# INNOVATIVE DATA TECHNOLOGY

# TDF 4050 TAPE FORMATTER OPERATIONS MANUAL

For 1050/1750 Series

1/2 inch, 7 and 9 Track

Tape Transports

and

2640-1 Standalone Chassis



#### CONTENTS

- 1.0 GENERAL INFORMATION
  - 1.1 TDF 4050 Specifications
  - 1.2 Physical Description
- 2.0 TAPE FORMATS
  - 2.1 2.7 NRZI
    - 2.8 2.16 PE
- 3.0 MOTION DELAYS
- 4.0 FORMATTER CONFIGURATION (OPTIONS)
- 5.0 BASIC OPERATIONS (COMMANDS)
- 6.0 CPU INTERFACE DESCRIPTION
  - 6.1 Basic Specifications
  - 6.2 Controller-to-Formatter Interface lines
  - 6.3 Formatter-to-Customer Controller Interface
- 7.0 TAPE TRANSPORT INTERFACE DESCRIPTION
- 21.0 SYSTEM DESCRIPTION 21.1 Mnemonics
- 22.0 MICROPROCESSOR
  - 22.1 Architecture
  - 22.2 Circuitry, Schematic
  - 22.3 Instruction Set
- 23.0 PROGRAM
  - 23.1-16 Program Sequences
  - 23.20-38 Subroutines
- 24.0 FORMATTER CIRCUITRY
  - 24.1 Scope
  - 24.2 System Clock and

Input/Outputs, Schematic(2)

- 24.3 Write and Timer Circuits Schematic 3
- 24.4 Read Input Circuits,

Schematic 4

- 24.5 PE Deskewing and Read
  Output Circuits, Schematic (5)
- 25.0 SERVICE
  - 25.1 Corrective Maintenance
  - 26.1 Formatter Interfaces

#### 1.0 GENERAL INFORMATION

The IDT TDF 4050 Tape Formatter is an electronic unit designed to be mounted inside the TD -1050 ½" digital tape transport.\* The formatter enables the generation and reading of ANSI, IBM and ECMA compatible tapes. The formatter works with both PE and NRZI mode tape transports, and with both 7-track and 9-track versions:

9-track: 1600 cpi PE and

800 cpi NRZI,

7-track: 800/200, 800/556

or 556/200 cpi NRZI.

In all cases the formatter performs the basic functions of formatting data written on tape and decoding data read from tape. Error checking is performed on data read back from tape.

The TDF 4050 Tape Formatter can control from one to four tape transports within a single mag tape system. The system can have a mix of single and dual stack heads, 9 or 7 track and PE and NRZI.

The formatter can handle tape transports operating either at full or half system speed. The standard speed range is from 7.5 ips to 75 ips.

The interfacing and operation of TDF 4050 Tape Formatter is compatible with industry standard. However, the formatter, being controlled internally by a programmable processor, can be reprogrammed to suit a customer's particular requirement.

All timing during start and stop and during read and write operation are controlled by one master crystal oscillator. There are no monostables and no R-C timing networks.

\* Mounts in 1050/1750 Series and 2640-1 stand-alone chassis.

#### 1.1 TDF 4050 SPECIFICATIONS

Recording Mode:

Dual mode, PE (phase encoded) and NRZI (non-return to zero inverted).

Tape Velocity:

Standard: 75, 45, 37.5, 25, 18.75, 12.5 and 7.5 ips.

Low (half speed): 37.5, 22.5, 18.75 and 12.5 ips.

Other speeds available by strapping or by change of master oscillator crystal.

Dual speed is stan-

dard.

Head Configuration: 9-track or 7-track,

9-track or 7-track, single or dual stack

head.

Tape transports with different head configurations can be controlled by one formatter (daisy-

chain).

Data Density:

9-track

PE: 1600 cpi NRZI: 800 cpi

7-track NRZI

One of three combinations may be selected:

800/556 cpi 800/200 cpi 556/200 cpi

Interblock gaps:

(IBG)

9-track: 0.6" nom. 7-track: 0.75" nom.

Compatibel

Tape Transport:

IDT 1050/1750

Electronics:

All solid state with crystal clock controlled synchronous microprocessor system.

Power:

+5V +5%, 6.0 Amps

max.

Operating

temperature:

 $+ 2^{\circ}C$  to  $+ 50^{\circ}C$ (35°F to 122°F)

Non-operating

temperature:

 $-45^{\circ}C$  to  $+71^{\circ}C$  $(-50^{\circ}F \text{ to } 160^{\circ}F)$ 

Operating

altitude:

0 - 6096 m

(0 - 20000 feet)

Relative humidity: 15% to 95% noncon-

densing.

Mounting:

Internally in the

TDI 1050.

Dimensions:

493 mm x 251 mm

(19.40" x9.87")

Weight:

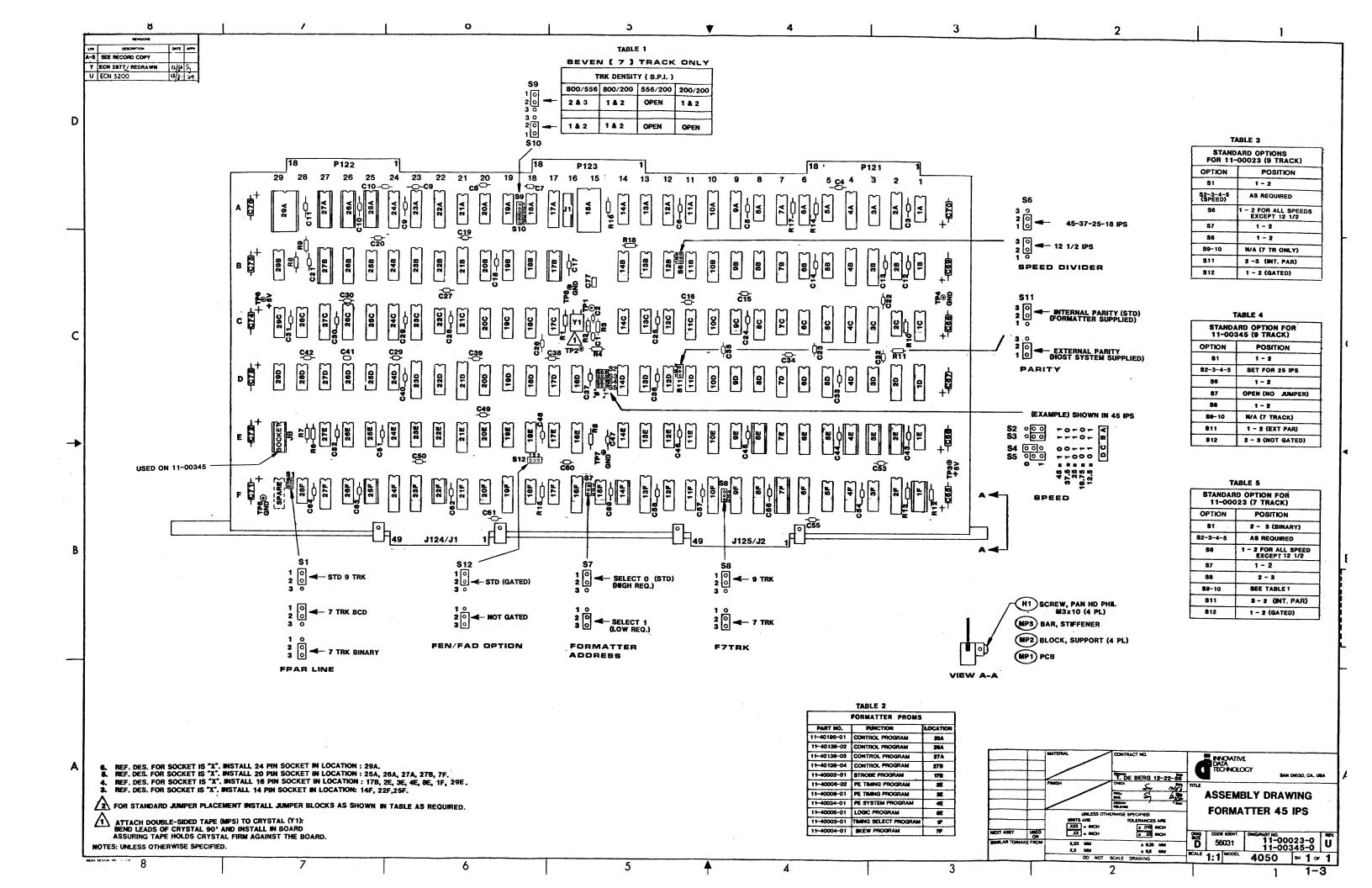
1.0 kg (2.2 pounds)

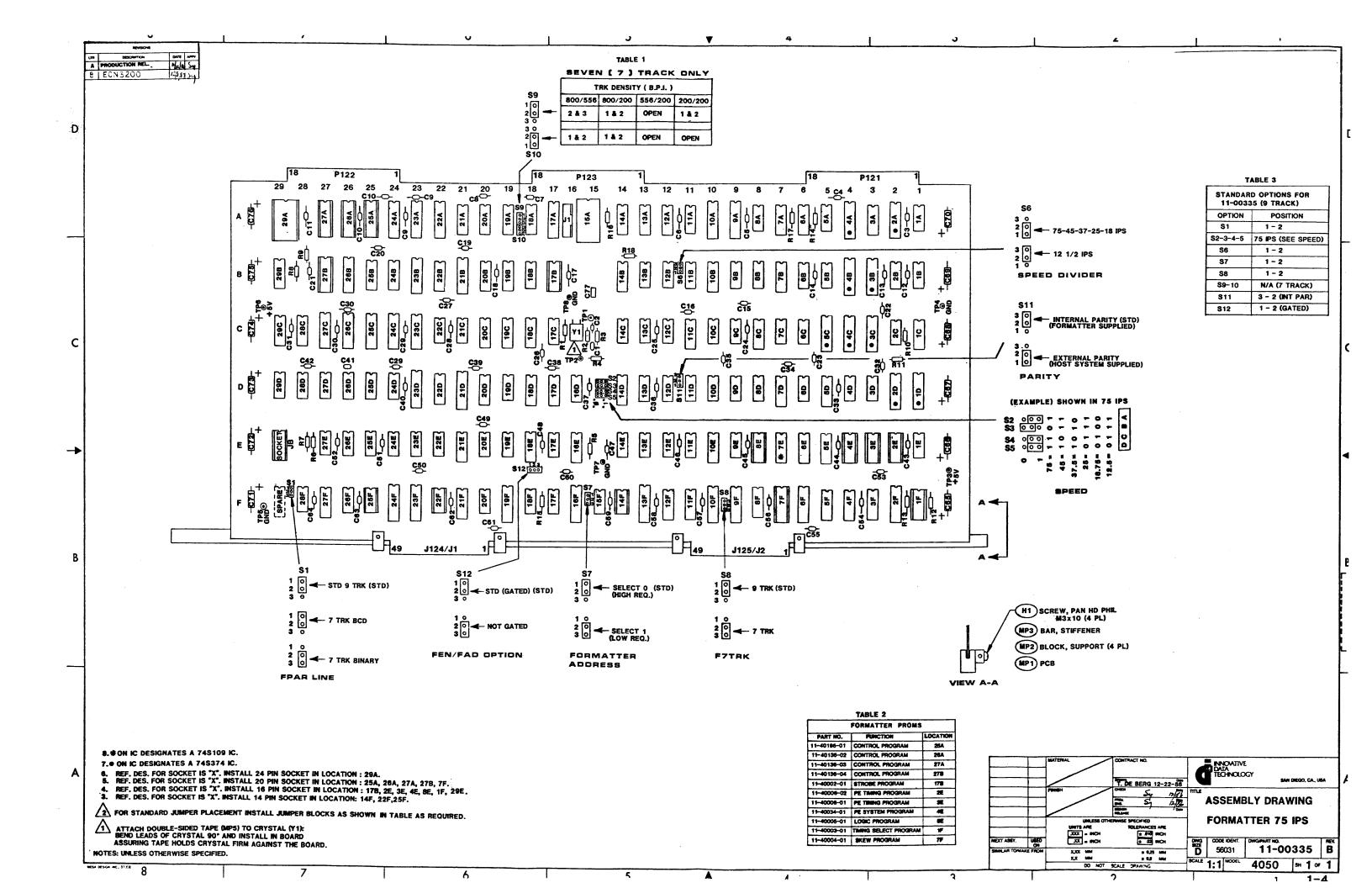
#### 1.2 PHYSICAL DESCRIPTION

The TDF 4050 Tape Formatter is contained on one PC board which is 493 mm x 251 mm  $(19.40" \times 9.87")$ , see Fig. 1.2.

The board is furnished with letters and numbers along the edges which are used as coordinates for locating a particular component. For example, the large IC U55 at the intersection of coordinates 15 and A is designated U55-15A on the circuit schematic.

The switch (strap) positions used to implement the available options are indicated by solid rectangles on Fig. 1.2. Details on setting the switches (placing the jumper plugs) are provided by Fig. 1.1.





#### 2.0 TAPE FORMATS

#### 2.1 CHANNEL NOTATION

The TDF 4050/TDI 1050 system is IBM, ANSI and ECMA compatible. However, the channel identifications used by IBM and ANSI/ECMA are different. This document uses the IBM notation. The relationship between IBM and ANSI/ECMA notation is given below.

The IBM channel 0 (ECMA/ANSI channel Z) contains the most significant bit in the data byte.

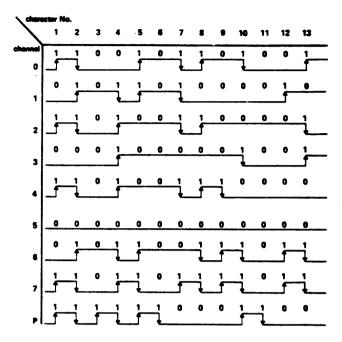


Fig. 2.1 NRZI, basic principle (9-track, odd parity)

#### 2.2 NRZI, BASIC PRINCIPLE

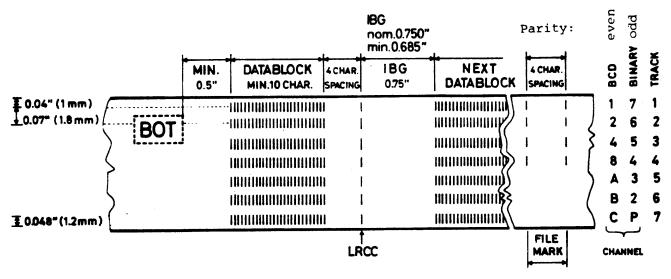
The basic principle of the NRZI (Non-Return to Zero Inverted) recording method is given in Figure 2.1. As this method gives a flux transition only when a "l" is written, a byte consisting of all zeros cannot be distinguished from no signal at all. Therefore, at least one bit in each byte (1 byte = 6 or 8 bits of data and 1 parity bit) must be "1", that is, either the parity bit is "l" for an all zeros character (odd parity) or an all zeros character must not occur. 9-track NRZI always employs odd parity securing at least one "1" bit in each byte. 7-track NRZI can have either odd or even parity. When a binary tape is written, the parity is odd, and when a BCD tape is written, the parity is even. In the latter case a character with all zeros cannot be used, as this will not be detected by the read electronics.

Figures 2.2 and 2.3 give the tape formats for 7-track NRZI and 9-track NRZI. In 7-track NRZI, channels 2 through 7 contain data bits (channel 2 contains the most significant bit).

Channels 0 and 1 are not used in 7-track NRZI. In 9-track NRZI channels 0 through 7 contain data bits (channel 0 contains the most significant bit).

In 9-track NRZI, the data density is always 800 cpi (characters per inch), (31.5 characters per millimeter).

In 7-track NRZI the data density can be either 200, 556 or 800 cpi (7.87, 21.89 or 31.5 characters per millimeter).



\*no all "0" character allowed

Fig. 2.2 NRZI 7-track tape format. 200/556/800 cpi.

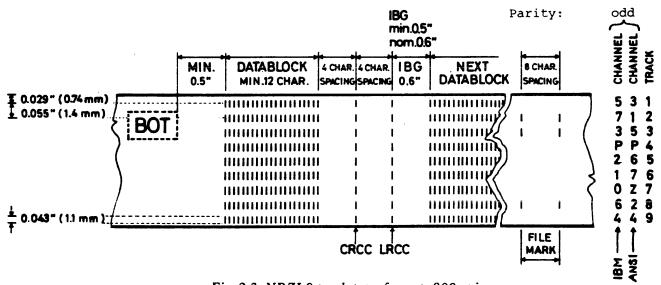


Fig. 2.3 NRZI 9-track tape format. 800 cpi.

#### 2.3 NRZI CHECK CHARACTER

In 7-track NRZI an LRCC (Longitudinal Redundancy Check Character) is written four character periods after the block containing the data bytes is terminated. Each bit of the LRC character is selected such that the total number of ones in a track, including the LRCC bit, is even. It is possible (in 7-track format only) for this character to be an all zeros character. A read data strobe is still generated, transferring the all zeros character.

In 9-track NRZI two check characters are generated. Four character periods after the data block, a CRCC (Cyclic Redundancy Check Character) is written.

The CRC character is used for error detection and correction. The CRC character may be an all zeros character. A read strobe is still generated. An LRC character is written four character periods after the CRC character (or eight character periods after last data character if the CRC character is all zeros). Each bit in the LRC character is selected such that the total number of "l" bits in a track (including the CRC and LRC character) is even. In the 9-track NRZI format, the LRC character will never be all zeros.

#### 2.4 NRZI, FILE MARK

The file mark or tape mark is a control block which consists of a single character with an LRC control character.

In 9-track NRZI the file mark contains
"l" bits in channels 3, 6 and 7 (IBM)
for both the data character and the LRC
character. The CRCC is zero.

In 7-track NRZI the file mark contains "1" bits in channels 4, 5, 6 and 7 for both the data character and the LRC\* character. The file mark will be separated by 3.56 inch (90.5 mm) from the previous record and by a normal interblock gap, respectively (0.75 inch (19.05 mm) in 7 track, and 0.6 inch (15.24 mm) in 9-track) from the following record.

In 7-track NRZI the parity of the file mark is always even. The TDF 4050 will indicate error when a file mark is detected while reading a binary tape with odd parity. (See note 2 on page 2-4.)

The ISO and DIN standards allow two different versions of the 7-track file mark, either the standard IBM/ANSI version or a version with "1" bits in channels 3, 4, 5, 6 and 7. The TDF 4050 will not recognize this last mentioned version as a file mark.

#### 2.5 NRZI, BLOCK SIZE

There is no hardware limitation in the TDF 4050 regarding the maximum size of a block written onto tape or read from tape. However, the minimum block size (except file marks) which can be handled by the TDF 4050 is 3 data characters. (see note 1, page 2-4).

#### 2.6 NRZI ERROR DETECTION

In the NRZI format, all read data deskewing is performed in the TDI 1050 read electronics. The formatter receives a 7- or 9-bit word from the tape transport and transfers this word to the customer interface. The Hard Error line is set true whenever one or more of the following errors are detected:

#### 9-track

- A data character is read from tape containing even parity. See note 2 on page 2-4.
- B) A CRC character is read from tape containing odd parity and there is an odd number of data characters in the block.
- C) A CRC character is read from tape containing even parity and there is an even number of data characters in the block.
- D) An entire track of data read from tape contains odd parity (including the CRC and LRC bit for this track).
- E) A track dropout occurs in such a way as to cause too long a gap between the data block and the CRC and LRC characters to be detected when the formatter interprets the dropout as an end-of-block condition.
- F) A track dropout occurs in such a way as to cause more than two check characters (CRC & LRC) to be detected when the formatter interprets the dropout as an end-of-block condition.
- G) A missing LRC character.

The formatter is able to detect errors A, B, C and D during both read forward and read reverse operation. Errors E, F and G will not be detected in read reverse mode.

#### 7-track

- A) A data character is read from tape containing even parity during an odd parity operation or odd parity during an even parity operation.

  Note that in 7-track NRZI the parity of the file mark pattern is always even. Hard error (parity error) will be indicated if the formatter operates in an odd parity read or read-afterwrite mode. See note 2 page 2-4.
- B) An entire track of data read from tape contains odd parity (including the LRC bit for this track).
- C) A track dropout occurs in such a way as to cause more than one check character (LRC) to be detected when the formatter interprets the dropout as an end-of-block condition. This error will not be detected in read reverse mode.

#### 9-track Operation

The formatter assumes that the IBG is detected when there has been an absence of signals for at least 26 character periods.

#### 7-track Operation

The formatter assumes that the IBG is detected when there has been an absence of signals for at least 14 (200 cpi) or 26 (556, 800 cpi) character periods.

For both 9 and 7-track operation, the detection test period becomes part of the total post-record delay generated by the formatter.

#### NOTE 1

A block size down to one data byte will be accepted and written by the formatter. However, if a complete block conforming to the file mark pattern is written, the read electronics will not accept this as a file mark during the read-after-write operation. (But during a normal read operation the block will be accepted as a file mark).

#### NOTE 2

Parity error is indicated by a pulse on the Hard Error line each time a byte with parity error is transferred. The pulse occurs at the same time and has the same width as the Read Data strobe.

True parity errors (not NRZI file marks) will also normally result in the Hard Error line being set true at the end of the block and staying true until the beginning of the next operation or a formatter clear.

#### 2.8 PE BASIC PRINCIPLE

The basic principle of the PE (Phase Encoded) recording method is given in Figure 2.4. As can be seen from this drawing, a "0" bit will always give a flux transition in one direction while a "1" bit gives a transition in the opposite direction. Where a bit is followed by a bit of the same value, the formatter will insert a phase transition between the two data bits. When reading a PE tape, the formatter decoding logic will ignore these phase transitions.

In the 4050 formatter, an oscillator controlled by a phase-locked loop to time the phase transitions has been avoided.

In such a system, the oscillator is locked on the data rate of one channel, and the oscillator output is then used to decode the other 8 channels. This gives a decoder logic of medium complexity. However, due to the use of a common oscillator, the channels are not fully independent. This in turn gives a lower tolerance for irregularities between the channels.

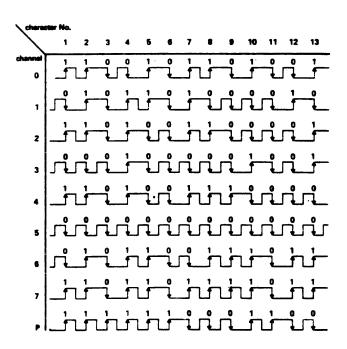


Fig. 2.4 PE, basic principle

The TDF 4050 employs a decoding system in which each channel is fully independent. Each channel has a digital counter and a set of look-up tables and buffer storage memory. This gives a very simple and flexible logic system.

In the read-after-write mode, look-up tables with tight tolerances are employed to ensure that the tape conforms to IBM/ANSI/ECMA-specifications. During read-only operations, other look-up tables with wide tolerances are employed, to enable reading even of tapes which do not conform strictly to the IBM/ANSI/ECMA-specifications.

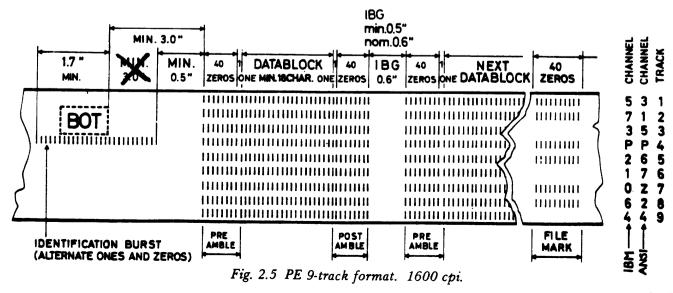
Details of the PE tape format are given in Figure 2.5. Channels 0 through 7 contain data bits, with the bit in channel 0 as the most significant bit. Channel P contains the parity bit, and odd parity is employed. No CRC or LRC characters are used. Each PE data block, however, is preceded by a preamble consisting of 40 bytes of all zeros followed by one byte of all ones. This is used (when reading forward) to establish synchronization. The all one byte identifies the end of the preamble and the start of the data bytes in the block.

Following each PE data block is a postamble which is the mirror image of the preamble, i.e. one byte of all ones and 40 bytes of all zeros. When reading in reverse, the postamble then precedes each block and is used to establish synchronization in the same manner as the preamble is used when reading forward. The PE format is identified by a burst of alternate ones and zeros in the parity track. This identification burst (ID-burst) is recorded at the beginning of the tape and embraces the BOT tab. Details of minimum distance are given in Figure 2.5.

#### 2.9 PE FILE MARK

The PE file mark as defined by IBM is ANSI/ECMA compatible, but is more limited. The ANSI/ECMA specifications defines a file mark as a special control block consisting of 64 to 256 flux reversals (at 3200 flux reversals per inch) in channels 2, 6 and 7. Channels 1, 3 and 4 are dc erased, but channels 0, P and 5 in any combination, may be dc erased or recorded in the manner stated for channels 2, 6 and 7. The TDF 4050 writes in IBM compatible file mark with 80 flux changes at 3200 flux reversals per inch in channels P, O, 2, 5, 6 and 7 with channels 1, 3 and 4 dc erased. This is also ANSI/ECMA compatible. When a read-after-write control is carried out, the TDF 4050 will only accept a file mark written in this manner. However, when a normal reading is carried out, the formatter will read any file mark which also is ANSI/ECMA compatible.

The PE file mark is preceded by a gap of approximately 3.56 inches (90.5 mm) followed by a normal interblock gap (IBG) of 0.6 inch (15.24 mm).



#### 2.10 PE READ DATA DESKEW

The TDF 4050 contains a deskew buffer for each track. Data may be skewed as much as 3.7 character periods. The HER line is set true if a larger amount of skew occurs. The operation is not terminated until the IBG is detected.

### 2.11 PE DATA SYNCRONIZATION AND PREAMBLE DETECTION

In read or read-after-write modes, the formatter will search for the preamble after a predetermined delay (ref. table 3.1). Channels 0 and 2 are monitored to establish the start of preamble. Depending upon bit skew, between 4 and 8 zeros must be detected before the formatter logic will accept the start of the preamble and enable the timing synchronization. 5 more zero bits are necessary in each channel before synchronization is established. Each channel is synchronized separately.

During preamble detection, all channels are monitored to detect dropout. If a dropout in all channels is detected, the formatter logic assumes that the few data pulses received were due to erroneous flux transitions occurring prior to the actual data record. The detection logic is reset, and the formatter starts searching for a new preamble.

After synchronization, all channels are monitored continuously. Any dropouts or dropins will be indicated by the error indication lines. Even if the HER is set true, the operation is not terminated until the formatter has detected the IBG.

#### 2.12 PE IBG DETECTION

The formatter assumes that the IBG is detected when there has been an absence of signals for at least 26 characters periods after a valid postamble detection. (The gap detection test period becomes part of the total post-record delay generated by the formatter.)

### 2.13 PE ERROR DETECTION AND CORRECTION

After synchronization has been established (between 4 and 8 zeros of the preamble), the formatter continuously measures the time difference between flux transitions in each channel separately. Dropouts or dropins are detected whenever this time difference falls outside predetermined boundaries in the formatter. If the dropout or dropin occurs on one track only, the formatter automatically corrects this track by using the odd parity information to recover the single track data.

During the remaining part of the record, no attempt is made to resynchronize the data on this track to the read logic.

If the dropout or dropin occurs on more than one track, the HER line is set true. The formatter will not terminate the operation until the end of block has been detected.

There are two error indication lines, <u>HER (Hard Error)</u> and <u>CER (Corrected</u> <u>Error.</u>

The <u>HER</u>-line is set true whenever one or more of the following errors are detected:

- A) Dropout or dropin on 2 or more channels.
- B) Skew buffer overflow.
- C) One or more channels fail to indicate a one ("1") bit at the end of the preamble.
- D) A false postamble detected in the middle of a block.
- E) One or more one-bits in the middle of the pre- or postamble.
- F) Even parity on read data.

The <u>CER</u>-line is set true whenever a single track dropout or dropin is detected. When operating in read-afterwrite mode, a CER-indication should always be followed by a rewrite of the block.

#### 2.14 PE BLOCK SIZE

There is no hardware limitation in the TDF 4050 regarding the maximum block length size written onto or read from tape. However, the minimum block size that can be handled by the TDF 4050 is 1 data character.

#### 2.15 ID BURST DETECTION

In PE read mode, before reading the first data block from BOT, the formatter will search for the ID burst pattern. When the ID burst is detected, or if the formatter cannot detect it within the specified time limit, the formatter will search for the gap between the ID burst and the first data block before enabling the normal read operation. This will give the read operation a clean start by being well removed from the potentially noisy area of the ID burst.

#### 2.16 GAP SIZE

To avoid tape run-out, the formatter contains a programmable gap time counter, which stops the operation if no read data has been detected after approx.
4.92 meters (16 feet) of tape movement. Optionally, this number may be changed from a minimum gap distance of 0.5 meter (1.6 feet) to a maximum gap of 143 meters (470 feet).

There is no hang-up of the formatter in this operation, which means that a new command may be transmitted immediately after the formatter busy line has been reset. This also means that there is no hang-up of the formatter if the BOT is reached in reverse mode and no read data bytes have been detected, but the FFBY line will stay true until the specified total gap time has elapsed.

### 3.0 MOTION DELAYS

To ensure a correct IBG, the formatter must wait a specified time after giving a tape motion command before any write or read sequence can be started. In the TDF 4050 all delays are controlled by a single crystal oscillator. Table 3.1 gives the standard time delays at 25 ips. To find delay times at lower or higher speeds, the delay times in the table should be multiplied by the ratio of 25 ips and the other speed, e.g. to find the pre-record delay time with a dual stack head in normal write operation at 45 ips with a 9-track tape drive, calculate as follows:

Delay time at 25 ips is from table 3.1 found to be 17 ms.

Delay time at 45 ips is then

17 ms x 
$$\frac{25}{45}$$
 = 9.44 ms

| Function                         |                                      |       | reco                   |          | Post-re                       |         |
|----------------------------------|--------------------------------------|-------|------------------------|----------|-------------------------------|---------|
|                                  |                                      | 9-t:  | ack                    | 7-track  | 9-track                       | 7-track |
| Write from BOT                   |                                      |       |                        | 256      | 3                             | 3       |
| Write Normal Single              | e stack head                         | 23    |                        | 28       | 3                             | 3       |
| Write Normal Dual                | stack head                           | 17    |                        | 22       | 3                             | 3       |
| Write File Mark                  |                                      | 150   |                        | 150      | 3                             | 3       |
| Read from BOT*                   | Read from BOT*                       |       |                        | 60       | 0                             | 0       |
| Read forward (Norma              | Read forward (Normal)                |       |                        | 12       | 0                             | 0       |
| Read reverse (Norma              | 11)                                  | 12    |                        | 12       | 3                             | 8       |
| Read reverse (Edit)              | Read reverse (Edit)                  |       |                        | 12       | 8                             | 13      |
| Start/stop time                  | Start/stop time                      |       |                        | 15       | 15                            | 15      |
| Time Delays, R<br>25 ips, PE mod | ead and Wri                          | te Fi | st I                   | ecord fr | om BOT at                     |         |
| Function                         | Delay<br>Before writing<br>ID (ms)   |       | ID<br>Duration<br>(ms) |          | Remainder<br>of Delay<br>(ms) | •       |
| Write First Record<br>from BOT   | 16                                   |       | 99                     |          | 140                           |         |
| Read First Record<br>from BOT    | st Record Delay Before ID sample (ms |       | ID sample period (ms)  |          | d                             |         |
|                                  | 54                                   |       |                        | 6        |                               |         |

<sup>\*</sup> NRZI only. For PE, see paragraph 2.15

Table 3.1 Pre-record and post-record delays at 25 ips

## 4.0 FORMATTER CONFIGURATION

The exact configuration and all options should be specified at the time of order. If necessary, a formatter can be reconfigured in the field.

#### 4.1 TAPE SPEEDS

The TDF 4050 can handle more than 7 different tape drive speeds. All drives must operate at the same standard speed or at half the speed (Low speed).

#### Standard

| speed |     | Low sp   | eed    |
|-------|-----|----------|--------|
| 75    | ips | 37.5     | ips    |
| 45    | ips | 22.5     | ips    |
| 37.5  | ips | 18.75    | ips    |
| 25    | ips | 12.5     | ips    |
| 18.75 | ips |          |        |
| 12.5  | ips | not appl | icable |
| 7.5   | ips | not appl | icable |

These speeds are selectable by strapping. The low speeds are remotely selectable (FSPEED).

Other speeds can be obtained by changing the system clock oscillator crystal.

The TDF 4050 formatters can read and write PE and NRZI at both high and low speeds.

#### 4.2 HEAD CONFIGURATION

The formatter can control tape transports with either single stack or dual stack head. This is controlled by the level of the SGL-signal from the selected tape transport.

#### 4.3 FORMATTER ADDRESS CONFIGU-RATION

The standard formatter is selected by a low <u>level</u> on the formatter address line, FAD. The formatter can be selected by a high on this line, by changing a strap.

#### 4.4 EXTERNAL PARITY GENERATION

The standard formatter generates the correct parity internally when writing data.

The formatter can be made to accept parity bits from the customer's controller, by changing a strap.

#### 4.5 ADAPTOR BOARD

The TDF 4050 is connected to the customer controller interface with two 50-lead flat cables, making it compatible with the new industry standard. Optionally, an adaptor board can be supplied, making the formatter connection compatible with the old industry standard.

#### 4.6 7-TRACK OPERATION

This is a standard feature which enables the formatter to operate with both 9 and 7-track tape transports.

There are 3 possible density combinations in 7-track, one of which must be specified when ordering. The combinations are selectable by the means of two straps.

9-track: 1600 cpi PE and 800 cpi NRZI

7-track: 800/200 cpi, 800/556 cpi or 556/200 cpi NRZI

#### 4.7 EVEN PARITY

This is a strap which enables a formatter configured to work with 7track tape transports to generate either even or odd parity on the tape. This is controlled by the FPAR line from the customer's interface.

### 4.8 READ-AFTER-WRITE TIMING CONTROL (OPTIONAL)

The standard TDF 4050 Tape Formatter generates and accepts data within ANSI, IBM and ECMA specifications. The readafter-write timing control option

enables the customer to specify that the formatter in the read-after-write mode shall work with tighter tolerances on the spacing between the bits on each track. This extra checking is performed only in read-after-write mode. When ordering this option, the customer must specify the timing tolerances. (PE only).

#### 4.9 GAP TIMING (OPTIONAL)

The gap timer, which terminates the operation if too long a gap is detected, is normally programmed to accept gaps up to 4.92 meters (16 feet). Optionally, this timer may be programmed to accept gaps of a value lying between 0.5 meters (1.6 feet) and 143 meters (470 feet).

#### 5.0 BASIC OPERATIONS

The TDF 4050 formatter is capable of executing the commands listed in table 5.1. However, if necessary, the formatter can be programmed to execute other commands specified by the customer. A short description of each standard command is given below.

| Command                 | FREV | FWRT | FUFN | FEDIT | FERASE |
|-------------------------|------|------|------|-------|--------|
| Read Forward            | н    | н    | н    | н     | н      |
| Read Reverse (Normal)   | L    | H    | H    | н     | н      |
| Read Reverse (Edit)     | L    | H    | H    | L     | н      |
| Write (Normal)          | H    | L    | H    | H     | н      |
| Write (Edit)            | н    | L    | н    | L     | н      |
| Write File mark         | н    | L    | L    | н     | H      |
| Erase (Variable Length) | н    | L    | H    | н     | L      |
| Erase (Fixed Length)    | н    | L    | L    | н     | L      |
| Space Forward           | Н    | H    | H    | н     | L      |
| Space Reverse           | L.   | H    | н    | н     | L      |
| File Search Forward     | H    | H    | L    | н     | н      |
| File Search Reverse     | L    | H    | L    | н     | H      |
| File Search Forward     | н    | H    | L    | H     | L      |
| (ignore data)           |      |      |      |       |        |
| File Search Reverse     | L    | H    | L    | H     | L      |
| (ignore data)           |      |      |      |       |        |
| H = High = False        |      |      |      |       |        |
| L = Low = True          |      |      |      |       |        |

Table 5.1

#### 5.1 READ FORWARD

This command causes the tape on the selected transport to be accelerated to the normal transport operating speed. The formatter will read the first record of data encountered and then decelerate the tape to a stop. All the necessary timing delays are generated by the formatter to secure proper head positioning in the interrecord gap. If necessary, a new read forward command can be transferred to the formatter during the tape deceleration time, thereby improving the access time to the next record. This is called "on-the fly" operation.

#### 5.2 READ REVERSE

This command is similar to the read forward command, but the tape motion is now in the reverse direction. "On-the-fly"-operation is possible. Whenever the tape transport detects the BOT-signal true during any reverse opera-

tion, the formatter logic is reset to the quiescent state. A new command can then be transferred to the formatter. The formatter busy signal will not be reset before the specified gap time has elapsed.

### 5.3 WRITE FORWARD (NORMAL) OPERATION

This command causes the formatter to accelerate the tape and after a predetermined time delay, to start transferring data from the customer controller to the transport. The formatter has no maximum restrictions on the block length. The data transfer continues until the last word command is given from the customer controller.

