Introduction

1.1. Scope

The aim of this document is to supply sufficient information to enable design engineers, familiar with large scale integrated digital circuit techniques, to interface their own systems to the Philips DCR 4 and MDCR 200 series cassette recorders.
This bulletin does not detail specific applications; one typical application is described in Section 3.
Section 2 starts with a summary of NRZ recording techniques and is biased towards the phase encoding method. The implementation of these techniques is also detailed in this section.
The last section details a typical application based around the 8085 and the MDCR; interface signal timing for both types of recorder is given together with a description and suggestions on how the interface signals would be used.
The level of description in Section 3 assumes that the designer is capable of developing his own system interface using an 8085 microprocessor; labelling routines, system to interface control, etc. are not dealt with.

Note: This document may be subject to change without prior notice.

1.2. Supplementary Documentation

The following publications are deemed necessary to fully understand the content of this document:

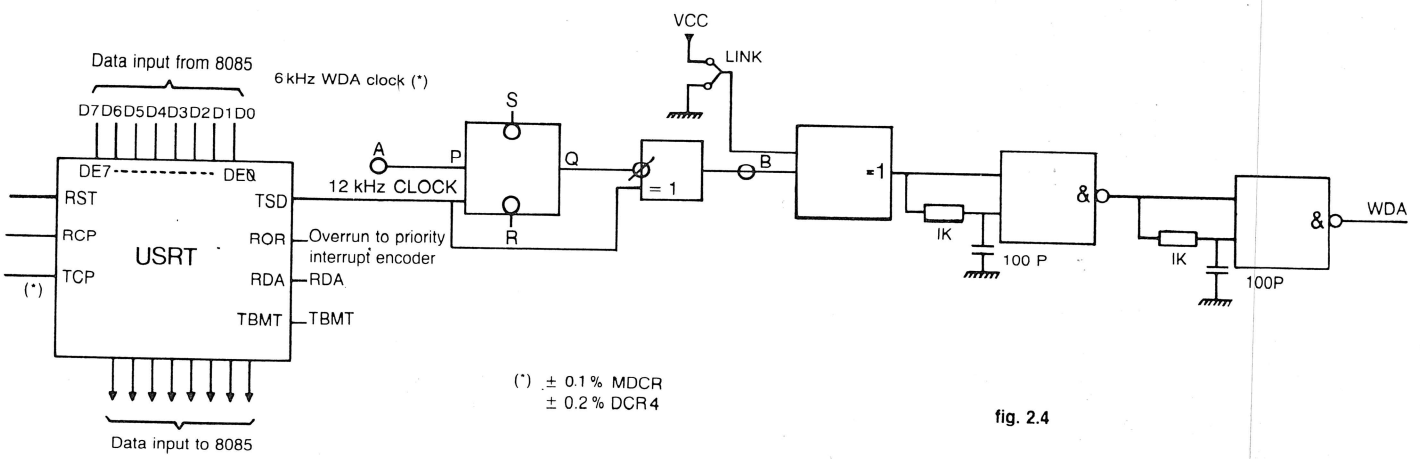
I) European Computer Manufacturers Association Standards EMCA-34 and EMCA-41.

II) Philips ELA
Digital Cassette Recorder type DCR 4A-4B-Service Manual.
Technical Information Manual Mini-DCR.

III) Intel Corporation
MCS-85 User's Manual.

1.3. ECMA-34 and 41 - A Summary

The interchange of digitally recorded magnetic tape cassettes from one system to another without the same standard would obviously create problems. The ECMA standards 34 and 41 were drafted in an attempt to give users a sound basis on which to build their system.
ECMA 34 is concerned with the physical properties (cassette dimensions, support planes, window sizes etc.) and digital data format and definitions (gap lengths, tape marks, postamble-preamble, CRC characters, method of recording etc.).
ECMA 41 is concerned with the structuring and identification of the data files recorded on the cassettes.
This document will not detail standards.



(*) ± 0.1% MDCR
± 0.2% DCR4

fig. 2.4

Recording techniques

2.1. NRZ (Non Return to Zero) Recording

There are basically three NRZ recording methods:

I) Level:
'Ones' and 'zeroes' are represented by opposite levels of recording.

II) Mark:
A change in the level of recording represents a 'one' and no change represents a 'zero'.

III) Space:
This method is the opposite of Mark.

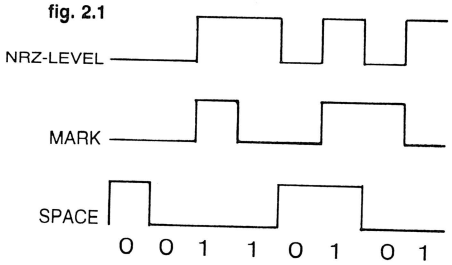
The disadvantages of NRZ recording are:

a) A separate clocktrack is required for reading data: this halves the tape storage capacity and tape skew can also create problems.

b) The recorded frequency can vary from d.c. upwards; read amplifiers require a wide bandwith – adversely affecting the signal to noise ration.

c) Bit to bit jitter is severely restricted.

fig. 2.1



2.2. PE (Phase Encoding)

PE can be regarded as a development of NRZ; the differences are as follows:

I) Level
A flux transition to the erase polarity represents a 'one' and a flux transition in the opposite direction represents a 'zero'. Additional transitions are required if successive bits have the same value.

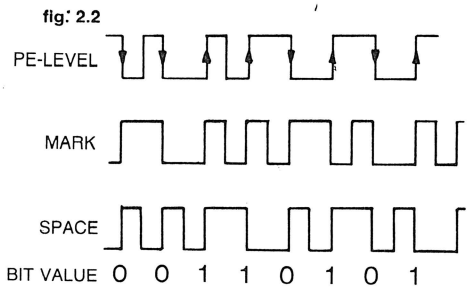
II) Mark and Space
A bit transition occurs at the beginning of each bit cell. An extra transition is given in the middle of the bit cell to indicate a 'one' for Mark PE, or 'zero' for Space PE.

