, the control counter is incremented.

Reading data: an IEC is sent and a data byte received. If the byte is 'AA' followed by the GAPS signal, the end of block is detected. Otherwise the address counter is incremented and the CRC locations updated.

Counter = 2

Writing data: an OEC and the first CRC character are sent and the control counter is incremented.

Writing tape mark: an OEC and an '00' byte are sent and the control counter is incremented.

Reading tape mark: an IEC is sent and a byte received. If the byte was '00', the control counter is incremented, otherwise the initial conditions are restored and the GAPS signal is awaited.

Counter = 3

Writing tape mark: an OEC and the second CRC character are sent and the control counter is incremented.

Counter = 4

Writing data or tape mark: an OEC and an 'AA' byte are sent and the control counter is incremented.

Reading data or tape mark: an IEC is sent and a byte received. If this is 'AA', followed by the GAPS signal, the tape mark counter is decremented. If the tape mark counter is zero, a disconnection command is given, otherwise the initial conditions are reset to search for another tape mark.

Counter = 5

An OEC and an '00' byte are sent to clear the write shift register and reset the CALL flip-flop. The control counter is incremented.

Counter = 6

The ERASE, FWD and WCD flip-flops are set. An OEC and preamble ('AA') are sent. After an inter-block gap is generated, the data block counter or the tape mark counter is decremented. When the relevant counter is zero, the DCR is disconnected, otherwise the initial conditions are restored to write the next tape mark or data block.

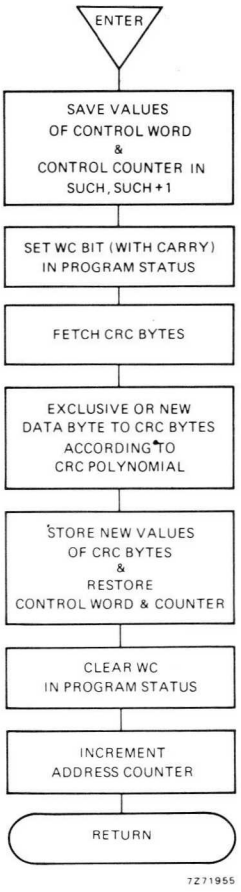
Cyclic redundancy check subroutine

The cyclic redundancy check is performed by software using the polynominal:

$$X^{16} + X^{15} + X^2 + 1.$$

A 16-bit remainder is generated during writing, and written at the end of each data block. The same CRC process during reading the data will produce the same 16-bit remainder, which then becomes zero when the CRC bytes are processed. If an error occurs during reading or writing, the final remainder is not zero and an error flag is set. The flow chart and simulated hardware are shown in Fig. 16.

(a)



(b)