If the selected transport has a single stack head, the formatter will start decelerating the tape after the proper delay. If the selected tape transport employs a dual stack head, the formatter will continue to move the tape forward until the whole record has been read by the read head. If no data has been detected within the specified gap time, the formatter terminates the operation, and starts waiting for a new command. Then after the proper time delay, the formatter starts to decelerate the tape.

"On-the-fly"-write operations, are allowed regardless of whether the selected tape transport has a single or a dual stack head configuration.

#### 5.4 READ REVERSE EDIT OPERATION

This command is similar to the read reverse command except that the formatter generates a different time delay to ensure proper head positioning in the selected tape transport.

### 5.5 WRITE FORWARD EDIT OPERATION

This operation is similar to the write forward operation except that the TWRS signal from the formatter to the tape transport is set true after the block has been written. This ensures that the write current in the selected tape transport is switched off gradually at the end of the write sequence, preventing erasure into the next data block. The write forward edit operation should be preceded by a read reverse edit operation for proper head positioning.

#### 5.6 WRITE FILE MARK OPERATION

This command causes the formatter to start the tape on the selected tape transport and after a predetermined time delay to generate the proper file mark pattern to be written on the tape. Formatter then decelerates the tape.

### 5.7 FRASE FORWARD OPERATION (FIXED LENGTH)

This command causes a 3.7" length of tape to be erased on the selected transport.

### 5.8 ERASE FORWARD OPERATION (VARIABLE LENGTH)

This command causes the formatter to accelerate the tape and erase the tape continuously until the "last-word" FLWD signal from the customer controller is set true. This terminates the erase operation. In the PE mode, the ID burst will be written even if an erase command from BOT is given.

#### 5.9 SPACE FORWARD

This operation is similar to a read forward operation, except that no read strobe signals are generated and no read data is supplied to the controller. File mark testing is performed on the read data, but not error checking.

#### 5.10 SPACE REVERSE

This operation is similar to the space forward operation, except that the formatter moves the tape in the reverse direction.

#### 5.11 FILE SEARCH FORWARD

This command causes the transport to perform a series of read forward commands

in the "on-the-fly"-mode. The operation is terminated either by the recognition of an ANSI/ECMA/IBM compatible file mark or the EOT tab. Tape is stopped following reading of a file mark similar to a normal read operation. If the EOT is encountered during a file mark search operation, the operation will terminate and the tape will be stopped at the end of the record currently being processed Data transfer and error detection are performed in the sameway as for the read forward operation.

#### 5.12 FILE SEARCH REVERSE

This operation is similar to a file search forward operation, but in the reverse direction. The formatter will not terminate the operation if the EOT is passed during this operation.

### 5.13 FILE SEARCH FORWARD (IGNORE DATA)

This operation is a combined file search forward and space forward operation. No data transfer takes place, and there is no error checking on the read data.

### 5.14 FILE SEARCH REVERSE (IGNORE DATA)

This operation is similar to a file search forward (ignore data) operation, but in the reverse direction.

#### 5.15 LOAD-ON-LINE

This operation enables a remote Load sequence.

#### 5.16 REWIND

This command causes the selected transport to rewind to BOT. In daisy-chained systems, one or more transports can be rewinding while the customer controller transfers data to or from another transport.

#### 5.17 OFF-LINE

This command places the selected transport in the off-line mode.

## 6.0 CPU INTERFACE DESCRIPTION

#### 6.1 BASIC SPECIFICATIONS

The formatter is plugged directly into the TDI 1050 formatter adaptor board; there are no cables between the formatter and the TDI 1050 electronics.

Two 50-lead flat cables are used to connect the TDF 4050 to the customer controller interface. Detailed specifications are given in Figure 6.1.

Optionally, the TDF 4050 can be supplied with an adaptor board, making it compatible with the old industry standard interface system (one 100-pin connector).

Signal names have been chosen to correspond to the logical true condition. This corresponds to a low level on the interface line. A disconnected or broken interface line is thus interpreted as a logical false signal by the formatter logic.

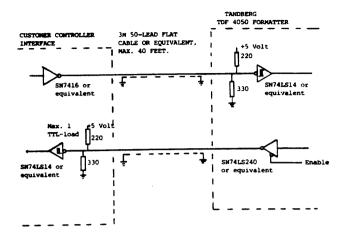


Fig. 6.1 Interface connections and specifications

#### NOTES

- Not more than 1 TTL load can be driven by each formatter interface driver.
- 2. If the Schmitt trigger SN74LS14 in the customer controller interface is replaced by a non-Schmitt-trigger input, the noise margin will be reduced. This degrading in signal-tonoise ratio becomes more significant as cable lengths increase to the maximum allowable length.
- 3. The flat cable is 3M 50-lead 28 AWG (3M part no. 3365-50) or equivalent. Two cables are used between the controller interface and the TDF 4050.
- 4. Maximum cable length is 40 feet (12 meters). If two formatters are connected in daisy-chain mode, the total maximum cable length is 40 feet, and the distance between the two formatters should be not more than 5 feet (1.5 meter).

#### TDF 4050 Controller Interface

| TDF 4050    |                 |                         |           |      |
|-------------|-----------------|-------------------------|-----------|------|
| Live<br>pin | Ground<br>pin   | Signal<br>Controller to | Formatter | Dwg. |
| J125-48     | J125-47         | Formatter Address       | (FFAD)    | 2    |
| J124-46     | J124-45 *       | Transport Address 0     | (FTADG)   | 2    |
| J125- 46    | J125-45         | Transport Address 1     | (FTAD.)   | 2    |
| J124-8      | J124-7          | Initiate Command        | (FG0)     | 2    |
| J124-18     | J124-17         | Reverse/Forward         | (FREV)    | 2    |
| J124-34     | J124-33         | Write/Read              | (FWRT)    | 2    |
| J124-42     | J124-41         | Write File Mark         | (FWFM)    | 2    |
| J124-38     | J124-37         | Edit                    | (FEDIT)   | 2    |
| J124-40     | J124-39         | Erase                   | (FERASE)  | 2    |
| J124-44     | J124-43         | Read Threshold 1        | (FRTH1)   | 2    |
| J124-36     | J124-35         | Read Threshold 2        | (FRTH2)   | 2    |
| J125-50     | J125-49         | Density select          | (FDEN)    | 2    |
| J124-45*    |                 | Parity select           | (FPAR)    | 2    |
| J124-20     | J124-19         | Rewind                  | (FREW)    | 2    |
| J]25-24     | J125-23         | Off-Line                | (FOFL)    | 2    |
| J124-4      | J12 <b>4-</b> 3 | Last Word               | (FLWD)    | 2    |
| J125-18     | J125-17         | Formatter Enable        | (FFEN)    | 2    |
| J124-22     | J124-21         | Write Data Parity       | (FWDP)    | 3    |
| J124-10     | J124-9          | Write Data 0            | (FWDO)    | 3    |
| J124-12     | J124-11         | Write Data l            | (FWD1)    | 3    |
| J124-30     | J124-2 <b>4</b> | Write Data 2            | (FWD2)    | 3    |
| J124-26     | J124-25         | Write Data 3            | (FWD3)    | 3    |
| J124-6      | J124-5          | Write Data 4            | (FWD4)    | 3    |
| J124-32     | J124-31         | Write Data 5            | (FWD5)    | 3    |
| J124-28     | J124-27         | Write Data 6            | (FWD6)    | 3    |
| J124-24     | J124-23         | Write Data 7            | (FWD7)    | 3    |
| J124-16     | J124-15         | Load On Line            | (FLOL)    | 2    |
|             |                 |                         |           |      |

<sup>\*</sup>These lines grounded except when working with a 7-track formatter.

| TDF 4050    |                  |                                   |           | ***        |
|-------------|------------------|-----------------------------------|-----------|------------|
| Live<br>pin | Ground<br>pin    | Signal<br>Formatter to controller |           | $\Diamond$ |
| J124-2      | J124-1           | Formatter Busy                    | (FFBY)    | 2          |
| J125-38     | J125-37          | Data Busy                         | (FDBY)    | 2          |
| J125-16     | J125-15          | CCG/IDENT                         | (FCCG/ID) | 2          |
| J125-12     | J125 <b>-</b> 11 | Hard Error                        | (FHER)    | 5          |
| J125-14     | J125 <b>-</b> 13 | File Mark                         | (FFMK)    | 2          |
| J125-28     | J125-27          | Ready                             | (FRDY)    | 2          |
| J125-44     | J125-43          | On-line                           | (FONL)    | 2          |
| J125-30     | J125-29          | Rewinding                         | (FRWD)    | 2          |
| J125-32     | J125-31          | File protect                      | (FFPT)    | 2          |
| J125-4      |                  | Load point                        | (FLDP)    | 2          |
| J125-22     | J125-21          | End of Tape                       | (FEOT)    | 2          |
| J125-26     | J125-25*         | NRZI                              | (FNRZ)    | 2          |
| J125-25*    |                  | 7-track                           | (F7TR)    | 2          |
| J124-14     | J124-13          | Single                            | (FSGL)    | 2          |
| J125-40     | J125-39          | Speed                             | (FSPEED)  | 2          |
| J125-36     | J125-35          | Demand Write Data strobe          | (FDWDS)   | 3          |
| J125-34     | J125-33          | Read strobe                       | (FRSTR)   | 5          |
| J125-1      |                  | Read Data Parity                  | (FRDP)    | 5          |
| J125-2      |                  | Read Data 0                       | (FRD0)    | 5          |
| J125-3      |                  | Read Data 1                       | (FRD1)    | 5          |
| J124-48     | J124-47          | Read Data 2                       | (FRD2)    | 5          |
| J124-50     | J124-49          | Read Data 3                       | (FRD3)    | 5          |
| J125-6      | J125-5           | Read Data 4                       | (FRD4)    | 5          |
| J125-20     | J125-19          | Read Data 5                       | (FRD5)    | 5          |
| J125-10     | J125-9           | Read Data 6                       | (FRD6)    | 5          |
| J125-8      | J125-7           | Read Data 7                       | (FRD7)    | 5          |
| J125-42     | J125-41          | Corrected Error                   | (FCER)    | 5          |

<sup>\*</sup>Normally grounded except for 7-track formatters.

### 6.2 CONTROLLER TO FORMATTER INTERFACE LINES

A number of signals to the formatter are routed unchanged to the selected tape transport. For a complete description of signal requirements and of the effects on the tape transport, see the TDI 1050 engineering specifications.

#### 6.2.1 FFAD - FORMATTER ADDRESS

This is a level which selects one of two possible formatters attached to the customer controller interface.

High = Address 0
Low = Address 1

The formatter's address is predetermined by a strap on the PC-board.

This line must stay stable during all operations. When selected, a formatter is connected to the customer controller, and all the controller/formatter interface lines are activated.

Unless otherwise specified the description given here of the other controller/formatter lines assumes that the formatter is selected.

### 6.2.2 FTADO, FTAD1 TRANSPORT ADDRESS

These two lines are levels which select one of up to four transports daisy-chained to the formatter. Both lines must remain stable during all operations. These lines are decoded by the formatter adaptor board as follows:

| TADO | TADI | ADDR |
|------|------|------|
| High | High | SLTO |
| High | Low  | SLTl |
| Low  | High | SLT2 |
| Low  | Low  | SLT3 |

#### 6.2.3 FGO - INITIATE COMMAND

This is a pulse which initiates the command specified by the command lines, Section 6.2.4. Table 5.1 gives a list of the different standard commands. On the trailing edge of FGO these lines are strobed into the formatter and the formatter busy signal, FFBY, is set low. No command should be transferred to the formatter whenever the data busy signal, FDBY, is low. The minimum pulse width for the FGO-signal is 0.2 µsec.

#### 6.2.4 COMMAND LINES

These lines specify a command to the formatter and should be held steady for 0.25 µsec before the leading edge, to 0.25 µsec after the trailing edge of FGO. The coding of these lines for different tape operations is given in table 5.1. Command definitions are given in section 5.

#### 6.2.4.1 FREV - REVERSE MODE

This line specifies forward or reverse tape motion.

Low = reverse
High = forward

#### 6.2.4.2 FWRT - WRITE MODE

This line specifies write or read operations.

Low = write High = read

#### 6.2.4.3 FWFM - WRITE FILE MARK

- In a write operation when FWRT is low, this line specifies a file mark to be written.
- In an erase operation when FERASE is low, this line specifies a fixed length erasure.
- In a read operation, when FWRT is high, this line specifies a file search.

#### 6.2.4.4 FEDIT - EDIT MODE

This line is used in two ways.

- Modifies read reverse stop delay to optimize head positioning when editing tapes.
- The transport write current is turned off (gradually) at the end of the record. This prevents the next data record from being partly erased, without generating turn-off noise.

#### 6.2.4.5 FERASE - ERASE MODE

1. If FERASE and FWRT are low, the formatter is positioned to execute a dummy write command. The formatter will go through all of the operations of a normal write command except that no data is recorded. A length of tape will be erased equivalent to the length of the dummy-record (as defined by FLWD).

Alternatively if the FERASE, FWRT and FWFM command lines are all low, the formatter is conditioned to execute a dummy write file mark command. A fixed length of tape of aproximately 3.7" will be erased.

During a space or file mark search command the erase line is used to inhibit the read strobes.

### 6.2.4.6 FTHRI - READ THRESHOLD LEVEL 1

This line is used for transports having a single stack head only and specifies the operating read circuit threshold level as follows:

Low = High threshold High = Normal threshold

FTHR1 will normally be made low only when it is required to perform a read control operation immediately after a write data operation.

### 6.2.4.7 FTHR2 - READ THRESHOLD LEVEL 2

This line specifies an extra low threshold level for the read electronics in the tape drive.

Low = Extra low threshold High = Normal threshold

FTHR2 will normally be made low only when it is required to recover very low amplitude data.

#### 6.2.4.8 FDEN - DENSITY SELECT

This line makes it possible to select between PE and NRZI format when operating with a 9-track tape drive or between 2 densities when operating with a 7-track tape drive. The operation is as follows:

#### 9-track

Low = PE 1600 cpi High = NRZI 800 cpi

#### 7-track

Low = 800 cpi or 556 cpi High = 556 cpi or 200 cpi

The formatter can be made to work with one of three possible density combinations. This is achieved with two straps on the formatter board. The three combinations are:

800/556 cpi, 800/200 cpi or 556/200 cpi.

The standard formatter will only accept a density change at the start of a new operation.

#### 6.2.5 FREW - REWIND COMMAND

This is a pulse (minimum 0.2 µsec) which causes the selected transport (provided it is ready and on-line) to rewind to BOT). This pulse is transmitted directly to the transport and will not cause the formatter to be set in the busy state.

#### 6.2.6 FOFL - OFF-LINE COMMAND

This is a pulse (0.2  $\mu$ sec minimum) which causes the selected transport to be turned off-line.

This pulse is transmitted directly to the transport and does not cause the formatter to become busy.

#### 6.2.7 FLWD - LAST WORD

- 1. During write operations, this line indicates that the next character to be strobed into the formatter is the last data character of the record. It should be set low by the controller interface at the same time that the last data character is placed on the interface lines.
- During a variable-length erase operation this signal indicates that the operation should be terminated.

#### 6.2.8 FFEN - FORMATTER ENABLE

This is a level which, when high, causes the formatter to reset to the quiescent state. This signal is not gated by FFAD, hence, if two formatters are connected to the interface both will be reset simultaneously.

This line may be used to disable the formatter if the controller power is lost or to clear the formatter logic in the case of illegal commands or unusual conditions.

### 6.2.9 FWD0-7 - WRITE DATA LINES

These eight lines transmit write data from the controller to the formatter. For 7-track operation, FWD0 and FWD1 are not used. The data bits appearing on the write data lines are written into the corresponding channels on tape. Table 6.1 gives the connection between the input write data lines and the track no. both for 9-track and 7-track operation. The first character of the data record should be available on these lines within one character period after

the FDBY -signal goes low and should remain stable until the trailing edge of the first FDWDS pulse issued by the formatter. The next data character must then be set up in less than half a character period after the trailing edge of the FDWDS-signal. See Fig. 6.2.

Subsequent characters will be processed in this way until FLWD is set true, indicating that the last character is being transferred.

#### 6.2.10 FWDP - WRITE PARITY LINE

This line is only used when the external parity generation option is used. This option requires that the customer generate the parity bit and apply this parity bit to FWDP. When operating with 9-track tape drives or in 7-track binary mode, the customer must generate odd parity on the basis of the 8, respectively 6, data lines and apply this parity bit to the FWDP line. When operating in 7-track BCD mode, the customer must generate even parity on the 6 data lines and apply this parity bit to the FWDP line.

Set-up and hold time requirements for the FWDP line are consistent with the requirements for the FWDO-7 lines.

The FWDP line corresponds to IBM channel P or ANSI track no. 4.

### 6.2.11 FLOL - LOAD AND ON LINE COMMAND

This command can be performed only if the selected transport has the load-on-line option. It is necessary to supply two FLOL pulses before the transport is placed on-line. The minimum pulse width for each pulse is 2.0 µsec. The two pulses must be separated by at least 1.0 sec. The first FLOL signal is a low pulse which can be given at any time after AC power is applied to the transport. The transport will then be caused to apply tape tension. When the second FLOL signal is applied the transport goes on-line. (See the TDI 1050 engineering specifications).

| Interface<br>lines |                | 9-track                  |            | 7-track               |        |     |
|--------------------|----------------|--------------------------|------------|-----------------------|--------|-----|
|                    | IBM<br>channel | ANSI/ECMA<br>Environment | ANSI Track | ANSI ASCII<br>Channel | Binary | BCD |
| FWD0               | 0              | E8                       | 7          | Z                     | _      | _   |
| FWD1               | 1              | E7                       | . 6        | 7                     | -      | -   |
| FWD2               | 2              | <b>E</b> 6               | 5          | 6                     | 2      | В   |
| FWD3               | 3              | <b>E</b> 5               | 3          | 5                     | 3      | A   |
| FWD4               | 4              | E4                       | 9          | 4                     | 4      | 8   |
| FWD5               | 5              | E3                       | 1          | 3                     | 5      | 4   |
| FWD6               | 6              | E2                       | 8          | 2                     | 6      | 2   |
| FWD7               | 7              | El                       | 2          | 1                     | 7      | 1   |
|                    |                |                          |            |                       |        |     |

NOTE. The FWDO and FWD1 lines must remain high during 7-track write operations.

Table 6.1

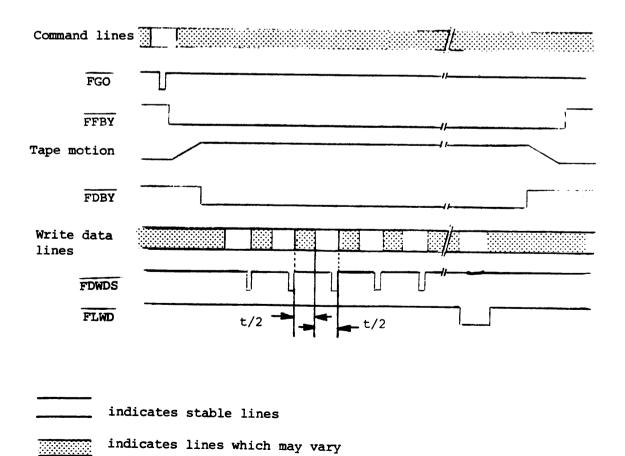


Fig. 6.2 Timing diagram, typical write operation. Not to scale.

#### 6.2.12 FPAR - EVEN PARITY SIGNAL

During 7-track operation the customer can select between binary operation with odd parity and BCD operation with even parity. The parity generation is controlled with the FPAR signal.

High = Odd parity (binary)
Low = Even parity (BCD)

This line is grounded for 9-track systems.

### 6.3 FORMATTER TO CUSTOMER CONTROLLER INTERFACE

#### 6.3.1 FFBY - FORMATTER BUSY

When a command is transferred to the formatter, this signal goes low at the trailing edge of FGO and remains low until tape motion ceases after the execution of the command. The controller will usually make use of this signal to inhibit further commands.

#### 6.3.2 FDBY - DATA BUSY

This signal goes low when the transport has reached operating speed, traversed the IBG, and the formatter is about to write data on the tape or read a signal from tape. This signal remains low until the data transfer is finished and the appropriate postrecord delay is completed. FDBY goes high at the same time that the capstan starts to decelerate the tape. A new command may be transferred to the formatter when the FDBY signal goes false. This is called "on-the-fly" operation. "On-the-fly" commands must be of the same READ/WRITE mode and the same tape direction. No attempt should be made to transfer commands to the formatter when FDBY is low (active).

# 6.3.3 FCCGID - CHECK CHARACTER GATE/IDENTIFICATION BURST DETECT

- This signal changes significance according to whether the NRZI or the PE mode is used. In the PE mode this line is pulsed if the identification burst on the tape is detected and when the tape operation is a read or read-after-write from BOT.
- 2. In the NRZI mode this line is set low by the formatter to indicate that the read character currently being transmitted to the controller is either a CRC or LRC character (9-track) or LRC character (7-track). During normal data transfer this line stays high.

#### 6.3.4 FHER - HARD ERROR

This line is set low if a read error has been detected by the formatter, as specified in sections 2.6 and 2.13. Except for parity errors this line will be set low as soon as an error occurs or at least at the end of the record, and will stay low until the next FGO signal is transmitted or until the FFEN-signal is set high. Parity error is indicated by a pulse on the HER line, one pulse each time a byte with parity error is transferred. The pulse occurs at the same time and has the same width as the FRDS strobe.

True parity errors (not NRZI file marks) will normally result in the HER line being set true at the end of the block and staying true until the next FGO is transmitted or until the FFEN signal is set high. However, if a PE data block has been written with even parity (employing the FWDP line) and the block except for the parity error is correctly written, the HER line will not be set at the end of the block, although parity error will be indicated by pulses on the line for each byte in the block. All error information will be reported to the controller before the FDBY signal goes false.

#### 6.3.5 FCER - CORRECTED ERROR

This line is active only when operating in PE mode. It is set low whenever a single track error occurs during a read or read-after-write operation. The signal will stay true until the next FGO signal is transmitted or until the FFEN signal is set high. If the FCER signal is set low during a read-after-write operation, the record should be rewritten.

#### 6.3.6 FFMK - FILE MARK

This line is used to indicate that the formatter read logic has detected a file mark. This may occur during any read forward or read reverse command, or during a write file mark command when operating in a read-after-write mode. The FFMK signal is pulsed after the complete file mark record has been read and the signal will stay true for at least 1.0 usec.

### 6.3.7 TRANSPORT STATUS AND CONFIGURATION LINES

There are seven status lines and four configuration lines transferred from the formatter to the customer controller interface. These lines indicate the status and configuration of the selected transport.

#### 6.3.7.1 FRDY - READY STATUS

This signal when true, indicates that the selected transport is ready to receive an external command.

#### 6.3.7.2 FONL - ON-LINE STATUS

This signal when true, indicates that the selected transport is on-line, that is, under remote control. When false, the transport is off-line and cannot be operated from the customer controller interface.

#### 6.3.7.3 FRWD - REWINDING STATUS

This signal when true, indicates that the selected transport is engaged in a rewind operation.

### 6.3.7.4 FFPT - FILE PROTECT STATUS

This signal when true, indicates that the selected transport is on-line and has a reel of tape without a write enable ring. No attempt should be made to write data onto tape when this signal is true. The formatter performs no test upon this signal. For servicing purposes it can be useful to perform a dummy write operation without a write enable ring on the tape reel, using a tape having a prewritten data pattern. By excluding the write part of the system in this way, errors in the read part may be exposed.

#### 6.3.7.5 FLDP - LOAD POINT STATUS

This signal is true when the selected tape transport detects the BOT tab on the tape and the initial load or rewind sequence is completed. When the tape is moving forward, the FLDP signal will remain true until the BOT tab can no longer be detected by the tape transport.

#### 6.3.7.6 EOT - END OF TAPE STATUS

This signal when true, indicates that the selected tape transport has detected the EOT tab on the tape. To preserve the original rise and fall times, this signal is not processed by the formatter.

#### 6.3.7.7 FNRZ - NRZI STATUS

This line when true, indicates that the selected tape transport and the formatter operate in NRZI mode. When false, this line indicates that the selected tape transport and the formatter operate in PE mode.

### 6.3.7.8 FSGL - SINGLE STACK

This line when true, indicates that the selected tape transport has a single stack head. This line conditions the formatter to generate the correct delays for the generation of the IBG and for head positioning.

#### 6.3.7.9 FSPEED - TAPE SPEED

This line when true, indicates that the selected tape transport operates in the low speed mode (half of normal operating speed). When false, this line indicates that the tape transport operates at normal speed.

#### 6.3.7.10 F7TR - 7-TRACK

This line when true, indicates that the tape transport has a 7-track head. This line will also condition the formatter to operate in 7-track mode. This line should be grounded (by strapping) for 9-track systems and must in such cases be ignored.

### 6.3.8 FDWDS - DEMAND WRITE DATA STROBE

This signal consists of a pulse for each data character to be written onto tape. The pulse width of the FDWDS pulses is 1.04 µsec at 75 ips and correspondingly wider at lower speeds. The first data character should be available on the write data input lines within one character period after the FDBY signal has been set true by the formatter and remain true until the trailing edge of the first FDWDS signal. Succeeding characters must then be placed on these lines within one half of a character period after the trailing edge of each FDWDS signal. During a write file mark command the required file mark pattern is generated internally by the formatter, and the FDWDS signal is not used.

During erase operation (variable length) this line will also be used. However, no data are strobed into the formatter or written onto tape. The customer controller may use this line to determine the length of tape which has been erased.

#### 6.3.9 FRSTR - READ DATA STROBE

This signal consists of a pulse for each character of read information (data, CRC, and LRC) to be transmitted to the customer controller interface and should be used to strobe the read

data lines FRDP, FRDO-7 into the controller read data register. The pulse width of this signal is 1.04 µsec at 75 ips and correspondingly wider at lower speeds. In NRZI mode, the transmission of check characters (9 track: CRC and LRC, 7 track: LRC) will be indicated by the check character gate line FCCG being set true. If the CRC character (9-track) or the LRC character (7-track) is an all zero character, the formatter still generates a FRSTR pulse.

The average time between pulses on the FRSTR line is given by

$$\frac{1}{S \times D}$$
 sec

Where S = tape speed (ips) and D = packing density (cpi)

The customer controller interface must be able to accept the whole block of data at the specified data rate.

Due to bit crowding, tape speed variation and signal dropout correction (PE), the customer controller interface must be able to receive characters at a rate which can vary between twice the nominal rate and half the nominal rate. (See note 3.)

### 6.3.10 FRDP, FRD0-7 READ DATA LINES

When operating with a 9-track tape transport, these 9 lines transmit read data from the formatter to the customer controller. When operating with a 7-track tape transport, FRDO and FRD1 are not used.

Each character read from tape is available by sampling these lines simultaneously with the FRSTR. At 75 ips data will be placed on the read data lines at least 1.4  $\mu$ sec. before the leading edge of the FRSTR pulse, and the set-up times increase correspondingly with lower tape speed. The data remains on the read data lines for approximately 0.9 character period. (See note 3.)

#### NOTE 3

In read reverse 7-track operation, if the LRC character is an all zeros character, the first three characters (including the "0" LRC character) is transmitted with an interval between each character of approx. 2  $\mu s$  at 75 ips (correspondingly longer intervals at lower speeds).

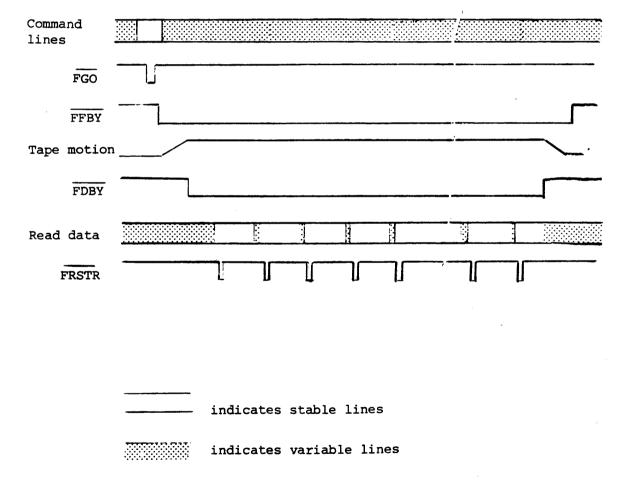


Fig. 6.3 Timing diagram, typical read operation. Not to scale.

### 7.0 TAPE TRANSPORT INTER-FACE DESCRIPTION

#### 7.1 INTERFACE

The TDF 4050 is plugged into the TDI 1050 Tape Transport via an I/O Interface Board that comes with the 1053 master configuration, see Fig. 7-1.

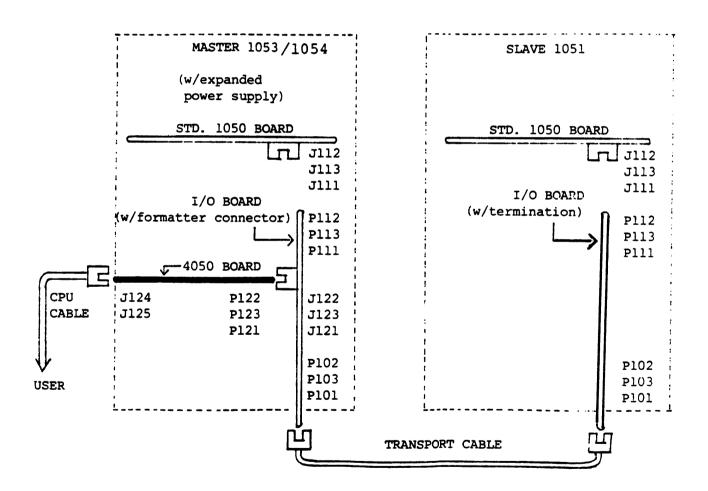


Fig. 7.1 Typical TDI-1050 System Configuration

| Live Pin   | Gnd Pin | Signal                     | Dwg. no. |
|------------|---------|----------------------------|----------|
| P121-V     | -       | TSLTO Select no. 0 Command | 2        |
| н          | 7       | TRWC Rewind "              | 2        |
| L          | 10      | TOFC Off-line "            | 2        |
| D          | 4       | TDDS Density Select "      | 2        |
| С          | 3       | TSFC Forward "             | 2        |
| E          | 5       | TSRC Reverse "             | 2        |
| В          | 2       | TOVW Overwrite "           | 2        |
| ĸ          | 9       | TSWS Set Write "           | 2        |
| I          | -       | TLOL Load-on-line "        | 2        |
| 18         | -       | TSLT1 Select no. 2 "       | 2        |
| A          | -       | TSLT2 Select no. 1 "       | 2        |
| J          | 8       | TSLT3 Select no. 3 "       | 2        |
| M          | 11      | TONL On-Line Status        | 2        |
| T          | 16      | TRDY Ready "               | 2        |
| R          | 14      | TLDP Load Point (BOT) "    | 2        |
| υ          | 17      | TEOT EOT "                 | 2        |
| F          | 6       | TDDI Data Density "        | 2        |
| N          | 12      | TRWD Rewind "              | 2        |
| P          | 13      | TFPT File Protect "        | 2        |
| s          | -       | 1                          |          |
| 15         | -       |                            |          |
| 14         | -       | } + 5 V                    |          |
| 16         | -       | )                          |          |
|            |         |                            |          |
| P123-3     | С       | TRDO Read Data 0           | 4        |
| 4          | D       | TRD1 " 1                   | 4        |
| 8          | J       | TRD2 " 2                   | 4        |
| 9          | ĸ       | TRD3 " 3                   | 4        |
| 14         | R       | TRD4 " 4                   | 4        |
| 15         | s       | TRD5 " 5                   | 4        |
| <u>1</u> 7 | U       | TRD6 " 6                   | 4        |
| 18         | v       | TRD7 " 7                   | 4        |
| 1          | A       | TRDP " P                   | 4        |
| 2          | В       | TRDS Read Strobe           | 4        |
| 10         | L       | TNRZ NRZI Status           | 2        |
| 11         | М       | T7TR 7-Track "             | 2        |
| 12         | N       | TSGL Single Gap "          | 2        |
| 13         | P       | TLSP Low Speed "           | 2        |
| 6          | -       | POCP Power-On Clear "      | 2        |
|            |         |                            |          |
| P122-M     | 11      | TWDO Write Data 0          | 3        |
| N          | 12      | TWD1 " 1                   | 3        |
| P          | 13      | TWD2 " 2                   | 3        |
| R          | 14      | TWD3 " 3                   | 3        |
| \$         | 15      | TWD4 " 4                   | 3        |
| T          | 16      | TWD5 " 5                   | 3        |
| U          | 17      | TWD6 " 6                   | 3        |
| v          | 18      | TWD7 " 7                   | 3        |
| L          | 10      | TWDP " P                   | 3        |
| Ē          | 5       | TRTHI Read Threshold 1     | 2        |
| A          | 1       | TWDS Write Strobe          | 3        |
| С          | 3       | TWRS Write Reset           | 3        |
| F          | 6       | TRTH2 Read Threshold 2     | 2        |
| H          | -       |                            |          |
| J          | -       | + 5 V                      |          |
| 7          |         |                            |          |
| 8          |         | 1 )                        |          |
|            | ļ       | 1 –                        | ,        |

#### 7.2 SIGNAL DESCRIPTION

Note that these signals are high when true. The action described assumes that the signal is true.

### 7.2.1 FORMATTER-TO-TRANSPORT SIGNALS

For details on signal requirements and on complete effect on tape transport, consult tape transport specifications.

TDDS, P121-D, Data Density Select.
Selects the High-Density Mode. Transits the formatter.

TLOL, Pl21-I, Load on-line.
Loads transport and places it on-line
by remote command. Transits the formatter.

TOFC, P121-L, Off-line-command.
Returns transport off-line by remote command. Transits the formatter.

# TOVW, P121-B, Overwrite. Conditions the transport to allow rewriting of a selected record. Transits the formatter.

TRTH1, P122-E, Read Threshold 1. Selects a high read threshold in single-stack transports (for amplitude check after a write operation). Transits the formatter.

TRTH2, P122-F, Read Threshold 2.

Selects an extra low read threshold in the transport (to recover weak signals).

Transits the formatter.

TRWC, Pl21-H, Rewind Command.
Commands tape to move reverse at high speed. Transits the formatter.

TSFC, P121-C, Synchronous Forward Command. Commands tape to go forward at synchronous (specified) tape speed. Is under program control.

#### TSLTx, Pl21, Select.

Selects one of up to four transports. Decoded by the formatter from a 2-bit remote select signal.

TSRC, P121-E, Synchronous Reverse Command. Commands tape to go backwards at synchronous (specified) speed unless tape is at Load Point. Is under program control.

TSWS, Pl21-K, Set Write Status. Enables write circuitry in the tape transport. Transits the formatter.

### TWDx, P122, Write Data. Constitute an 8-bit data byte plus

a parity bit to be recorded on tape. Derived directly from input write data under program control, except for the parity bit which in a standard confirguration is generated by the formatter.

TWDS, 122-A, Write Data Strobe. Effects actual recording of the write data present on the TWDx lines. Generated by the formatter.

#### TWRS, 122-C, Write Reset.

Resets the transport write circuits and causes a flux transition to the polarity of the interblock gap for all those tracks not already so magnetized. (The ensuing flux transitions constitute the LRCC when operating in NRZI mode.)

#### 7.2.2 TAPE TRANSPORT-TO-FORMATTER SIGNALS

POCL, P123-6, Power-On-Clear Pulse.
Will clear the formatter, specifically the program counter, instruction register, command and control registers.

TDDI, P121-F, Data Density Indication. Will select the highest of the two possible densities in 7-track operation.

# TEOT, P121-U, End of Tape. Indicates the detection of an EOT marker on the tape. Transits the formatter, but is monitored by the formatter to stop a search operation before tape run-out.

# TFPT, Pl21-P, File Protect. Indicates the presence of a write ring on the mounted tape reel. Transits the formatter.

#### TLDP, P121-R, Load Point.

Indicates the detection of a BOT marker on the tape. Transits the formatter, but is tested by the formatter and used extensively in the program.

#### TLSP, P123-13, Low Speed.

Indicates that the tape transport is set to run at half the tape speed specified for the mag tape system. Will cut the system clock rate in half to maintain proper timing for this speed.

#### TNRZ, P123-10, NRZI mode.

Indicates that the transport is operating in the NRZI mode. Transits the formatter, is tested by the formatter program and effects operation by the formatter in the NRZI mode.

#### TONL, P121-M, On-Line.

Indicates that the transport is on-line. Transits the formatter.

#### TRDx, Pl23, Read Data.

Constitute an 8-bit data byte plus a parity bit being read off the tape. Processed in software by the formatter in the NRZI mode, processed in formatter by the PE microcontroller logic in the PE mode.

#### TRDS, P123-2, Read Data Strobe.

Strobes the NRZI read data into the formatter input registers and effects program processing.

#### TRDY, Pl21-T, Ready.

Indicates that the transport is loaded, on-line and not rewinding. Transits the formatter and enables the formatter GO signal which strobes the command registers.

#### TRWD, Pl21-N, Rewinding.

Indicates that the transport is doing a rewind operation. Transits the formatter.

#### T7TR, P123-11, 7-Track.

Indicates that the transport is equipped with a 7-track head. Conditions the formatter for 7-track operation and transits the formatter if so strapped.