The advantages of phase encoding are summarised thus:

a) The technique provides self clocking facilities since at least one flux transition occurs for each bit: bit-to-bit jitter problems are reduced.

b) Only two frequencies are recorded thus reducing the wide bandwidth required for NRZ and consequently improving the signal to noise ratio.

fig. 2.2



2.3. Data Locked Circuit for PE Data (DCR 4)

The circuit shown in Fig. 2.3. can be used to produce clock pulses that are locked onto the phase encoded data signal RDA from the DCR 4.

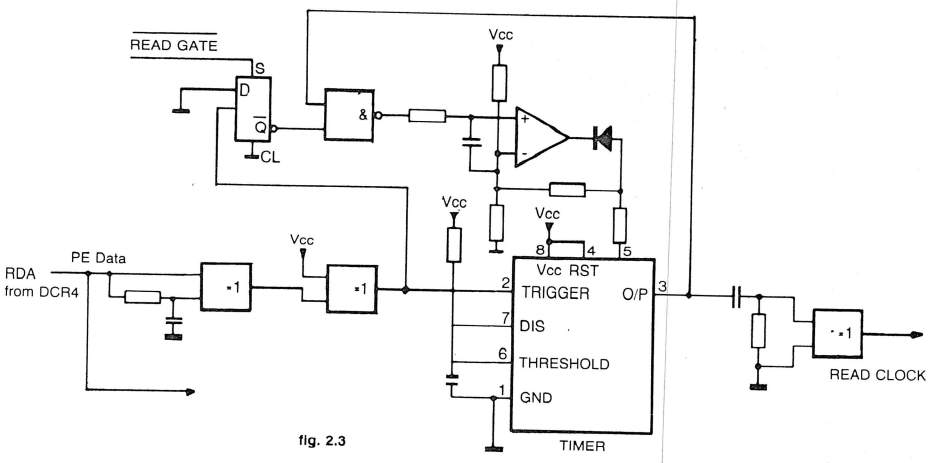


fig. 2.3

The circuit is synchronised by the preamble to give an indication of the bit value as each clock pulse goes negative.
The circuit required for the DRC 4 is almost identical to that built already in to the MDCR. In both cases the trailing edge of the read clock pulses can be used to clock data (RDA) into the receive shift register of the USRT.

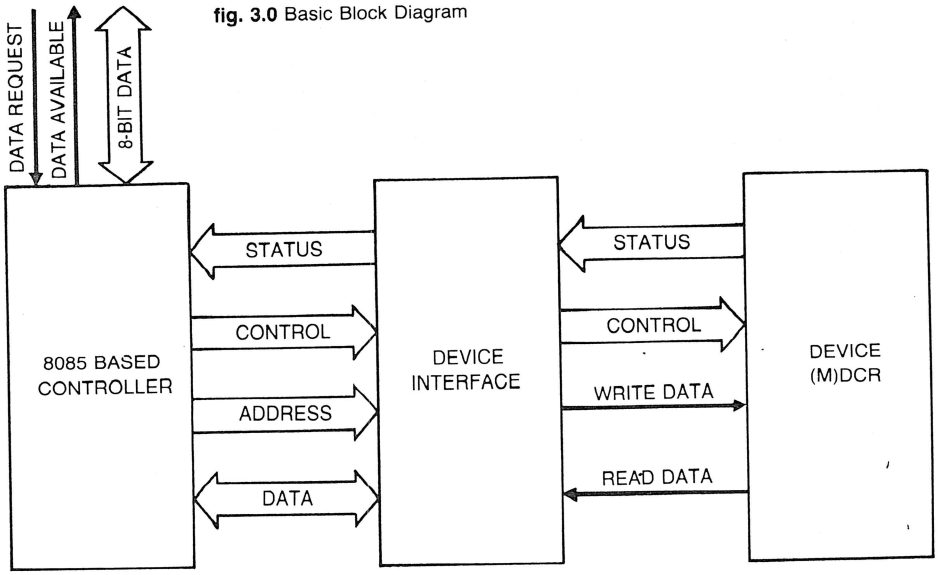
2.4. Writing PE Data (CMOS only) (DCR 4 and MDCR)

The schematic Fig. 2.4. on page 2 shows output TSD (Transmit Serial Data) connected to the WDA line via two exclusive OR gates; the first gate is 'strobed' with 6 kHz (± 0.2 % or ± 1 %) transition signal to produce phase encoded data which is fed to the second gate. The other input to the second EXOR gate is connected to Vcc or ground to ensure correct polarity on the WDA line to the recorder. The output is fed to the DCR via a 'glitch' remover.
NOTE:
The clock pulses, TCP for strobing data from the USRT and the 6 kHz data strobe must be generated from the same source.

Application information

BASIC BLOCK DIAGRAM DESCRIPTION

fig. 3.0 Basic Block Diagram



3.1. General

The 8085 Controller should be designed to receive and transmit 8-bit parallel data in an continuous mode; any further facilities required (e.g. variable block length, line mode select, labelling features etc.) could be designed into the firmware of the Controller but are not the subject of this document.
Since ECMA specifies a maximum data block length of 256-8 bit characters, this description will assume that maximum block lengths are written.

3.2. Data Storage Sequence (write) (DCR 4 + MDCR)

8-bit parallel data input to the Controller, under control of the Data Request and Data Available signals would, typically be stored in a predefined buffer in RAM that already contains any required additional characters (labels, flags, carriage return – line feeds etc.). As the data is received the CRC character should be formulated and appended to the data block. When the data storage buffer is full, the Controller should enter an interrupt orientated routine to write data to the device interface and also receive another block of data input (use could be made of a second storage buffer in RAM for this purpose).

3.3. Interfacing the 8085 data bus with the read and write data logic (DCR 4 + MDCR)

Two hardware methods of generating phase encoded data are shown in figures 2.4. and 3.3. The timing of shift pulses for the 16-bit shift register could be realised with 8085 firmware. However, utilisation of the USRT (COM 2601 of SMC micro-system corp.) is recommended purely from a costing level; this device is an ideal method for interfacing both read and write data to the 8085 data bus.
Some aspects of its use can be summarised as follows:
The TBMT (Transmit Buffer Empty) signal can be used as a high priority interrupt to the 8085; as a result of the interrupt, another character is loaded into the USRT.
The sync register can be used for pre-amble and post-amble transmission and recognition on receipt.
The TCP (Transmitter Clock) timing 6 kHz, must be derived from the 6 kHz signal used for clocking the write data logic.
The RDA (Receive Data Available) signal can be used to interrupt the 8085 for Read Data Storage.