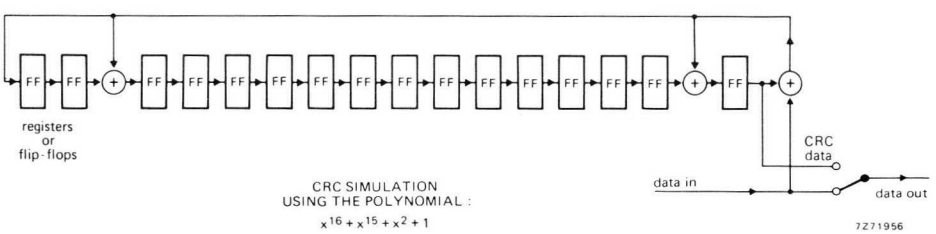


Fig. 16 Flow chart of the CRC subroutine (a) and simulated hardware (b).

```
TWIN ASSEMBLER VER 2.0                                    PAGE 0001

LINE ADDR  OBJECT  E SOURCE

0001               *JOSE LUIS CORTINAS 770202-9.30
0002               *
0003               *CONNECTION ROUTINE FOR DIGITAL CASSETTE RECORDER
0004               *
0005               *IT IS A PART OF THE MAIN PROGRAM
0006               *FIVE OPERANDS ARE TRANSFERRED FROM A TABLE TO THE
0007               *MEMORY LOCATIONS (CML---CML+5), THE STATUS OF THE
0008               *DIGITAL CASSETTE RECORDER IS TESTED AND
0009               *THE RIGHT SIGNALS ARE GENERATED TO THE DCR TO
0010               *PERFORM THE REQUIRED FUNCTION
0011               *
0012               *THE FIVE OPERANDS ARE:
0013               *  OPER1: 1 BYTE: CONTROLWORD
0014               *  OPER2: 1 BYTE: NUMBER OF BYTES OF FIRST BLOCK TO
0015               *                 BE WRITTEN ON TAPE
0016               *  OPER3: 1 BYTE: NUMBER OF TAPEMARKS TO BE WRITTEN
0017               *                 ON TAPE OR SEARCHED FOR
0018               *  OPER4: 1 BYTE: NUMBER OF DATABLOCKS TO BE WRITTEN
0019               *                 ON TAPE OR READ FROM TAPE
0020               *  OPER5: 2 BYTES:ADDRESS OF FIRST BYTE IN MEMORY TO BE WRITTEN
0021               *                 ON TAPE OR READ FROM TAPE
0022               *
0023               * CONTROLWORD        FUNCTION                TO BE DEFINED OPERANDS
0024               * ===========        ========                ======================
0025               *
0026               *    H'90'    WRITE DATA FROM BEGIN OF TAPE       OPER2,4,5
0027               *    H'80'    WRITE DATA SOMEWHERE ON TAPE        OPER2,4,5
0028               *    H'10'    READ DATA FROM BEGIN OF TAPE        OPER4,5
0029               *    H'00'    READ DATA SOMEWHERE ON TAPE         OPER4,5
0030               *    H'D0'    WRITE TAPEMARK(S) FROM BEGIN OF TAPE OPER3
0031               *    H'C0'    WRITE TAPEMARK(S) SOMEWHERE ON TAPE  OPER3
0032               *    H'50'    SEARCH TAPEMARK(S) FROM BEG. OF TAPE OPER3
0033               *    H'40'    SEARCH TAPEMARK(S) SOMEWH. ON TAPE   OPER3
0034               *
0035               *
0036               ************************************************
```

```
LINE ADDR  OBJECT  E SOURCE

0038               * DEFINITIONS OF SYMBOLS:
0039               *
0040 0000          R0   EQU   0          PROCESSOR REGISTERS
0041 0001          R1   EQU   1
0042 0002          R2   EQU   2
0043 0003          R3   EQU   3
0044 0080          S    EQU   H'80'      PSU: SENSE
0045 0040          F    EQU   H'40'           FLAG
0046 0020          II   EQU   H'20'           INTERRUPT INHIBIT
0047 0007          SP   EQU   H'07'           STACKPOINTER
0048 00C0          CC   EQU   H'C0'      PSL: CONDITION CODE
0049 0020          IDC  EQU   H'20'           INTER DIGIT CARRY
0050 0010          RS   EQU   H'10'           REGISTER BANK SELECT
0051 0008          WC   EQU   H'08'           1=WITH, 0=NO CARRY
0052 0004          OVF  EQU   H'04'           OVERFLOW
0053 0002          COM  EQU   H'02'           1=LOG., 0=ARITH. COMP.
0054 0001          C    EQU   H'01'           CARRY/NO BORROW