#### TSGL, P123-12, Single Gap.

Indicates that the transport is equipped with a single gap head. Transits the formatter and conditions it for single-gap operation.

# SECTION 21 SYSTEM DESCRIPTION

#### CONTENTS

- 21.0 SYSTEM DESCRIPTION
- 21.1 MNEMONICS

#### 21.0 SYSTEM DESCRIPTION

In the TDF 4050 Formatter, a specially constructed, fast microprocessor is used to control all formatting tasks, and most of the signal processing involved lies in software, stored in ROMS.

Formatter hardware, exclusive of the microprocessor hardware, is in the following referred to as the formatter system. All formatter fuctions and operations are synchronized to a precision system clock.

All pertinent inputs and outputs to and from the CPU and Tape Transport interface are fed onto, or off, a central, common Data Bus DBx, see Fig. 21.1. Strobes Ax, Bx out of the microprocessor control this signal flow and operate various registers, latches and gates in the formatter system.

All commands, status signals and read/ write data enter this bus, and all decoding, processing and distribution are handled via this bus by the microprocessor, with certain exceptions:

Some command and status signals need no intervention by the formatter and pass right through.

NRZI formatting utilizes standard packages for parity bit generation/ checking and CRC character generation in the write mode, but is otherwise exclusively processed by the microprocessor firmware (firmware = ROMs containing program software).

For PE formatting, the read operation is special and employs a unique circuit scheme for deskewing read data signals. Performing this task in microprocessor firmware is precluded because of the high transfer rate in PE and the multiplicity of 9 channels. The deskew circuits are fully digital, however, and the PE write operation is handled completely by software.

The control signals for the special circuitry is handled via the Data Bus, using microprocessor signals Ax for strobing. However, some formatter system status signals (Mx) are fed directly to the microprocessor outside the Data Bus. These are signals tested for purposes of program branching. The Data Bus is at such time occupied by the accompanying jump address.

A timing counter, under the control of the microprocessor, is provided for the generation of time delays and write strobes.

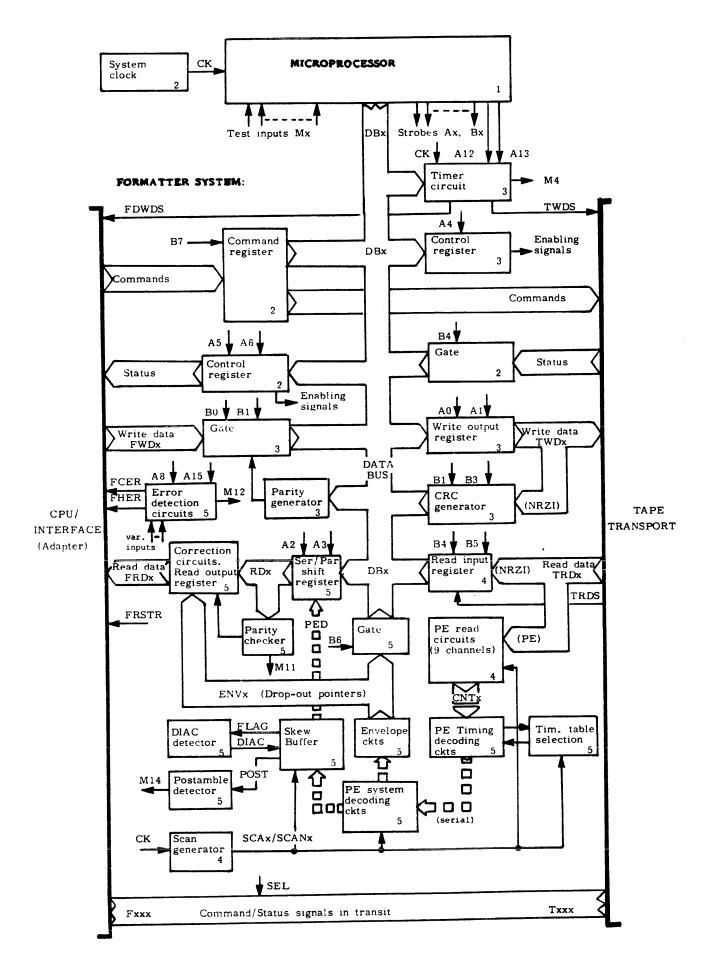


Fig. 21.1 TDF 4050 Block Diagram

# 21.1 MNEMONICS

| Mne-<br>monic | .Description  | Mne-<br>monic | Description   |
|---------------|---|---------------|---|
| Ax            | uP strobe signals. Strobes uP (uP = 4050 microprocessor) signals DBx from Data Bus into formatter system, see Table 22.5. | DCK           | Timer Circuit output clock signal at double character bit rate.   |
| ABx           | uP Address Bus signals. Selects<br>Ax and Bx strobes or Scratch Pad<br>locations.   | DIAC          | Data in All Channels. Goes true every time a complete data byte has been assembled in the Skew Buffer, including the full preamble "1" byte (see FLAG). |
| Bx            | uP strobe signals. Strobes signals from the formatter system onto the Data Bus as DBx, see Table. 22.5.                   | DISCL         | Disable system clock.  ALU output (result) bits.  |
| Cx            | Internal uP control strobes, see Table 22.2. Moves data between the various uP functional units.                          | DROP-<br>OUT  | Goes true if there is a drop-<br>out on the track in question.<br>The ENV signal will go false<br>and thereafter maintain DROP-                         |
| CBx           | uP Control Bus signals. Selects ALU function, Cx or Mx, etc.  |               | OUT false for that track for the remainder of the block.  |
| CE            | Chip Enable. Activates Scratch Pad Memory.  | ECK           | Enabled Clock signal.   |
| CK            | System Clock Pulse. Width 70 nsec. Rate: see Table 1 on   | EFWDS         | Enable FDWDS (demand next word) signal.   |
| GI FII D      | Schematic no. 2.  | ENFL          | Enable (Soft error indicator) Flip-Flop.  |
| CLEAR         | Clears all relevant registers when formatter is not enabled (FFEN = 0) or when power goes on (POCP = 1).                  | ENHE          | Enable Hard Error indication in case of multiple-track drop-outs.   |
| CKCRC         | Clock enable signal for the CRC generator.  | ENLRC         | Enable LRC character, i.e. TWRS strobe generation.  |
| CNTx          | Counter Ouput: Bits indicate time interval between flux transitions in PE read data.                                      | ENRST         | Enable FRSTR (read strobe) signal.  |
| CORR          | Corrected Bit. Causes formatter to ignore parity error.   | ENSE          | Enable Soft Error detection of single-track drop-outs.  |
| CRCG          | CRC gate. Causes formatter to ignore parity error.  | ENV           | Envelope, i.e. a data signal is present, and it is not a dropout track. (Derived from the   |
| DAx           | Accumulator data bits.  |               | DROP-OUT signal on the previous scan cycle.)  |
| DBx           | Data Bus signals. Represent read/write data, timing data, jump addresses, status and                                      | ENVx          | ENV for channel x.  |
|               | command bits. For common bit allocations, see Table 22.5, 22.7.   | EPAR          | Even Parity. (Used in 7-track operation).   |

| Mne-<br>monic      | Description   | Mne-<br>monic  | Description  |
|--------------------|---|----------------|--|
| ETWDS              | Enable TWDS (write strobe)  | MULT           | Multiple-track drop-out.   |
|                    | signal.   | NRZS           | NRZI Status gating signal.   |
| EXCLK              | External Clock signal. Can be applied when DISCL = 1.   | OVER-          | Skew Buffer overflow.  |
| Fxxx               | Formatter/CPU signals, see paragraphs 6.2 and 6.3.  | PARE           | Parity error.  |
| FLAG               | Goes true after a preamble "1"  | PCx            | Program Counter bits. Address to uP Program ROM.   |
|                    | has been detected or after a<br>new valid bit has entered the<br>skew buffer. (DIAC goes true         | PED            | PE read data (in serial form).   |
|                    | when FLAG proves true for each channel in one and the same scan cycle.)                               | PEDET          | PE data block detected (M7-DDET = 1, NRZS = 1).  |
| GO                 | Start read/write operation (FGO, FBSY, TRDY all = 1).   | PERST          | PE reset. Keeps Edge Detector flip-flops in the PE Read Circuits cleared until the onset of a PE read operation. |
| LADD               | Load Address, for jump or call operation.   | POCP           | Power-On Clear Pulse.  |
| LEVEL              | Read Data logical level from PE READ Circuit.   | POST           | A postamble "1" detected in current channel.   |
| LLEVEL             | Last (previous) LEVEL signal.   | POSTD          | Postamble detected. (POST is true for each channel in one and  |
| LLOWR              | Last (previous) LOWR signal.  |                | the same scan cycle.)  |
| LOAD               | Load new data bit (A data trans-<br>ition has taken place since last<br>being scanned). (Synchronized | PREMD          | Preamble detected. (Set by the first DIAC signal in a block.)  |
| LOWR               | VALID signal.)  Lower Range of PE timing tables   | RBx            | uP program ROM output signals, see Table 22.8 for bit allocations.   |
|                    | in Ul28, Ul07 PE Timing Decoder PROMs.  | READ           | Enable Read Operation.   |
| LSELl              | Last (previous) SELl signal.  | REGCL          | Register Clear pulse.  |
| LSPED              | Synchronized TLSP signal.   | RREQ           | Read Request. Transport read   |
| LT <sub>O</sub>    | Last (previous) $T_0$ $T_1$ signals.  | R∕₩            | strobe received.  Scratch Pad read out command   |
| LTREND             | Last (previous) TREND signal.   |                | signal. Is false when data is written into the Scratch Pad.  |
| LVALID             | Last (previous) VALID data signal.  | s <sub>0</sub> | Parameters indicative of bit spacing quality.  |
| Mx                 | Input signal to jump test multiplexer. See Table 22.3.  | Sx             | Scratch Pad Locations, see<br>Table 22.7. (Not a signal.)  |
| MOTION             | Move tape (forward or reverse). Enables the TSFC and TSRC   | SCAx           | Scan strobe, binary coded.   |
| 21_4               | signals.  | SCANx          | Scan strobe, decoded.  |
| / 1 <del></del> // |   |                |  |

| Mne-<br>monic        | Description   | Mne-<br>monic | Description   |
|----------------------|---|---------------|---|
| SEL                  | Formatter selected; addressed and enabled by CPU (FFAD, FFEN = 1). Gates formatter input and output signals.  | WRITE         | Enable write operation.  Condition for a jump is met. |
| SELl                 | Select PROM 1.  |               |   |
| SENV                 | Synchronized ENV signal. Is delayed relative to original DROP-OUT signal, but so is eventual PE data byte with which it operates.                                     |               |   |
| SFC                  | Synchronous Forward Command.  |               |   |
| SLADD                | Synchronized LADD signal. Being used to extend the LADD signal to the next clock pulse.   |               |   |
| SLEVEL               | Synchronized LEVEL signal.  |               |   |
| SLLD                 | The SLEVEL signal clocked by (and'ed with) the LOAD signal.   |               |   |
| SONE                 | Set logical "l" signal. Used when needed in the absence of a true FLAG signal, e.g. in the case of a single-track drop-out being corrected.                           |               |   |
| SRREQ                | Synchronized RREQ signal.   |               |   |
| STRBx                | Signal presence in PE read channels. Used for data block detection and gap qualification.   |               |   |
| SUP                  | Suppress. A transitory pulse between program steps to prevent previously active output circuits to the Data Bus from short-circuiting the next active output circuit. |               |   |
| To<br>T <sub>1</sub> | Parameters indicative of timing table suitability.  |               |   |
| Txxxx                | Transport/Formatter signals, see paragraph 7.2.   |               |   |
| TREND                | Indicates longer term variation in the bit rate.  |               |   |
| VALID                | Data bit deemed valid by the PE system decoding circuitry.  |               |   |

# **SECTION 22 MICROPROCESSOR**

## CONTENTS

- 22.0 MICROPROCESSOR
- 22.1 ARCHITECTURE

  - 22.1.1 PC & PROGRAM ROM
    22.1.2 ROM Output Circuits
    22.1.3 Control

  - 22.1.4 ALU
  - 22.1.5 Scratch Pad Memory
- 22.2 CIRCUITRY, Schematic 1
- 22.3 INSTRUCTION SET
  - 22.3.1 Data Transfer
  - 22.3.2 ALU Operations 22.3.3 Branching

  - 22.3.4 Programming Examples

# 22.0 MICROPROCESSOR

#### 22.1 ARCHITECTURE

A block diagram of the microprocessor is shown on Fig. 22.0. The diagram shows five sections numbered 22.1.1 through 22.1.5 which refer to respective areas on schematic no. 1. Descriptions of circuit detail are provided on the page facing the schematic. A functional description will be given below in respective paragraphs 22.1.1 to 22.1.5.

The 8-bit data bus DBx (x = 0 through 7) constitutes the main communication line between the microprocessor and the outside world and between the various components in the microprocessor itself.

On this bus, data to be processed (in the ALU) is received from the outside and processed data delivered to the outside. The bus also carries data to and from the internal scratch pad memory (RAM) and presents the program counter (PC) with jump addresses from the program memory (ROM) or from the return address register. In the latter case the Data Bus includes two more bits to provide a full 10-bit address.

## 22.1.1 PC & ROM

The program counter (PC) is the heart of the system and supplies the program memory (ROM) with the address of the instruction to be executed. In the ROM are stored the various instructions necessary to carry out all the functions of the microprocessor. Along with the instructions are stored data and address information necessary to execute the instructions.

For a sequence of instructions which are to follow each other step by step, the program counter is incremented by one for each step. However, provisions have been made in the system to preset the program counter and to be able to jump to another location in the memory and start on a different sequence of instructions. Whether to jump or not is determined by the current sequence.

It may either call for an outright jump (unconditional jump) or for a test of

some selected operating condition (conditional jump). The outcome of the test will determine whether to continue the current sequence or jump. In the case of a conditional jump, the address of the first instruction in the new sequence will be present on the data bus (DBx) ready to be loaded into the program counter. If the outcome of the test implies that no jumping shall take place, the program counter continues to be incremented by one. If there is to be a jump, loading will take place (LADD true), and the program counter will present the ROM with the address of the first instruction in the new sequence. After the jump, the program counter will resume being incremented by

If the new sequence is a subroutine and it is desirable to return to the previous sequence after the subroutine has been executed, a return address register allows the present address + 1 (PCx) in the program counter to be stored prior to the jump. This is accomplished by a Call instruction in the program. The subroutine sequence will then contain a Return instruction at the end of the sequence which will cause the program counter to be loaded with the address stored in the return address register. thus making possible the continuation of the original sequence after having completed the subroutine.

## 22.1.2 ROM Output Circuits

The 16-bit output bus, RBx, carries the selected instruction, data and address words from the ROM and distributes this information via buffer registers to respective control (CBx), data (DBx) and address (ABx) buses.

The instruction word on the control bus (CBx) is used

- to move data between the various functional components of the microprocessor: CBl2-15 is decoded to provide the internal Cx microprocessor strobes
- 2) to control operations on the data: CB8-10 selects ALU function
- 3) to address and select condition inputs for the jump test: CB10-13 selects inputs Mx.

- 4) to input or output data to the Data
  Bus and to set/reset control flipflops in the formatter system: Indirectly through the use of Cx
  strobes to operate the Ax and Bx
  strobe generators when CBll is false.
- 5) to input or output data to the scratch pad memory: Indirectly through the use of Cx strobes when CBll is true.

The address bus (ABx) is used

- to direct the Ax and Bx strobe generator outputs to appropriate points in the formatter system and
- 2) to select memory location in the scratch pad memory.

The data bus output (DBx) of the ROM is used

- to supply control words to the formatter system, directed to the appropriate control registers by the Ax strobes.
- supply file mark, pre/postamble,
   ID burst, write data.
- 3) to supply file mark compare data, mask-out data and other data to the ALU.
- 4) to supply timing data to the timer circuit and the scratch pad memory.
- 5) to provide jump addresses. In the latter case, the DBx bus is augmented by 2 bits from the control bus (CBx) to provide a 10-bit address word.

## 22.1.3 CONTROL

Control of microprocessor operation by the control bus (CBx) is exercised via an instruction decoder ROM which decodes a 4-bit part of the instruction word (CB12-15) into distinct strobe signals (Cx).

Control and strobe signals Ax, Bx to the formatter system on the other hand, are generated by suitable demultiplexers, enabled by CB11. addressed by the address bus signals (ABx) and activated by Cx. In

general, the Ax strobes will deliver data (DBx) from the microprocessor out. to the formatter system, whereas the Bx strobes will fetch data (DBx) from the formatter system into the microprocessor.

Another 4-bit part of the instruction word (CB10-13) is used to select input status signals (Mx) for the jump test. These status signals, with the exception of the MO-ZERO signal, come from various points in the formatter system and are tested by the microprocessor program to determine which one of two alternative program sequences to follow. In practice, a jump will ensue if the multiplexer output signal LADD (Load jumping Address) transmitted to the program counter (PC) is true.

#### 22.1.4 ALU

The arithmetic function to be performed in the Arithmetic Logic Unit (ALU) is controlled directly by a 3-bit part of the instruction word, CB8-10. The unit performs arithmetic or logic operations on data (DAx) stored in the accumulator register and on data (DBx) on the data bus. These operations may include addition, subtraction, logic "AND", etc., and the result (DRx) is subsequently stored in the accumulator (ACC).

The result can also be examined for jump test purposes. For example, a status condition in the formatter system, strobed in as a word on the data bus (DBx), can be compared in the ALU with a reference pattern word fetched previously from the program ROM and stored in the accumulator. A match can be detected by subtracting the two words, in which case the result is zero and makes output signal MO-ZERO go true. A jump will ensue when this signal input to the multiplexer is selected by CB10-13.

## 22.1.5 SCRATCH PAD MEMORY

The scratch pad memory (RAM) is used for temporary storage of data and as an aid in providing long time delays and bit counts. Types of data may include command and status configurations, read/and write data (see Fig. 22.7).

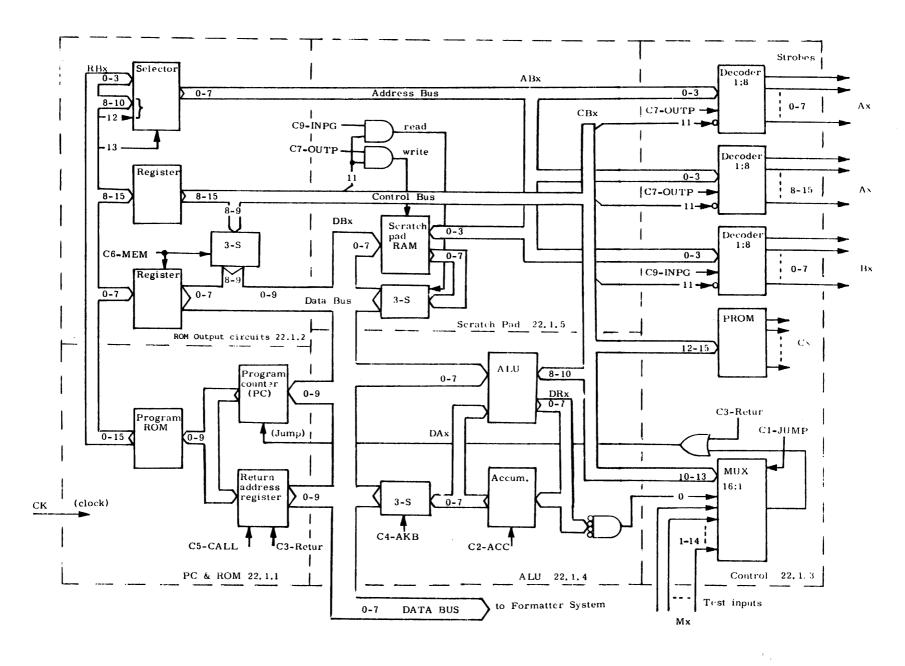


Fig. 22.0 Microprocessor Block Diagram

#### 22.2.1 PC & ROM

#### PROGRAM COUNTER

The 10-bit program counter (PC) (U71-73) is incremented by input clock signal CK. When a jump is to take place, the program counter is loaded with the new address from the data bus (DBx) by strobe signal LADD from the test multiplexer.

#### PROGRAM ROM

The program ROM (U67-70) consists of four 512-word by 8-bit packages, providing storage for 1K of 16-bit program words (RBx).

#### RETURN ADDRESS REGISTER

The transfer of the current address in the program counter (PC) to the return address register (Ul09, 75) is enabled by the CALL instruction signal C5-CALL (coming from the instruction decoder ROM). By the time this instruction is being executed the program counter setting has already advanced by one to PC + 1, which is actually the correct return address.

The RETURN instruction signal C3-RETURN is eventually used to place the contents of the return address register on the Data Bus DBx. From the Data Bus the address will be loaded into the PC by the LADD signal.

#### 22.2.2 ROM FAN-OUT

## ADDRESS BUS (ABx)

Since the Data bus (DBO-7) and the Control Bus (CB8-15) occupy all available bit positions on the ROM bus (RBO-15), particular bit positions for the remaining Address Bus cannot be firmly allocated and must be selectable (U52) according to the occasional availability of unused bit positions. Thus, when RB13 is true, ABx represent RB8-12 (except RB11); and when false, RBO-3.

## CONTROL BUS (CBx)

The control bus register Ulll is clocked by clock signal CK. When the LADD (Load Jumping Address) signal is active (true), the control bus register is prevented from being loaded. This way any instruction signal immediately following the jump instruction is being ignored until the instruction at the new jumped-to address is well established. For timing relations, see Fig. 22.1.

To obtain a 10-bit jump address on the Data Bus (DBx), two additional bits RB8-9 are introduced via the Control Bus register Ulll (as CB8-9) and a 3-state gate U58 (as DB8-9).

## DATA BUS (DBx)

The data bus register Ull3 is clocked by clock signal CK. The register is prevented from being loaded by LADD under the same circumstances as for the control bus register.

The latched data is placed on the Data Bus (DBx) by instruction signal C6-MEM. The SUP signal temporarily suppresses this action for the finite time it takes any other latch to remove data from the data bus, (i.e. to turn off), thereby avoiding short-circuits between the respective output circuits. (See Fig. 22.1)

#### 22.2.3 CONTROL

## FORMATTER STROBE DECODERS

The Address Bus signals (ABx), together with the respective C7-OUTG and C9-INPG instruction signals, select outputs Ax, Bx. Signals Ax strobes signals from the microprocessor Data Bus (DBx) into the formatter system, and signals Bx strobe signals from the formatter onto the microprocessor Data Bus (DBx). The latter are momentarily suppressed by the SUP gate signal to prevent short-circuiting between registers which are simultaneously being switched off and on.

CBll may be regarded as a part of the instruction decode signal. These signals, ABx and CBll, are also applied to the Scratch Pad Memory, and CBll determines whether the C7-OUTP/C9-INP signal is an inputting/outputting instruction or a scratch pad read/write instruction.

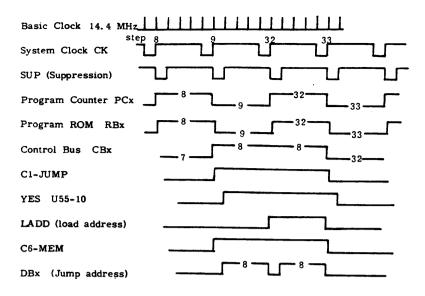


Fig. 22.1 Time Relationships

| Cl-JUMP  | Strobes the Test Input Multi-<br>plexer U55 for program branching<br>purposes. | C6-MEM  | Strobes an up to 10-bit data word RBx from memory onto Data Bus. This word can be a branch address, a time-base, a file |
|----------|--|---------|---|
| C2-ACC   | Strobes ALU result on DRx bus into accumulator U157.                           |         | mark pattern, etc.  |
|          |  | C7-OUTP | Strobes data from the Data Bus  |
| C3-RETUR | Strobes return address in Return Address Registers U109, U75 onto Data Bus.    |         | into the Scratch Pad Memory (if CBll = 1) or strobes out the Ax outputting strobes (if CBll = 0).                       |
| C4-AKB   | Strobes accumulator data DAx onto Data Bus.                                    | C9-INPG | Strobes data from the Scratch<br>Pad Memory onto the Data Bus<br>(if CBll = 1) or strobes out                           |
| C5-CALL  | Strobes present address on PCx bus into Return Address Registers U109, U75.    |         | the Bx inputting strobes (if CBll = 0).   |

|             |        |      |            |     |     |     |      |     |      | Cx  |       |     |      |
|-------------|--------|------|------------|-----|-----|-----|------|-----|------|-----|-------|-----|------|
| used        |        | CE   | 3 <b>x</b> |     | 1 . | C9- | C7-  | C6- | C5-  | C4- | C3-   | C2- | Cl-  |
| for         | CB15-  | -14- | -13-       | -12 |     | INP | OUTP | MEM | CALL | AKB | RETUR | ACC | JUMP |
| instruction | pin 13 | 12   | 11         | 10  | pin | 9   | 7    | 6   | 5    | 4   | 3     | 2   | 1    |
| Tzz         | 0      | 0    | 0          | 0   |     | 1   | 1    | 0   | 1    | 1   | 1     | 1   | 1    |
|             |        | _    | _          | -   |     | _   | _    |     | _    | _   | _     | -   | *    |
| ZZZ         | 0      | 1    | 0          | 0   |     | 1   | 1    | 0   | 1    | 1   | 1     | 0   | 1    |
| OUT, SUT    | 0      | 1    | 1          | X   |     | 1   | 0    | 0   | 1    | 1   | 1     | 1   | 1    |
| AUT, RUT    | 0      | 0    | 1          | X   |     | 1   | 0    | 1   | 1    | 0   | 1     | 1   | 1    |
| INz         | 0      | 1    | 0          | 1   |     | 0   | 1    | 1   | 1    | 1   | 1     | 0   | 1    |
| Jzz         | 1      | 0    | X          | X   |     | 1   | 1    | 0   | 1    | 1   | 1     | 1   | 0    |
| Czz         | 1      | 1    | X          | X   |     | 1   | 1    | 0   | 0    | 1   | 1     | 1   | 0    |
| RTN         | 0      | 0    | 0          | 1   |     | 1   | 1    | 1   | 1    | 1   | 0     | 1   | 1    |

Table 22.2 Instruction Decoder U54 - Truth Table Table

|        |    |             |     |          |     | U62        | U61        | U63        |                             |
|--------|----|-------------|-----|----------|-----|------------|------------|------------|-----------------------------|
|        |    |             |     | \ c7-0   | UTP | 0          | 0          | 1          | selects Ax or Bx group      |
|        |    |             |     | C9-I     |     | 1          | 1          | 0          |                             |
|        |    |             |     | AB       | 3   | 1          | 0          | 1          | selects A0-7 or A8-15       |
| ABx A  | в2 | ABl         | AB0 | CB       | 11  | 0          | 0          | 0          | selects AxBx or Scratch Pad |
| U61-63 |    |             |     | U61-63 \ |     |            |            |            |                             |
| pin    |    | 2           | 1   | pin out  |     | Ax         | Ax         | Bx         | selects individual Ax or Bx |
|        |    | <del></del> |     |          |     |            |            |            |                             |
|        | 0  | 0           | 0   | 15       |     | A0         | <b>A8</b>  | BO         |                             |
|        | 0  | 0           | 1   | 14       |     | Al         | <b>A</b> 9 | Bl         |                             |
|        | _  |             |     |          |     |            |            |            |                             |
|        | 0  | 1           | 0   | 13       | l   | A2         | A10        | B2         |                             |
|        | 0  | 1           | 1   | 12       |     | A3         | All        | в3         |                             |
|        | 1  | 0           | 0   | 11       |     | Α4         | A12        | В4         |                             |
| •      | _  | U           | J   |          |     |            |            |            |                             |
|        | 1  | 0           | 1   | 10       |     | <b>A</b> 5 | A13        | <b>B</b> 5 |                             |
|        | 1  | 1           | 0   | 9        |     | <b>A6</b>  | Al4        | в6         |                             |
|        | 1  | 1           | 1   | 8        |     | A7         | A15        | в7         | _                           |
|        |    |             |     |          | -   |            |            |            | •                           |

22.4 Formatter Strobe Decoders U61-63 - Truth Table

|            | `                     | \ DBx    | DB7 ( | DB6 | DB5     | DB4   | DB3      | DB2      | DB1            | DB0      | Destination,<br>Schem no. |
|------------|-----------------------|----------|-------|-----|---------|-------|----------|----------|----------------|----------|---------------------------|
|            | Muite Data            | TWDx     | 7     | 6   | 5       | 4     | 3        | 2        | 1              | TWDO     | 3                         |
| AO<br>Al   | Write Data Write Data | TWDx     | ,     |     |         |       |          |          | TWDP           |          | 3                         |
| A1<br>A2   | Read Data             | FRDx     |       |     |         |       | 2        | 1        | 0              | FRDP     | 5                         |
| A2         | Reset RREQ latch      |          |       |     |         |       |          |          |                |          | 4                         |
| <b>A</b> 3 | Read Data             | FRDx     |       |     |         | 7     | 6        | 5        | 4              | FRD3     | 5                         |
| AS         | Read Strobe           | RSTR     |       |     |         |       |          |          |                |          |                           |
| <b>A4</b>  | Write Control         | ,        |       |     | CKCRC   | ENCRC | ENLRC    | EFDWS    | ETWDS          | WRITE    | 3                         |
| A4<br>A5   | Read Control          | - 1      |       |     |         | ļ     | 1        | ENFL     |                | CRCG     | 2                         |
|            | Main Status           |          |       |     | NRZS    | !     | MOTION   | READ     | FDBY           | FFBY     | 2                         |
| A6         | Set Gap Test La       | tch (M9) |       |     |         | [     |          |          |                |          | 4                         |
| A7<br>——   |                       |          |       |     | <b></b> |       | -        | -        | <del> </del> - |          | 5                         |
| <b>A8</b>  | Set HER flip-f        |          |       | 1   |         |       |          |          |                |          | 2                         |
| <b>A9</b>  | Reset INREQ la        |          |       |     |         | 1     |          |          |                |          | 4                         |
| A10        | Set M7-DDET fl        | ip-flop  |       |     |         |       |          |          |                |          | 4                         |
| All        |                       |          |       | 1   |         |       |          |          |                |          | 3                         |
|            | Load Time Base        |          |       |     |         |       |          |          | 1              | ]        | 3                         |
| A13        | Reset TIMING 1        | .atch    | H     |     | 1       |       |          |          |                | 1        | 3                         |
| A14        |                       |          |       |     |         |       |          |          | 1              |          | -                         |
| <b>A15</b> | Reset HER/CER         | f-flops  |       |     |         |       | <u> </u> | <u> </u> |                | <u> </u> | 5                         |
| B0         | Write Data            | FWDx     | 7     | 6   | 5       | 4     | 3        | 2        | 1              | FWD0     | 3                         |
|            | Set Parity lat        | ch       |       |     |         |       |          |          |                |          |                           |
| Bl         | Write Data            | FWDx     |       |     |         |       | 1        |          | FWDP           | 1        | 3                         |
|            | CRC char.             | CRCx     | 1     | 1   |         |       |          |          |                | CRCP     |                           |
| в2         |                       |          |       |     |         |       |          |          |                |          |                           |
| в3         | CRC char.             | CRCx     | 7     | 6   | 5       | 4     | 3        | 2        | 1              | CRC0     | 3                         |
| В4         | Read Data, De         | ns. TRDx | TSGL  | K   | H       | TDDI  | 2        | 1        | 0              | TRDP     | 2,4                       |
| В5         | Read Data             | TRDx     | 1     | 2   | PυO     | 7     | 6        | 5        | 4              | TRD3     | 4                         |
| в6         | PE Env.               | ENVx     | 7     | 6   | 5       | 30 4  | 2        | 1        | 0              | ENVP     | 5                         |
| в7         | Write Command         | s        |       |     | READ    | ERASE | WFM      | EDIT     | WRITE          | FWD      | 2                         |

Table 22.5 Formatter Strobes Ax, Bx

| U55        | CBx<br>-pin |   |   | 11<br>14 |   | Jump if                                   | origin,<br>schem. no. | jump/call instruction |
|------------|-------------|---|---|----------|---|---|-----------------------|-----------------------|
| Signal U55 | -pin        |   |   |          |   |   |                       |                       |
| MO-ZERO    | 8-          | 0 | 0 | 0        | 0 | ALU all "0"s<br>detected                  | 1                     | JZE,CZE               |
| Ml-DINRQ   | 7           | 0 | 0 | 0        | 1 | Inputting requested (New command arrived) | 2                     | JIL,CIL               |
| M2-NRZ     | 6           | 0 | 0 | 1        | 0 | NRZI mode                                 | 2                     | JNZ, CNZ              |
| M3-7TR     | 5           | 0 | 0 | 1        | 1 | 7-track                                   | 2                     | J7T,C7T               |
| M4-TIME    | 4           | 0 | 1 | 0        | 0 | Timing not completed                      | . 3                   | JNT, CNT              |
| M5-LPD     | 3           | 0 | 1 | 0        | 1 | at load point (BOT)                       | 2                     | JLP,CLP               |
| M6-RREQ    | 2           | 0 | 1 | 1        | 0 | Read requested (New read strobe arrived)  | 2                     | JRR,CRR               |
| M7-DDET    | 1           | 0 | 1 | 1        | 1 | Data block detected                       | 4                     | JDT,CDT               |
| M8-LWD     | 23          | 1 | 0 | 0        | 0 | Last word coming                          | 2                     | JLW,CLW               |
| M9-GAPD    | 22          | 1 | 0 | 0        | 1 | Gap detected                              | 4                     | JGD,CGD               |
| M10-EOT    | 21          | 1 | 0 | 1        | 0 | EOT detected                              | 2                     | JEO,CEO               |
| Mll-PARCK  | 20          | 1 | 0 | 1        | 1 | Parity even                               | 5                     | JPE,CPE               |
| M12-HER    | 19          | 1 | 1 | 0        | 0 | Hard error detected                       | 5                     | JHE, CHE              |
| M13-PREMD  | 18          | 1 | 1 | 0        | 1 | Preamble detected                         | 5                     | JPR,CPR               |
| M14-POSTD  | 17          | 1 | 1 | 1        | 0 | Postamble detected                        | 5                     | JPD,CPD               |
| +5V "1"    | 16          | 1 | 1 | 1        | 1 | Unconditional jump                        | -                     | JUN, CUN              |

Table 22.3 Multiplexer U55 - Truth Table

## MULTIPLEXER, JUMP TEST INPUTS Mx

Address input to the multiplexer is provided by the Control Bus (CBx) which selects one of the inputs if the CI-JUMP Enable is true.

The JUMP output signal YES or the C3-RETURN instruction signal will produce the Load Jump Address signal LADD. This signal is synchronized by CK and extended to the next clock cycle by flip-flop U46, see timing diagram Fig. 22.1. The signal is also applied to the Control Bus and Data Bus registers and effectively blocks data on the next step immediately following the jump instruction. This ensures completion of the jump and the orderly transition to the next instruction at the new address.

## 22.2.4 ALU & ACCUM

## ALU

From inputs A and B, the Arithmetic Logic Unit will generate an output R according to function selected by instruction signals CB8-10.

e various functions available are listed in the ALU Function Table 22.6.

The A input will be whatever data is stored in the accumulator ACC, and the B input will be whatever data is appearing on the Data Bus (DBx). The output R is fed to a NAND gate and will produce a true MO-Zero signal whenever this output is zero. The MO-Zero signal is connected to the Test Input Multiplexer where it is available for jumping purposes.

|                 |   |                 | СВх | 10 | 9 | 8 | Ins | truct | ion |
|-----------------|---|-----------------|-----|----|---|---|-----|-------|-----|
| Function        |   | u59 <b>-6</b> 0 | pin | 7  | 6 | 5 | Tzz | ZZZ   | INZ |
| Clear           |   |                 |     | 0  | 0 | 0 |     | CLR   |     |
| B minus A minus | 1 |                 |     | 0  | 0 | 1 |     | MSU   |     |
| A minus B minus | 1 |                 |     | C  | 1 | 0 | TSB | SUB   | INS |
| A plus B        |   |                 |     | 0  | 1 | 1 | TAD | ADD   | INP |
| A ⊕ B (Xor)     |   |                 |     | 1  | 0 | 0 | TXR | XRA   | INX |
| A J B (Or)      |   |                 |     | 1  | 0 | 1 | TOR | ORA   | INO |
| A n B (And)     |   |                 |     | 1  | 1 | 0 | TAN | ANA   | INA |
| Preset          |   |                 |     | 1  | 1 | 1 |     | SET   |     |

Table 22.6 ALU U59-60 - Function Table

## **ACC**

The accumulator (ACC) receives input data from the ALU only, and this data is entered into the accumulator by instruction signal C2-ACC. The accumulator can be loaded with 0 by giving the ALU a Clear command and generating the C2-ACC instruction. The ALU is then in effect ignoring the Data Bus (DBx) input. With a zero in the accumulator, a transfer of data from the Data Bus to the accumulator can now be accomplished by generating the C2-ACC instruction and giving the ALU e.g. an A U B instruction.

Other instruction signals which appear simultaneously with C2-ACC are the C6-MEM or the C9-INPG signals, that is, data from the accumulator is introduced to the ALU together with data on the Data Bus (DBx) respectively from the program memory or from the formatter system/scratch pad.