3.4. Interface Signal Timing

Detailed timing diagrams for the MDCR and both versions of DCR 4 are given in Appendix A. This paragraph generalises the usage of these signals.

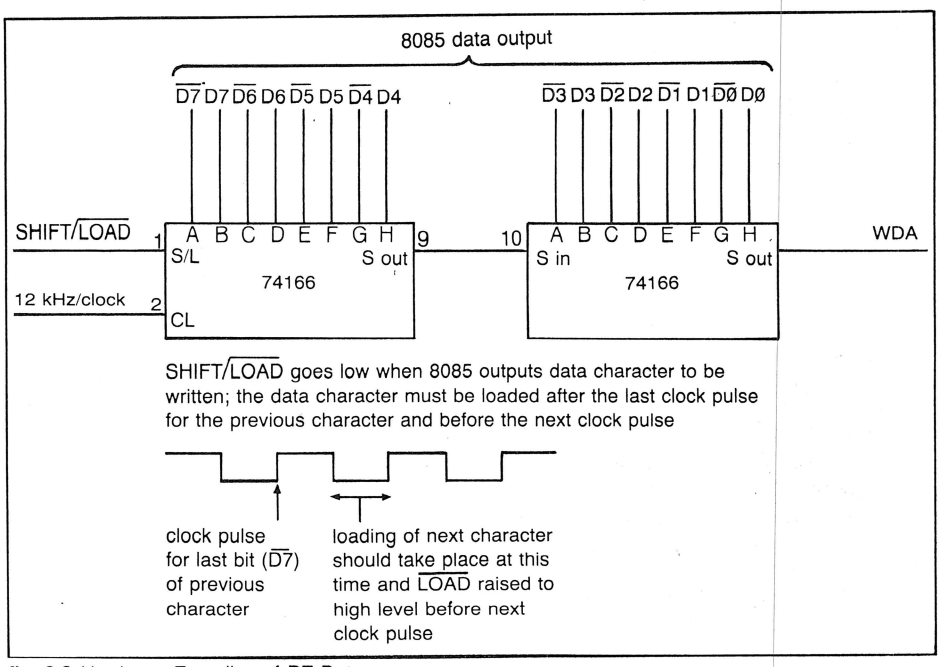


fig. 3.3 Hardware Encoding of PE Data

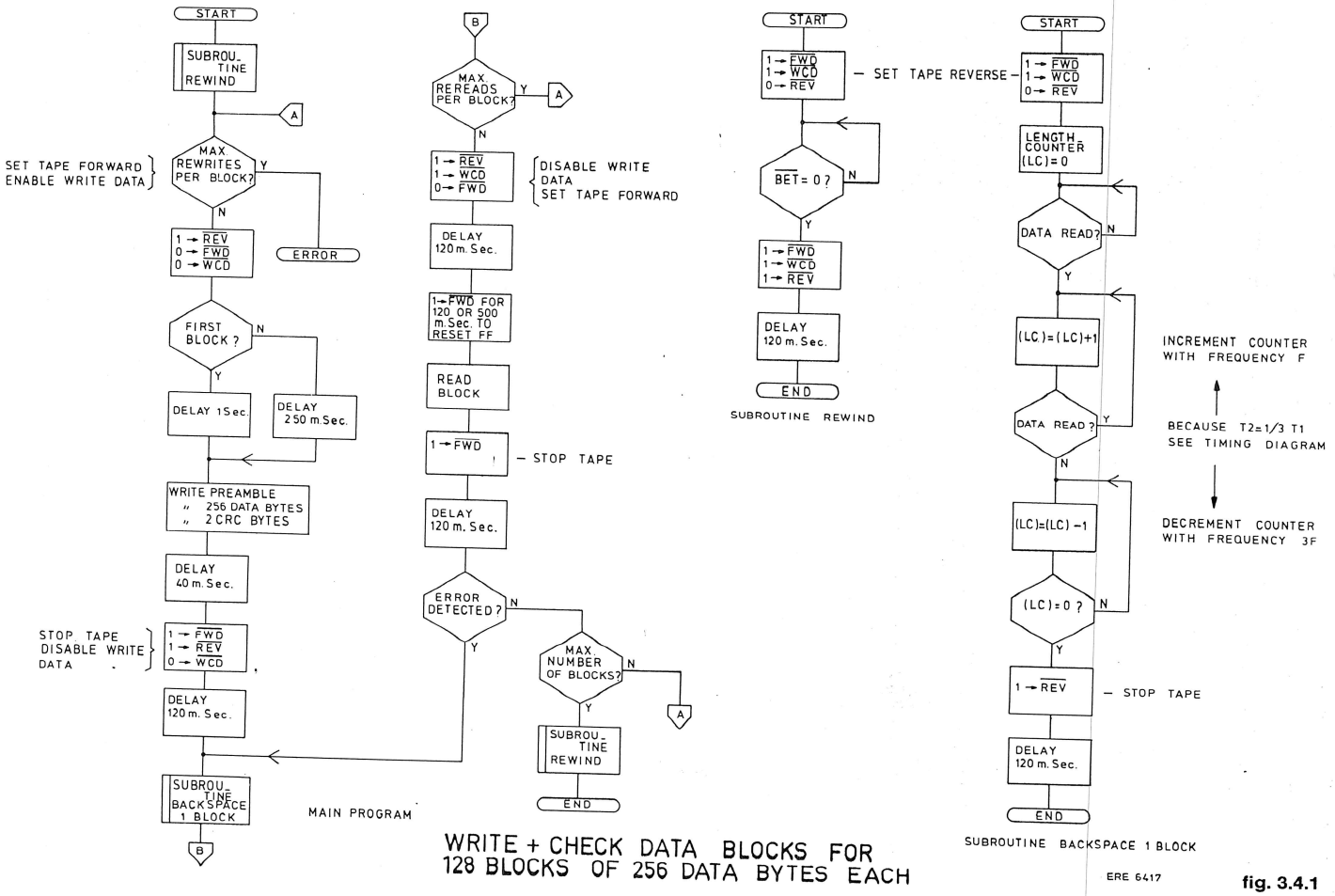
3.4.1. Proposal for MDCR (partly useable for DCR 4)

After inserting and loading a mini cassette in the MDCR. CIP, and WEN if fitted, go 'low'; these signals can be used to indicate that the MDCR is ready for operation.
The flowcharts in fig. 3.4.1. show the basic operating routines of the MDCR; these flowcharts and the timing diagram given in fig. 3.4.1. give sufficient detail for interfacing purpose. A brief description of the routines is given below.