0055 0000          Z    EQU   0          BRANCH COND: ZERO
0056 0001          P    EQU   1                       POSITIVE
0057 0002          N    EQU   2                       NEGATIVE
0058 0000          EQ   EQU   0                       EQUAL
0059 0001          GT   EQU   1                       GREATER THAN
0060 0002          LT   EQU   2                       LESS THAN
0061 0003          UN   EQU   3                       UNCONDITIONAL
0062 0000          A1   EQU   0                       ALL BITS ARE 1
0063 0002          N1   EQU   2                   NOT ALL BITS ARE 1
0064               *
0065 0000          DCR  EQU   0          DIGITAL CASSETTE RECORDER
0066 00A0          STAT EQU   H'A0'      STATUS COMMAND
0067 0040          CX   EQU   H'40'      CONNECTION COMMAND
0068 0080          OEC  EQU   H'80'      OUTPUT EXCHANGE COMMAND
0069 0020          IEC  EQU   H'20'      INPUT EXCHANGE COMMAND
0070 00E0          DX   EQU   H'E0'      DISCONNECTION COMMAND
0071               *
0072               *
0073 0000               ORG   H'7F0'
0074 07F0          CML  RES   6          CASSETTE MEMORY LOCATIONS
0075 07F6          RET  RES   2          KEEPS ADDRESS FIRST OPERAND
0076 07F8          LOC  RES   2          SAVE LOCATIONS OF R0 AND PSL
0077 07FA          CRC  RES   2          ADDRESSES OF CRC CHARACTERS
0078 07FC          SUCH RES   2          SAVES R2,R1 DURING CRC SUBROUTINE
0079               ************************************************
```

```
TWIN ASSEMBLER VER 2.0                                    PAGE 0003

LINE ADDR  OBJECT  E SOURCE

0081 07FE                ORG     H'600'
0082 0600 7510           CPSL    R5          SELECT REGISTERBANK #0
0083 0602 7702           PPSL    COM         LOGICAL COMPARE
0084 0604 0406           LODI,R0 <$+7        FETCH ADDRESS HIGH FIRST OPERAND
0085 0606 0508           LODI,R1 >$+5        FETCH ADDRESS LOW FIRST OPERAND
0086 0608 1F0620         BCTA,UN CCXR        CALL DCR CONNECTION ROUTINE
0087                  *THE FOLLOWING TABLE IS AN EXAMPLE FOR WRITING A BLOCK
0088                  *OF 32 BYTES FOLLOWED BY 5 BLOCKS OF 256 BYTES.
0089                  *THE FIRST BYTE IS FETCHED FROM MEMORY ADDRESS H'1000'.
0090 060B 80             DATA    H'80'       CONTROLWORD
0091 060C 20             DATA    H'20'       BYTECTR (32)
0092 060D 04             DATA    H'04'       NUMBER OF TAPEMARKS
0093 060E 06             DATA    H'06'       NUMBER OF DATABLOCKS
0094 060F 1000           DATA    H'10,00'    ADDRESS OF FIRST BYTE
0095                  *
0096 0611                ORG     H'620'
0097                  *
0098 0620 CC07F6      CCXR STRA,R0 RET         SAVE ADDRESS FIRST OPERAND
0099 0623 CD07F7           STRA,R1 RET+1
0100                  *
0101 0626 0706             LODI,R3 6           LOAD R3 WITH # OF OPERANDS
0102 0628 0FC7F6      TROP LODA,R0 *RET,R3,-   TRANSFER OF OPERANDS INTO
0103 062B CF67F0           STRA,R0 CML,R3      CASSETTE MEMORY LOCATIONS CML....CML+5
0104 062E 5B78             BRNR,R3 TROP
0105                  *
0106 0630 20               EORZ    R0
0107 0631 CC07FA           STRA,R0 CRC         CLEAR CRC LOCATIONS
0108 0634 CC07FB           STRA,R0 CRC+1
0109 0637 57A0        CSTR REDE,R3 STAT+DCR    READ DCR STATUS