With the desired data in the accumulator, operation between this data and the new data on the Data Bus (DBx) can be performed in the ALU and the result (R) tested on. If the new data (from Program ROM) is accompanied by the C2-ACC signal, the result is entered into the accumulator.

The Accumulator output may be dumped onto the Data Bus using the C3-AKB instruction signal. The latter signal operates in conjunction with the C7-OUTG instruction signal (only) and will make the data available to the formatter system if control bus signal CB11 is false, and to the scratch pad memory if CB11 is true.

## 22.2.5 SCRATCH PAD MEMORY

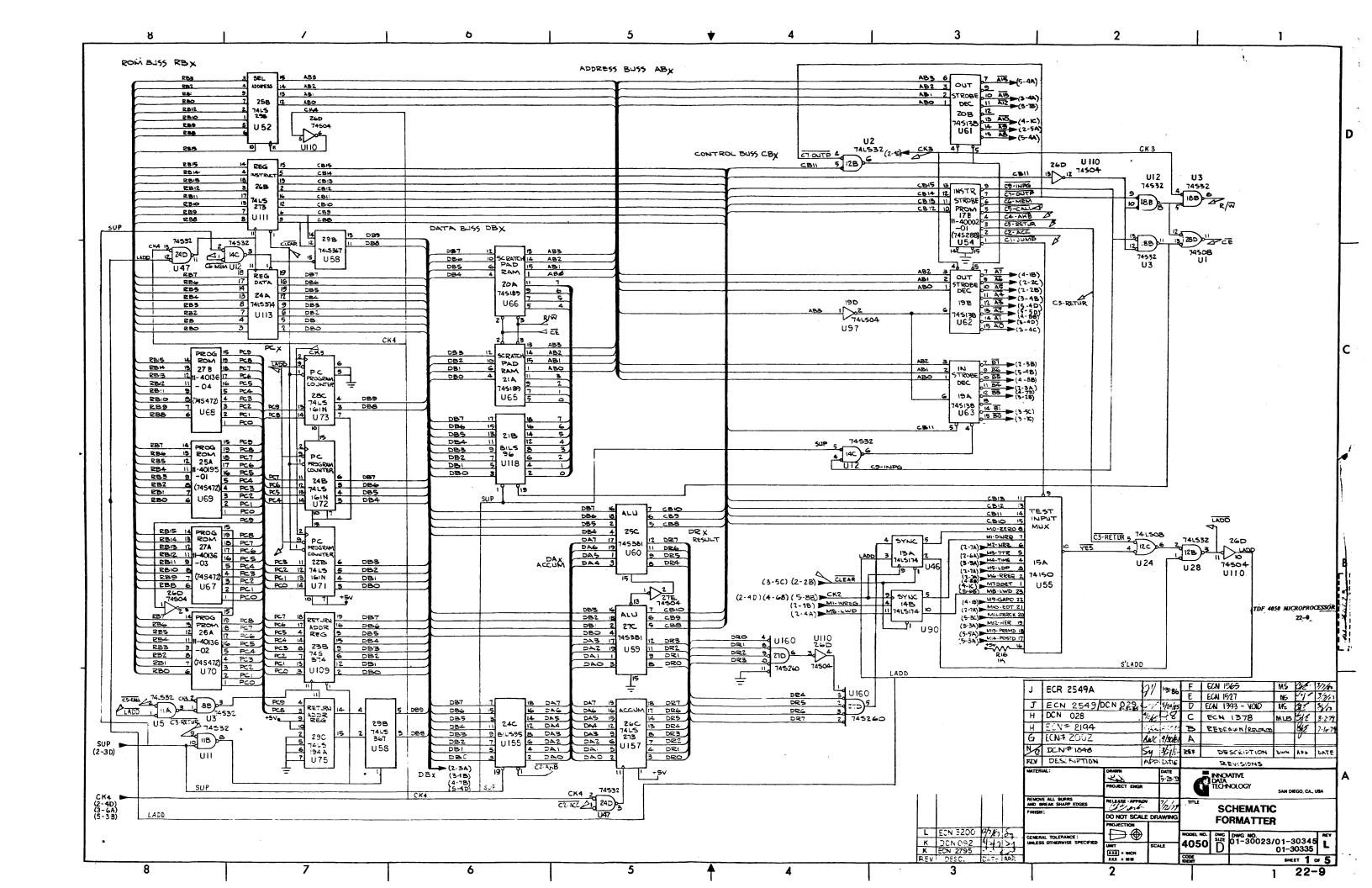
The Scratch Pad Memory consists of two 64-bit RAMs, providing storage for 16 8-bit words. Addressing is provided by the Address Bus (ABx).

The write signal is provided by instruction signal C7-OUTG and the read signal by instruction signal C9-INPG. The latter signal is gated with the transient suppress SUP signal. These signals are in turn enabled by control bus signal CBll, which multiplexes the instruction and address signals between the scratch pad memory (CBll true) and the Ax, Bx decoders (CBll false).

| RB13 |            | CBl | 1 = | 1 |   | CBll selects Scratch Pad Me    | mory  |        |            |        |                  |      |       |       |
|------|------------|-----|-----|---|---|--------------------------------|-------|--------|------------|--------|------------------|------|-------|-------|
| 0    | RBx        | ,3  | 2   | 1 | 0 | RB13 selects applicable RBx    | group | to th  | e ABx      | bus    |                  |      |       |       |
| 1    | RBx        | 12  | 10  | 9 | 8 |                                | (msb) |        |            |        | DBx              |      |       | (lsb) |
|      | ABx        | 3   | 2   | 1 | 0 | Used for storing               | DB7   | DB6    | DB5        | DB4    | DB3              | DB2  | DB1   | DB0   |
| Sx:  | so l       | 0   | 0   | С | 0 | Input Command                  | SGL   | BOT    | 1          | ERASE  | WFM              | Edit | Write | Fwd   |
|      | Sl         | Ċ   | ō   | ō | 1 | Delay Count LSB                |       |        | l          |        |                  | l    |       |       |
| l    | S2         | 0   | ō   | 1 | 0 | " "MSB                         |       | l      | 1          | }      |                  | 1    |       |       |
| ì    | <b>S</b> 3 | 0   | 0   | 1 | 1 | NRZI/PE mode                   |       |        | PE         | }      |                  | l    |       |       |
| 1    | S4         | 0   | 1   | 0 | 0 | PE write data l                | TWD7  | TDW6   | TDW5       | TWD4   | TWD3             | TWD2 | TWD1  | TWDO  |
| 1    | 1          |     |     |   |   | NRZI read data l               |       | 1      | 1          |        | TRD2             | TRD1 | TRD0  | TRDP  |
| -    | <b>S</b> 5 | 0   | 1   | 0 | 1 | PE write data 2                |       |        | 1          | _      | 1                |      | TWDP  |       |
| 1    | ļ          |     |     |   |   | NRZI read data 2               | TRD1  | TRD2   | <b>*</b> ) | TRD7   | TRD6             | TRD5 | TRD4  | TRD3  |
|      | s6         | 0   | 1   | 1 | 0 | Ramp Delay, LRC/CRC timing     |       |        | Į          |        |                  |      |       |       |
|      | S7         | 0   | 1   | 1 | 1 | Gap length, LSB (GPCNT), ID    | read  | count  |            |        |                  |      |       |       |
| į    | S8         | 1   | 0   | 0 | 0 |                                |       |        |            | ansfer | comple           | eted |       |       |
| 1    | 59         | 1   | 0   | 0 | 1 | - " - MSB " ,00                |       |        |            | char.  |                  |      |       |       |
|      | SA         | 1   | 0   | 1 | 0 | Search indicator               | AA i  | F EOT, | FF if      | file n | nark             |      |       |       |
|      | SB         | 1   | 0   | 1 | 1 | Gap count (GPTST), (DROPT)     | l     |        |            |        |                  |      |       |       |
|      | sc         | 1   | 1   | 0 | 0 | Byte count                     | l     |        |            | DE -   |                  |      | _     |       |
|      | SD         | 1   | 1   | 0 | 1 |                                | •     | parit  | y erro     | r, FF  | ir n <b>ar</b> c | erro | E     |       |
| 1    | SE         | 1   | 1   | 1 | 0 | Gap count(GPTST), Read ind.(DU | ALR)  |        |            |        |                  |      |       |       |
| 1    | SF         | 1   | 1   | 1 | 1 | Post-record delay              |       |        |            |        |                  |      |       |       |

\*) TRDO U TRDP

Table 22.7 Scratch Pad Locations Sx



#### 22.3 INSTRUCTION SET

The instruction set includes three types of instructions:

- Data Transfer: Move data between ROM,

scratch pad and for-

matter system.

- ALU operation: Perform arithmetic or

logical operations on data from ROM, scratch-

pad or formatter

system.

- Branching: Jump conditionally or

unconditionally, call subroutines and return.

This gives rise to eleven possible operations, see Table 22.8. Memory bit allocations for the various control, address and data fields are also listed in Table 22.8. A complete listing of all instructions is given in Table 22.9.

## 22.3.1 DATA TRANSFER

These instructions move data  $\underline{m}$  from the ROM or from the Accumulator out to desired areas in the formatter system selected by outputting strobe signals  $A\underline{x}$ , (Table 22.5) or into the scratch pad memory to location Sx (Table 22.7).

#### 22.3.2 ALU OPERATIONS

These instructions fetch data m from the ROM and perform arithmetic and logical operations on them, see ALU Function Table 22.6. Other data come from desired areas of the formatter system selected by inputting strobes Bx (Table 22.5) or from location Sx (Table 22.7) in the scratch pad memory.

The data (DBx) is fed to the B input of the ALU whereas the accumulator data (DAx) is fed to the A input. The output result (DRx) will normally be loaded into the accumulator except for the  $T_{\rm ZZ}$  instructions, which leave the accumulator unchanged.

Data can be entered into the accumulator by first clearing the ALU with a CLR A instruction, and then entering the data with an Or instruction: either ORA m, INO A,x or INO S,x from respectively the ROM, the formatter or the scratch pad memory.

The  $T_{ZZ}$  instruction permits operation with the same operand (stored unchanged in the accumulator) repeatedly without having to load the accumulator anew every time. This instruction is particularly useful for testing purposes. The ALU outputs are connected to a NAND gate and by effecting a zero output of the ALU, the NAND gate will present a gate signal MO-ZERO to the jump test Multiplexer which can be sampled for jump condition test purposes.

#### 22.3.3 BRANCHING

These instructions will, if certain conditions are met, disrupt the normal program flow and cause it to continue from a distant address in the ROM.

A conditional jump instruction is accompanied by a test of the conditions in question. If the conditions are not met, no jump will take place, and normal program flow will not be disrupted.

A jump condition is tested in the jump Test Multiplexer. The jump instruction contains an address field which selects an appropriate input to the Multiplexer (Table 22.3). The MO-ZERO test result from the ALU is among these inputs. A jump will ensue if the selected input is true.

An unconditional jump instruction (JUN, CUN) will be followed by a jump regardless.

For a call instruction, the jump execution is accompanied by the loading of the current address into a return address register. There is only one such register.

The return instruction RTN at the end of a subroutine will load the PC counter with the address saved by the call instruction and cause control to be resumed by the main program.

| Data Transfer AUT x  RUT x |          | ROM data (m) to formatter (Ax)  ROM data (m) to scratch pad (Sx)  ACC data to formatter (Ax)  ACC data to scratch pad (Sx)  ROM data (m) to ALU, ACC unchanged | 0 0 | 1 0 0 | 1 1 1 | ABx | 0 0 | as ABx,<br>as ABx,<br>addresses Sx<br>as ABx,<br>selects Ax<br>as ABx. | as DBx to  RBx r  (DAx to    | o formatter  Scratch Pad  not used Formatter) | C6-MEM, C7-OUTP  C6-MEM, C7-OUTP  C4-AKB, C7-OUTP |
|----------------------------|----------|--|-----|-------|-------|-----|-----|--|------------------------------|---|---|
| Data Transfer AUT x  RUT x | A + F    | ACC data to formatter (Ax)  ACC data to scratch pad (Sx)   | 0   | 0     |       | -   | 0   | addresses Sx<br>as ABx,<br>selects Ax                                  | RBx r<br>(DAx to             | not used<br>Formatter)                        |   |
| AUT x                      | A + S    | ACC data to scratch pad (Sx)   |     |       | 1     |     | 0   | selects Ax   | (DAx to                      | Formatter)                                    | C4-AKB,C7-OUTP                                    |
|                            |          |  | 0   | 0     | 1     | "   | 1   | as ARY.  | DD                           |   | <del></del>                                       |
| Tzz m                      | M,A + A  | DOM data (m) to NIII DOC unchanged   |     |       |       |     | -   | addresses Sx   |                              | not used<br>Scratch Pad)                      | C4-AKB, C7-OUTP                                   |
|                            | 1        | NOW data (m) to ALO, Acc unchanged   | 0   | 0     | 0     | 0   | 0   | as CBx,<br>controls ALU  | as DE                        | Ex to ALU                                     | C6-MEM (testing)                                  |
| ALU ZZZ B                  | M,A + A  | ROM data (m) to ALU, load ACC  | 0   | 1     | 0     | 0   | 0   | as CBx,<br>controls ALU  | as DE                        | x to ALU                                      | C6-MEM,C2-ACC                                     |
| Operation INZ A,           | x P+A    | Formatter data (Bx) to ALU, load ACC   | 0   | 1     | 0     | 1   | 0   | as CBx,<br>controls ALU  | RBx not used<br>(DBx to ALU) | as ABx,<br>selects Bx                         | C9-INP,C2-ACC                                     |
| INZ S,                     | x S+A    | Scratch pad data (Sx) to ALU, load ACC   | 0   | 1     | 0     | 1   | 1   | as CBx,<br>controls ALU  | RBx not used<br>(DBx to ALU) | as ABx,<br>addresses Sx                       | C9-INP,C2-ACC                                     |
| Jzz aa                     | aaa JUMP | Jump to address aaaaa  | 1   | 0     |       | CBx |     | lects<br>put   | Jump Address, a              | as CBx and DBx                                | C6-MEM,C1-JUMP                                    |
| Branching Czz aa           | aaa CALL | Jump to address aaaaa, save current address  | 1   | 1     |       | CBx |     | lects<br>put   | Jump Address, a              | s CBx and DBx                                 | C6-MEM,C1-JUMP<br>,C5-CALL                        |
| RTN                        | RETURN   | Return to saved address  | 0   | 0     | 0     | 1   |     | RBx not  | used, (Return A              | Address as DBx)                               | C3-RETUR  |

Table 22.8 Instruction types and program ROM bit allocation

Instruction Decoder (see Cx column or Table

22.2)

| Operation                                      | Mnemonic   | 15               | 14                                      | 13                                      | 12  | 11                                      |   | Bx Bi   |             | tteri<br>7 (                    |   | 4                                       | 3                                       | 2                                       | 1                                       | 0               | Hex.not.   | Comments  | Dest.  | Schem<br>no.   |
|--|--|------------------|---|---|---|---|---|---|-------------|---------------------------------|---|---|---|---|---|-----------------|--|---|--|--|
|  | OUT x,m  | as (             | CBx<br>ects                             | Сх                                      | ABx   | CB11                                    | as A  | ABX<br>ects A   |             |                                 | as D                                    | Bx t                                    | o fo                                    | rmati                                   | er                                      |                 |  | C6-MEM,C7-OUTP activated  | Ax   |  |
| M → F  Memory to  Formatter, Restination Ax    | OUT 0,m OUT 1,m OUT 2,m OUT 2,m OUT 5,m OUT 6,m OUT 6,m OUT 9,m OUT 9,m OUT D,m OUT C,m OUT E,m  | 00000000000000   | 1 | 1 | 0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>1<br>1<br>1<br>1<br>1 | 00000000000000                          | 0<br>0<br>0<br>1<br>1<br>1<br>1<br>0<br>0<br>0<br>1<br>1<br>1<br>1<br>1 | 0 0 1<br>1 0 0<br>0 1<br>1 1<br>0 0 0<br>0 1<br>1 1<br>0 0 0<br>1 1 1<br>0 0 1<br>1 1 0 0<br>1 1 1                  | 0<br>0<br>0 | 0<br>0<br>0<br>0<br>0<br>0<br>0 | 135                                     | m4.                                     | 00000                                   | m <sub>2</sub>                          | 0 0 0 0 0 0 m1                          | ,m <sub>O</sub> | 60 61 62 63 64 65 66 67 00 70 00 71 00 72 00 73 00 74 75 76 77 00  | TWDx Tape Write data 1 TWDx Tape Write data 2 FFDx Read output data 1 FFDx Read output data 2 Write control reg. Read control reg. Main status reg. Set M9-GAPD Set HER (Hard Error) Reset MI-INREQ Set M7-DDET (not used) Preset timing counter Reset M4-TIME (not used) Reset HER, set CER (See Table 22.5) | A0<br>A1<br>A2<br>A3<br>A4<br>A5<br>A6<br>A7<br>A8<br>A9<br>A10<br>A11<br>A12<br>A13<br>A14<br>A15 | 3<br>4,5<br>5<br>3<br>2<br>2<br>4<br>5<br>2<br>4<br>3<br>3 |
|  | SUT x,m  | as (             | CBx<br>ects                             |   | ABx   | CB11                                    | as<br>sel   | ABX<br>ects S   | ×           |                                 | as Di                                   | Bx to                                   | scr                                     | atch                                    | pad                                     |                 |  | C6-MEM,C7-OUTP activated  | S×   |  |
| M → S  Memory to Scratch Pad, location Sx      | SUT 0,m<br>SUT 1,m<br>SUT 2,m<br>SUT 2,m<br>SUT 3,m<br>SUT 4,m<br>SUT 5,m<br>SUT 6,m<br>SUT 7,m<br>SUT 8,m<br>SUT 9,m<br>SUT 8,m<br>SUT A,m<br>SUT C,m<br>SUT C,m<br>SUT C,m | 00000000000000   | 1 | 1 | 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1                                 | 1 | 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1   | 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1 1 1 0 0 1 |             | <sup>m</sup> 6                  | <b>33</b> 5                             | <sup>m</sup> 4,                         | . <sup>m</sup> 3                        | <sup>10</sup> 2                         | m <sub>1</sub>                          | <b>m</b> o      | 68 69 68 60 60 65 78 79 75 75 75 75 75   | Input command Delay LSB Delay MSB NRZI/PE (bit 5) TWDx, NRZI TRDx TWDx, NRZI TRDx Ramp delay.LRC/CRC timing GPCNT. ID read count " LRC/CRC detected " CRC all "0"s Search indicator Gap count for gap test Byte count Error Indicator Gap count, Read ind. Post-record delay (See Table 22.7)                 | 60<br>51<br>52<br>53<br>54<br>55<br>56<br>57<br>58<br>59<br>SA<br>SB<br>SC<br>SD<br>SE<br>SF       |  |
|  | AUT ×  | as<br>sel        | CBx<br>ects                             | С×                                      | ABx   | СВ11                                    | as A  | ABX<br>ects A   |             |                                 | (:                                      | not                                     | used)                                   | ı                                       |   |                 |  | C4-AKB, C7-OUTP activated   | Ax   |  |
| A → F  Accum. to  Formatter, destination Ax    | AUT 0 AUT 1 AUT 2 AUT 3 AUT 4 AUT 5 AUT 6 AUT 7 AUT 8 AUT 9 AUT A AUT B AUT C AUT D AUT E  | 00000000000000   | 0 | 1 | 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1                           | 000000000000000000000000000000000000000 | 0 0 0 1 1 1 0 0 0 1 1 1 1 1   | 0   |             | ) 0                             |   | 000000000000000000000000000000000000000 | 0 0 0 0                                 | 000000000000000000000000000000000000000 | 0 |                 | 20 00<br>21 00<br>22 00<br>23 00<br>24 00<br>25 00<br>26 00<br>27 00<br>30 00<br>31 00<br>32 00<br>33 00<br>34 00<br>35 00<br>36 00<br>37 00 | TWDx Tape write data 1 TWDx Tape write data 2 FRDx Read out data i FRDx Read out data 2 Write control reg. Read control reg. Main status reg. Set M9-GAPD Set HER (Hard ERROR) Reset M1-INREQ Set M7-DDET (not used) Preset timing counter Reset M4-TIME (not used) Reset HER, Set CER (See Table 22.5)       | A0<br>A1<br>A2<br>A3<br>A4<br>A5<br>A6<br>A7<br>A8<br>A9<br>A10<br>A11<br>A12<br>A13<br>A14<br>A15 | 2<br>4<br>3<br>3   |
|  | RUT x  | as (             | CBx<br>ects                             |   | ABx   | CBll                                    | as /  | ABX<br>ects S   |             |                                 | (1                                      | not                                     | used)                                   |   |   |                 |  | C4-AKB, C7-OUTP activated   | Sx   |  |
| A -> S Accum. data to Scratch Pad, location Sx | RUT 0 RUT 1 RUT 2 RUT 3 RUT 4 RUT 5 RUT 6 RUT 7 RUT 8 RUT 9 RUT A RUT B RUT C RUT C RUT E  | 0000000000000000 | 000000000000000                         | 1 | 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1                       | 1 | 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1   | 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 1 0 0 1                                       |             |                                 | 000000000000000000000000000000000000000 | 000000000000000000000000000000000000000 | 000000000000000000000000000000000000000 | 000000000000000000000000000000000000000 | 000000000000000000000000000000000000000 |                 | 28 00<br>29 00<br>2A 00<br>2B 00<br>2C 00<br>2D 00<br>2E 00<br>2F 00<br>3B 00<br>3B 00<br>3B 00<br>3C 00<br>3D 00<br>3E 00<br>3F 00          | Input command Delay LSB Delay MSB NRZI/PE (6 + 5) bit 5 PE TWDX, NRZI TRDX PE TWDX, NRZI TRDX Ramp delay. LRC/CRC timing GPCNT. ID read count " LRC/CRC detected " CRC all "0"s Search indicator Gap count for gap test Byte count Error indicator Gap count, Read ind. Post-record delay (See Table 22.7)    | S0<br>S1<br>S2<br>S3<br>S4<br>S5<br>S6<br>S7<br>S8<br>S9<br>SA<br>SB<br>SC<br>SD<br>SE<br>SF       |  |

Fig. 29a Instruction Set, Data Transfer

|  |  | 1  |             | RBx bit pa   | ttern                            |   | T  |   | 7   |                              |
|--|--|--|-------------|--|----------------------------------|---|--|---|---|------------------------------|
| Operation  | Mnemonic   |  | 2 11        | 10 9 8 7   | 6 5                              | 3 2 1 0   | Hex.not.   | Comments  |   |                              |
| M,A → A  | Tzz m  | as CBx<br>selects Cx   | not<br>used | as CBx sel-<br>ects ALU func   | as DB                            | to ALU  |  | C6-MEM activated  |   |                              |
| Memory<br>to ALU.<br>(Accumu-<br>lator un-<br>changed) | TSB m TAD m TXR m TOR m TAN m  | 0 0 0 0  | 0 0         | 0 1 0 m <sub>7</sub> 0 1 1 1 0 0 1 0 1 1 0 1   | ™a6 ma5 ma4                      | , m <sub>3</sub> m <sub>2</sub> m <sub>1</sub> m <sub>0</sub>   | 02<br>03<br>04<br>05<br>06                         | If A minus m minus 1 A plus m A ⊕ m A + m A ⋅ m (If A = OOH, then M1-ZERO = 1)  |   |                              |
|  | 222 B  | as CBx<br>selects Cx   | not<br>used | as CBX sel-<br>ects ALU func   |                                  | x to ACC  |  | C6-MEM and C2-ACC activated   |   |                              |
| M,A → A  Memory to ALU.(Accu- mulator loaded)          | CLR m<br>MSU m<br>SUB m<br>ADD m<br>XRA m<br>ORA m<br>ANA m<br>SET A | 0 1 0 0<br>0 1 0 0 | 0 0 0 0     | 0 0 0 0 0 0 0 0 0 1 m <sub>7</sub> 0 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 0 0 0 1 1 1 0 | 0 0 0                            | m <sub>3</sub> m <sub>2</sub> m <sub>1</sub> m <sub>0</sub>   | 40 00<br>41<br>42<br>43<br>44<br>45<br>46<br>47 00 | A = 00 Hex A = m minus A minus 1 A = A minus m minus 1 A = A plus m A = A (*) m A = A + m A = A + m A = F Hex (If A = 00H, then M1-ZERO = 1)  | Dest.   | Schem no.                    |
|  | INZ A,x  | as Cbx<br>selects Cx   | CB11        | as CBx sel-<br>ects ALU func   | (not used)                       | as ABx<br>selects Bx  |  | C9-INP and C2-ACC activated   | Bx  |                              |
| F → A  | INS A,x<br>INP A,x<br>INX A,x<br>INO A,x<br>INA A,x                  | 0 1 0 1<br>0 1 0 1<br>0 1 0 1<br>0 1 0 1<br>0 1 0 1                                  | 0<br>0<br>0 | 0 1 0 0<br>0 1 1 0<br>1 0 0 0<br>1 0 1 0<br>1 1 0 0  | 0 0 0<br>0 0 0<br>0 0 0<br>0 0 0 |   | 52<br>53<br>54<br>55<br>56                         | A = A minus P minus 1<br>A = A plus F<br>A = A 	 F<br>A = A + F<br>A = A - F  |   |                              |
| Formatter<br>to<br>ALU                                 |  |  |             |  |                                  | 1 0 0 0 1 1 0 1 0 1 1 1 1 0 0 1 1 1 1 1   | 08<br>09<br>0A<br>0B<br>0C<br>0D<br>0E<br>0F       | FWDx CPU Write data 1 FWDx CPU Write data 2 (not used) CRC character NRZI Read data 1, Density NRZI Read data 2 PE Env. Command input register  | BG<br>B1<br>B2<br>B3<br>B4<br>B5<br>B6<br>B7  | 3<br>3<br>2,4<br>4<br>5<br>2 |
|  |  | as CBx   | 1 1         | as CBx sel-  |                                  | selects Bx  |  | (See Table 22.5)  |   | ļ                            |
|  | INZ S,X  | selects Cx   | CB11        | ects ALU fund  | (not used)                       | selects Sx  |  | C9-INP and C2-ACC activated   | S×  |                              |
|  | INS S,x<br>INP S,x<br>INX S,x<br>INO S,x<br>INA S,x                  | 0 1 0 1<br>0 1 0 1<br>0 1 0 1<br>0 1 0 1<br>0 1 0 1                                  | 1 1 1       | 0 1 0 0<br>0 1 1 0<br>1 0 0 0<br>1 0 1 0<br>1 1 0 0  | 0 0 0<br>0 0 0<br>0 0 0<br>0 0 0 |   | 5A<br>5B<br>5C<br>5D<br>5E                         | A = A minus S minus 1<br>A = A plus S<br>A = A ③ S<br>A = A + S<br>A = A · S  |   |                              |
| S → A Scratch pad to ALU                               |  |  |             |  |                                  | 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 1 | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F    | Input command Delay LSB Delay MSB NRZS/PE (6 + 5) bit 5 PE TWDX, NRZI TRDx PE TWDX, NRZI TRDx Ramp delya.LRC/CRC timing GPCNT. ID read count " LRC/CRC detected " CRC all "0"s Search indicator Gap count for gap test Byte count Error indicator Gap count, Read ind. Post-record delay (See Table 22.7) | \$0<br>\$1<br>\$2<br>\$3<br>\$4<br>\$5<br>\$6<br>\$7<br>\$8<br>\$9<br>\$A<br>\$B<br>\$C<br>\$D<br>\$E<br>\$E<br>\$E<br>\$E<br>\$E<br>\$E<br>\$E<br>\$E<br>\$E<br>\$E<br>\$E<br>\$E<br>\$E |                              |

Fig. 29b Instruction Set, ALU Operation

|                            |   |   | _                                |    |   |   |  | F  | Bx E       | it             | atte           | rn         |                  |                 |                |                       |                |                       | T                       | 1  |  | Scher  |
|----------------------------|---|---|----------------------------------|----|---|---|--|--|------------|----------------|----------------|------------|------------------|-----------------|----------------|-----------------------|----------------|-----------------------|-------------------------|--|--|--|
| Operation                  | Mnemonic  | 15                                      | 14                               | 4  | 13                                      | 12  | 11   | 10   | 9          | 8              | 7              | 6          | 5                | 4               | 3              | 2                     | 1              | 0                     | Hex.not.                | Comments   | Dest.  |  |
|                            | Jzz addr.   |   | CB:                              |    | as<br>sel                               | CBx<br>ects   | Мх   |  |            |                |                | as Di      | Bx J             | and description | addre          | 255                   |                |                       |                         | C6-MEM,C1-JUMP activated. Jump if  | Мх   |  |
| Jump<br>conditio-<br>nally | JZE JIL JNZ JTT JNT JLP JRT JDT JLW JGD JED JPE JHE JPR JPD JUN | 111111111111111111111111111111111111111 | 0                                |    | 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 | 0<br>0<br>0<br>0<br>1<br>1<br>1<br>1<br>0<br>0<br>0<br>1<br>1<br>1      | 0<br>0<br>1<br>1<br>0<br>0<br>1<br>1<br>0<br>0<br>1                | 0 1 0 1 0 1 0 1 0 1                            | <b>3</b> 9 | <sup>m</sup> 8 | <sup>m</sup> 7 | <b>m</b> 6 | <sup>m</sup> 5,  | 4               | <sup>m</sup> 3 | <b>m</b> <sub>2</sub> | m <sub>1</sub> | <b>m</b> <sub>0</sub> | 8 8 8 8 9 9 9 A A B 8 8 | Zero Input command reg. is loaded NRZI = 1 7-track = 1 No timer indication At load point RREQ = 1 Data block detected LWD = 1 Command detected ECT = 1 Parity Error Hard Error Postamble detected Jump unconditionally             | MO<br>M1<br>M2<br>M3<br>M4<br>M5<br>M6<br>M7<br>M8<br>M9<br>M10<br>M11<br>M12<br>M13<br>M14<br>M15 | 1<br>2<br>2<br>3<br>2<br>2<br>4<br>2<br>4<br>2<br>5<br>5 |
|                            | Czz addr.   | 1                                       | CB:                              |    | as (                                    | CBx<br>ects   | Mx   |  |            |                | a              | s DE       | 3x Ca            | 11 4            | addre          | ss                    |                |                       |                         | C6-MEM, C1-JUMP,C5-Call.   | Mx   |  |
| Call<br>conditio-<br>nally | CZE CIL CNZ C7T CNT CNT CLP CRR CDT CCBD CBO CPE CCP CPP CUN    | 1 1 1 1 1 1 1 1 1 1 1 1 1 1             | 11 11 11 11 11 11 11 11 11 11 11 |    | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1       | 0<br>0<br>0<br>0<br>1<br>1<br>1<br>0<br>0<br>0<br>0<br>1<br>1<br>1<br>1 | 0<br>0<br>1<br>1<br>0<br>0<br>1<br>1<br>0<br>0<br>1<br>1<br>0<br>0 | 0<br>1<br>0<br>1<br>0<br>1<br>0<br>1<br>0<br>1 | <b>"</b> 9 | <b></b> 8      | 7              | <b>m</b> 6 | <sup>10</sup> 5, |                 | <sup>m</sup> 3 | <sup>m</sup> 2        | m <sub>1</sub> | <b>®</b> O            | C C C D D E E E F F     | Zero Input command reg. is loaded NRZI = 1 7-track = 1 No timer indication At load point RREQ = 1 Data block detected LWD = 1 Gap detected At EOT Parity Error Hard Error Postamble detected Call unconditionally, (PC+1 is saved) | M0<br>M1<br>M2<br>M3<br>M4<br>M5<br>M6<br>M7<br>M8<br>M9<br>M10<br>M11<br>M12<br>M13<br>M14<br>M15 | 1 2 2 2 3 2 2 4 2 4 2 5 5 5 5                            |
|                            |   | as                                      | СВх                              | se | el.                                     | Сх  |  |  |            |                |                | (n         | ot u             | sed)            |                |                       |                |                       |                         | C3-RETUR activated   |  |  |
| Return                     | RTN   | 0                                       |                                  | )  | 0                                       | 1   | 0  | 0  | 0          | 0              | 0              | 0          | 0                | 0               | 0              | 0                     | 0              | 0                     | 10 00                   | Return to previous program   | addres   | 59+1   |

Fig. 29c Instruction Set, Branching

## 22.3.4 PROGRAMMING EXAMPLES

-1. Storing input commands in Scratch Pad Memory

One of the very first actions in the formatter program is to set up the initial operating conditions, among others to heed the input commands to the formatter from the CPU.

One group of such commands are the write commands controlled by input strobe B7, see schematic 2 and Table 22.5. These commands are to be stored in the scratch pad register for use later on. To do that, they are at first brought into the accumulator via the data bus and the ALU and then from there loaded into location S0 in the scratch pad memory.

This takes more than two instructions since the data is moved via the ALU and is automatically operated on by whatever is presently stored in the ACC. To prevent alteration of the data by such operation, the ALU is first cleared using instruction CLR A, thereby loading a 0 into the accumulator. Data on the data bus is then entered into the ALU by an input instruction which will not alter the data by operating on it with 0. The data will then end up unaltered in the accumulator.

INO A,7 is such an instruction, which will perform an OR operation on the 0 and the write commands inputted by B7, resulting in the latter being now stored in the accumulator, as intended.

The data can now be transferred to the scratch pad memory using the RUT 0 instruction which will transfer the data from the accumulator to location SO in the RAM.

However, in the case of the write command byte, bits 6 and 7 are not used (see Table 22.5), and it is desirable to mask out these bits. This is accomplished by loading the accumulator with 1's in the bit positions of interest and 0's in the positions to be masked out, using instruction ORA 3FH. Since 3F is 0011 1111 in hexadecimal notation, and the left most bits are the most significant bits, 0's are thus obtained for bit position 6 and 7. The data to be loaded into the accumulator are subsequently AND'ed with 3F using instruction INA A,7 instead of INO A,7, essentially passing on only bits 0 through 5 to the accumulator.

The resulting program steps, loading the write commands into the scratch pad memory, are therefore:

ALU ACC

CLR A Clear ALU enter 0

ORA 3FH OR 0 and 3F enter 3F

INA A,7 AND 3F and B7 data enter bits 0-5

RUT 0 Enter Acc. data into location S0

## -2. Testing jump (branch) conditions

Normally, the decision to jump requires only one instruction, e.g. JEO "address", which results in a jump to the specified address if the simultanously addressed multiplexer input M10-EOT is true. (The M10-EOT input indicates whether or not the EOT marker is being detected in the selected mag tape unit).

However, not all jump conditions can be tested by this direct method. Not, for example, the forward command. This particular test object is represented by bit 0 in the B7 write command byte (see Table 22.5). This bit can be examined in the ALU if the byte is present in the accumulator, by e.g. instruction TAN 01H which masks out all bits but bit 0 and performs an AND operation on this bit. The result (from the NAND gate at the

ALU output) is clocked into the MO-ZERO latch (U46), and this multiplexer input is reached by jump instruction JZE "address".

Unless the write command byte already is in the accumulator it must be fetched from the scratch pad memory, and the program steps may thus be:

|               | ALU                  | ACC :                   |  |  |  |  |  |
|---------------|----------------------|-------------------------|--|--|--|--|--|
| CLR A         | Clear ALU            | enter 0                 |  |  |  |  |  |
| INO S,0       | OR 0 and S0 byte     | enter SO byte           |  |  |  |  |  |
| TAN OLE       | AND SO byte and 01   | (SO byte still in acc.) |  |  |  |  |  |
| JZE (address) | Test MO-ZERO and jum | p if true               |  |  |  |  |  |

#### -3. Setting up a time delay

To set up a time delay, a unit of time and a count number (multiplication factor) must be selected such that the count number times the unit time equals the desired delay. A subroutine (See DELAY, Paragraph 23.20) will take care of the count bookkeeping and return to the normal program at the end of the delay.

The time unit is selected by presetting a time counter in the formatter, using strobe Al2. The count number is a two byte hexadecimal number which is transferred to locations Sl and S2 in the scratch pad memory.

The necessary steps are therefore the following:

| OUT C, m | Strobe Al2 presets the counter to m,. (for  |
|----------|---|
|          | counter to m <sub>1</sub> . (for 10 usec at 75 ips:m=208= DOH). This program step |
|          | may already have been made  |
|          | earlier in the program.   |

SUT 1, m<sub>2</sub> Load Least significant count byte (LSB) m<sub>2</sub> into location S1.

SUT 2, m<sub>3</sub>

Load Most significant count byte (MSB) m<sub>3</sub> into location S2.

CUN DELAY An unconditional jump to subroutine DELAY.

## 22.3.5 INSTRUCTION MATRIX

A source/destination instruction matrix, as shown in Table 22.10, gives a very usefull overview of the various instructions and their functions. The source from which the data originates, is listed in the vertical column, and the destination to which the data is transferred, is shown in the horizontal row. The table indicates all instructions available to transfer data from one source to a selected destination, including those which at the same time perform an operation with an operand in the accumulator.