I) **Initialisation**
The firmware in the system should be initialised (can be entered via a CIP interrupt) to set all command lines to the 'high' level ($\overline{FWD}$ $\overline{WCD}$ $\overline{REV}$) and reset the CIP signal latch (see fig. 3.4.2.). Whether preparing to Read or Write Data the Rewind Subroutine should be entered first.

II) **Rewind Subroutine (MDCR only)**
All signals are deactivated unless stated otherwise. The reverse signal is activated ($\overline{REV} = \phi$ until the beginning/end of tape signal is activated ($\overline{BET} = \phi$). The reverse signal is then deactivated and the routine times out the stop time (120 milliseconds) before returning from the call.

III) **Backspace One Block (MDCR only)**
The reverse signal is activated ($\overline{REV} = \phi$) and the routine waits for Read Data to be input. Whilst RDA is active the routine increments a counter (starting from zero). When no more RDA is input the routine decrements the counter, at three times the rate, and when the counter reaches zero the reverse signal is deactivated. The routine then waits 120 milliseconds before returning from the call.



WRITE + CHECK DATA BLOCKS FOR 128 BLOCKS OF 256 DATA BYTES EACH

ERE 6417

fig. 3.4.1

IV) **Initial Gap**
To ensure that the tape is well clear of the beginning of the tape a delay of one second allows the tape to transport approximately 12 inches before writing the initial gap. (on the DCR 4 the initial gap is written immediately following the BET hole).
The WCD line is held low (high for the DCR 4) and WDA line held low to write the initial gap.

V) **Reading Data**
When the receive shift register is full (8 bits) the byte is transferred to the receive buffer register in the USRT and the RDA, Receive Data Available, signal pin 26 of the USRT goes to '1'. This RDA signal can be used to give a priority interrupt to the 8085.
A firmware routine should then be entered that reads the data from the USRT, calculates partial CRC and checks for end of block.
If the preamble character is loaded into the USRT Receiver Synch Register during initialisation, the USRT will signal receipt of each AA byte via the SCR pin 17 output; this can be used to signal the start of read data blocks from the MDCR.
'No data' can be used to signal end of block/check CRC.

VI) **Writing Data**
Before writing data to the MDCR the following conditions must be fulfilled.

I) Cassette loaded with write enable plugs

$\overline{CIP} = \overline{WEN} = \phi$

II) The write data buffer in microprocessor RAM must be full (i.e. data block and CRC word).

III) The transmitter character register in the USRT must be loaded with the pre/post amble byte AA Hex.

The tape should then be transported to the Begin of Tape and Initial Gap written.
After writing the initial gap the write clock signal should be applied to the USRT; this starts writing of the preamble – the first byte of data can be loaded at this time. 8085 interrupts should be enabled and each time the TBMT Interrupt is received the firmware should output the next data byte.

3.5. Read after Write Error Checking DCR 4B

Referring to the DCR 4B waveforms given in Appendix A of this document, it can be seen that the RDA signal is received some 20 m.secs after starting to write data. A simple way of testing that data is being written, would be to use the RDA signal to trigger a retriggerable single-shot, having an activation time slightly longer than the bit cell time (166 μsecs); if the single shot de-actives a write error status flag could be set and the operation retried further along the tape – See 'Retry Counters'
This method does not detect single drop-outs; if this feature is required then logic is required to perform a CRC check. The 9401 data sheet contains a brief description of CRC generation and checking.