0110 0639 F710             TMI,R3  H'10'
0111 063B 987A             BCFR,A1 CSTR        BRANCH IF NOT READY
0112 063D 0E07F0           LODA,R2 CML         FETCH CONTROLWORD
0113 0640 9A04             BCFR,N  TSOT        BRANCH IF READING MODE
0114 0642 F702             TMI,R3  H'02'
0115 0644 1871             BCTR,A1 CSTR        BRANCH IF WEN SIGNAL NOT ACTIVE
0116 0646 F610        TSOT TMI,R2  H'10'
0117 0648 1806             BCTR,A1 TBOT        BRANCH IF STARTING THE TAPE
0118 064A 0420             LODI,R0 H'20'
0119 064C D440             WRTE,R0 CX+DCR      SEND FORWARD COMMAND
0120 064E 1B21             BCTR,UN TWRM
0121                  *
0122 0650 F780        TBOT TMI,R3  H'80'
0123 0652 9806             BCFR,A1 SFC2        BRANCH IF LEADER
0124 0654 0480             LODI,R0 H'80'
0125 0656 D440             WRTE,R0 CX+DCR      SEND REWIND COMMAND
0126 0658 1B5D             BCTR,UN CSTR
0127 065A 0420        SFC2 LODI,R0 H'20'
0128 065C D440             WRTE,R0 CX+DCR      SEND FORWARD COMMAND
```

```
LINE ADDR  OBJECT  E SOURCE

0130 065E 0508          LODI,R1 8         DELAY FOR PASSING LEADER
0131 0660 3B25     LOP  BSTR,UN CBTR          (1,4 SEC)
0132 0662 F97C          BDRR,R1 LOP
0133               *
0134 0664 57A0     STR  REDE,R3 STAT+DCR  SEND STATUS REQUEST
0135 0666 1A7C          BCTR,N  STR       BRANCH IF NO TAPE HOLE
0136 0668 02            LODZ    R2
0137 0669 9A06          BCFR,N  TWRM      BRANCH IF READING
0138 066B 0430          LODI,R0 H'30'
0139 066D D440          WRTE,R0 CX+DCR    SET ERASE, FORW AND WCD FF
0140 066F 3B16          BSTR,UN CBTR      CALL BEGIN OF TAPE SUBROUTINE 175 MS
0141               *
0142 0671 02       TWRM LODZ    R2
0143 0672 1A0A          BCTR,N  SPE       BRANCH IF WRITING
0144 0674 070B          LODI,R3 11
0145 0676 3B11          BSTR,UN LOP8      CALL START READING DELAY SUBROUTINE 20 MS
0146 0678 0421          LODI,R0 H'21'
0147 067A D440          WRTE,R0 CX+DCR    SET RGT FF AND FORWARD
0148 067C 1B04          BCTR,UN CIAL
0149 067E 0432     SPE  LODI,R0 H'32'
0150 0680 D440          WRTE,R0 CX+DCR    SET ERASE, FORW, WCD AND CALL FF
0151               *
0152 0682 0706     CIAL LODI,R3 6         LOAD R3 WITH NUMBER OF OPERANDS
0153 0684 9F87F6        BXA     *RET      RETURN TO ADDRESS (RET) +(R3)
0154               *
0155               *
0156               *
0157               *    DELAY SUBROUTINES
0158               *
0159 0687 074D     CBTR LODI,R3 77        DELAY TIME 175 MS 1 MHZ CLOCK
0160 0689 20       LOP8 EORZ    R0
0161 068A F87E     LOP7 BDRR,R0 LOP7
0162 068C FB7C          BDRR,R3 LOP7
0163 068E 17            RETC,UN
0164               *
0165               *
0166               * SAVE AND RESTORE SUBROUTINES
0167               *
0168 068F CC07F8   SAVE STRA,R0 LOC       SAVE (R0) IN LOC
0169 0692 7710          PPSL    RS        SELECT REG. BANK #1
0170 0694 13            SPSL              (PSL) INTO R0
0171 0695 CC07F9        STRA,R0 LOC+1     SAVE (PSL) IN LOC+1
0172 0698 17            RETC,UN           RETURN
0173               *
0174 0699 0C07F9   REST LODA,R0 LOC+1     FETCH (PSL) IN R0
0175 069C C1            STRZ    R1        (R0) TO R1
0176 069D 93            LPSL
0177 069E 0C07F8        LODA,R0 LOC       FETCH OLD VALUE R0