#### Destination

|                      | Transfer        | of Data           | Transfer          | and operation | Transfer of |  |
|----------------------|-----------------|-------------------|-------------------|---------------|-------------|--|
| Source               | Formatter<br>Ax | Scratch Pad<br>Sx | ALU/Accum.<br>ACC | ALU           | Control     |  |
| Program ROM (m)      | OUT x,m         | SUT x,m           | zzz m             | Tzz m         |             |  |
| Formatter Syst. (Bx) |                 |                   | INz A,x           |               | 1           |  |
| Scratch Pad (Sx)     |                 |                   | INZ S,x           |               | 1           |  |
| Accumulator (ACC)    | AUT x           | RUT x             |                   |               |             |  |
| (Non-returning)      |                 |                   |                   |               | Jzz a       |  |
| (Returning)          |                 |                   |                   |               | Czz a/RTN   |  |

Table 22.10 Instruction Matrix

# SECTION 23 PROGRAM

| CONTE | NTS                                |                                      |
|-------|------------------------------------|--------------------------------------|
|       | PROGRAM                            | 23.20 DELAY SUBROUTINE               |
|       |                                    |                                      |
| 23.1  | START SEQUENCE                     | 23.21 IDWT2 SUBROUTINE               |
| 23.2  | IDWRT SEQUENCE                     | 23.22 GPTST SUBROUTINE               |
| 23.3  | COMDC SEQUENCE                     | 23.23 DENST SUBROUTINE               |
| 23.4  | WCOMD SEQUENCE                     | 23.24 GPCNT SUBROUTINE               |
| 23.5  | RCOMD SEQUENCE                     | 23.25 EOTCH SUBROUTINE               |
| 23.6  | ERASE SEQUENCE                     | 23.26 COMWT SUBROUTINE (incl. COMMW) |
| 23.7  | PWFLM SEQUENCE                     | 23.27 PREMW SUBROUTINE               |
| 23.8  | PREAD SEQUENCE                     | (incl. PMIDL)                        |
| 23.9  | PWBGN SEQUENCE                     | 23.28 DATWT SUBROUTINE               |
| 23.10 | NZBGN SEQUENCE                     | 23.29 DROPT SUBROUTINE               |
|       | 23.10.1 NZFLM sequence, file marks | 23.30 RWDAT SUBROUTINE               |
|       | 23.10.2 NSWRT sequence, NRZI data  | 23.31 RDATA SUBROUTINE               |
| 23.11 | WLBYT SEQUENCE                     | 23.32 TRANS SUBROUTINE               |
| 23.12 | NREAD SEQUENCE                     | 23.33 TRANB SUBROUTINE               |
| 23.13 | DTRAN SEQUENCE                     | 23.34 NDLAY SUBROUTINE               |
| 23.14 | NZGAP SEQUENCE                     | 23.35 CRCET SUBROUTINE               |
| 23.15 | RREVM SEQUENCE                     | 23.36 NOCKC SUBROUTINE               |
| 23.16 | DREND SEQUENCE                     | 23.37 SDLAY SUBROUTINE               |
|       |                                    | 23.38 ERTST SUBROUTINE               |
|       |                                    | 23.39 DUALR SUBROUTINE               |

## 23.0 PROGRAM

Most of the formatter logic is built into the microprocessor software. The program, stored on ROMs, consists of close to 1 k 16-bit words. The program provides testing of input commands and status signals and the consequent branching to the proper sequences which will bring about the desired formatter operations. These operations vary as to format (PE/NRZI), purpose (read/write/erase/edit), movement (forward/reverse/search) and contents (data, file mark, ID Burst), see List of Commands, Table 23.1.

Accordingly, the program is broken up into corresponding subprograms or sequences, each capable of handling its particular set of requirements, see Fig. 23.2.

A formatter operation always starts with the START sequence. After the operation is completed, the program returns to this sequence, in which registers are cleared and the initial conditions for a new command are set up. The program then enters a standby state waiting for the new command.

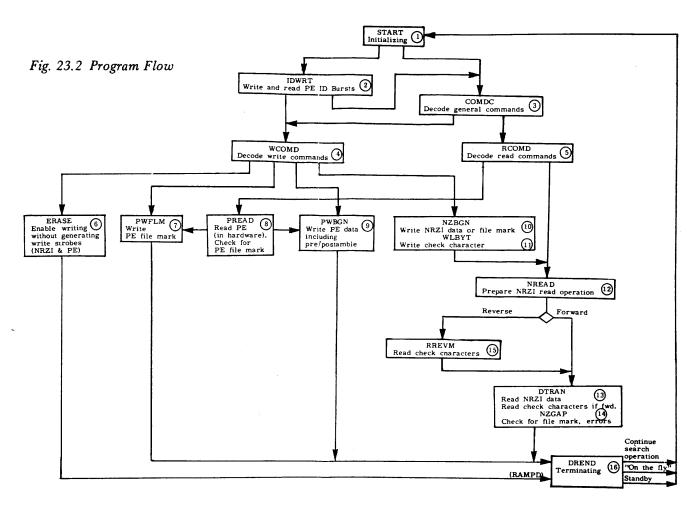
If the tape is at load point and the format selected is PE, the program immediately enters the IDWRT sequence which is concerned with reading or writing the PE Identification Burst. Otherwise, the program enters the COMDC sequence, in which further command decoding takes place.

At this point the program splits into two major parts, the WCOMD and RCOMD command decode sequences, respectively concerned with writing and reading.

The WCOMD write command decode sequence is followed by the various actual sequences: either the ERASE, the PWFLM (write PE File Mark), the PWBGN (write PE data) or the NZBGN (write NZRI) sequence. The NZBGN sequence branches internally into a NWFLM (write NRZI file mark) and a NSWRT (write NRZI data) sequence. The ERASE sequence is included here because all write operations are preceded by erasing, and an erase operation is simply a write operation without write strobes.

| Command                              | ERASE<br>DB4 | FILE<br>MARK<br>DB3 | EDIT<br>DB2 | WRITE<br>DBl | FWD<br>DBO |
|--------------------------------------|--------------|---------------------|-------------|--------------|------------|
| Read Forward                         | 1            | 0                   | 0           | 0            | 1          |
| Read Reverse, Normal                 | 1            | 0                   | 0           | 0            | 0          |
| Read Reverse, Edit                   | 1            | 0                   | 1           | 0            | 0          |
| Write, Normal                        | 1            | 0                   | 0           | 1            | 1          |
| Write, Edit                          | 1            | 0                   | 1           | 1            | 1          |
| Write File Mark                      | 1            | 1                   | 0           | 1            | 1          |
| Erase, Variable Length               | 0            | 0                   | 0           | 1            | 1          |
| Erase, Fixed Length                  | 0            | 1                   | 0           | 1            | 1          |
| Space Forward                        | 0            | 0                   | 0           | 0            | 1          |
| Space Reverse                        | 0            | 0                   | 0           | 0            | 0          |
| File Search, Forward                 | 1            | 1                   | 0           | 0            | 1          |
| File Search, Reverse                 | 1            | 1                   | 0           | 0            | 0          |
| File Search, Forward,<br>Ignore Data | 0            | 1                   | 0           | 0            | 1          |
| File Search, Reverse, Ignore Data    | 0            | 1                   | 0           | 0            | 0          |

Table 23.1 List of Commands



The various write sequences are terminated by the DREND and RAMPD sequences which appropriately terminate the write operations, and respectively provide the proper post-record delays and the cessation of tape movement before returning to the START sequence.

The RCOMD read command decode sequence is followed by the PREAD (read PE) and the NREAD (read NRZI) sequences. A NRZI block of data is not, as with PE, symmetrical in its format structure.

In read reverse, the check characters arrive first, and a special sequence RREVM is provided to handle the check characters before entering the data transfer sequence DTRAN. The RREVM sequence splits internally into sequences for respectively 7-track and 9-track operation.

NRZI data characters and check characters in the forward mode are handled by the DTRAN sequence. Finally, the read sequences are terminated by the DREND and RAMPD sequences, as was the case for the write sequences.

The PREAD sequence is very simple because the actual read operation is performed in hardware. The PREAD sequence utilizes parts in common with the PWFLM and PWBGN sequences.

Note that all timing delays in the program are given in values of a 75 ips tape drive. However, the actual delays depend on the internal clock frequency, and this will be set according to selected tape speed, and ensure proper delays.

In addition to the sequence groups shown on Fig. 23.2 and described in paragraphs 23.1 - 16, there is a number of subroutines called out in the program to perform often repeated operations. These are described in paragraphs 23.20 - 38.

For each sequence, two flow charts are shown in addition to the program listing. One flow chart is functional and gives an explanatory description of the major events in the sequence. The other flow chart gives a detailed description of the program listing. The functional flow charts will normally give a self-explanatory account of the program operation, and the following text is meant only to be supplementary comments to the flow charts.

## 23.1 START SEQUENCE

The START sequence serves to initialize the program and to ready the formatter for action.

The main control registers and the input-loaded flip-flop (M1-INREQ) will initially have been cleared by external clear signals (Formatter Enable FFEN = 1 or Power-On-Clear POCP = 0), but since the program can return to the START sequence on its own, it must contain provisions to reset these registers by itself.

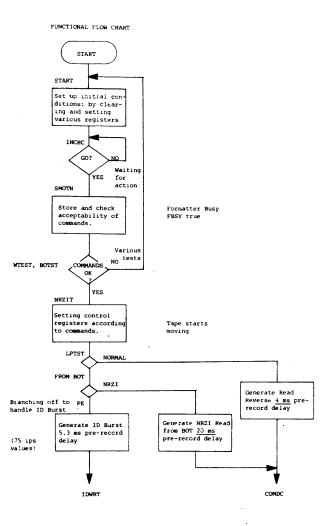
As soon as a command for a new operation has been received (Ml-INREQ = 1), the Formatter Busy line FSBY is set true, and the program then tests for non-valid commands: Read Reverse from BOT, or Write Reverse. All other commands are accepted, Table 23.1 not being exclusive.

When, in the NRZIT statement, the formatter enables tape motion, the NRZI status line NRZS is set true. A test on selected format is made, however, immediately afterwards, and if it proves to be PE, the status line is corrected and set false. The program is written this way to save bytes in the program memory.

Other byte saving measures include the early storing of pre-record delay values, in some cases only part of the total value.

The START sequence can be entered from one of the subroutines, the GPTST gap test routine. This will prevent tape run-out if read status for any reason is lacking during a gap test.

The DELAY subroutine generates a time delay according to a preset time base (SET10, strobe A12) and a multiplication factor stored in scratch pad positions S1 and A2. (See Paragraph 23.20).



| 0000 | 7100    | START  | OUT  | 9,00H  | RESET INPUT LOADED BIGNAL          |
|------|---------|--------|------|--------|------------------------------------|
| 0001 | 6600    |        | OUT  |        |                                    |
|      |         |        |      | 6,00H  | RESET MAIN STATUS REGISTER         |
| 0002 | 6400    | INTIM: | OUT  | 4,00H  | RESET WRITE CONTROL REGISTER       |
| 0003 | 6500    |        | OUT  | 5,00H  | RESET READ CONTROL REGISTER        |
| 0004 | 8406    | INCHC: |      |        |                                    |
|      |         | INCHC: | JIL  | SMOTN  | WAITING FOR NEW COMMAND            |
| 0005 | BC04    |        | JUN  | INCHC  |                                    |
| 0006 | 6601    | SMOTN: | OUT  | 6,01H  | ISET FORMATTER BUSY.               |
|      |         |        | 001  | 0,010  |                                    |
|      |         | 1      |      |        | INOTE THAT WE HERE ASSUME THAT WE  |
|      |         | ;      |      |        | MARE OPERATING IN NRZI MODE. IF IT |
|      |         | •      |      |        | THE OF MARKS THE HALL HODE, IF IT  |
|      |         |        |      |        | IS PE MODE; THIS WILL BE CORRECTED |
|      |         | i      |      |        | ILATER.                            |
| 0007 | 4000    |        | CLR  | Α      | , _ · · · · _ · · ·                |
| 0008 | 453F    |        |      |        | -141.014 41.00 0.00 1 1 1          |
|      |         |        | ORA  | 3FH    | MASK OUT BIT 6 & 7                 |
| 0009 | 5607    |        | INA  | A.7    | LOAD INPUT COMMAND REGISTER INTO   |
|      |         | ;      |      |        | ACCUMULATOR                        |
| 000A | 2800    | ,      | 0.17 |        |                                    |
|      |         |        | RUT  | 0      | STORE COMMAND INTO REGISTER 80     |
| 000B | 0601    |        | TAN  | 01H    | TEST UPON FORWARD MODE ( BIT 0 )   |
| 000C | 800E    |        | JZE  | WTEST  |                                    |
|      |         |        |      |        | REVERSE IF ZERO                    |
| 000D | BC12    |        | JUN  | NRZIT  | FORWARD MODE                       |
| 000E | 0602    | WTEST: | TAN  | 02H    |                                    |
| 000F | 8011    | W.C.   |      |        | TEST UPON WRITE MODE ( BIT 1 )     |
|      |         |        | JZE  | BOTST  | READ MODE IF ZERO                  |
| 0010 | BCOO    |        | JUN  | START  | WRITE REVERSE IS NOT ALLOWED       |
| 0011 |         | DOTOT- |      |        |                                    |
|      | 9400    | BOTST: | JLP  | START  | REVERSE FROM BOT IS NOT ALLOWED    |
| 0012 | 6609    | NRZIT: | OUT  | 6,09H  | SET MOTION                         |
| 0013 | 7 4 DO  | SET10: | OUT  | C.ODOH |                                    |
|      |         | OCTIV! |      |        | SET-UP FOR 10 MICROSECONDS TIMING  |
| 0014 | 8818    |        | JNZ  | NRZIM  | ;TEST UPON NRZI MODE               |
| 0015 | 6820    |        | SUT  | 3,20H  | SET PE MODE                        |
| 0016 | 6629    |        |      |        |                                    |
|      |         |        | OUT  | 6,29H  | SET CORRECT STATUS                 |
| 0017 | BC19    |        | JUN  | STAR1  |                                    |
| 001B | 6B00    | NRZIM: | SUT  |        | ACET MOTE MODE                     |
|      |         |        |      | 3,00Н  | SET NRZI MODE                      |
| 0019 | 6F00    | STAR1: | SUT  | 7,00H  | RESET TIMING REGISTER 0            |
| 001A | 7800    |        | SUT  | в,оон  | RESET TIMING REGISTER 1            |
| 001B | 7900    |        |      | 0,000  |                                    |
|      |         |        | SUT  | 9,00H  | RESET TIMING REGISTER 2            |
| 001C | 7A00    |        | SUT  | A,00H  | RESET FILE SEARCH REGISTER         |
| 001D | 7 CO 0  |        | SUT  |        | DECET DATE SAME                    |
|      |         |        |      | C,00H  | RESET BYTE COUNTER                 |
| 001E | 4000    |        | CLR  | Α      |                                    |
| 001F | 5504    |        | INO  | A,4    | LOAD STATUS PATTERN                |
| 0020 | 4680    |        |      |        |                                    |
| 0020 | 7000    |        | ANA  | 80H    | MASK OUT ALL BITS EXCEPT BIT       |
|      |         | ;      |      |        | :7 ( SINGLE STACK HEAD INDI-       |
|      |         | ;      |      |        | ;CATOR BIT                         |
|      |         | ,      |      |        |                                    |
| 0021 | 5 D 0 0 |        | INO  | 5,0    | ;LOAD COMMAND BYTE                 |
| 0022 | 2800    |        | RUT  | 0      | ;STORE COMMAND. BIT 7 NOW          |
|      |         |        |      | •      |                                    |
|      |         |        |      |        | ; INDICATES SINGLE/DUAL STACK HEAD |
| 0023 | 9428    | LPTST: | JLP  | NTEST  |                                    |
| 0024 | 6990    |        | SUT  | 1,90H  |                                    |
| 0025 | 6A01    |        |      |        | 1007 UD 540 / WELL 1000 / -        |
|      |         |        | SUT  | 2,01H  | SET-UP FOR 4 MILLISECOND DELAY     |
| 0026 | FCD5    | CDLAY: | CUN  | DELAY  |                                    |
| 0027 | BC9E    |        | JUN  | COMDC  | RETURN FROM DELAY                  |
| 0028 |         | NTEGT  |      |        | THE FORM FROM DELAT                |
|      | 4000    | NTEST: | CLR  | A      |                                    |
| 0029 | 4540    |        | ORA  | 40H    | SET BOT-BIT                        |
| 002A | 5 DO 0  |        | INO  |        |                                    |
|      |         |        |      | S,0    | FOR ACC WITH COMMAND REG.          |
| 002B | 2800    |        | RUT  | 0      | COMMAND REGISTER NOW CONTAINS      |
|      |         | ;      |      |        | ; INFORMATION ABOUT BOT            |
| 002C | 8832    | •      | JNZ  | DOAMC  |                                    |
|      |         |        |      | D20MS  | ;NRZI FROM BOT?                    |
| 002D | 7E09    |        | SUT  | E,09H  | SET READ ID COUNTER                |
| 002E | 6912    |        | SUT  |        |                                    |
| 002F |         |        |      | 1,12H  |                                    |
| 0026 | 6 A 0 2 |        | SUT  | 2,02H  | SET-UP FOR 5,3 MILLISECOND         |
|      |         | i      |      |        | DELAY. NOTE THAT WE DO NOT         |
|      |         |        |      |        |                                    |
|      |         | ;      |      |        | DISTINGUISH BETWEEN READ OR        |
|      |         | ;      |      |        | WRITE WHEN GOING FROM BOT          |
|      |         | ;      |      |        | IN PE MODE. THIS IS DONE           |
|      |         |        |      |        |                                    |
|      |         | i      |      |        | LATER IN THE PROGRAM               |
| 0030 | FCD5    |        | CUN  | DELAY  |                                    |
| 0031 | BC35    |        | JUN  |        |                                    |
|      |         |        |      | IDWRT  |                                    |
| 0032 | 69 DO   | D20MS: | SUT  | 1,0D0H |                                    |
| 0033 | 6A07    |        | SUT  | 2,07H  | SET-UP FOR FIRST PART OF PRE-      |
|      |         |        |      | _ ,    |                                    |
|      |         | ;      |      |        | DELAY WHEN GOING FROM BOT IN       |
|      |         | ;      |      |        | :NR7I MODE                         |

0034 BC26

23-2

00 → SB

Reset byte

count reg.

NO

BOT M5-LPD=1

(BOTST:

YES

START

 $(\mathbf{1})$ 

INITIALIZING

## 23.2 IDWRT SEQUENCE

The IDWRT sequence performs the reading and the writing of the PE ID Burst. The ID pattern consists of alternate "0"s and "1"s in the P-channel, all other channels being DC-erased. The ID Burst lasts for a period of 33 ms and is then followed by a 46.6 ms gap to the first data block. (Note: 75 ips values are used throughout.)

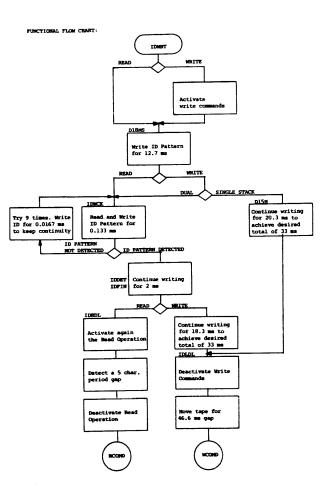
12.7 ms after the start of the ID operation, the written or read ID pattern is checked. If it is correct, the ID detected signal FCCG/ID is set true for 2 ms. If not, the program will try nine times before giving up, using SE to keep count.

There is no spacial read-only sequence. The program goes through the motions of writing, but without actually doing so, since the write strobes are not enabled.

The IDWT2 subroutine generates the "0"s and "1"s for the number of times that is stored in the Sl and S2 registers. The character period is preset by the Al2 strobe. (See Paragraph 23.21.) With the write strobe disabled (ETWDS=0) the IDWT2 is used to generate a delay of 4 msec.

In the read-only mode a subroutine, GPTST, is used to detect the gap between the ID Burst and the first data block. (See Paragraph 23.22.) The gap count value loaded into SB determines the number of 2.5 time periods which the gap should last.

The DUALR subroutine is used to condition the operation for reading unless the transport has a single stack head and is in the write mode. (See Paragraph 23.39.)

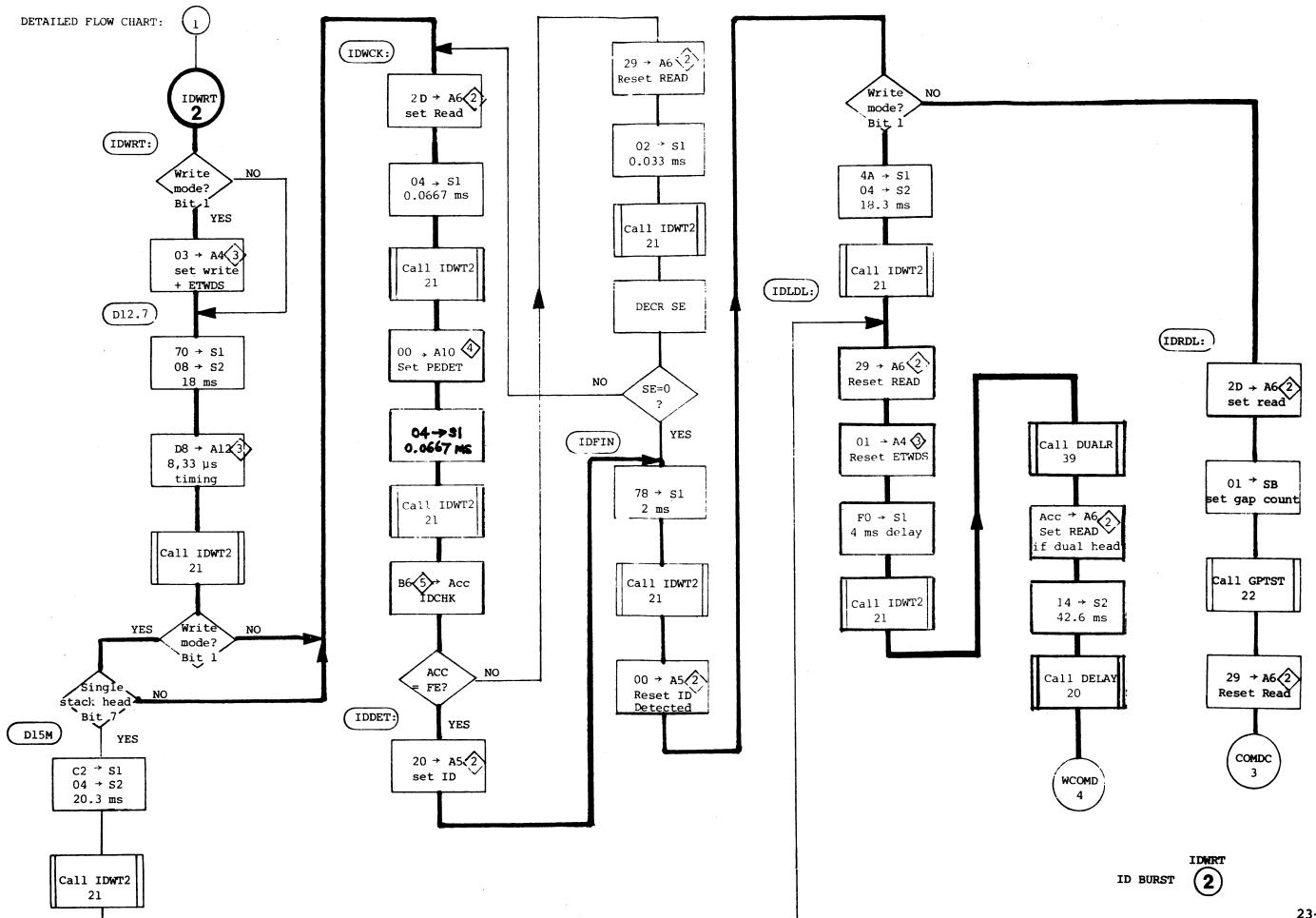


IDWRT: CLR INO TAN JZE 0035 0036 0037 0038 0039 003A 003B 003C 003D D12.7 ISET WRITE AND ETWOS 4,03H 1,0FAH 2,02H OUT SUT SUT D12.7: ISET-UP FOR 12,7 MILLISECONDS DELAY 74D8 FC48 5D00 C,0D8H IDWT2 ISET-UP FOR 8,33 MICROSECONDS TIMING OUT CUN INO TAN JZE TAN JZE JUN SUT SUT INE KNOW THAT ACC. IS ZERO NOW 8.0 IME KNOW THAT ACC. 18 ZERO NOW ITEST UPON WRITE MODE ( BIT 1 ) INO WRITE IF ZERO ITEST UPON BINGLE BTACK HEAD ISINGLE STACK HEAD IF ZERO IDUAL STACK HEAD 003F 0040 0041 0602 805D 0680 02H IDWCK 80H D20.3 IDWCK 0044 0045 0046 0047 69 C2 6A04 FC48 BC77 1,0C2H 2,04H IDWT2 IDLDL D20.3: SET-UP FOR 20,3 MILLISECONDS DELAY JUN OUT IDWCK 662D SET READ 005D 6,2DH SET-UP FOR 0,066 MILLISECOND 005E 6904 SUT 1,04H 005F FC48 0060 7200 CUN IDWT2 A,00H SET PEDET. THIS IS NECESSARRY BECAUSE THE READ ELECTRONICS #WILL NEVER ACCEPT A PE START #WITH ALL CHANNELS DC ERASED #EXCEPT THE P-CHANNEL. ;SET-UP FOR 0,066 MILLISECOND;WRITING 0061 6904 SUT 1,04H IDWT2 0062 FC48 ; WE KNOW THAT ACC IS ZERO ; LOAD ENVELOPE PATTERN INTO 0063 5506 INO A,6 ACCUMULATOR ;TEST IF INFO ONLY IN P-CHANNEL OK IF ZERO 0064 04FE 0065 8080 0066 6629 OFEH IDDET 6,29H TXR JZE OUT ;SET-UP FOR 0,033 MILLISECONDS ;WRITING 0067 6902 SUT 1,02H 0068 FC48 CUN INO IDWT2 ;LOAD ID READ COUNTER. WE KNOW THAT TACC IS ZERO HERE 0069 5D0E 006A 4200 SUB DECREMENT ACC NO ID-DETECTION AT ALL IF ZERO JZE RUT JUN SUT CUN 006B IDWCK TRY ONCE MORE 006D BC5D SET-UP FOR 2 MILLISECONDS WRITING 006E 6978 FC48 1,78H IDHT2 ; RESET ID DETECTED SIGNAL ( IF IT ; IS SET ). 0070 6500 OUT 5,00H INO TAN JZE SUT SUT 0071 5D00 0072 0602 TEST UPON WRITE MODE (BIT 1 ) 0073 0074 80 D0 694A SET-UP FOR 18.3 MILLISECONDS NRITING 0075 6AQ4 2,04H 0076 FC48 IDWT2 CUN 0076 FC48 0077 6629 0078 6401 0079 69F0 007A FC48 007B FCA9 007C 2600 007D 6A14 007E FCD5 007F BD1D RESET READ OUT OUT SUT CUN CUN AUT SUT CUN JUN RESET ETHOS
ISET 4 MILLISECONDS DELAY DUALR SET READ IF NOT SINGLE STACK 0080 6520 0081 BC6E 00D0 662D 00D1 7801 00D2 FC82 OUT JUN OUT 5,20H IDFIN IDDET: SET ID DETECTED 6,2DH SET READ SUT CUN OUT 00D3 6629 RESET READ TO CLEAR THE READ 6,29H 00D4 BC9E JUN COMDC

5,0 02H

(LOAD COMMAND (TEST UPON WRITE MODE ( BIT 1 )

23 - 4

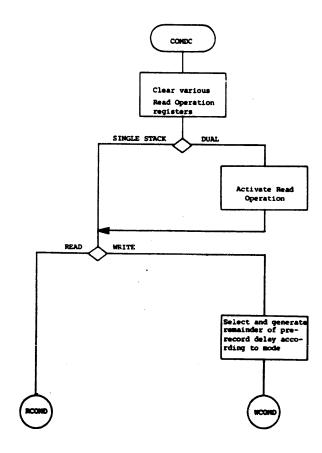


## 23.3 COMDC SEQUENCE

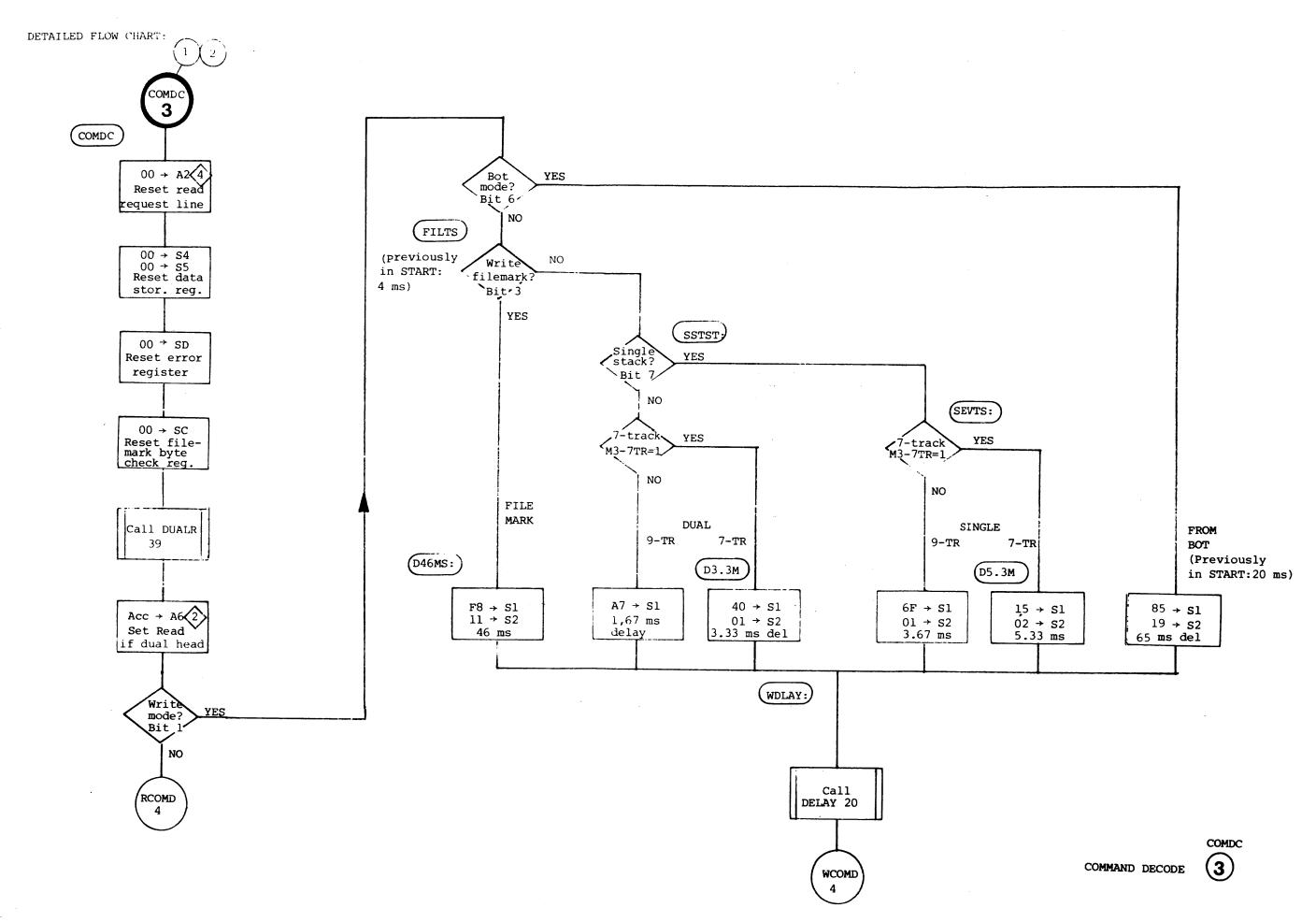
The COMDC sequence decodes commands and status signals and generates the remainder of the pre-record delay according to the decoded mode of operation.

The read strobe flip-flop (U25,4) is reset by the A2 just in case it has been set by noise in the start-up phase.

FUNCTIONAL FLOW CHART:



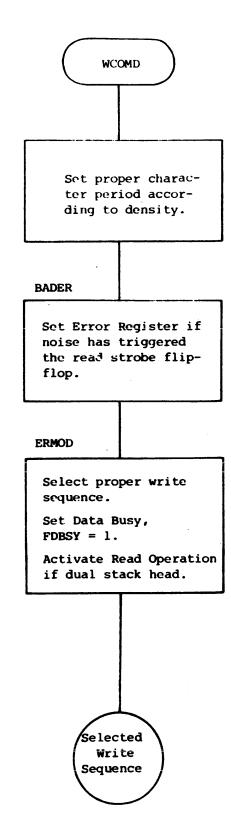
| 009E  | 6200          | COMDC: | OUT   | 2,00H    | IRESET READ REG.   |
|-------|---------------|--------|-------|----------|--|
| 009F  | 4 C00         |        | SUT   | 4,00H    | RESET DATA REGISTER 1  |
| 00A0  | 4 D0 0        |        | SUT   | 5,00H    | RESET DATA REGISTER 2  |
| 00A1  | 7000          |        | SUT   | D,00H    | RESET NAZI ERROR REGISTER  |
| 00A2  | 7 CO 0        |        | PUT   | C,00H    | IRESET NRZI FILEMARK BYTE  |
|       |               | ı      |       | -        | COUNT REGISTER   |
| 00A3  | FCAE          |        | CUN . | DUALR    |  |
| 00A4  | 2400          |        | AUT   | 6        | ISET READ IF NOT SINGLE & MRITE  |
| 00A5  | 4000          |        | CLR   | A        |  |
| 00A6  | 5000          |        | INO   | 5,0      |  |
| 00A7  | 0602          |        | TAN   | 02H      | ITEST UPON WRITE MODE ( BIT 1 )  |
| BACO  | 8120          |        | JZE   | RCOMD    | IREAD ONLY IF ZERO   |
| OOAP  | 0640          |        | TAN   | 40H      | ITEST UPON BOT MODE ( BIT 6 )  |
| OOAA  | 8088          |        | JZE   | FILTS    | INO BOT START-UP IF ZERO   |
| BAOO  | 4985          |        | SUT   | 1,85H    |  |
| DOAC  | 6A19          |        | SUT   | 2,19H    | ISET-UP FOR 65 MILLISECONDS  |
|       |               | ī      |       | -        | IDELAY = LAST PART OF PREDELAY   |
| OOAD  |               | 1      |       |          | INHEN STARTING FROM BOT  |
|       | B CBF         |        | JUN   | HDLAY    |  |
| 00BB  | 0608          | FILTS: | TAN   | 08H      | ITEST UPON WRITE FILEMARK (BIT 1)  |
| 00BC  | BOC1          |        | JZE   | SSTST    | INO FILENARK IF ZERO   |
| OOBD  | 69F8          |        | SUT   | 1.0FBH   |  |
| OOBE  | 6A11          |        | SUT   | 2,11H    | ISET-UP FOR 46 MILLISECONDS DELAY  |
|       |               | ï      |       | •        | IFILEMARK DELAY.   |
| OOBF  | FCD5          | WDLAY: | CUN   | DELAY    |  |
| 00 C0 | <b>B</b> D1 D |        | JUN   | MCOMD    |  |
|       |               |        |       |          |  |
| 00 C1 | 0480          | SSTST: | TAN   | 80H      | ITEST UPON SINGLE MODE ( BIT 7 )   |
| 00 C2 | 80 C9         |        | JZE   | SEVTS    | ISINGLE STACK HEAD IF ZERO   |
| 00 C3 | ■ CC4         |        | J7T   | D3.3M    |  |
| 00C4  | 69A7          |        | SUT   | 1.0A7H   | SET-UP FOR 1,67 MILLISECONDS DELAY   |
| 00C5  | BCBF          |        | JUN   | WOLAY    |  |
| 00 C6 | 694D          | D3.3M: | SUT   | 1 . 4 DH |  |
| 00C7  | 6A01          |        | SUT   | 2,01H    | ISET-UP FOR 3,33 HILLISECONOS DELAY  |
| 00 CB | 8 CBF         |        | JUN   | HOLAY    |  |
| 00 C9 | BCCD          | SEVTS: | J7T   | D5.3M    |  |
| 00CA  | 696F          |        | SUT   | 1,6FH    |  |
| 00 CB | 6A01          |        | SUT   | 2.01H    | SET-UP FOR 3,47 MILLISECONDS DELAY   |
| 00 CC | BCBF          |        | JUN   | HOLAY    | The state of the s |
| 00CD  | 4715          | D5.3M: | SUT   | 1.15H    |  |
| 00 CE | 6 <b>A</b> 02 |        | SUT   | 2,02H    | ISET-UP FOR 5,33 MILLISECONDS DELAY  |
| 00 CF | 8 CBF         |        | JUM   | HOLAY    |  |



## 23.4 WCOMD SEQUENCE

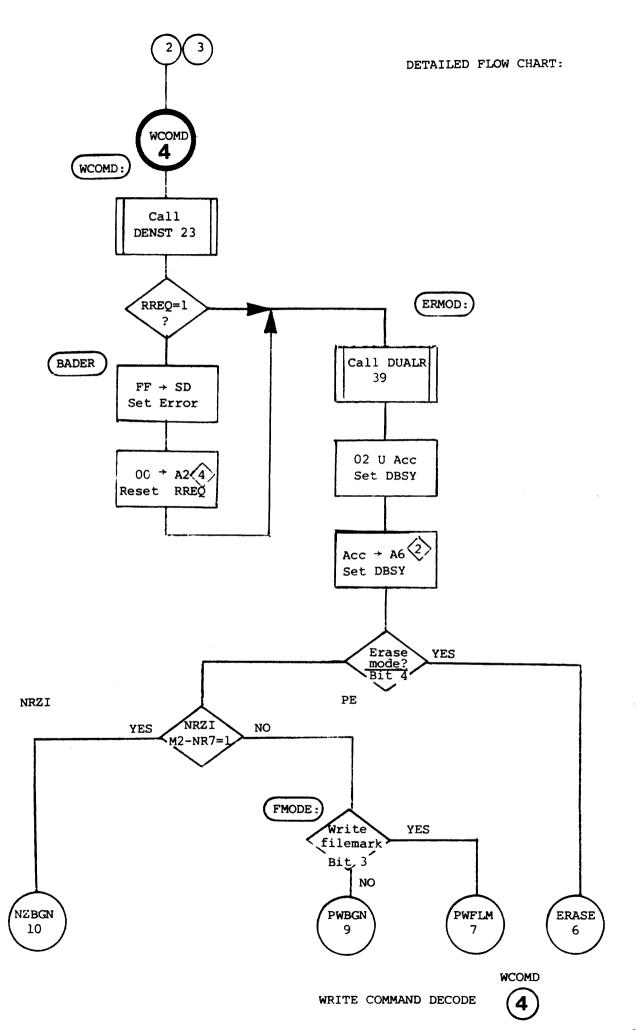
The WCOMD sequence decodes the write command and selects the proper write sequence to follow. The pre-record delay was completed in the COMDC sequence.