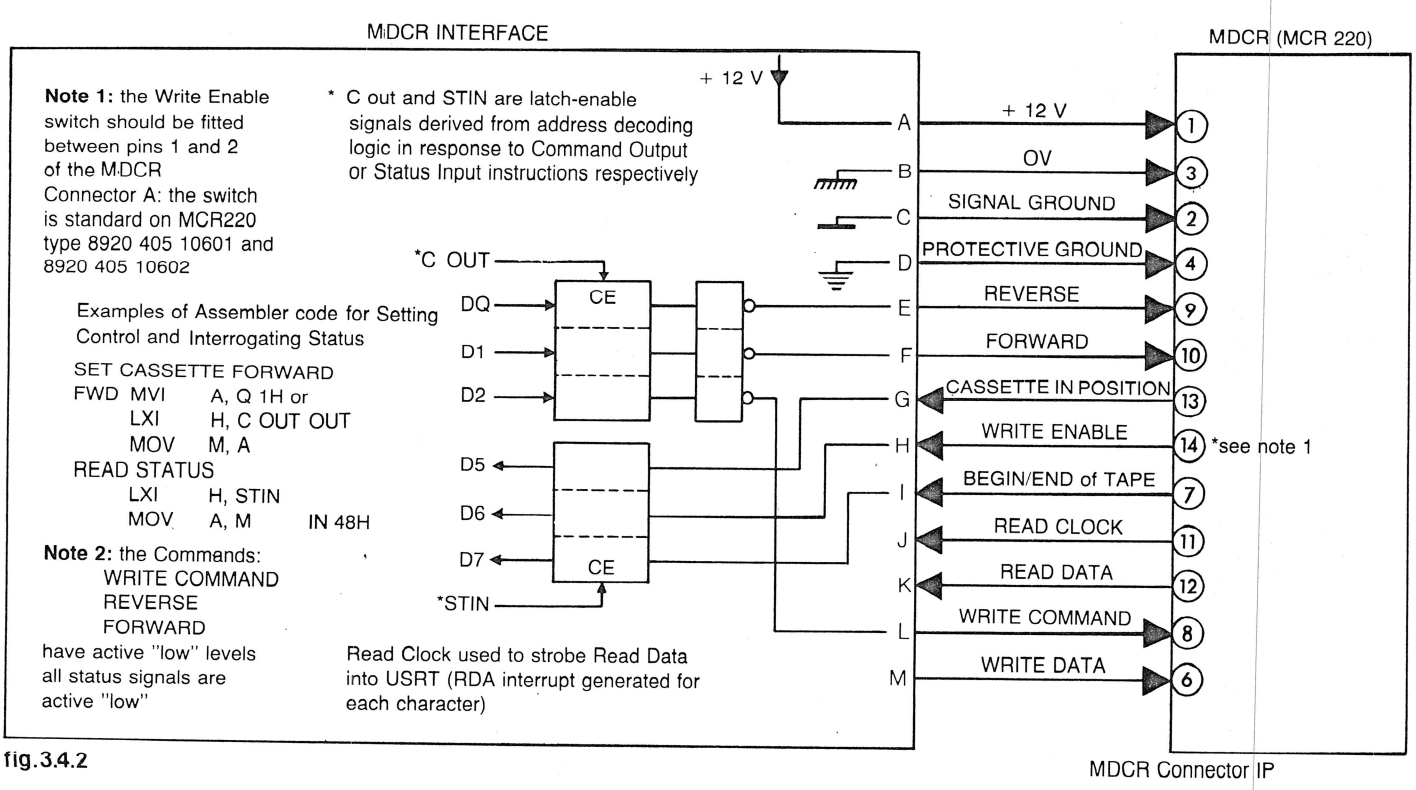
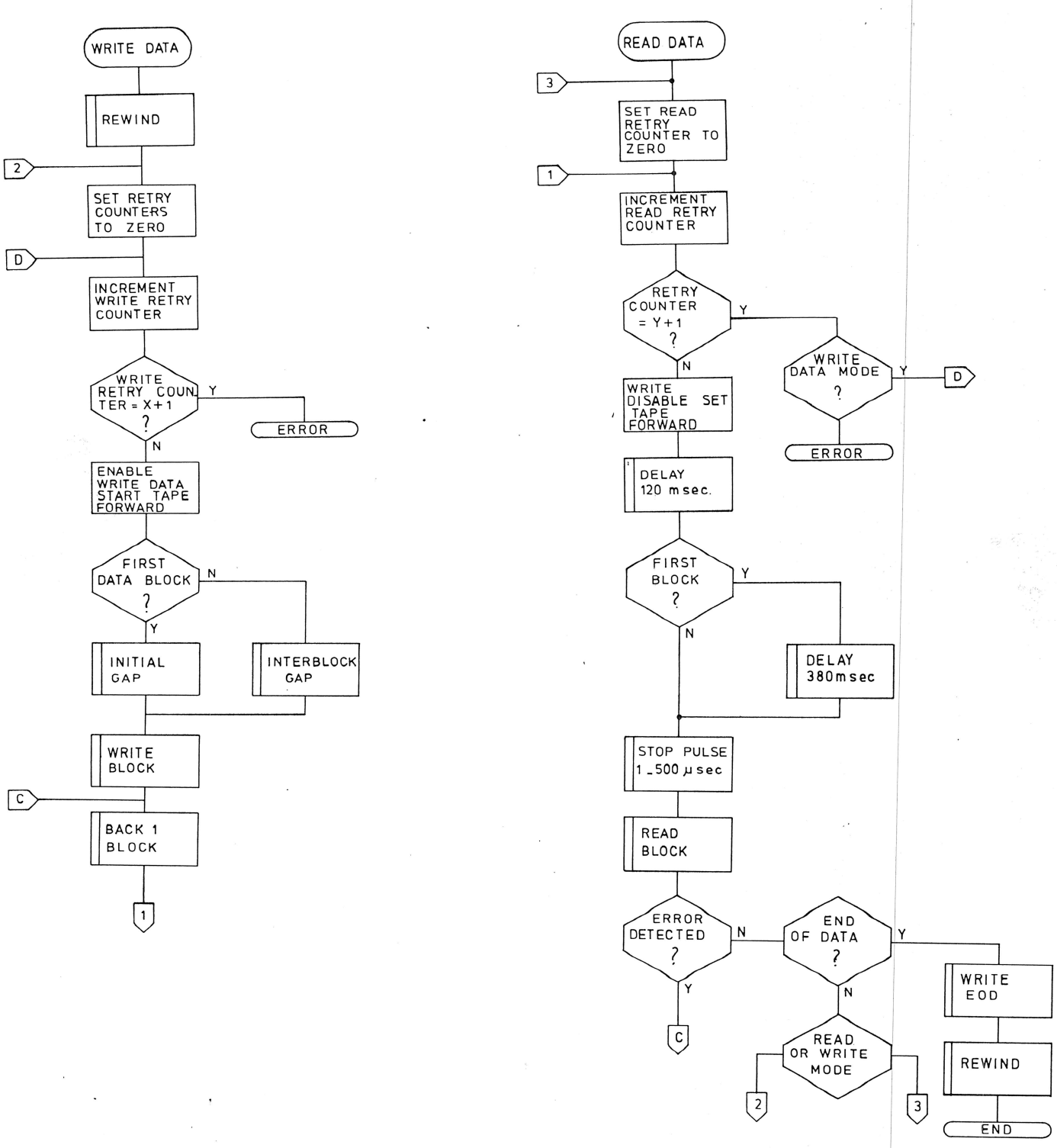


fig.3.4.2



ERE 6461

fig. 3.4.3

3.6. Retry Counters (DCR 4 + MDCR)

Two counters are required for read and write retries.
Typically, a count of ten retries is regarded as sufficient. The read and write routines should contain firmware that automatically increments and tests for a maximum count after detecting a read or write error. If the maximum count is not reached the tape should be backspaced one block and the read or write retried.
On the DCR 4, since no reverse is possible, a flag should be used to indicate a read-after-write error detected in the previous block and that the previous block has been rewritten.

So:

Assuming block n-1 has a read after write error, block n starts with a flag to indicate the fact. After reading block n the flag in block N + 1 is used to indicate whether the second attempt was successful or not and so throughout the tape.