0178 06A1 45C0          ANDI,R1 H'C0'     RECONSTRUCT CC IN PSL
0179 06A3 7510          CPSL    RS        SELECT REG. BANK #0
0180 06A5 17            RETC,UN           RETURN
```

```
LINE ADDR  OBJECT  E SOURCE

0182                *  CYCLIC REDUNDANCY CHECK SUBROUTINE
0183                *
0184                *INITIALISATION PROCEDURE
0185                *
0186 06A6 CE07FC    CRCS   STRA,R2 SUCH    SAVE CONTROLCTR IN SUCH
0187 06A9 CD07FD           STRA,R1 SUCH+1  SAVE CONTROLWORD IN SUCH+1
0188 06AC 7708             PPSL    WC      ROTATE WITH CARRY
0189 06AE 0D07FB           LODA,R1 CRC+1   OLD LSB REMAINDER
0190 06B1 0E07FA           LODA,R2 CRC     OLD MSB REMAINDER
0191                *
0192                *EXECUTION OF CRC
0193                *
0194 06B4 0708          LODI,R3 8
0195 06B6 22        LBL1 EORZ    R2         EXCL. OR R2 AND R0 TO R0
0196 06B7 D0             RRL,R0            MSB TO CARRY
0197 06B8 D1             RRL,R1            ROTATE CARRY IN R1
0198 06B9 D2             RRL,R2
0199 06BA 22             EORZ    R2        REPAIR R0
0200 06BB F501           TMI,R1  H'01'     TEST IF EXOR'S WERE ACTIVE
0201 06BD 9804           BCFR,A1 LBL2      BRANCH IF NOT ACTIVE
0202 06BF 2504           EORI,R1 H'04'     APPLY FEEDBACK
0203 06C1 2680           EORI,R2 H'80'
0204 06C3 FB71      LBL2 BDRR,R3 LBL1
0205                *
0206                *DISCHARGE OF SYNDROME
0207                *
0208 06C5 CE07FA           STRA,R2 CRC     NEW MSB REMAINDER
0209 06C8 CD07FB           STRA,R1 CRC+1   NEW LSB REMAINDER
0210 06CB 0E07FC           LODA,R2 SUCH    RESTORE R2 AND R1
0211 06CE 0D07FD           LODA,R1 SUCH+1
0212 06D1 7508             CPSL    WC
0213                *
0214                *INCREMENT OF ADDRESS COUNTER
0215 06D3 0702      IAC  LODI,R3 2
0216 06D5 0F47F4    LOP5 LODA,R0 CML+4,R3,- FETCH L/U HALF ADDRESSCTR
0217 06D8 D805           BIRR,R0 NEXT
0218 06DA CF67F4         STRA,R0 CML+4,R3   RESTORE L/U HALF ADDRESSCTR
0219 06DD 5B76           BRNR,R3 LOP5
0220 06DF CF67F4    NEXT STRA,R0 CML+4,R3   RESTORE L/U HALF ADDRESSCTR
0221 06E2 17             RETC,UN
0222                *
```

```
LINE ADDR  OBJECT  E SOURCE

0224                *SUBROUTINE FOR INHIBIT READING DURING GAP
0225                *
0226 06E3 0420      IGR  LODI,R0 H'20'
0227 06E5 D440           WRTE,R0 CX+DCR     RESET RGT FLIPFLOP
0228 06E7 0719           LODI,R3 25
0229 06E9 3F0689         BSTA,UN LOP8       DELAY FOR 50 MSEC
0230 06EC 0421           LODI,R0 H'21'
0231 06EE D440           WRTE,R0 CX+DCR     SET RGT FLIPFLOP
0232 06F0 17             RETC,UN
0233                *
0234                *
0235                *SUBROUTINE FOR GAPCHECK
0236                *
0237 06F1 0710      GAPC LODI,R3 H'10'      DELAY OF 150 USEC FOR GAPCHECK
0238 06F3 FB7E           BDRR,R3 $
0239 06F5 57A0           REDE,R3 STAT+DCR   READ DCR STATUS
0240 06F7 F740           TMI,R3  H'40'      TEST FOR GAPS SIGNAL
0241 06F9 17             RETC,UN
0242                *
```

```
LINE ADDR  OBJECT  E SOURCE

0244                *INTERRUPT ROUTINE FOR DIGITAL CASSETTE RECORDER
0245                *
0246 06FA                ORG     H'477'
0247 0477 3F068F         BSTA,UN SAVE        CALL SAVE SUBROUTINE