The DENST subroutine selects proper timer period (Al2, 3) and post-record delay according to pre-selected density. (See Paragraph 23.23.) FUNCTIONAL FLOW CHART:



23-8

| 0110 | FCEB   | WCOMD:  | CUN | DENST  | SET CORRECT DENSITY             |
|------|--------|---------|-----|--------|---------------------------------|
| 011E | 992A   |         | JRR | BADER  | ; IF TRUE = BAD ERASE           |
|      | FCAE   | ERMOD:  | CUN | DUALR  | ,                               |
| 011F |        | Emilion |     | 02H    | SET DOSY LINE                   |
| 0120 | 4502   |         | ORA |        |                                 |
| 0121 | 2400   |         | AUT | 6      | SET DOSY STATUS                 |
| 0122 | 4000   |         | CLR | A      |                                 |
| 0123 | 5 DO 0 |         | INO | 5.0    |                                 |
|      |        |         |     |        | :TEST UPON ERASE MODE ( BIT #4) |
| 0124 | 0610   |         | TAN | 10H    |                                 |
| 0125 | 8138   |         | JZE | ERASE  | FERASE MODE IF ZERO :           |
| 0126 | BAOB   |         | JNZ | NZBGN  | INRZI MODE?                     |
| 0127 | 040B   |         | TAN | 08H    | ITEST UPON FILEMARK ( BIT 3 )   |
|      |        |         |     | PMBGN  | INO FILEMARK WRITING IF ZERO    |
| 0128 | 818D   |         | JZE |        |                                 |
| 0129 | 8069   |         | JUN | PWFLM  | IPE WRITE FILEMARK OPERATION    |
|      |        |         |     |        |                                 |
| 012A | 7 DFF  | BADER:  | SUT | D,OFFH | SET HARD ERROR REGISTER         |
| 0128 | 4200   |         | OUT | 2.00H  | RESET RREG LINE                 |
| 0120 | ADIF   |         | JUN | ERMOD  |                                 |
|      |        |         |     |        |                                 |



23-9

## 23.5 RCOMD SEQUENCE

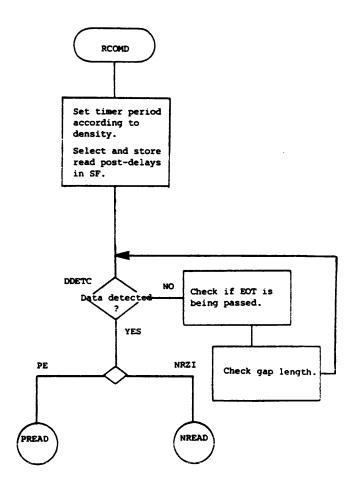
The RCOMD sequence decodes the read command and selects the proper read sequence. Furthermore, the read post-record values are stored in SF using the DENST subroutine.

While the program is waiting for the first read data, the possible appearance of EOT is being checked. If detected, the Search Indicator (SA) is set. (See Paragraph 23.25 on the EOTCH subroutine.)

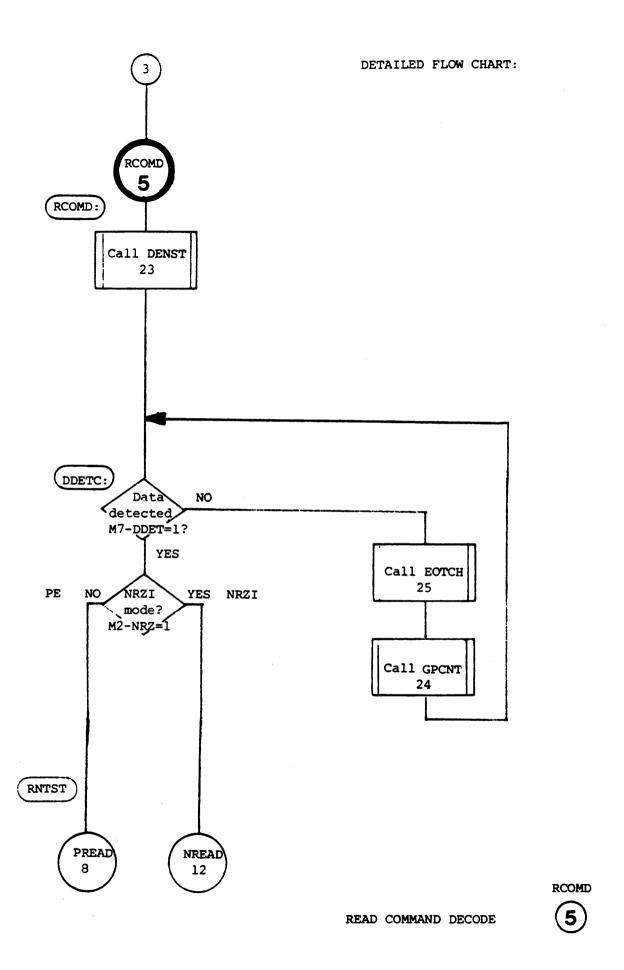
The gap length is also being checked, using subroutine GPCNT. This subroutine specifies a certain maximum allowable length (period) beyond which it will stop tape motion and return to the START sequence. The length is for the standard TDF 4050 selected to be 4.37 meters, but can be reprogrammed if desired, see Paragraph 23.24. (Whereas the GPCNT subroutine is used to check that a gap does not exceed a certain maximum value, the use of the GPTST subroutine can be said to ascertain a gap of a certain minimum value.)

Note that by returning the program to the START sequence when the gap period is exceeded, there will be no hang-up necessitating a master clear signal to continue operation. As soon as the FBSY line goes false, the formatter will be ready for a new command. This also applies when the tape drive stops tape motion in reverse at BOT, since it is the gap travel time which is checked, not the actual gap length.

FUNCTIONAL FLOW CHART:



| 0120<br>012E | FCE8<br>9032 | RCOMD:<br>DDETC: | JDT | DENST | SET CORRECT DENSITY  |
|--------------|--------------|------------------|-----|-------|----------------------|
| 012F         | FD34         |                  | CUN | EOTCH | -                    |
| 0130         | FD4B         |                  | CUN | GPCNT | TCALL GAP COUNT TEST |
| 0131         | BD2E         |                  | JUN | DDETC |                      |
| 0132         | BATE         | RNTST:           | JNZ | NREAD | INRZI READ?          |
| 0133         | BOEB         |                  | JUN | PREAD | IPE READ OPERATION   |



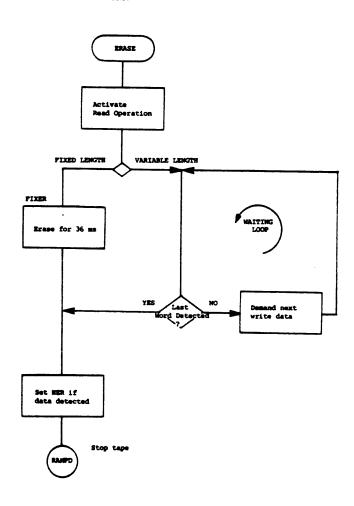
## 23.6 ERASE SEQUENCE

The ERASE sequence results in an erase operation by forwarding the write command (FWRT/TSWS) to the tape transport, and by omitting to generate write strobes (TWDS).

There are two types of erase operations. The first one is a fixed length (2.7 inch) type operation, the second one a variable type which keeps erasing until it is terminated by the Last Word signal FLWD. The (F)WFM command signal selects type of operation.

The FLWD signal is only transmitted on demand, by the FDWDS signal. The latter signal must therefore be continuously repeated, and at the proper synchronous rate. This rate is at the double timing counter rate wherefore the timing latch (U44,3) is set twice in the waiting loop.

PUNCTIONAL FLOW CHART:



| 0138   | 0408  | ERAGE:  | TAN | 08H     | (TEST UPON WFH HOOE ( BIT 3 )       |
|--------|-------|---------|-----|---------|-------------------------------------|
|        |       | •       |     |         | INE KNOW SO IS IN ACCUMULATOR HERE. |
| 0137   | 8140  |         | JZE | VAREI   | VARIABLE ERAGE IF ZERO              |
| 013A   | 74EC  | FIXER   | OUT | C.OECH  | ISET 4,147 MICROSECONDS TIMING      |
| 0138   | 49 CO |         | SUT | 1.0 COH | the three properties that           |
| 013C   | 6A21  |         | SUT | 2,21H   | 1057-10 COR 1/ MILL 1557-100 CO     |
| 013D   | FC05  |         | CUN | DELAY   | 19ET-UP FOR 36 MILLISECONDS DELAY   |
| 013E   | 7577  | ERAST:  |     |         |                                     |
|        |       | EMMO!   | JDT | HARDE   | IBAD ERABE IF JOT-1.                |
| 01JF   | 9F98  |         | JUN | DREND   |                                     |
|        |       | •       |     |         |                                     |
| 0140   | FD05  | VARE1:  | CUN | COMMIT  | IDUNTY WRITE TO AVOID TIME RACE.    |
|        |       | 1       |     |         |                                     |
| 0141   | A149  | VARER:  | JLH | WREQ5   | ILAST WORD?                         |
| 0142   | 6404  |         | OUT | 4.04H   | ISET EFWOR                          |
| 0143   | 9143  | UREQ1:  | JNT | UREQI   | HAIT FOR TIME INDICATOR             |
| 0144   | 7500  |         | OUT | D.00H   | RESET TIME INDICATOR                |
| 0145   | 6400  |         |     |         |                                     |
|        |       |         | OUT | 4 ,00H  | RESET EFHOS                         |
| 0146   | 7146  | WREQ4 : | JNT | WREQ4   | IMAIT FOR TIME INDICATOR            |
| 0147   | 7500  |         | OUT | D.00H   | RESET TIME INDICATOR                |
| 0148   | BD41  |         | JUN | VARER   | IONCE HORE!                         |
| 0149   | 9149  | UREQ5 : | JNT | HREQ5   | HAIT FOR TIME INDICATOR             |
| 014A   | 803E  |         | JUN | ERAST   | IMMTI LOW ITUE THOTONION            |
| ** *** | 4036  |         | 300 | ERADI   |                                     |

DETAILED FLOW CHART: ERASE ERASE:  $00 \rightarrow A4,3$ Reset EFWDS NO YES WFM WREQ4: = 1? Time Bit 3 NO latch set M4-TIME= 1 VAREL YES EC + A12 3  $00 \rightarrow A13,3$ Call COMWT 4.167 usec Reset time 26 timing latch FIXER: VARER: C0 → S1 21 → S2 36 ms delay NO YES Last word? M8-LWD=1 for PE WREQ5:  $04 \rightarrow A4,3$ Call DELAY NO set FFWDS 20 ĺatch set M4-TIME=1 ERAST: YES (WREQ3: Time NO latch set YES Data Detected M4-TIME=1 <m7-DDET=1 YES  $00 \rightarrow A13,3$ NO Reset time latch HARDE DREND 16 **ERASE** 

**ERASE OPERATION** 

#### 23.7 PWFLM SEQUENCE

The PWFLM sequence writes a PE File Mark pattern. This pattern consists of 40 "0"s in channels P,0,2,5,6 and 7 with channel 1,3 and 4 DC-erased, and is IBM compatible.

A subroutine COMWT is used to generate the phase transition part of the "0" data signals. In actuality, in order to obtain the correct polarity imprinted on the tape, the program must generate "1"s, the opposite polarity. The COMWT routine synchronizes these "1"s and then generates synchronized "0"s for the phase transition part. (COMWT, see Paragraph 23.26.)

Synchronization to the required bit rate is accomplished by having the program wait for the bit rate time latch to be set before going further.

## 23.8 PREAD SEQUENCE

The PREAD sequence performs the normal PE read test operation. Note that whereas the NRZI read operation mostly lies in software, the PE read operation mostly employs special hardware (Schematics no. 4 and 5).

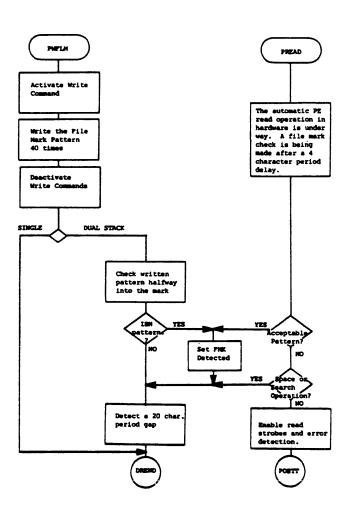
Because of the 8-bit limitation of standard available registers, the 9-bit read words are broken up in two parts and handled one after the other.

Reading and transfer of PE read data in hardware start automatically as soon as a data block has been detected (M7-DDET = 1) in the RCOMD sequence.

The program in the meantime checks to see if the received data is a file mark. If not, it proceeds to the POSTT sequence, waiting for the postamble to be detected (M14-POSTD).

In cheking for a filemark in the read mode, the program allows for 8 possible file marks, whereas in the write mode (see PWFLM) only the IBM standard file mark is acceptable. Other standards allow erasure in any of tape tracks 1, 4 and 7. Therefore, the PREAD sequence ignores the corresponding 5, P and 0 bits.

FUNCTIONAL FLOW CHART:



23-15

DBSY

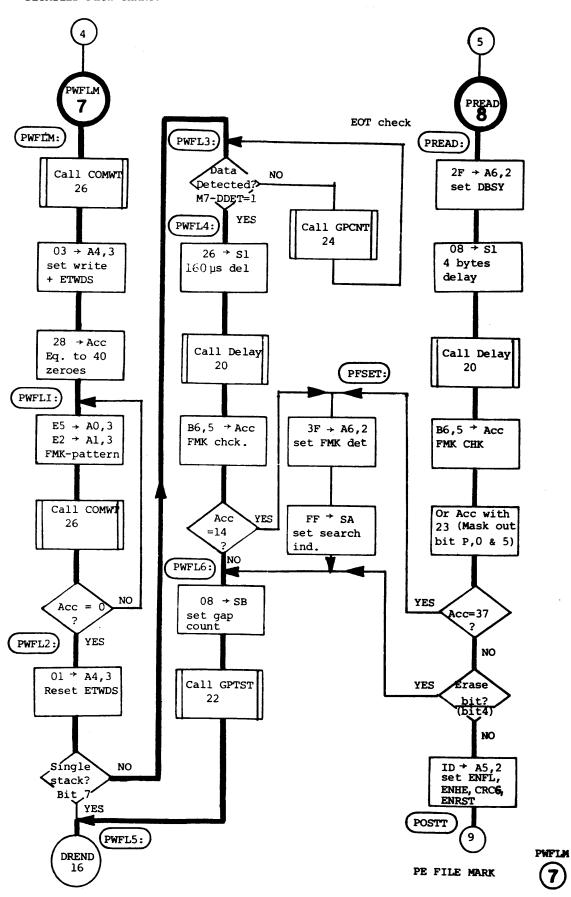
| Page   | Erroneous Text                                    | Correct Text  |
|--------|---|---------------|
| 6-7    | FDWDS   | FDWDS         |
|        | FLWD  | FLWD          |
| 22-8   | DB3   | DB3<br>SO WFM |
| 22-14  | 1000  08<br>.   .<br>.   .<br>1111  0F (8 places) | 0000  00      |
| 23-5   | 18 ms   | 12,7 ms       |
| 23-9   |   | YES           |
| 23-9   | DBSY  | FDBY          |
| 23-12  | Activate<br>Read Operation                        |               |
| 23-1.3 | EFWDS FFWDS ,3 (4 places)                         | EFDWS FFDWS   |

**FDBY** 

| 0169 | FD85   | PWFLM:      | CUN .      | COMUT       | ;THIS IS A DUMMY WRITE OPERATION,<br>;WHICH IS NECESSARY TO AVOID TIMING<br>;RACE. OF COURSE, NO DATA IS WRITTEN<br>;ONTO TAPE DURING THIS OPERATION. |
|------|--------|-------------|------------|-------------|---|
| 016A | 6403   |             | OUT        | 4,03H       | SET WRITE AND ETWOS   |
| 016B | 4000   |             | CLR        | A           |   |
| 016C | 4528   |             | <b>ORA</b> | 28H         | THIS IS EQUIVALENT WITH 40 ZEROES   |
| 016D | 60E5   | PWFL1:      | OUT        | 0,0E5H      |   |
| 016E | 61E2   | ;           | OUT        | 1,0E2H      | ;LOAD FILEMARK PATTERN INTO WRITE<br>;REGISTER  |
| 016F | FD85   |             | CUN        | COMMT       | START WRITING   |
| 0170 | 0500   |             | TOR        | 0 <b>0H</b> | #OPERATION COMPLETED?   |
| 0171 | 8173   |             | JZE        | PWFL2       | OPERATION FINISHED WHEN ZERO  |
| 0172 | BD6D   |             | JUN        | PWFL1       | CONCE MORE!   |
|      |        | i           |            |             |   |
| 0173 | 6401   | PWFL2:      | OUT        | 4,01H       | RESET ETWOS   |
| 0174 | 5000   |             | INO        | 5,0         | ;WE KNOW ACC IS ZERO HERE   |
| 0175 | 0880   |             | TAN        | 80H         | (TEST UPON SINGLE MODE ( BIT 7 )  |
| 0176 | 8181   |             | JZE        | PWFLS       | SINGLE STACK IF ZERO  |
| 0177 | 9 D7 A | PWFL3:      | JOT        | PWFL4       | ;WAITING FOR DATA   |
| 0178 | FD4B   |             | CUN        | GPCNT       |   |
| 0179 | BD77   |             | IUN        | °₩FL3       |   |
| 017A | 6926   | PWFL4:      | UT         | 1,26H       | SET-UP FOR 160 MICROSECONDS DELAY   |
| 0178 | FCD5   | ;           | CUN        | DELAY       | THIS IS TO AVOID READ PROBLEMS AT THE BEGINNING OF THE FILEMARK PATTERN   |
| 017C | 5506   |             | INO        | A,6         | ;LOAD FILEMARK REGISTER INTO ACC. WE  |
|      |        | ;           |            |             | KNOW THAT ACC. IS ZERO HERE.  |
| 017E | 0414   |             | TXR        | 14H         | CORRECT FILEMARK PATTERN?   |
| 017E | 8182   |             | JZE        | PFSET       | CORRECT FILEMARK PATTERN IF ZERO  |
| 017F | 7808   | PWFL6:      | SUT        | 8,08H       | SET GAP COUNTER   |
| 0180 | FCB2   |             | CUN        | GPTST       |   |
| 0181 | 8F98   | PWFL5:      | JUN        | DREND       |   |
| 4100 |        | j<br>DEGET. |            |             |   |
| 0182 | 663F   | PFSET:      | OUT        | 6,3FH       | SET FILEMARK DETECTED SIGNAL  |
| 0183 | 7 AFF  |             | SUT        | A,0FFH      | LUAD SEARCH INDICATOR REGISTER  |
| 0184 | BD7F   |             | JUN        | PWFL6       |   |
|      |        |             |            |             |   |

| 01EB | 662F   | PREAD: | OUT | 6,2FH | SET DOSY                         |
|------|--------|--------|-----|-------|----------------------------------|
| 01EC | 6908   |        | SUT | 1.08H | SET 4 BYTES DELAY                |
|      |        | ;      |     | - •   | THIS DELAY IS PUT IN HERE        |
|      |        |        |     |       | TO ENSURE THAT TE DON'T          |
|      |        | •      |     |       | TRY TO READ RIGHT AT THE         |
|      |        |        |     |       | BEGINNING OF THE FILEMARK        |
| 01ED | FCD5   | •      | CUN | DELAY | ,                                |
|      | 5506   |        | INO | A,6   | :LOAD FILEMARK REGISTER          |
| 01EF | 4523   |        | ORA | 23H   | MASK OUT BIT P.O AND 5           |
| 01F0 | 0437   |        | TXR | 37H   | TEST IF WE HAVE GOT ONE OF 8     |
|      |        | ;      |     |       | POSSIBLE FILEMARKS PATTERN.      |
|      |        | į      |     |       | REMEMBER, THIS IS A READ ONLY    |
|      |        | ;      |     |       | ;OPERATION.                      |
| 01F1 | 8182   |        | JZE | PFSET | FILEMARK DETECTED IF ZERO        |
| 01F2 | 4000   |        | CLR | A     |                                  |
| 01F3 | 5 DO 0 |        | INO | 5,0   |                                  |
| 01F4 | 0610   |        | TAN | 1 Ó H | ;TEST UPON ERASE MODE ( BIT #4 ) |
|      |        | ;      |     |       | ; IF ERASE MODE IS TRUE; WE ARE  |
|      |        | ;      |     |       | ;EITHER DOING A SPACE OR FILE    |
|      |        | ;      |     |       | ;MARK ( IGNORE DATA ) SEARCH.    |
|      |        | ;      |     |       | ; IN BOTH CASES; NO ERROR        |
|      |        |        |     |       | DETECTION OR READ STROBE GENE-   |
|      |        | ;      |     |       | RATION ARE ALLOWED.              |
| 01F5 | 817F   |        | JZE | PWFL6 | SPACE OR FILE SEARCH ( IGNORE    |
|      |        | ;      |     |       | ;DATA ) IF ZERO.                 |
| 01F6 | 651D   |        | OUT | 5,1DH |                                  |
| 01F7 | BD06   |        | JUN | POSTT | JUMP TO POSTAMBLE TEST           |

# DETAILED FLOW CHART:



PE READ TEST



# 23.9 PWBGN SEQUENCE

The PWBGN sequence performs the normal PE writing operation.

A subroutine PREMW, is utilized to write the 40 postamble "0"s. The same subroutine is later used to write the 40 postamble "0"s. The first part of this subroutine writes the phase transitions, and the second part, PMIDL, writes the data transitions, see paragraph 23.7. When writing the preamble, the program enters the subroutine at the PMIDL point, starting as appropriate with a data transition.

The COMWT subroutine (Paragraph 23.26) is used to write the preamble all "1" byte.

Note that the inverse polarity is used, as was the case when writing file marks. This is the polarity needed to obtain a polarization of the tape medium in agreement with international standards. Doing the inversion in software saves hardware.

In writing data in the PE format, a phase transition shall be inserted if the next bit is of the same polarity. The next bit is therefore made available at this time in order to determine whether or not to insert a phase transition.

To generate the phase transitions, however, a simple scheme is used which avoids the use of Exclusive-Or operations or similar logic. This scheme consists of trying to write, at an instant proper for a phase transition, a bit having a polarity inverse of that of the next bit.

Then, if the next bit is of the same polarity as the present bit, the phase transition bit becomes an actual phase transition since its polarity then is the opposite of the present one. If, on the other hand, the next bit is of the opposite polarity of that of the present bit, the phase transition bit will have the same polarity as the present bit, and no flux transition will ensue on the tape. This will thus conform to the requirements of the PE format.

Since in practice we have to invert all data bits in software, there is no need to perform an extra inversion to obtain the phase bits: they can be used in the true from in which they appear.

Thus, the program generates phase transitions by feeding the transport the next bits by their original polarity, and then on the next clocking generates data transitions by inverting the same bits.

The write data operation begins with the PWBYT sequence in which data are fetched from the BO, Bl registers and transferred to the AO, Al registers and S4, S5 registers via the accumulator. A subroutine, DATWT, is used to synchronize the phase transition bits now stored in the AO, Al register, and then on the next clocking to transfer the data bits from S4, S5 to AO, Al in inverted form. The subroutine generates the FDWDS demand data strobe during the phase transition. (DATWT, see Paragraph 23.28).

This sequence is repeated until the LWD last word line goes true, after which the postamble is generated. As soon as the read circuitry detects data, the error detection circuits are enabled.

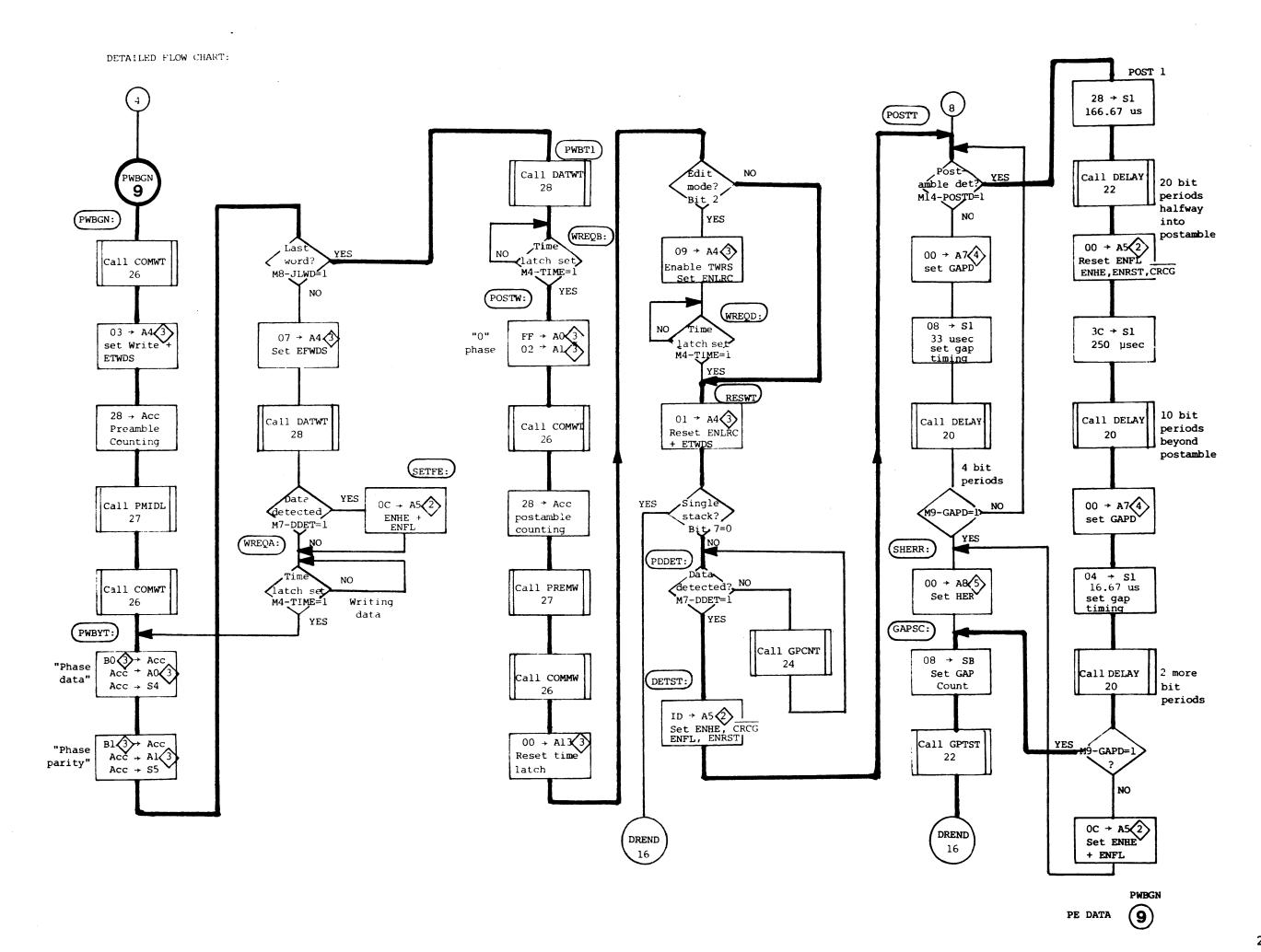
The COMWT subroutine writes the postamble all "1"s byte and the PREMW follows to write the 40 postamble all "0"s bytes. This time there is no need to start in the PMIDL. However, since the last part of the PREMW ends with a data transition, it becomes necessary to add a phase transition. To this end the last part of the COMWT subroutine is used, starting at COMMW.

To secure a return to the proper gap polarization of the tape, a TWRS signal is generated which resets all the write flip-flops on the tape transport. Normally, however, they will already have been put in the reset state by the COMMW operation. (In Edit mode we skip this provision).

Note that timing in the PE write operation is so critical that there can be no extra program steps in going from the PWBGN sequence into the PWBYT sequence.

| PUNCTIONAL FLOW CHART:   | 018D<br>018E   | FD05          | PUBGNI      | CUN        | 4,03H           | ITHIS IS A DUMMY WRITE OPERATION TO (AVOID TIMING PROBLEMS. NO DATA IS INRITTEN. ISET WRITE AND ETHOS   |
|--|----------------|---------------|-------------|------------|-----------------|---|
|  | 018F           | 4000          |             | CLR        | A T             | · - ··- <del>-</del>  |
|  | 0190           | 4528          |             | ORA        | 28H             | ITHIS IS EQUIVALENT WITH THE 40 ZEROEB<br>FOR THE PREAMBLE  |
| PHRIGH   | 0191           | FDAA          |             | CUN        | PMIDL           | INE HAVE TO MAKE A CALL TO THE MIDDLE IPART OF WRITE PREAMBLE SUB-ROUTINE ITO MAKE A CORRECT START OF THE IPREAMBLE   |
|  | 0192           |               | •           | CUN        | COMMIT          | SWRITE DATA ALL ONES  |
| Activate Write Commands  | 0193<br>0194   | 4000<br>5500  | PMBYT:      | CLR<br>INO | A,0             | ;LOAD INPUT REGISTER 0 INTO ACC. THIS<br>;REGISTER CONTAINS THE DATA PATTERN<br>;IN TRUE FORM, HHICH HEAMS THAT ME<br>;CAN JUST TRANSFER THE BYTE TO THE<br>;OUTPUT REGISTER INVERE THE DATA LINES  |
| Write 40 Preamble<br>Zeros   |                |               | ;           |            |                 | ; ARE INVERTED ONCE MORE. THIS WILL<br>; THEN GENERATE THE PHASE TRANSITION.  |
| COMWT Write 1 Preamble ones  | 0195<br>0196   | 2000<br>2C00  | ;           | AUT<br>RUT | 0               | LOAD PHASE TRANSITION   STORE DATA TO ENABLE WRITING THE   DATA TRANSITION LATER ON.  |
| Enable Error Det.  | 0197           | 4000          | •           | CLR        | Α.              |   |
| PWBYT Last NO  Circuitry if Data Detected Write PE Data                    | 0198           | 5501          | ;           | INO        | A,1             | ;LOAD INPUT REGISTER 1 INTO ACC. THIS ;REGISTER CONTAINS THE PARITY BIT ;IN TRUE FORM, MHICH MEANS THAT ME CAN ;JUST TRANSFER THE BIT TO THE OUTPUT ;REGISTER MHERE IT HILL BE INVERTED ;ONCE MORE. THIS GENERATE THE PARITY ;PHASE TRANSITION. |
| word coming? PWBT1 YES   | 0199<br>019A   | 2100<br>2000  | •           | AUT<br>RUT | 1<br>5          | ;LOAD PARITY PHASE TRANSITION<br>;STORE PARITY BIT TO ENABLE WRITING  |
|  | 0198           | A1A3          | ;           | JLW        | PWBT1           | THE PARITY "DATA" TRANSITION LATER ON.  |
| Write Last Word  | 019C           | 6407          |             | OUT        | 4,07H           | SET EFHOS. THIS IS NOT THE LAST WORD.   |
| POSTW  | 019D<br>019E   | FD85<br>90A1  |             | JDT        | DATWT<br>SETFE  | ;CALL DATA WRITE SECTION<br>;DATA DETECTED?   |
|  | 019F           | 919F          | WREGA:      | JNT        | WREGA           | ;WAITING FOR THE DATA TRANSITION TO<br>;BE WRITTEN  |
| Write 1 postamble<br>ones  | 91 <b>A</b> 0  | BĐ93          |             | JUN        | PWBYT           | ONCE MORE   |
|  | 01A1           | 631D          | SETFE:      | OUT        | 5,1DH           | ;DATA IS DETECTED. SET ENHE,<br>;ENRST, ENFL AND *CRCG.   |
|  | 01A2           | 809F          | •           | JUN        | HRENA           | iciansi, ciare may rened.   |
| Write 40 post-<br>amble zeros  | 01A3           | F085          | PWBT1:      | CUN        | DATHT           | ;WRITE LAST DATA BYTE   |
|  | 01 44          | 91A4          | WREGB:      | JNT        | WREGE           | HAITING FOR THE LAST DATA BYTE TO BE WRITTEN  |
| RESWI  | 01 A5<br>01 C0 | BDC0<br>60FF  | POSTH:      | JUN<br>OUT | POSTW<br>0,0FFH |   |
| Deactivate<br>Write Commands   | 01 C1<br>01 C2 | 6102<br>FD85  |             | OUT        | 1,02H<br>COMMT  | ¡LOAD PHASE TRANSITION  |
|  | 01 C3<br>01 C4 | 4000<br>4528  |             | CLR        | A<br>28H        | ;ALL ONES BYTE HRITTEN<br>;THIS IS EQUIVALENT TO THE 40   |
| SINGLE DUAL STACK PPDET  |                |               | ;           |            | PREMI           | ZEROES OF THE POSTAMBLE.  |
| Check presence<br>and length of<br>postamble and de-<br>tect a 22.5 period | 01 C5<br>01 C6 | FDA6<br>FDB7  | ;           | CUN        | COMPA           | WRITE LAST TRANSITION. NOTE THAT ME HAVE TO JUMP INTO THE MIDDLE OF THE COMMON WRITE SECTION  |
| gap,   | 01 C7          | 7500          | ŧ           | OUT        | D,00H           | ;TO GENERATE THE CORRECT TRANSITION.<br>RESET TIME INDICATOR  |
| Terminate  | 01 CB<br>01 C9 | 4000<br>5000  |             | CLR<br>INO | ŝ,o             |   |
| (DREND)  | 01 CA<br>01 CB | 0604<br>81 CE |             | JZE        | 04H<br>RESWT    | (TEST UPON EDIT MODE ( BIT 2 )  |
|  | 01 CC<br>01 CD | 6409<br>91 CD | WREQD:      | OUT<br>JNT | 4,09H<br>WREQD  | ;EMABLE WARS GENERATION<br>;WAITING FOR THE WARS SIGNAL   |
|  | 01 CE          | 6401          | RESHT:      | OUT        | 4,01H           | RESET ENHARS AND ETHOS. THE MRITE SIGNAL IS KEPT TRUE, TO FENABLE THE USE OF THE CORRECT TIMING PROMS DURING THE READ PROCESS.  |
|  | 01 CF<br>01 D0 | 0680<br>8398  | -           | TAN<br>JZE | 80H<br>DREND    | TEST UPON SINGLE MODE ( BIT 7 )   |
|  | 01 D1<br>01 D2 | 9004<br>FD4B  | PDDET:      | JDT<br>CUN | DETST           | MAITING FOR THE FIRST DATA BYTE TO ARRIVE   |
|  | 01 D3<br>01 D4 | BDD1<br>651D  | DETST:      | OUT        | PDDET<br>5,1DH  | ;DATA IS DETECTED. SET ENHE,<br>;ENRST, ENFL AND *CRCG  |
|  | 0105           |               | j<br>Boetti | JUN        | POSTT           | •   |
|  | 01D6<br>01D7   | 6700          | POSTT:      | JPD        | POST1<br>7,00H  | POSTAMBLE DETECTED?   |
|  | 01D9           | 6908<br>FCD5  |             | SUT<br>CUN | 1,08H<br>DELAY  | SET SHORT DELAY   |
|  |                | A5DC          | ;           | J60        | SHERR           | ;IF TRUE; GAP HAS BEEN DETECTED<br>;HITHOUT POSTAMBLE DETECTED. THIS<br>;IS A HARD ERROR ( TOTAL<br>;DROPOUT OR HISSING POSTAMBLE)  |
|  | 01 DC          | 8006<br>7000  | SHERR:      | OUT        | POSTT<br>8,00H  | ONCE HORE!  |
|  |                | 7808<br>FC82  | GAPSC:      | SUT        | B,OBH<br>GPTST  | SET GAP COUNTER   |
|  | 01DF           | BF9B<br>6928  | POST1:      | JUN        | DREND<br>1,28H  | ;SET-UP FOR 166,67 MICROSECONDS DELAY   |
|  | 01E1           | FCD5<br>4500  |             | CUN        | DELAY<br>5,00H  | RESET ENFL.ENHE, ENRST AND +CRCG.   |
|  | 01E3           | 693C          |             | SUT        | 1,3CH           | IF EVERYTHING IS OK WE ARE IN THE MIDDLE OF THE POSTAMBLE NOW.  |
|  |                | FCD5<br>6700  |             | CUN        | DELAY<br>7,00H  | ISET GAPD. HE SHOULD NOW HAVE PASSED  |
|  |                |               | ï           |            |                 | THE POSTAMBLE. IF THE GAPD SIGNAL. THE RESET BY DATA TRANSITIONS NOW,   |
|  | 01E6           |               | 1           | SUT        | 1,04H           | SOMETHING IS WRONG. SET GAP TIMING  |
|  | 01E7<br>01EB   | A5 DD         |             | JGD        | DELAY<br>GAPSC  | FINE, NO EXTRA DATA TRANSITIONS   |
|  | 01E9           | 650C          |             | OUT        | 5,0CH           | SET ENHE AND ENFL. SOMETHING IS WRONG   |
|  |                |               |             |            | -               |   |

23-16



#### 23.10 NZBGN SEQUENCE

The NZBGN sequence handles the NRZI write operations and splits into the NZFLM file mark sequence and the normal NSWRT data block sequence. First, two dummy write operations are executed to avoid timing race.