NOTE:
A block that was wrong during RAW can be right whilst performing read. (Due to different threshold levels)

3.7. CRC Generation and Checking (DCR 4 + MDCR)

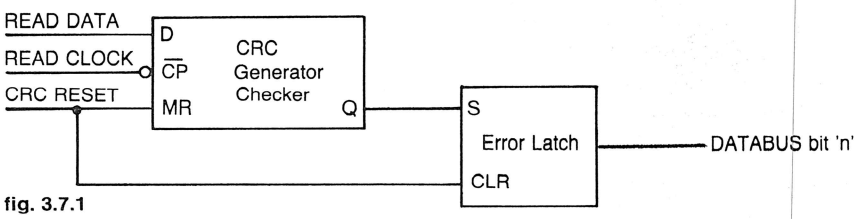


fig. 3.7.1

A typical method of handling the Cyclic Redundancy check character would be to generate the CRC character, whilst writing the data block into the Write Buffer, using 8085 firmware and checking during read data using hardware (various CRC Generator/Checker chips are available). The Fairchild 9401 chip is outlined in the schematic above; if required the 9401 chip can also be used to generate the CRC character. The 9401 should have a master reset 'high' pulse applied during the last bit of the preamble (if the USRT is used for assembling read data bytes, the Synch Character Received signal – generated on detection of the Preamble character – can be used to trigger a CRC Reset pulse generator). The Error Latch can be sampled and tested by the 8085, at the end of reading the data block, to determine whether a read retry should be made nor not.
Since ECMA-34 specifies the CRC polynomial CRC-16 ($x^{16} + x^{15} + x^2 + 1$), the select code inputs to the 9401 should all be grounded.
For more information on the 9401 the reader should consult the Fairchild Data Sheet for this device.

3.8. Interrupt and Interrupt Vector Generation Logic (DCR 4 + MDCR)

The 8085 Controller should be interrupt orientated in order to ensure that the Read and Write data character buffers (in this case the USRT) are emptied and filled at the correct time.
Interrupts and their respective vectored addresses can be generated via the Intel 74148/8 bit to Octal Priority encoder chip.
Naturally, Write Data interrupts should have top priority, followed by Read Data interrupts; a practical manner of realising this is shown in the schematic below..

In this schematic the interrupt priority is as follows:

1st – Transmit Buffer Empty (TBMT) – USRT ready to be loaded with next Write Data character.

2nd – Read Data Available (RDA) – the Read Data Buffer in the USRT is full and should be read before the next interrupt ROR.

3rd – Receiver Over-Run (ROR) – indicates that the previously received Read Data character was not read in time.

4th – Synch Character Received – can be used as a 'preamble detected' signal when reading data blocks – can also be used to detect postamble.

It is possible to connect up to eight inputs to the 74148 chip; this leaves another four lower priority interrupts that can be allocated to whatever additional facilities the designer wishes.

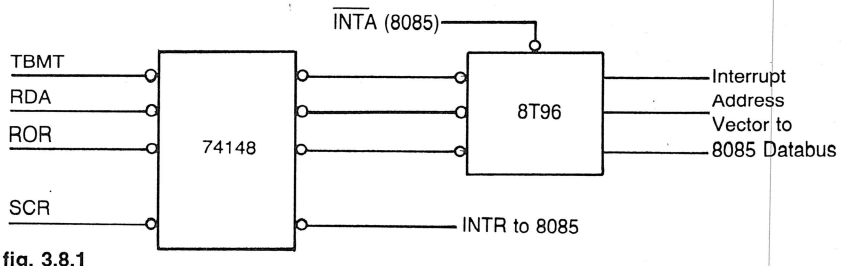
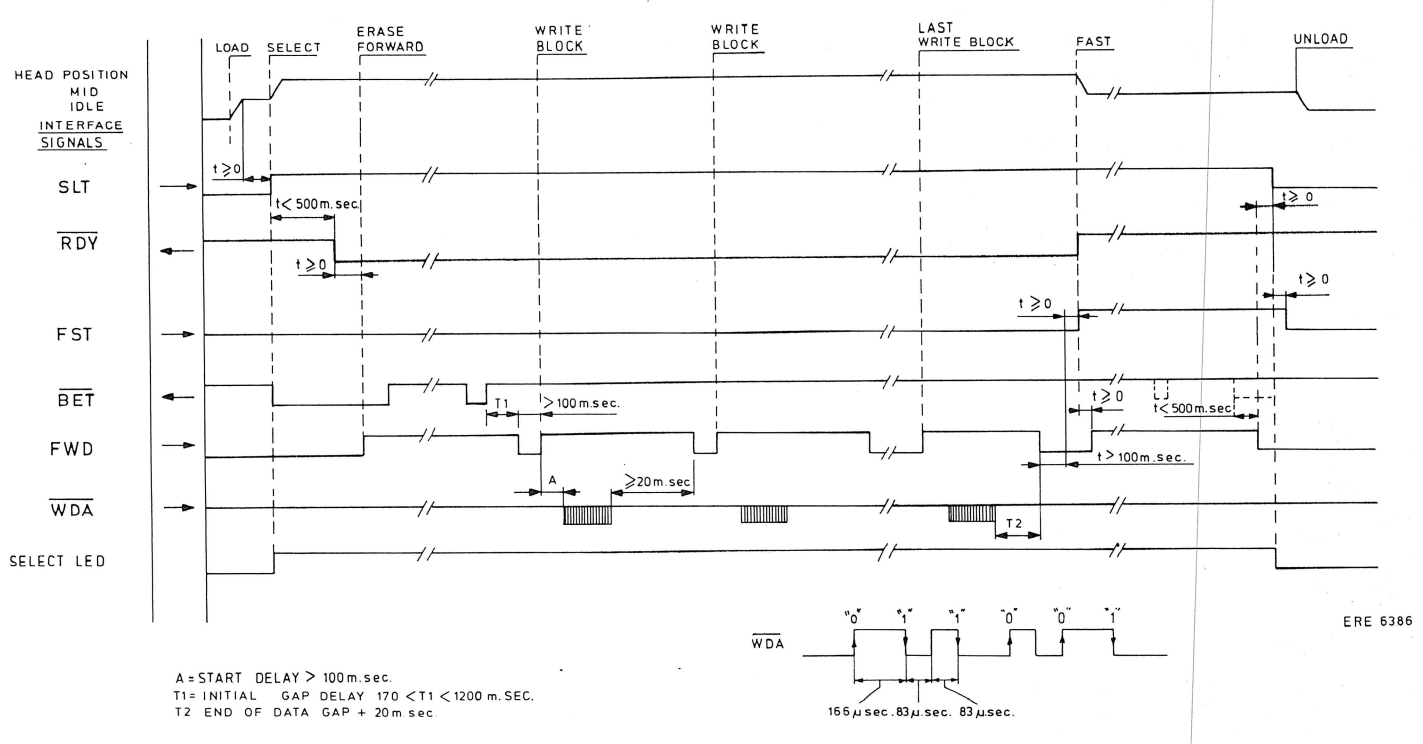