0248 047A 7702           PPSL    COM         LOGICAL COMPARE
0249 047C 0E07F0         LODA,R2 CML         SET CONTROLWORD
0250 047F 02             LODZ    R2          TRANSFER R2 TO R1
0251 0480 C1             STRZ    R1
0252 0481 45F8           ANDI,R1 H'F8'       CLEAR LOWER 3 BITS
0253 0483 4607           ANDI,R2 H'07'       CLEAR UPPER 5 BITS
0254 0485 E606           COMI,R2 H'06'
0255 0487 1C058B         BCTA,EQ ERR         BRANCH IF CONTROLCTR EQUAL 6
0256 048A 7508           CPSL    WC          ADD WITHOUT CARRY
0257                *
0258 048C 02             LODZ    R2
0259 048D 9C04F0         BCFA,Z  DAB         BRANCH IF CONTROLCTR IS NOT ZERO
0260 0490 01             LODZ    R1
0261 0491 9A20           BCFR,N  REPP        BRANCH IF READING MODE
0262                *
0263 0493 04AA           LODI,R0 H'AA'
0264 0495 D480           WRTE,R0 DEC+DCR     WRITE PREAMBLE
0265                *
0266 0497 071C           LODI,R3 28
0267 0499 3F0689         BSTA,UN LOP8        CALL START WRITING DELAY SUBR. (60 MSEC)
0268                *
0269 049C 0424           LODI,R0 H'24'
0270 049E D440           WRTE,R0 CX+DCR      SET FORWARD AND WCD, RES ERASE FF
0271 04A0 F508       AL  TMI,R1  H'08'
0272 04A2 9C0560         BCFA,A1 INCC        BRANCH IF TEMP FLAG NOT SET
0273 04A5 3F06F1         BSTA,UN GAPC        CALL GAPCHECK SUBROUTINE
0274 04A8 9C05AF         BCFA,A1 RIC         BRANCH IF NO GAP
0275 04AB 2508           EORI,R1 H'08'       CLEAR TEMP FLAG
0276 04AD 3F06E3         BSTA,UN IGR         INHIBIT READING DURING GAP
0277 04B0 1F05AF         BCTA,UN RIC
0278                *
0279 04B3 5420       REPP REDE,R0 IEC+DCR    READ FIRST BYTE
0280 04B5 E4AA           COMI,R0 H'AA'
0281 04B7 1867           BCTR,EQ AL          BRANCH IF PREAMBLE
0282 04B9 1F05AF         BCTA,UN RIC
0283                *
0284 04BC 3F06E3     NTST BSTA,UN IGR        RESET RGT DURING GAP
0285 04BF 0702           LODI,R3 2
0286 04C1 0F47FA     NL  LODA,R0 CRC,R3,-
0287 04C4 9825           BCFR,Z  ERR         BRANCH IF CRC BYTES ARE NOT ZERO
0288 04C6 5B79           BRNR,R3 NL
```

```
LINE ADDR  OBJECT  E SOURCE

0290 04C8 0C07F5          LODA,R0 CML+5      FETCH ADDRESSCTR LOW
0291 04CB 188C            BCTR,Z  LOZ        BRANCH IF ZERO
0292 04CD A401            SUBI,R0 1          DECR. ADDRESSCTR LOW
0293 04CF 188A            BCTR,Z  LOZA       BRANCH IF ZERO
0294 04D1 F800            BDRR,R0 $+2        DECREMENT ADDRESSCTR LOW
0295 04D3 CC07F5          STRA,R0 CML+5      RESTORE ADDRESSCTR LOW
0296 04D6 1F059C          BCTA,UN DOBC
0297 04D9 F800       LOZ  BDRR,R0 $+2        DECREMENT ADDRESSCTR LOW
0298 04DB F800       LOZA BDRR,R0 $+2        DECREMENT ADDRESSCTR LOW
0299 04DD CC07F5          STRA,R0 CML+5      RESTORE ADDRESS CTR LOW
0300 04E0 0C07F4          LODA,R0 CML+4      FETCH ADDRESSCTR HIGH
0301 04E3 F800            BDRR,R0 $+2        DECREMENT ADDRESSCTR HIGH
0302 04E5 CC07F4          STRA,R0 CML+4      RESTORE ADDRESSCTR HIGH
0303 04E8 1F059C          BCTA,UN DOBC
0304                 *
0305 04EB 2520       ERR  EORI,R1 H'20'      SET CRC ERROR BIT
0306 04ED 1F05BD          BCTA,UN DICO
0307                 *
0308 04F0 E601       DAB  COMI,R2 H'01'
0309 04F2 9C053D          BCFA,EQ CRCH       BRANCH IF CONTROLCTR UNEQUAL TO ONE
0310                 *