# 23.10.1 NZFLM SEQUENCE, FILE MARKS

NRZI file marks consist of only two bytes. The second byte is an LRC check character, necessarily identical to the first byte, bacause its primary purpose is to restore the proper gap polarization on the tape.

Note that it is not necessary to load the write parity register Al this time, since the parity bit is always zero for both 7-track and 9-track file marks.

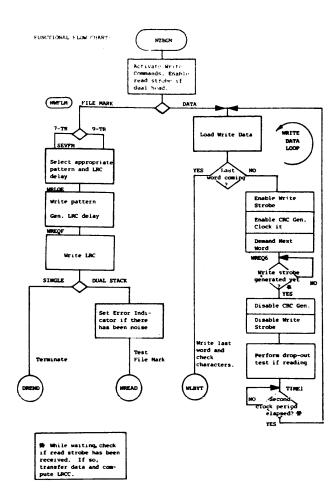
A gap test operation is performed to check the presence of noise just preceding the file mark: Fist the gap test flip-flop (U25,4) is set. If it is still set by the time its output M9-GAPD is checked, there has been no noise.

### 23.10.2 NSWRT SEQUENCE, NRZI DATA

In writing NRZI data, the program sets up a timing counter clock rate which is twice the bit rate, as was the case in PE. The clocking period, which for PE was 4.167 usec, is now 8.333 usec (whereas the bit period is 16.67 usec.) This finer timing is done to know exactly when the next byte is loaded in from the host computer and also to avoid problems during last word checking. The write strobe is only enabled every other clocking period.

The various steps in the NRZI write sequence is evident from the flow charts.

Keep in mind that the read head will read and start reading the written record some 2 msec after the start of writing (for 9-track heads at 75 ips; 4 msec for 7-track). This event is marked by the setting of the M7-DDET flip-flop (4), which is eventually reset when the read operation is deactivated. The DROPT



subroutine, used to detect dropouts, is in effect passive until this flip-flop is set (DROPT, Paragraph 23.29).

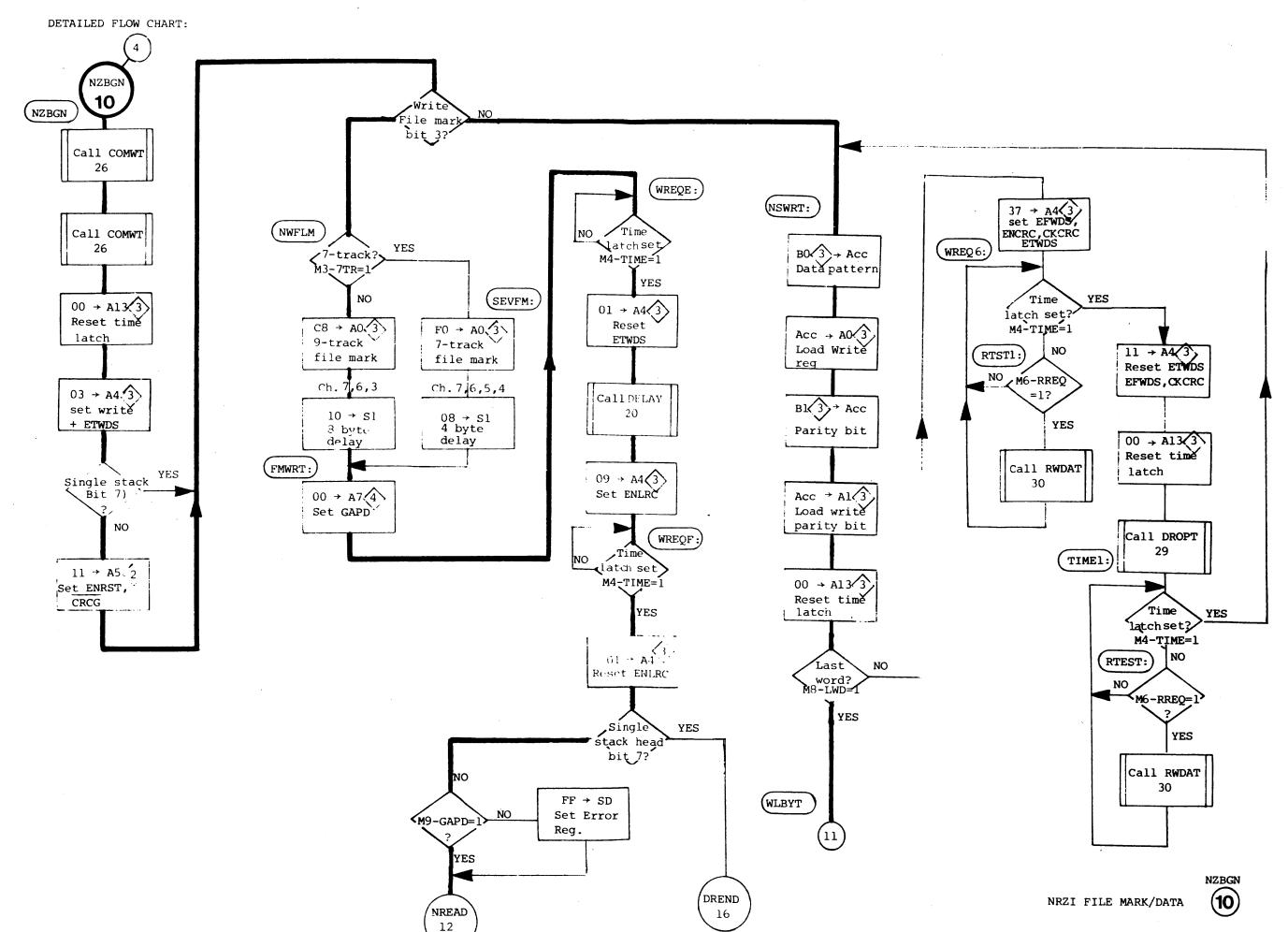
Arrival of read data is also manifested by the setting of the M6-RREQ flip-flop (4), being triggered by the TRDS read strobes. This flip-flop, however, is reset by the A2 strobe every time read data is being transferred (5).

The A2 strobe both resets this flipflop (4) and strobes the (first part of the) read data into the read shift register (5).

In the NSWRT sequence, the detection of read strobes by the RREQ signal is used to signal the transfer of read data using the RWDAT subroutine. This routine also checks vertical parity and keeps a running computation of the final LRC characters which it stores in the S4, S5 registers (RWDAT, Paragraph 23.30).

IDUMY WRITE OPERATION INCH DUMMY WRITE OPERATION RESET TIMING INDICATOR ISINGLE STACK MODE? 0212 8214 0213 6511 0214 0608 ISINGLE STACK IF ZERO ISET ENRET AND \*CRC6 ITEST UPON WRITE FILEMARK IMODE ( BIT 3 ) 5,11H 08H NZBG1: 0215 8217 NBMRT NMFLM SEVFM JZE INO FILEMARK IF ZERO FILEMARK OPERATION 01F8 8E08 01F9 60C8 NWFLM: ILOAD 9-TRACK FILEHARK PATTERN IINTO WRITE DATA REGISTER ISET-UP FOR 8 BYTE PERIODS DELAY OUT 0,0CBH 01FA 6910 FMURT: ISET GAPD HALTING FOR THE FIRST BYTE TO IBE WRITTEN 01FB 4700 01FC 91FC OUT JNT 7,00H 01FD 6401 01FE FCD5 01FF 6409 0200 9200 OUT CUN OUT JNT 4,01H DELAY 4,09H WREQF ISET ENLRC IMAITING FOR THE LRC BYTE TO IBE WRITTEM IRESET ENLRC IACC IS ZERO HERE ITEST UPON SINGLE MODE ( BIT 7 ) WREGF: 0201 6401 4,01H 0202 5D00 0203 0680 INO 5,0 80H 0204 8398 0205 A698 0206 7DFF 0207 BE98 0208 60F0 0209 6908 DREND NREAD D,OFFH NREAD 0,OFOH SINGLE STACK IF ZER JZE JGD SUT JUN OUT SUT JUN CLR INO AUT OK IF GAPD STILL IS I ILOAD 7-TRACK FILEMARK PATTERN ISET-UP FOR 4 BYTE PERIODS DELAY 020A BDFB 0217 4000 0218 5500 0219 2000 NELIRT | LOAD WRITE DATA INTO ACC. | LOAD WRITE DATA INTO WRITE REG. 021A 4000 021B 5501 021C 2100 CLR INO AUT | LOAD PARITY WRITE DATA INTO ACC. HRITE REGISTER RESET TIME INDICATOR LAST BYTE? 021D 7500 021E A23A 021F 0220 0221 TUC TUC TUC CUN JNT SET WRITE, ETWOS, EFWOS, CLCRC & ENCRC HAITING FOR DATA TO BE WRITTEN IRESET ETHOS, EFHOS AND CLCRC 0222 7500 0223 FE76 0224 9226 D,00H DROPT RTEST RESET TIME INDICATOR HALTING FOR FIRST PART OF NEXT TIME1: 0225 BE17 0226 DA2A 0227 BE24 0228 DA2A JUN CRR JUN CRR JUN NSHRT RHDAT RTEST: READ DATA DETECTED? RTST1:

23-18



#### 23.11 WLBYT SEQUENCE

The WLBYT sequence is a continuation of the NZBGN sequence. It writes the last word in the NRZI block and generates the CRC and LRC characters. The LRC character is not, however, required for 7-track operation.

The check characters are placed 4 character periods apart. Since we operate with double clocking rate, that means 8 clock periods apart. (This half character period of 8.33 usec should not be confused with the system clock period of 0.208 usec.)

After having written the last word, the WLBYT sequence conditions the CRC generator. This requires a few extra operations which are not, as it happens, performed by the CRC Generator package used, the MC 8500 (Ul17,3). This device requires for proper operation one additional shift with all inputs low after the last data character has been shifted in.

Therefore, the first step is to generate an all "0"s character. This is fed to CRC Generator via write data output registers AO, Al. However, the new CRC character output is inverted by the B3 output gates, and a software inversion operation is necessary to convert the character back to true form.

Another circuit complication is caused by the CRC generator Bl output gate for the parity bit. This gate has the strobe Bl in common with the gate for the input parity bit from U50, which already has been assigned DBl. The CRC parity bit has therefore been assigned DBO instead of the desired DBl, which must be used later in order to load write output register Al. Consequently, after the CRC parity bit has been brought into the accumulator, it is shifted left one place to the bit 1 position. It is also re-inverted and finally loaded into the write output register by the Al strobe.

The S6 scratch pad register is used to properly time the writing of the CRCC. This register is set to 3 when the program enters the timing sequence, although only 2 clock periods have so far elapsed. By the time it encounters the first S6 test, 3 clock periods have elapsed, but the register has been in-

cremented to 4. That is, S6-1 clock periods have elapsed. The program then cycles the timing loop until S6 = 7 (after six clock periods) when it branches off, waits a clock period and enables the CRC write strobe on what is then not the 8th, but the 7th clock pulse. The final writing of the CRCC appropriately takes place on the 8th clocking after one more cycle.

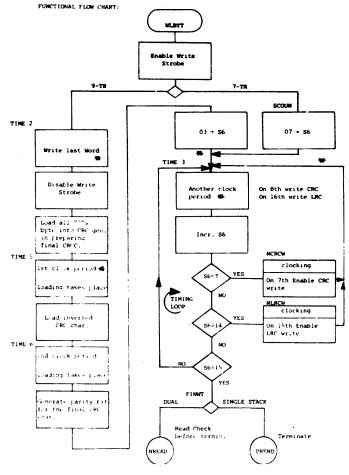
On this cycle S6 is incremented to 8, and the program will go on cycling through the timing loop again until S6 = 14, when it branches off for the LRC character. On the 15th clock count, the LRC write strobe is enabled, and the writing takes place on the next cycle at the 16th clock count. Only then is the S6 register incremented from 14 to 15 and thus qualifying for exit from the WLBYT sequence.

Note that there has been no generation and loading of a particular LRC character. This character automatically comes into being when the LRC write strobe TWRS resets all write amplifiers in the tape transports. Only those channels not already in the proper reset state will be reset and caused to have a flux reversal on the tape. The resulting flux pattern constitutes the LRC character.

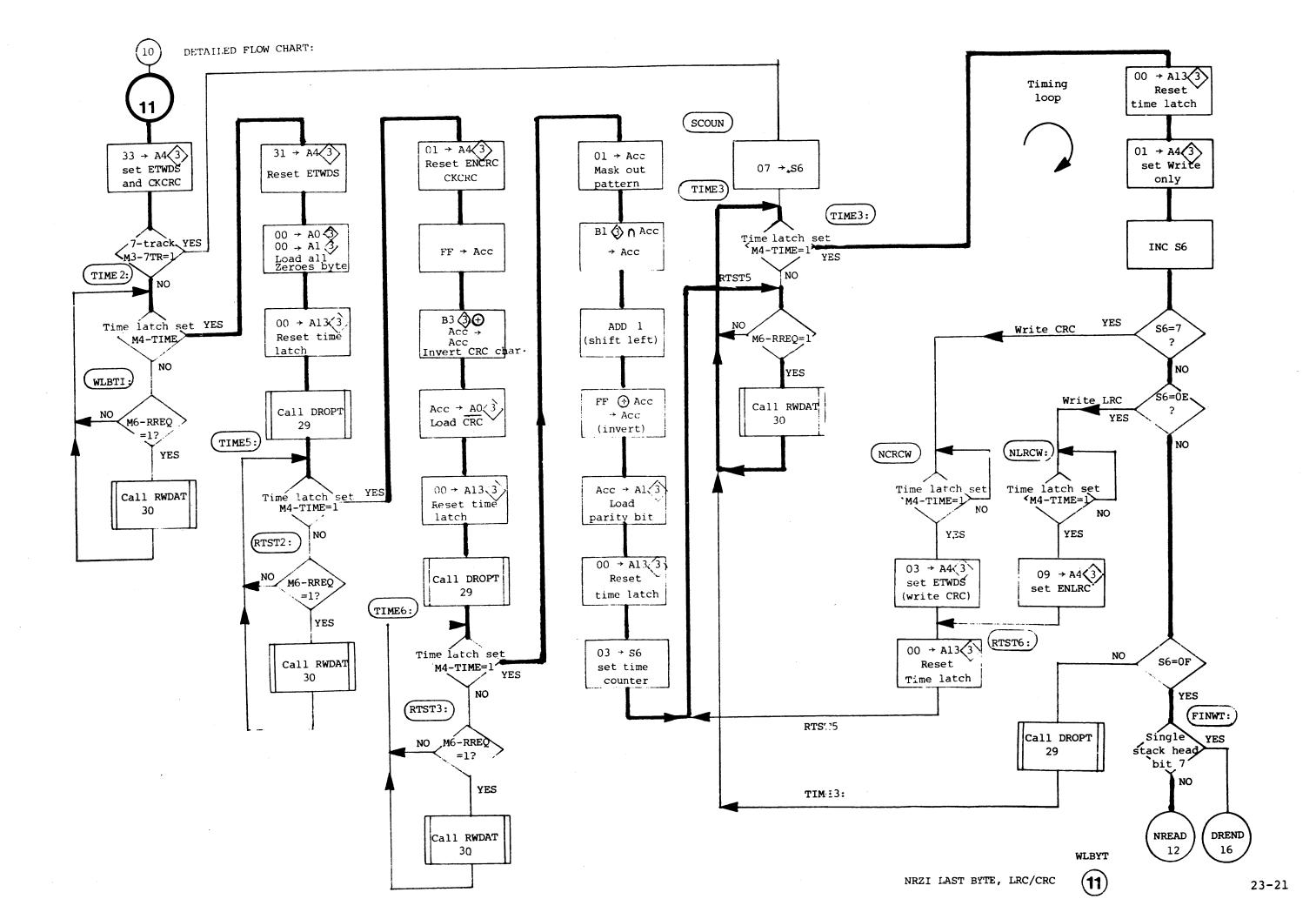
For 7-track operation, there is no CRC character, and the LRC character shall appear 8 clock periods after the last word, which in this case has not actually been written yet. In order to use the same timing sequence as for 9-track operation, the S6 register is preset to 7, thereby skipping 7 clock periods, and avoiding the CRC loop.

On the eigth clock count, the last word is written, and the program goes into the timing loop counting another 8 clock periods before it writes an LRC in the same manner as for the 9-track operation.

| 023A    | 6433  | WLET1: | TUR   | 4,33H  | SET ETWDS AND CLORG                     | 0259 | 4E07   | SCOUN:   | SUT | 6,07H      | INDICATES HE SHALL NOT          |
|---------|-------|--------|-------|--------|---|------|--------|----------|-----|------------|---------------------------------|
| 0238    | 8E59  |        | JIT   | SCOUN  | SEVEN TRACK OPERATION?                  |      |        | ;        |     |            | SHRITE THE CRC CHARACTER        |
| 0236    | 9253  | IIME.: | JNI   | WLBT1  | WAITING FOR LAST DATA BYTE              | 025A | 9269   | TIME3:   | JNT | RTSTS      |                                 |
|         |       | i      |       |        | ;10 BE WRITTEN                          | 025B | 7500   |          | OUT | D,00H      | RESET TIMING INDICATOR          |
| 0236    | 6431  |        | OUT   | 4,31H  | RESET ETWOS                             | 025C | 6401   |          | OUT | 4,01H      | ISET WRITE ONLY                 |
| 023E    | 5000  |        | OUT   | 0.00H  |   | 025D | 4000   |          | CLR | Α΄         |                                 |
| 0231    | : 100 |        | OUT   | 1,00H  | FLOAD ALL ZEROES CHARACTER              | 025E | 5 D0 6 |          | INO | 5,6        |                                 |
| ) 2 4 0 | 7500  |        | OUT   | D,00H  | RESET TIMING INDICATOR                  | 025F | 4301   |          | ADD | 01H        | IINCREMENT SA                   |
| 9241    | FE/6  |        | CL'N  | DROFT  | TEST FOR DROP-OUTS                      | 0260 | 2E00   |          | RUT | 6          | ISTORE SA                       |
| 0245    | 9255  | TIME5: | JNT   | RTST2  | WAITING FOR THE ALL-ZERGES              | 0261 | 0407   |          | TXR | 07H        | 186 = 07?                       |
|         |       | ;      |       |        | CHARACTER TO BE CLOCKED INTO            | 0262 | 824B   |          | JZE | NCRCH      | TIME FOR GENERATING THE         |
|         |       | :      |       |        | THE CRC REGISTER                        |      |        |          |     |            | CRC CHARACTER IF ZERO           |
| 0243    | 6401  | •      | OUT   | 4,01H  | FESET ENCRY AND CLURC                   | 0263 | 040E   | •        | TXR | 0EH        | 154 = 0E?                       |
| 6244    | 4700  |        | SET   | Α.     | , | 0264 | 824F   |          | JZE | NLRCH      | ITIME FOR GENERATING THE        |
|         | 5403  |        | INX   | A.3    | TINP 3 EXCLUSIVE-ORED WITH ACC.         |      |        |          |     | THE IN COM | ILRC CHARACTER IF ZERO          |
|         | 3403  | :      |       | 7,5    | THIS PERFORMS AN INVERTING OF           | 0265 | 440F   | •        | XRA | 0FH        | ISA = OF?                       |
|         |       | •      |       |        | ALL THE DATA BITS OF THE CRC            | 0266 | 8272   |          | JZE | FINHT      |                                 |
|         |       | ;      |       |        | CHARACTER.                              |      | FE76   |          | CUN | DROPT      | HARITING FINISHED IF ZERO       |
| 0.44    | 2000  | •      | AUT   | 0      | LOAD WRITE DATA REGISTER WITH           | 0268 | BESA   |          | JUN | TIME3      | ITEST FOR DROP-OUTS             |
| V. 40   | 2000  |        | -01   | •      | THE DATA PART OF THE CRC                | V200 | DEJA   |          | JUN | 111452     | IONCE MORE!                     |
|         |       | :      |       |        |   |      |        | •        |     |            |                                 |
|         | 7500  | •      | OUT   | F. AA  | CHARACTER                               |      |        | •        |     |            |                                 |
|         |       |        |       | D,00H  | RESET TIMING INDICATOR                  |      |        | ;        |     |            |                                 |
|         | FE76  |        | CUN   | DROPT  | TEST FOR DROP-OUTS                      |      | DA2A   | RTST5:   |     | RWDAT      | READ DATA DETECTED?             |
| 0249    | 9257  | TIME:  | INT   | RTST3  | HAITING FOR THE DATA PART               | U26A | BESA   |          | JUN | TIME3      |                                 |
|         |       | •      |       |        | OF THE CRC CHARACTER TO BE              |      |        | i        |     |            |                                 |
|         |       | i      |       |        | FLOADED                                 | 0268 | 9268   | NCRCH:   | JNT | NCRCH      | MAITING FOR THE THE LAST        |
|         | 4000  |        | CLR   | A      |   |      |        | i        |     |            | TIME INDICATOR BEFORE THE       |
|         | 4501  |        | ORA   | 01H    | SET BIT O TO 1                          |      |        | ĭ        |     |            | CRC CHARACTER CAN BE            |
| 0240    | 5601  |        | INA   | A,1    | LOAD PARITY BIT OF CRC CHARACTER        |      |        | ï        |     |            | WRITTEN.                        |
|         |       | ;      |       |        | INTO ACCUMULATOR IN POSITION 0          |      | 6403   |          | OUT | 4,03H      | SET ETHOS ( WRITE CRC )         |
| 65.4P   | 4301  |        | ADD   | 61 H   | SHIFT PARITY BIT OF CRC BYTE            |      | 7500   | RTST4:   |     | D,00H      | RESET TIMING INDICATOR          |
|         |       | ï      |       |        | PONE PLACE TO THE LEFT                  | 026E | BE69   |          | JUN | RTST5      |                                 |
|         | 44FF  |        | XRA   | OFFH   | ; INVERT THE CRC PARITY BIT             |      |        | <b>i</b> | _   |            |                                 |
| 024F    | 2100  |        | AUT   | 1      | LOAD THE CRE PARITY BIT INTO            | 026F | 926F   | NLRCH:   | JNT | NLRCH      | HAITING FOR THE LAST            |
|         |       | ;      |       |        | THE WRITE PARITY REGISTER               |      |        | ï        |     |            | TIME INDICATOR SEFORE THE       |
| 0250    |       |        | OUT   | D,00H  | RESET TIMING INDICATOR                  |      |        | i        |     |            | ILRC CHARACTER CAN BE GENE-     |
| 0.751   | 6E03  |        | SUT   | 6,03H  | (LOAD 03 INTO S6                        |      |        | ĭ        |     |            | RATED                           |
| 0252    | BE69  |        | JUN   | RISTS  |   | 0270 | 6409   |          | OUT | 4,09H      | ;SET ENLRC. ENABLE LRC-         |
| 0253    | DAZA  | WLBT1: | CRR , | RWEAT  | READ DATA DETECTED?                     |      |        | i        |     |            | ;GENERATION                     |
| 0254    | BE3C  |        | JUN   | 1 TWF? |   | 0271 | 8E4D   |          | JUN | RTST6      |                                 |
|         |       | ;      |       |        |   |      |        | ĭ        |     |            |                                 |
| 0255    | DA2A  | RIST2: | CRR   | RWDAT  | READ DATA DETECTED?                     |      |        | 1        |     |            |                                 |
| 0256    | BE42  |        | ソログ   | TIMES  |   | 0272 | 5 DO 0 | FINHT:   | INO | 5,0        | INE CAN DO THIS BECAUSE HE KNOW |
|         |       | ;      |       |        |   |      |        |          |     | -          | MOST SIGNIFICANT PART OF THE    |
|         |       | i      |       |        |   |      |        | i i      |     |            | ACCUMULATOR IS ZERO AT THIS     |
| 0257    | DACA  | RT513: | CRR   | RWUAT  | READ DATA DETECTED?                     |      |        |          |     |            | POINT                           |
| 0258    | BE49  |        | JUN   | TIME   |   | 0273 | 0480   | •        | TAN | BOH        | ITEST UPON SINGLE MODE ( BIT 7  |
|         |       |        |       |        |   | 0274 | 8378   |          | JZE | DREND      | ISINGLE STACK IF ZERO           |
|         |       |        |       |        |   | 0275 | BE98   |          | JUN | NREAD      | IDUAL STACK HEAD                |



<sup>#</sup> If read trade has been receive data is transferred and current SPCC computed.



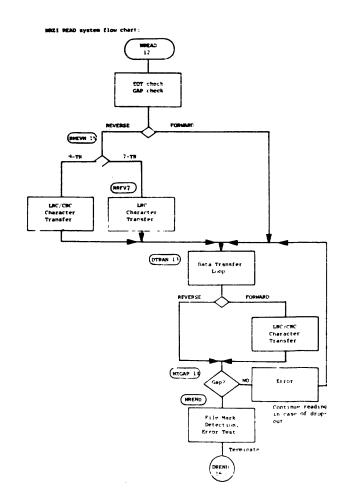
### 23.12 NREAD SEQUENCE

The NREAD sequence determines the nature of the required NRZI read operation and selects the proper read sequence.

Unless a data block has already been encountered, the program checks for the presence of EOT using the EOTCH subroutine, and checks the time it takes to reach the block, using the GPCNT subroutine.

Note that no data transfer takes place if the Erase line is true in the forward mode.

An overall flow chart for the NRZI read sequences is shown, including the NREAD (12), the RREVM(15), the DTRAN(13), and the NZGAP (14) sequences. When reading reverse the block end check characters appear first and must be handled first.



| 0298           | 9 <b>E</b> 9C | NREAD: | JOT        | SETOS          |  |              |              |         |            |                 |   |
|----------------|---------------|--------|------------|----------------|--|--------------|--------------|---------|------------|-----------------|---|
| 0279           | FD34          | MEND.  | CUN        | EOTCH          | TEST FOR EUT   |              |              |         |            |                 |   |
| 029A           | FD4B          |        | CUN        | GPCNT          | TEST FOR TOO LONG A SAP  | 02E7         | 5 D0 8       | SOTST   | INO        | 8,8             | INE HAVE NOW PASSED THE TIME PERIOD                                 |
| 029B           | 8E98          |        | JUN        | NREAD          |  |              |              | 1       |            |                 | IDURING WHICH A CHECK CHARACTER                                     |
| 029C           | 660F          | SETBS: | OUT        | 6,0FH          | SET DBSY   |              |              | !       |            |                 | IMAY ARRIVE. BE EQUAL TO 01 NEANS                                   |
| 029D           |               |        | SUT        | в,00н          |  |              |              | :       |            |                 | THAT HE HAVE READ BOTH THE CRC                                      |
| 029E           | 7900          |        | SUT        | 9,00H          | RESET TIMING REGISTERS   |              |              | •       |            |                 | IAND THE LRC CHARACTER IF OPE-                                      |
| 029F<br>02A0   | 4000<br>5000  |        | CLR<br>INO | A .            |  |              |              | i       |            |                 | TRATING WITH A P-TRACK SYSTEM, OR<br>THE LRC CHARACTER ONLY IF OPE- |
| 02A1           | 0601          |        | TAN        | 5,0            | TEST HEAV COMMEN MARK  |              |              | i       |            |                 | FRATING WITH A 7-TRACK SYSTEM.                                      |
| 02A2           | 8302          |        | JZE        | 01H<br>RREVM   | TEST UPON FORMARD MODE ( BIT 1 ) FREAD REVERSE IF ZERO   |              |              | i       |            |                 | THEREFORE, WE GO ON TO THE  |
| 02A3           | 0610          |        | TAN        | 10H            | TEST UPON ERASE MODE ( BIT +4 )  |              |              | i       |            |                 | INRZI READ END OPERATION.   |
| 02A4           | 82C5          |        | JZE        | DTRAN          | INO DATA TRANSFER IF ZERO  |              |              |         |            |                 | IF SO EQUAL OO, THE READING   |
| 02A5           | 6511          |        | OUT        | 5,11H          | SET *CRCG AND ENRST  |              |              | ş       |            |                 | FOR THE CRC CHARACTER IS  |
| 02 <b>A</b> 6  | BEC5          |        | JUN        | DTRAN          |  |              |              | •       |            |                 | FINISHED! BUT WE STILL HAVE TO                                      |
|                |               |        |            |                |  |              |              | :       |            |                 | HAIT FOR THE LRC CHARACTER.   |
| 02C5           | 6905          | DTRAN: | SUT        | 1,05H          | SET-UP FOR 2,5 CHARACTER   |              |              | •       |            |                 | THIS WILL ONLY HAPPEN FOR A   |
|                |               | ;      |            |                | PERIODS GAP TIMING.  |              |              | ï       |            |                 | 19-TRACK SYSTEM.  |
| 02 C6          | 9ACB<br>FEB5  | RREGT: | JRR        | CDATT          | ;DATA WAITING?   |              |              | í       |            |                 | IMEN THE CRC CHARACTER HAS BEEN IREAD! HE TEST UPON THE CONTENTS    |
| 02 C7<br>02 CB | 0500          |        | CUN<br>TOR | NDLAY<br>00H   | RETURN FROM DELAY  |              |              | •       |            |                 | OF THE SP REGISTER TO SEE IF  |
|                | B2CE          |        | JZE        | GAPDT          | DELAY FINISHED IF ZERO   |              |              | •       |            |                 | THE CRC HAS AN ALL ZEROES CHARACTER.                                |
| 02 CA          |               |        | JUN        | RREGT          | TOLERI I INIBILO II LENO   |              |              | ;       |            |                 | IF IT HAS, WE TRANSFER AN   |
| 02 CB          | FEA7          | CDATT: |            | RDATA          | START READ DATA TRANSFER.  |              |              | Ŧ       |            |                 | ALL ZEROES CHARACTER TO THE   |
| 02CC           | FEBA          |        | CUN        | TRANS          | FINISH DATA TRANSFER. TEST   |              |              | ;       |            |                 | READ DATA OUTPUT REGISTERS.   |
|                |               | ;      |            |                | FOR PARITY ERRORS.   |              |              | į       |            |                 |   |
| 02CD           | BECS          |        | JUN        | DTRAN          | ONCE MORE!   | 0350         | 4401         | i       | <b></b>    |                 |   |
| 02CE           |               | GAPDT: | INO        | 5,0            |  | 02E9         |              |         | XRA<br>JZE | 01H<br>S9TST    | LRC CHARACTER READ?   |
| 02 CF          |               |        | TAN        | 01H            | TEST UPON FORMARD MODE ( BIT 1 )   | 02EA         |              |         | CUN        | NOCKC           | ILRC READ IF ZERO   |
|                | 8359<br>0610  |        | JZE<br>TAN | NZGAP<br>10H   | REVERSE IF ZERO  |              | 7900         |         | SUT        | 9.00H           | WAIT FOR LRC  |
|                | 82D4          |        | JZE        | CWAIT          | (TEST UPON ERASE MODE ( BIT #4 )   | 02EC         | FF20         |         | CUN        | SDLAY           | A SHORT DELAY IS NECESSARY  |
| 0203           |               |        | OUT        | 5,30H          | RESET +CRCG, SET CCHG/ID   |              |              | ;       |            |                 | HERE TO AVOID THAT THE PARITY                                       |
|                |               | :      |            | •,••           | THE REASON FOR THIS IS THAT  |              |              | i       |            |                 | FERROR SIGNAL IS SENT. REMEMBER                                     |
|                |               | į      |            |                | THE CRC-CHARACTER OR THE 7-TRACK   |              |              | 3       |            |                 | THAT HE ARE SENDING AN ALL  |
|                |               | i      |            |                | ;LRC CHARACTER MAY HAVE EVEN   | 4250         | FEBF         | ;       | •          |                 | IZEROES CHARACTER   |
|                |               | i      |            |                | ;PARITY.   | 02EE         |              |         | CUN        | ERTST           | CALL ERASE TEST   |
| 0204           |               | CHAIT: |            | 1,08H          | SET 4 CHARACTER PERIODS GAP TIMING.  | 02EF         | 4631<br>2500 |         | ANA<br>AUT | 31H<br>5        | AND WITH 31   |
| 0205           | BED7          |        | J7 T       | LRCHT          | ;NO CRC IF 7-TRACK OPERATION   |              |              | :       |            | ,               | IF NOT A SPACE OR SEARCH ( IG-<br>INORE DATA ) OPERATION, THE       |
| 02D6<br>02D7   | 8E08<br>7801  | LRCWT: | JUN<br>SUT | CRCWT<br>8,01H | ; INDICATES THAT HE ARE WAITING  |              |              | i       |            |                 | +CRCG LINE IS SET TRUE.   |
| 4201           | , 601         | i i    | 301        | 0,010          | FOR AN LRC CHARACTER.  | 02F0         | 6908         | ·       | SUT        | 1.08H           | SET DELAY   |
| 0208           | FE85          | CRCWT: | CUN        | NOLAY          | TO THE CITY OF THE CONTROL OF THE CO | 02F1         | BED7         |         | JUN        | LRCHT           |   |
| 02D9           | 0500          |        | TOR        | 00H            | ;ACC IS ZERO?  | 02F2         | 5009         | STTST:  | INO        | 5,9             | HE KNOW ACCUMULATOR IS ZERO HERE                                    |
| 02DA           |               |        | JZE        | SETST          | ;DELAY FINISHED WHEN ZERO  | 02F3         | 82F5         |         | JZE        | NOLRC           |   |
|                | 9ADD          |        | JRR        | CRDAT          |  | 02F4<br>02F5 | BF39<br>BEFB | NO. 84. | JUN        | NZGAP           |   |
| 02DC           |               |        | JUN        | CRCHT          |  | 02F6         | 7 DFF        | NOLRC:  | J7T<br>SUT | ZELRC<br>D.OFFH | -DET FESOR SECTIONS   |
| 0200           |               | CRDAT: |            | RDATA          | START DATA TRANSFER  | 02F7         | BF59         |         | JUN        | NZGAP           | ISET ERROR REGISTER   |
| 02DE<br>02DF   |               |        | AUT        | 3              | START RDS GENERATION   | 02F8         | FEFF         | ZELRC:  | CUN        | NOCRC           | TRANSFER ALL ZEROES CHARACTER                                       |
| 02E0           | 4000<br>5009  |        | CLR<br>INO | ŝ,,            |  | 02F9         | FEB1         |         | CUN        | RDAT1           | INCREMENT SC. ACCUMULATOR IS  |
| 02E1           |               |        | JZE        | ONDET          | ; IF ZERO, WE HAVE RECEIVED THE FIRST  |              |              | ï       |            |                 | LOADED WITH INPUT 5 HERE. BUT                                       |
|                |               |        |            |                | BYTE IN THIS PERIOD AND EVERY-   |              |              |         |            |                 | THAT DOES NOT MATTER.   |
|                |               | į      |            |                | THING IS OK.   | 02FA         | 8F39         |         | JUN        | NZGAP           |   |
| 02E2           | 7 DFF         |        | SUT        | D,OFFH         | LOAD ERROR REGISTER. HE HAVE   |              |              |         |            |                 |   |
|                |               | i      |            |                | RECEIVED MORE THAN ONE BYTE IN   |              |              |         |            |                 |   |
|                |               | ŧ      |            |                | THIS PERIOD.   |              |              |         |            |                 |   |
| 02E3           | 4511          | _      | OUT        | 5,11H          | SET +CRCG AND ENRST. START NORMAL  |              |              |         |            |                 |   |
|                |               | •      |            |                | ;DATA TRANSFER ONCE MORE. THE GAP<br>;WE DETECTED WAS PROBABELY A  |              |              |         |            | m <sub>b</sub>  | o data buto can duration i  |

# 23.13 DTRAN SEQUENCE

The DTRAN sequence controls the transfer of NRZI data bytes read from tape. It also includes the transfer of check characters in the forward mode.