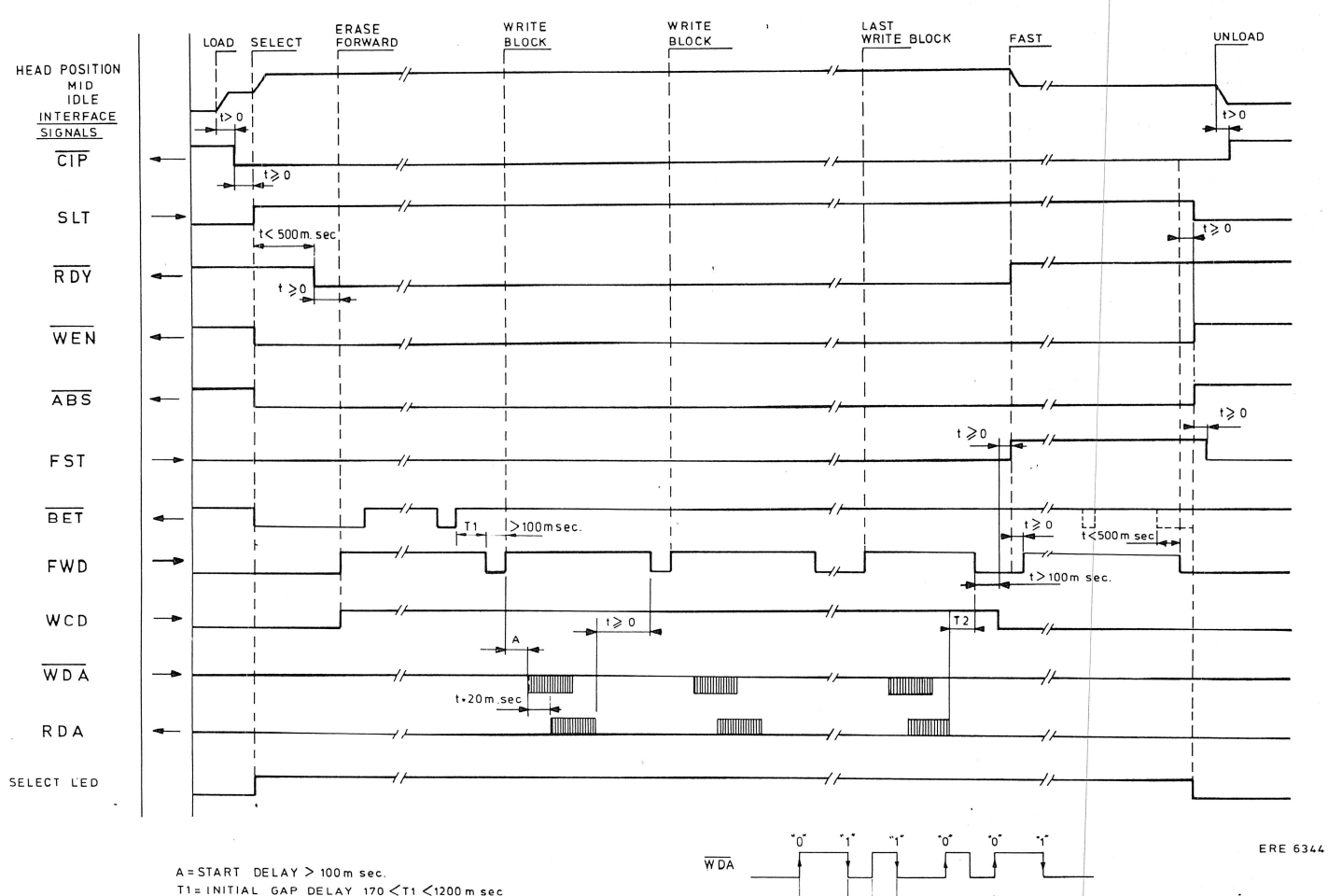
fig. 3.8.1

APPENDIX A



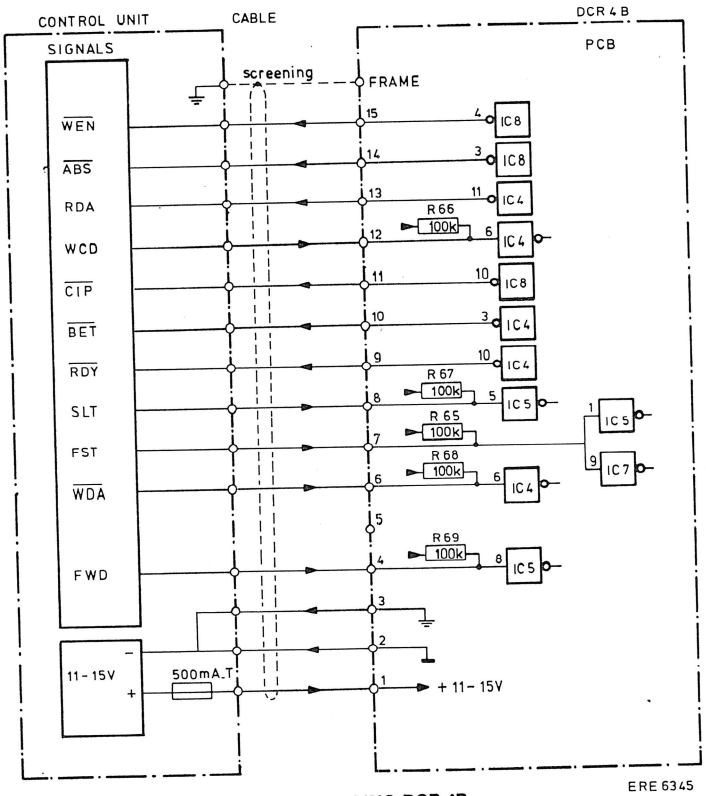
A = START DELAY > 100 m.sec.
T1 = INITIAL GAP DELAY 170 < T1 < 1200 m. SEC.
T2 END OF DATA GAP + 20 m sec