0311 04F5 F540            TMI,R1  H'40'
0312 04F7 9813            BCFR,A1 FDB        BRANCH IF BYTE IS DATA
0313                 *
0314 04F9 01         DELT LODZ    R1
0315 04FA 9A06            BCFR,N  RZB        BRANCH IF READING MODE
0316                 *
0317 04FC 20              EORZ    R0
0318 04FD D480            WRTE,R0 DEC+DCR    OUTPUT A ZERO BYTE
0319 04FF 1F0560          BCTA,UN INCC       BRANCH TO INCREMENT CONTROLCTR
0320                 *
0321 0502 5420       RZB  REDE,R0 IEC+DCR    INPUT A BYTE
0322 0504 1C0560          BCTA,Z  INCC       BRANCH TO INCREMENT CONTROLCTR
0323 0507 6508            IORI,R1 H'08'      SET TEMPORARY FLAG
0324 0509 1F05AF          BCTA,UN RIC
0325                 *
0326 050C 01         FDB  LODZ    R1
0327 050D 9A19            BCFR,N  NEW        BRANCH IF READING MODE
0328                 *
0329 050F 0C87F4          LODA,R0 *CML+4     FETCH BYTE TO BE WRITTEN
0330 0512 D480            WRTE,R0 DEC+DCR    OUTPUT A BYTE
0331 0514 3F06A6          BSTA,UN CRCS       CRC SUBROUTINE AND INCREMENT ADDR. CTR
0332                 *
0333 0517 0F07F1          LODA,R3 CML+1      FETCH BYTECTR.
0334 051A FB06            BDRR,R3 COL
0335 051C CF07F1          STRA,R3 CML+1      IF ZERO RESTORE AND
0336 051F 1F0560          BCTA,UN INCC       INCREMENT CONTROLCTR
```

```
LINE ADDR  OBJECT  E SOURCE

0338 0522 CF07F1     COL  STRA,R3 CML+1      RESTORE BYTECTR.
0339 0525 1F05B9          BCTA,UN RERE
0340                 *
0341 0528 5420       NEW  REDE,R0 IEC+DCR    READ A BYTE
0342 052A CC87F4          STRA,R0 *CML+4     STORE THE READ BYTE
0343 052D E4AA            COMI,R0 H'AA'
0344 052F 9806            BCFR,EQ  CALL      BRANCH IF NOT AA
0345 0531 3F06F1          BSTA,UN GAPC       CALL GAPCHECK SUBROUTINE
0346 0534 1C04BC          BCTA,A1 NTST       BRANCH TO CRC CHECK IF GAP
0347 0537 3F06A6     CALL BSTA,UN CRCS       CRC SUBROUTINE AND INCREMENT ADDR.CTR
0348 053A 1F05B9          BCTA,UN RERE       BRANCH TO RESTORE REGISTERS
0349 053D E602       CRCH COMI,R2  H'02'
0350 053F 9807            BCFR,EQ SECH       BRANCH IF CONTROLCTR UNEQUAL TO 2
0351                 *
0352 0541 F540            TMI,R1 H'40'
0353 0543 9807            BCFR,A1  EXT       BRANCH IF BYTE IS DATA
0354                 *
0355 0545 DE04F9          BIRA,R2 DELT       INCREMENT CONTROLCTR.
0356                 *
0357 0548 E603       SECH COMI,R2 H'03'
0358 054A 9808            BCFR,Z  POST       BRANCH IF CONTROLCTR UNEQUAL TO 3
0359                 *
0360 054C 0E67F8     EXT  LODA,R0 CRC-2,R2
0361 054F D480            WRTE,R0 DEC+DCR    OUTPUT CRC CHARACTERS
0362 0551 1F0560          BCTA,UN INCC
0363                 *
0364 0554 E604       POST COMI,R2 H'04'
0365 0556 982E            BCFR,Z  LIT        BRANCH IF CONTROLCTR UNEQUAL TO 4
0366                 *
0367 0558 01              LODZ    R1
0368 0559 9E056A          BCFA,N  POTM       BRANCH IF READING MODE
0369                 *
0370 055C 04AA            LODI,R0 H'AA'
0371 055E D480            WRTE,R0 DEC+DCR    OUTPUT POSTAMBLE
0372 0560 8601       INCC ADDI,R2 H'01'      INCREMENT CONTROLCTR
0373 0562 01              LODZ    R1         TRANSFER R1 TO R0
0374 0563 82              ADDZ    R2         INCLUDE CONTROLCTR