TIMING DIAGRAM DCR 4A

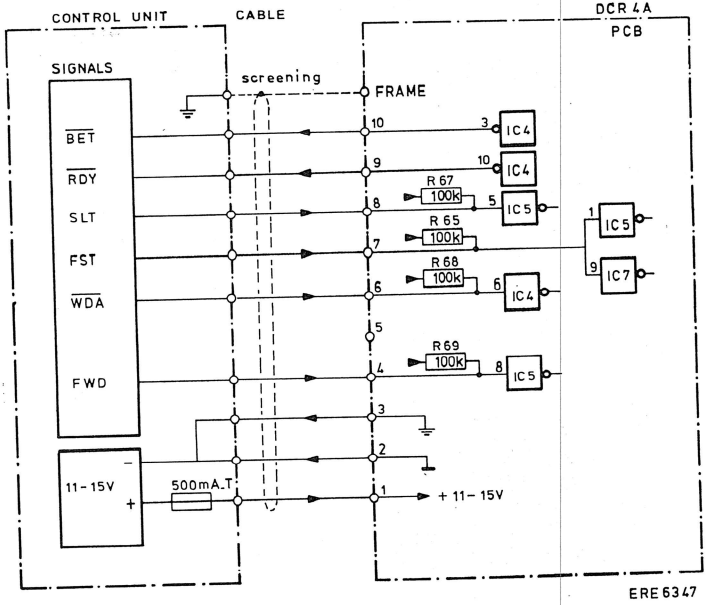


A = START DELAY > 100 m sec.
T1 = INITIAL GAP DELAY 170 < T1 < 1200 m sec
T2 = END OF DATA GAP

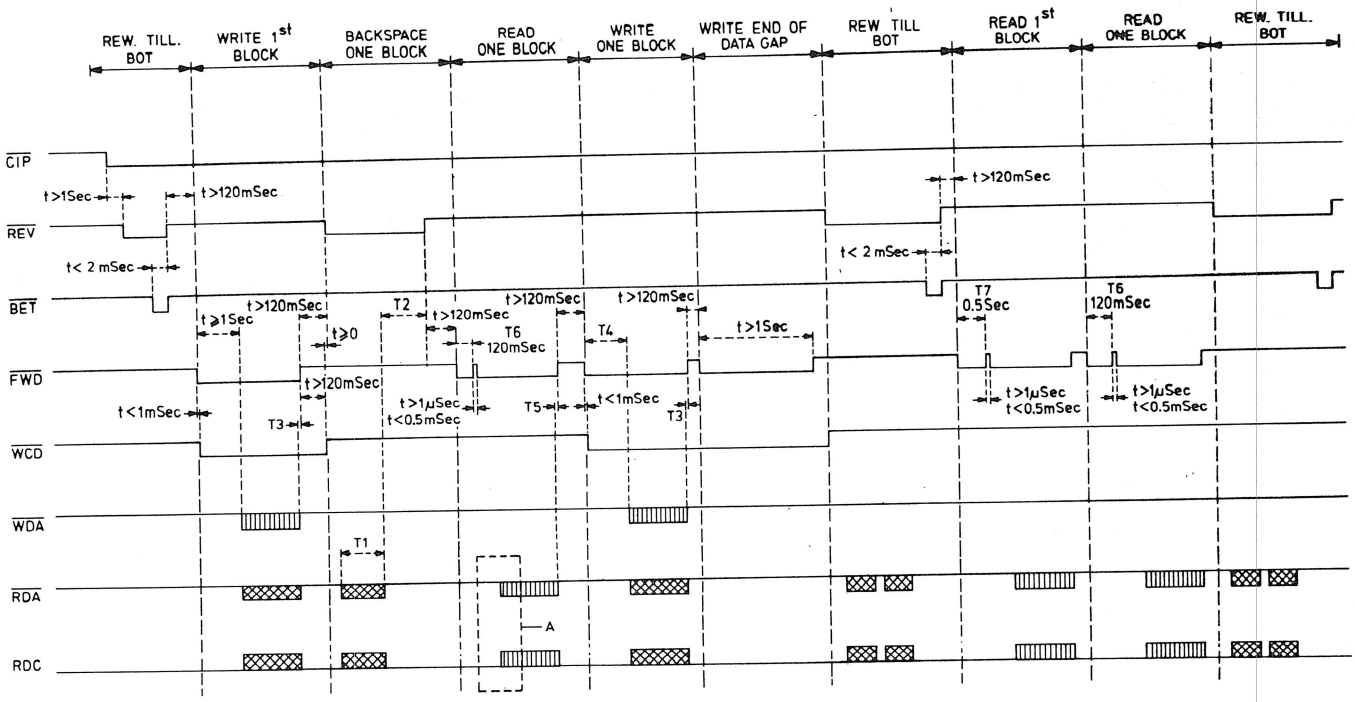
TIMING DIAGRAM DCR 4B

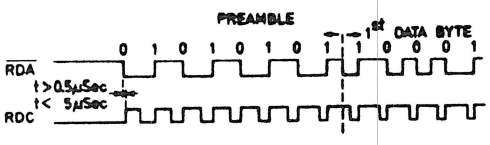


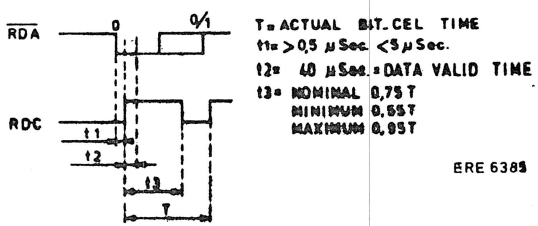
INTERFACING DCR 4B



INTERFACING DCR 4A







TIMING DIAGRAM MDCR

SCHEMATIC DIAGRAM MDCR