0375 0564 CC07F0          STRA,R0 CML        SAVE COMPLETE CONTROLWORD
0376 0567 1F05B9          BCTA,UN RERE
0377                 *
0378 056A 5420       POTM REDE,R0 IEC+DCR    READ BYTE
0379 056C E4AA            COMI,R0 H'AA'
0380 056E 9C05AF          BCFA,EQ RIC        BRANCH IF NOT AA
0381 0571 3F06F1          BSTA,UN GAPC       CALL GAPCHECK SUBROUTINE
0382 0574 9C05AF          BCFA,A1 RIC
0383 0577 0C07F2     DM   LODA,R0 CML+2      FETCH TAPE MARK COUNTER
0384 057A F882            BDRR,R0 ONCE
0385 057C 1B3F            BCTR,UN DICO
0386 057E CC07F2     ONCE STRA,R0 CML+2      RESTORE TAPE MARK COUNTER
0387 0581 3F06E3          BSTA,UN IGR        RESET RGT DURING GAP
0388 0584 1B20            BCTR,UN RWT
```

```
LINE ADDR  OBJECT  E SOURCE

0390 0586 20        LIT  EORZ   R0
0391 0587 D480           WRTE,R0 OEC+DCR    OUTPUT A ZERO BYTE
0392 0589 1B55           BCTR,UN INCC       BRANCH TO INCREMENT CONTROLCTR
0393 058B 0430      ERA  LODI,R0 H'30'
0394 058D D440           WRTE,R0 CX+DCR     SET ERASE, FORW AND WCD FF
0395 058F 04AA           LODI,R0 H'AA'
0396 0591 D480           WRTE,R0 OEC+DCR    OUTPUT PRE/POSTAMBLE
0397 0593 072F           LODI,R3 47
0398 0595 3F0689         BSTA,UN LOP8       CALL GAP DELAY SUBROUTINE (100 MSEC)
0399 0598 F540           TMI,R1  H'40'
0400 059A 185B           BCTR,A1 DM         BRANCH IF TAPEMARK
0401 059C 0C07F3    DOBC LODA,R0 CML+3      FETCH DATABLOCK-COUNTER
0402 059F F882           BDRR,R0 MOR
0403 05A1 1B1A           BCTR,UN DICO
0404 05A3 CC07F3    MOR  STRA,R0 CML+3      RESTORE DATA BLOCK COUNTER
0405 05A6 01        RWT  LODZ   R1
0406 05A7 9A06           BCFR,N  RIC        BRANCH IF READING MODE
0407 05A9 8501      AAD  ADDI,R1 1          SET ONE TO CONTROLCTR
0408 05AB 0424           LODI,R0 H'24'      SET WCD AND FORWARD, RES ERASE FLIPFLOP
0409 05AD D440           WRTE,R0 CX+DCR
0410 05AF CD07F0    RIC  STRA,R1 CML        RESTORE CONTROLWORD
0411 05B2 20             EORZ   R0          CLEAR R0
0412 05B3 CC07FA         STRA,R0 CRC        CLEAR CRC LOCATIONS
0413 05B6 CC07FB         STRA,R0 CRC+1
0414 05B9 3F0699    RERE BSTA,UN REST       RESTORE REGISTERS, STATUS
0415 05BC 37             RETE,UN            RETURN TO MAIN PROGRAM, ENABLE INTERRUPT
0416                *
0417 05BD 01        DICO LODZ   R1
0418 05BE 9A0D           BCFR,N  DISC       BRANCH IF READING
0419 05C0 0410           LODI,R0 H'10'
0420 05C2 D440           WRTE,R0 CX+DCR     RESET FORWARD, SET WCD AND ERASE
0421 05C4 0713      FIN  LODI,R3 19
0422 05C6 3F0689         BSTA,UN LOP8       CALL STOP FORWARD DELAY SUBROUTINE (40MS)
0423 05C9 D4E0           WRTE,R0 DX+DCR     DISCONNECTION COMMAND
0424 05CB 1B62           BCTR,UN RIC
0425                *
0426 05CD D4E0      DISC WRTE,R0 DX+DCR     DISCONNECTION COMMAND
0427 05CF 1B73           BCTR,UN FIN
0428                *
0429                *
0430 0477                END    H'477'

 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.
AS53	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 ASCII • ASCII to BCD • Hexadecimal to ASCII • ASCII to Hexadecimal
AS55	Fixed Point Decimal Arithmetic Routines	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.

Printed in The Netherlands Date of release: 9 December 1977 T.D. 9399 500 06401