In the reverse mode the check character transfer, being handled by the RREVM sequence (Paragraph 23.15), will already have taken place by the time the program enters the DTRAN sequence. Hence, after the data transfer operation is completed, the program exits straight off to the NZGAP sequence, bypassing the forward mode check character transfer sequence in the DTRAN.

The general data transfer sequence utilizes three subroutines for the transfer operation. These are used repeatedly in a program loop until all data is transferred and a gap is detected which exceeds the maximum allowable gap between data bytes, that is, 2.5 character periods. At this point a check character, which shall appear after 4 character periods, is assumed to come next, and the program branches off to the check character transfer part of the DTRAN sequence. However, in the case of a read reverse operation, the detected gap is assumed to be the interblock gap and the program exits to the NZGAP sequence as explained.

The data byte gap duration is monitored by the NDLAY subroutine (Paragraph 23. 34). This subroutine is similar to the DELAY routine, but requires an outside closing loop to complete the delay. This allows the delay timing to be interrupted whenever a read strobe is received (RREQ = 1) and to branch out of the delay loop to execute the transfer of the data byte accompanying the read strobe. Only if the delay is completed, that is, if the delay exceeds 2.5 character periods, will the program break out of the data transfer loop.

To execute the data transfer, the RDATA and TRANS subroutines are used (Paragraphs 23.31 and 32). Together they perform the same function as the RWDAT subroutine except that the RDATA subroutine increments the SC byte count register if the number of bytes detected is less than 3. This register is used later to qualify the block: Bad block if SC is less than 2, file mark if SC = 2, and data block if SC is larger than 2.

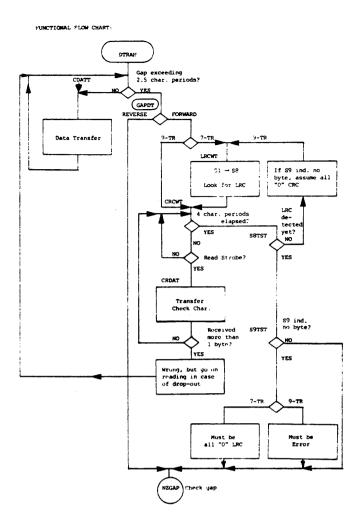
In general, the RDATA subroutine loads the read output registers and increments the SC register, and the TRANS subroutine strobes the data byte out, and performs a vertical parity check. The NDLAY subroutine, in addition to branching out 2.5 character periods after the last read strobe, also performs an EOT check. This check will be performed very often since the rate at which NDLAY is cycled through is much higher than the data bit rate seen by the RREQ read strobe detection circuit.

The check character transfer loop used in the remainder part of the DTRAN sequence is similar to the data transfer part and is used for both the CRC and LRC characters. The mechanism of breaking out of this loop once both characters have been detected employs the S8 register. This register is set to Ol whenever there is only one character to search for, such as in 7-track operation or after the CRC haracter has been detected in the 9-track operation. A test of the S8 register is then made after this search has been made, and if S8 = 01, the program goes on to finish the DTRAN sequence.

When the program enters the CRCWT check character transfer loop, 2.5 character periods have already elapsed since the last character, and this gap constitutes the minimum required before the first check character arrives. This character, which in both 7-track and 9-track operation may be an undetectable all "0"s byte, is assumed to arrive within an additional 4 character periods, for a total of 6.5 periods since the last data byte. To monitor the check character gaps, the NDLAY subroutine is used, much in the same manner as it was used in checking the data byte gaps in the data transfer loop. The S8 test is placed in the NDLAY closing loop.

In the case of an all "0" byte, the program will break out of the NDLAY delay loop after 6.5 periods in total without having entered the CRDAT check character transfer loop, since no read strobe will accompany such a byte

If the byte is not all "0"s, it will be duly transferred, using the RDATA sub-routine and the A3 strobe instruction used in the TRANS subroutine. (The TRANS subroutine is not used here



because the parity check contained therein is not wanted here).

At this point in the CRDAT loop, the S9 register is tested. In normal operation, the program will break out within 4 character periods and reset the S9 register to 00. This register is set to 01 only during the CRDAT sequence, in the CRCET subroutine which is part of the sequence. Hence, if the S9 is 01 at this test point, it means that a second byte has been detected before the elapse of the full 4 character periods since a first byte set the S9 register. It is then assumed that this byte must be a data byte and that the initial gap exceeding 2.5 character periods was due to a drop-out. The error register is therefore set, and the program is returned to the DTRAN entry point in order to resume reading and transferring data bytes.

The CRCET subroutine will, in addition to registering the detection of a byte in S9, check and modify the S4 parity register, see Paragraph 23.35.

In 7-track operation, the LRC character may be an all "0"s byte, but not so in 9-track operation. On the other hand, in 9-track operation the CRC character may be an all "0"s byte.

Normally, the program will cycle the check character transfer loop once in the 7-track mode and then exit to the S9TST sequence. If however, the LRC is an all "0"s byte, the transfer loop is not cycled, the S9 is not set, and the program exits to the S9TST sequence after 4 character periods. In this sequence a test on S9 is made, and if S9 = 00, an all "0"s LRC is assumed and transferred at this point.

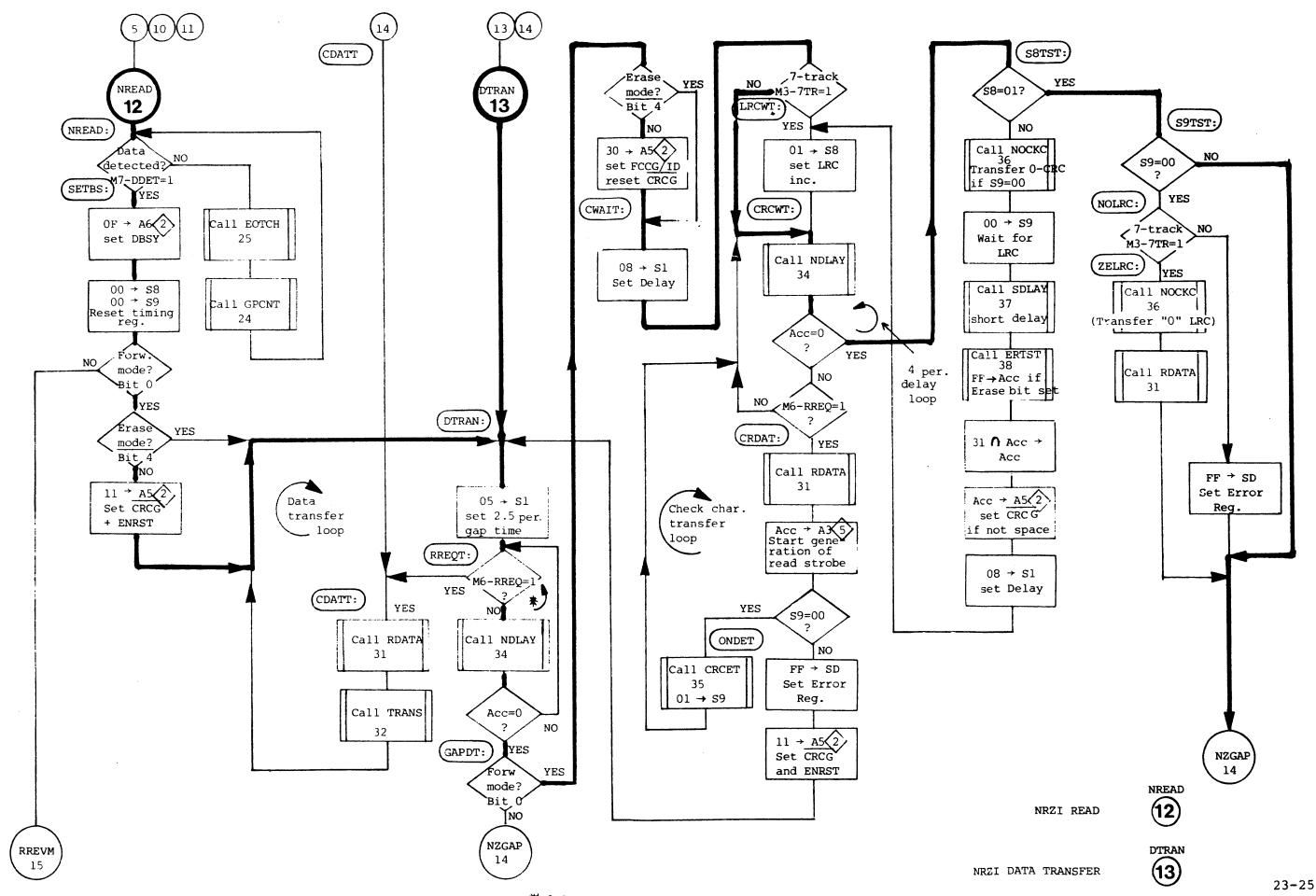
In the 9-track mode, the S8 register is not set until the first CRC character search cycle has been completed, and the program will make one extra search cycle for the LRC character before exiting to the final S9TST sequence.

The CRC search cycle will normally include cycling the transfer loop, setting the S9 register. Not, however, for an all "0"s byte. In that case, S9 will remain 00. After the S8 test, beginning on the LRC search cycle, the S9 register is therefore tested to determine whether or not a CRC byte was detected. If not detected, an all "0"s CRC byte is assumed and consequently transferred. This is taken care of by the NOCKC subroutine (Paragraph 23.36).

An all "0"s byte will generate a parity error signal, and the CRCG-gate, which controls the transmission of parity error signals, must be prevented from enabling such transmission <u>until</u> the byte has been safely transferred. This is accomplished by introducing a short delay using the SDLAY subroutine (Paragraph 23.37).

Parity error signal transmission is enabled for the LRC character, but not in the space or search/ignore data modes. The actual mode is determined by the ERTST subroutine (Paragraph 23.38).

The program continues its search for an LRC character in the same manner as for the CRC character: Waiting for a read strobe, transferring the character, waiting for the end of the NDLAY delay and finally exiting to the S9TST sequence. If S9 = 00, indicating failure to detect an LRC character, the error register is set.



### 23.14 NZGAP SEQUENCE

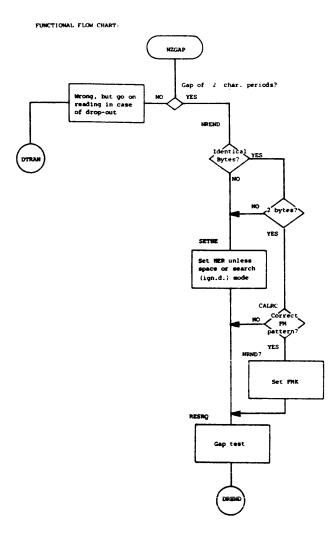
The NZGAP sequence is a continuation of the NREAD sequence and checks the gap following the data block. If the gap is at least 2 character periods long, the program goes on to check for file marks and errors. If on the other hand, there are signals received within this period, the gaps previously believed in the DTRAN sequence to be check character space gaps, are assumed to have been drop-outs, and the program returns to the DTRAN sequence.

The check for errors starts with a check of the longitudinal parity, examining the S5 and S4 register which contains the results for each read channel of exclusive-oring (in the RDATA subroutine) all previous bits (cfr. Paragraph 23.30). The respective longitudinal parity bits shall be "0". This also holds true for file mark bits.

In the program the data part of the S5 is examined first, and then the S4. The four most significant bits in S4 shall also be "0", and the entire S4 byte is examined. Of the 9 available channels in the 7-track mode it is channel 0 and 1 which are not used, and these two are masked out.

If the S5, S4 test is passed and there is no error registered in SD, the first condition for file mark is checked: The number of bytes in the block. This has previously been recorded in the SC register by the RDATA subroutine and shall be 2 for a NRZI file mark.

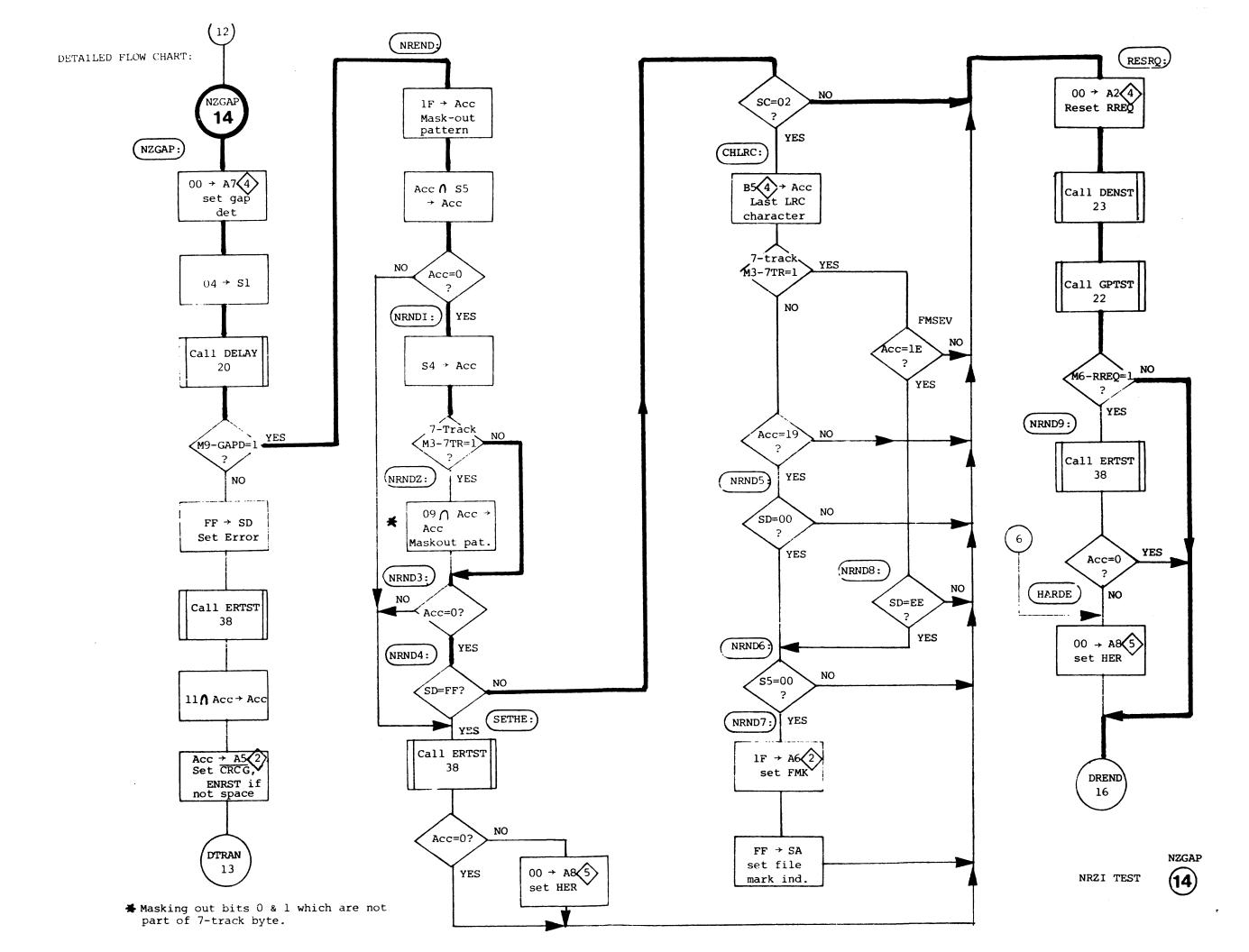
The next file mark test is the file mark pattern itself. For this test the entire B5 byte can be used, and the B4 part need not be used, since both the P and 0 channel bits is "0" for any type of file mark. For 7-track and 9-track formats the pattern shall have "l"s in channels 4, 5, 6, 7, respectively 3, 6, 7. Furthermore, for a 7-track file mark the parity shall be even, and the SD shall contain EE as the result of the parity check in the TRANS subroutine.



The NZGAP sequence is finished off by a 25 period gap test using the GPTST subroutine. In addition, a check is made on read strobes. Detection of read strobes is ignored only if in space or search/ignore data mode (as tested by ERTST).

|                | 0358<br>0358<br>0358<br>0358<br>0358<br>0358<br>0358<br>0358 | 7 DFF FEBF 4411 2500 8ECS 4000 451F 5505 8367 8F6C 0500 834E 8F6B 0500 834E 8F8B 0500 NRN 834E 8F8B 0500 04FF 838D 8F69 5D0D 04FF 8381 8F71 5D0D 04FF 8387 8F71 5D0D NRN 4000 5D0C 0402 8376 8571 8571 8587 8571 85887 85887 8571 85887 | SUT CUM JGD SUT CUM ANA ANA ANA ANA ANA ANA ANA ANA JUN O1: INO J7T JUN JUN O4: INO TXR JUN O4: INO TXR JUN O4: INO TXR JUN OX TXR JUN OX TXR JZE JUN ANA JUN OX TXR JUN OX TXR JZE JUN TXR JZE JUN | 7,00H 1,04H 0ELAY NREND D,0FFH ERTST 11H 5 DTRAN A 1FH SETHE ST,4 NRND1 SETHE NRND2 SODH OFFH SETHE A 5,0CH CHLRC RESRO A 5,5 FRISE INND SETHE A 5,0CH CHLRC RESRO A 5,5 FRISE SETHE | ISET GAP DE ISET 2,0 PEI ISET 2,0 PEI ISET 2,0 PEI ISET EARD EI ISET HARD EI ISET HARD EI ISET HARD ER ISET HARD ER ISET HARD ER ISET FILE MA ISET FILE MA ISET FILE MA ISET FILE MA ISET GAP CO ISET HARD EI ISET HA | E ABBURE THE IS DETECTE TO THE IS DETECTE TO THE IS DETECTE TO THE ING IN SPACE OF THE IS DETECTED TO THE IS DETECTED. | P DELAY AT A D ER F ENOBE.  EPT 0 AND 1 HERE HERE HERE HERE HERE HERE HERE HER |       |      |
|----------------|--|---|---|--|--|--|--|-------|------|
| bit location:  | 7  | 6   | 5   | 4  | 3  | 2  | 1  | 0     |      |
| S4             | L  | L   | L   | L  | ch. 2  | ch. l  | ch. 0  | ch. P |      |
| 7-tr mask      | 0  | 0   | 0   | 0  | 1  | 0  | 0  | 1     | (09) |
| S5, B5         | ch. 1  | ch. 2   | ch.PUO  | ch. 7  | ch. 6  | ch. 5  | ch. 4  | ch. 3 |      |
| Data mask      | 0  | 0   | 0   | 1  | 1  | 1  | 1  | 1     | (1F) |
| 7-tr file mark | 0  | 0   | 0   | 1  | 1  | 1  | 1  | 0     | (1E) |
|                |  |   |   |  |  |  |  |       |      |

23-26



### 23.15 RREVM SEQUENCE

The RREVM sequence is part of the NREAD sequence and handles the check character transfer prior to the data transfer in the reverse read mode.

The program for the forward mode check character transfer in the DTRAN sequence was able to use many common smaller sequences for 7 and 9 track modes of operation. In the RREVM sequence, these modes are handled by completely separated branches.

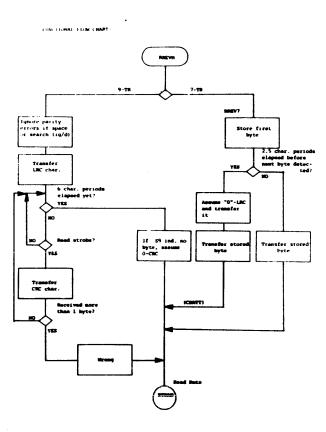
In 9-track operation, the first byte arrived is assumed to be the LRC character and is promptly transferred. The CRC byte is expected to be detected within 6 character periods, and is assumed to be an all "0"s byte in the absence of a read strobe during this waiting period. The program sequence is similar to the corresponding one in the DTRAN sequence.

In the 7-track mode, matters are complicated by the fact that the LRC character may be an all "0"s byte, and it is impossible therefore to determine whether the first detected byte is an LRC character or a data character before the gap to the next byte is known to exceed the minimum required for an LRC character.

The first byte is therefore stored while the program goes on to examine the gap. If the gap exceeds 2.5 character periods, the detected byte is qualified to be an LRC character and transferred as such. If the gap is less than 2.5 periods, the LRC is assumed to be an all "0"s byte, and such a "0" byte is transferred, before the stored byte is transferred. The latter is now assumed to be a data byte and is transferred after a short delay.

Using CRCG, parity error signal transmission is disabled during the LRC character transfer, and the SDLAY subroutine is used as before to keep it so disabled until the transfer is safely completed.

The TRANB subroutine is used to transfer the first byte instead of RDATA, since this byte has already been stored in S4, S5, and in the case of an all "0"s LRC character is no longer available in



the B4, B5 registers. In this case, the read strobe for the next byte has already been received, and the program must consequently use the CDATT entry point in DTRAN sequence to secure the transfer of the accompanying data byte.

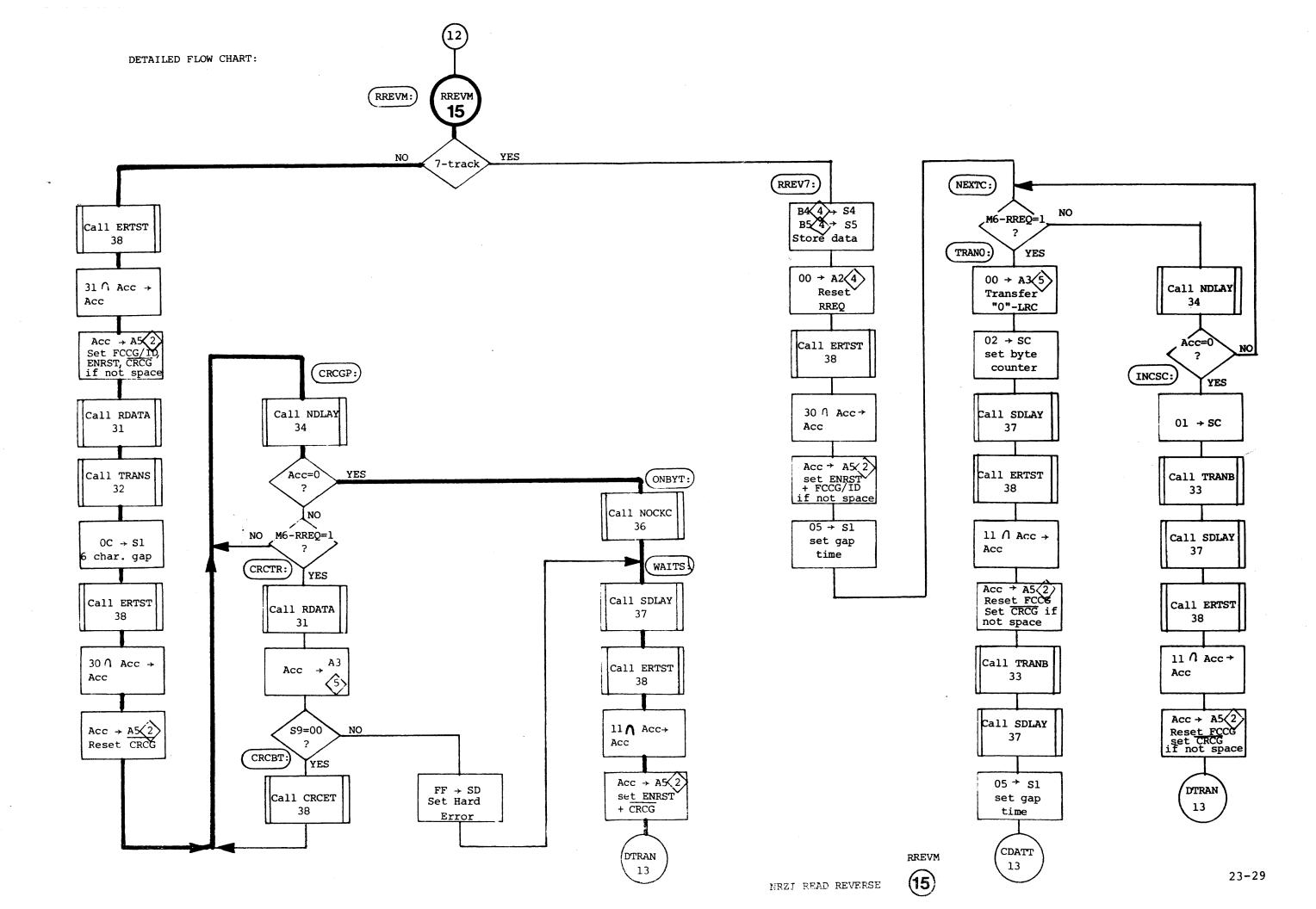
|  |  | ş   |  |  | ISO GREAT.   |
|--|--|---|--|--|--|
| 0303   | FEBF   | i   | CUN  | ERTST  | CALL ERASE TEST  |
| 0304   | 4631   |   | ANA  | 31H  | IF ACC IS FF, THIS GIVES 31<br>IF NO SPACE, THIS WILL SET THE  |
| 0305   | 2500   |   | AUT  | 5  | IF NO SPACE, THIS WILL SET THE CCH6/ID, ENRST AND +CRCG LINES.   |
| 0306   | FEA7   | •   | CUN  | RDATA  | TRANSFER LRC CHARACTER   |
| 0307<br>030B   | FEBA<br>690C   |   | CUN<br>SUT   | TRANS  | START ROS-GENERATION SLOAD &-CHARACTER PERIODS DELAY   |
| 0309<br>030A   | FEBF<br>4630   |   | CUN  | ERTST  | CALL ERASE TEST SECTION  |
| 0308   | 2500   |   | AUT  | 30H<br>5   | IF NO SPACE OPERATION, THIS  |
|  |  | •   |  |  | WILL RESET THE +CRC6 LINE.   |
|  |  | i   |  |  | THIS IS NECESSARRY TO AVOID FALSE PARITY ERROR INDICATION.   |
| 030C<br>030D   | FE85<br>0500   | CRCGP:                                    | CUN  | NDLAY<br>00H   | ;DELAY FINISHED?   |
| 030E   | 831A   |   | JZE  | ONBYT  | DELAY FINISHED?  |
| 030F<br>0310   | 9811<br>BF0C   |   | JRR<br>JUN   | CRCTR<br>CRCGP   | CRC BYTE DETECTED?   |
| 0311   | FEA7   | CRCTR:                                    | CUN  | RDATA  | START CRC DATA TRANSFER  |
| 0312<br>0313   | 2300<br>4000   |   | AUT<br>CLR   | 3  | START RDS-GENERATION   |
| 0314   | 5D09   |   | INO  | 5,9  |  |
| 0315<br>0316   | 8318<br>7DFF   |   | JZE<br>SUT   | CRCBT<br>D,OFFH  | CRC BYTE RECEIVED IF ZERO SET ERROR REGISTER. MORE THAN  |
|  |  | i   |  | •,•  | IONE BYTE DETECTED IN THE CRC  |
| 0317   | 8F18   | ;   | JUN  | WAITS  | TIME GAP.  |
| 0318   | FF4E   | CRCBT:                                    | CUN  | CRCET  | START CRC TESTING  |
| 0319<br>031A   | BF0C<br>FEFB   | ONBYT:                                    | JUN<br>CUN   | CRCGP<br>NOCKC   |  |
| 031B   | FF20   | WAITS:                                    | CUN  | SDLAY  | START SHORT DELAY  |
|  |  | i   |  |  | TO AVOID TIMING PROBLEMS THEN GENERATING THE RDS-  |
|  |  | i   | <b>-</b>   |  | ;STROBES.  |
| 031C<br>031D   | FEBF<br>4611   |   | CUN<br>ANA   | ERTST<br>11H   |  |
| 031E   | 2500   |   | AUT  | 5  | THIS WILL SET THE ENRST AND  |
|  |  | ;<br>;                                    |  |  | ;+CRCG LINES IF HE ARE NOT<br>;OPERATING IN SPACE MODE   |
| 031F   | BEC5   |   | JUN  | DTRAN  | START NORMAL DATA TRANSFER   |
|  |  |   |  |  |  |
|  |  |   |  |  |  |
| 0326   | 4000   | RREV7:                                    | CLR  | •  | U OOD R AND 3 CHANGE   |
| 0327<br>032 <b>8</b>   | 5504<br>2000   | RREV7:                                    | INO<br>RUT   | A,4  | ILOAD P AND 2 CHANNEL<br>ISTORE DATA INTO SA   |
| 0327<br>0328<br>0329   | 5504<br>2 COO<br>4000  | RREV7:                                    | INO<br>RUT<br>CLR  | A.4<br>A   | STORE DATA INTO 84   |
| 0327<br>0328<br>0329<br>032A   | 5504<br>2 CO0<br>4000<br>5505  | RREV7:                                    | INO<br>RUT<br>CLR<br>INO   | A,4<br>A<br>A,5  | STORE DATA INTO SA  LOAD THE MEST OF THE DATA CHARACTER  |
| 0327<br>0328<br>0329   | 5504<br>2 COO<br>4000  |   | INO<br>RUT<br>CLR  | A.4<br>A   | STORE DATA INTO SA  LOAD THE WEST OF THE DATA ICHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD   |
| 0327<br>0328<br>0329<br>032A<br>032B<br>032C   | 5504<br>2C00<br>4000<br>5505<br>2D00<br>4200   |   | INO<br>RUT<br>CLR<br>INO<br>RUT<br>OUT   | A,4<br>A<br>A,5<br>5<br>2,00H  | STORE DATA INTO SA  LOAD THE NEST OF THE DATA CHARACTER STORE DATA INTO SS   |
| 0327<br>0328<br>0329<br>032A<br>032B<br>032C<br>032C   | 5504<br>2C00<br>4000<br>5505<br>2D00<br>6200<br>FEBF<br>4630   | i   | INO<br>RUT<br>CLR<br>INO<br>RUT<br>OUT<br>CUN<br>ANA   | A,4<br>A<br>A,5  | STORE DATA INTO 84  LOAD THE WEST OF THE DATA CHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD HALF OF "0"-LRC CHARACTER.   |
| 0327<br>0328<br>0329<br>032A<br>032B<br>032C   | 5504<br>2C00<br>4000<br>5505<br>2D00<br>6200<br>FEBF   | ;   | INO<br>RUT<br>CLR<br>INO<br>RUT<br>OUT   | A,4<br>4<br>A,5<br>5<br>2,00H<br>ERTST   | STORE DATA INTO 84  LOAD THE WEST OF THE DATA CHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD CHALF OF "O"-LRC CHARACTER.  SET ENRST AND CCHS/ID IF NOT  |
| 0327<br>0328<br>0329<br>032A<br>032B<br>032C<br>032C   | 5504<br>2C00<br>4000<br>5505<br>2D00<br>6200<br>FEBF<br>4630   | i   | INO<br>RUT<br>CLR<br>INO<br>RUT<br>OUT<br>CUN<br>ANA   | A,4<br>A<br>A,5<br>5<br>2,00H<br>ERTST<br>30H  | STORE DATA INTO 84  LOAD THE WEST OF THE DATA CHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD HALF OF "O"-LRC CHARACTER.  SET ENRST AND CCHS/ID IF NOT IN SPACE MODE SET-UP FOR 2,5 CHARACTER PERIODS  |
| 0327<br>0328<br>0329<br>032A<br>032B<br>032C<br>032C   | 5504<br>2C00<br>4000<br>5505<br>2D00<br>6200<br>FEBF<br>4630<br>2500   | ;   | INO<br>RUT<br>CLR<br>INO<br>RUT<br>OUT<br>CUN<br>AMA<br>AUT  | A,4<br>A<br>A,5<br>5<br>2,00H<br>ERTST<br>30H  | STORE DATA INTO 84  LOAD THE MEST OF THE DATA CHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD HALF OF "O"-LRC CHARACTER.  SET ENRST AND CCHS/ID IF NOT IN SPACE HOOE   |
| 0327<br>0328<br>0329<br>032A<br>032B<br>032C<br>032C<br>032F<br>0337   | 5504<br>2C00<br>4000<br>5305<br>2D00<br>6200<br>FEBF<br>4630<br>2500<br>6905<br>9836<br>FE85   | i<br>i                                    | INO RUT CLR INO RUT OUT CUN ANA AUT SUT JRR CUN  | A,4<br>A<br>A,5<br>5<br>2,00H<br>ERTST<br>30H<br>5<br>1,05H<br>TRANO<br>NOLAY  | STORE DATA INTO 84  LOAD THE WEST OF THE DATA CHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD CHALF OF "O"-LRC CHARACTER.  SET ENRST AND CCHS/ID IF NOT IN SPACE MODE SET-UP FOR 2,5 CHARACTER PERIODS CAP TIMINS CO"-LRC IF READ REQUEST EQUAL TO 1   |
| 0327<br>0328<br>0329<br>032A<br>032C<br>032C<br>032C<br>032F<br>0337   | \$504<br>2C00<br>4000<br>5305<br>2D00<br>6200<br>FEBF<br>4630<br>2500<br>6705  | i<br>i<br>NEXTC:                          | INO RUT CLR INO RUT OUT CUN ANA AUT SUT JRR  | A,4<br>A,5<br>5<br>2,00H<br>ERTST<br>30H<br>5<br>1,05H<br>TRANO  | #STORE DATA INTO 84  !LOAD THE MEST OF THE DATA !CHARACTER !STORE DATA INTO SS !RESET READ REQUEST. LOAD !HALF OF "O"-LRC CHARACTER.  !SET ENRST AND CCH6/ID IF NOT !IN SPACE MODE !SET-UP FOR 2,5 CHARACTER PERIODS !GAP TIMING !"O"-LRC IF READ REQUEST EQUAL TO 1 !DELAY FINISHED? !THE FIRST CHARACTER RECEIVED HAS  |
| 0327<br>0328<br>0329<br>032A<br>032B<br>032C<br>032C<br>032F<br>0337<br>0333   | \$504<br>2000<br>4000<br>5305<br>2D00<br>6200<br>FEBF<br>4630<br>2500<br>6705<br>PB36<br>FE85<br>0500<br>8340  | i<br>i                                    | INO<br>RUT<br>CLR<br>INO<br>RUT<br>OUT<br>CUN<br>ANA<br>AUT<br>SUT<br>JRR<br>CUN<br>TOR<br>JZE   | A,4<br>A<br>A,5<br>5<br>2,00H<br>ERTST<br>30H<br>5<br>1,05H<br>TRANO<br>NDLAY<br>00H<br>INCSC  | STORE DATA INTO 84  LOAD THE WEST OF THE DATA CHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD HALF OF "O"-LRC CHARACTER.  SET ENRST AND CCHS/ID IF NOT IN SPACE MODE SET-UP FOR 2,5 CHARACTER PERIODS GAP TIMING "O"-LRC IF READ REQUEST EQUAL TO 1 IDELAY FINISHED?   |
| 0327<br>0328<br>0329<br>032A<br>032B<br>032C<br>032C<br>032F<br>0337   | 5504<br>2000<br>5305<br>2D00<br>6200<br>FEBF<br>4630<br>2500<br>6705<br>9B36<br>FE85<br>0500   | i i i i t TRANO:                          | INO RUT CLR INO RUT OUT CUN AMA AUT SUT JRR CUN TOR  | A,4<br>A<br>A,5<br>5<br>2,00H<br>ERTST<br>30H<br>5<br>1,05H<br>TRANO<br>NDLAY  | #STORE DATA INTO 84  #LOAD THE WEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCH6/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMINS #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED HAS #AN LRC CHARACTER IF ZERO #START TRANSFERRING OF THE   |
| 0327<br>0328<br>0329<br>032A<br>032B<br>032C<br>032C<br>032F<br>0333<br>0333<br>0334   | 5504<br>2000<br>5505<br>2D00<br>6200<br>FEBF<br>4630<br>2500<br>6705<br>PB36<br>FEB5<br>0500<br>8340<br>BF31   | i<br>i<br>i<br>MEXTC:                     | INO<br>RUT<br>CLR<br>INO<br>RUT<br>OUT<br>CUN<br>ANA<br>AUT<br>SUT<br>JRR<br>CUN<br>TOR<br>JZE<br>JUN  | A,4 A A,5 S 2,00H ERTST 30H S 1,05H TRANO NDLAY 00H INCSC  | #STORE DATA INTO 84  *LOAD THE MEST OF THE DATA *CHARACTER *STORE DATA INTO SS *RESET READ REQUEST. LOAD **HALF OF "O"-LRC CHARACTER.  **SET ENRST AND CCH6/ID IF NOT **;IN SPACE MODE **SET-UP FOR 2,5 CHARACTER PERIODS **GAP TIMING **;TO"-LRC IF READ REQUEST EQUAL TO 1  **DELAY FINISHED? **THE FIRST CHARACTER RECEIVED MAS **JAN LRC CHARACTER IF ZERO   |
| 0327<br>0328<br>0329<br>032A<br>032C<br>032C<br>032C<br>032F<br>0337<br>0333<br>0334<br>0333<br>0334   | 5504<br>2000<br>5305<br>2000<br>6200<br>FEBF<br>4630<br>2500<br>6705<br>7836<br>FE85<br>0500<br>8340<br>8531<br>6300<br>7C02<br>FF20   | i i i i t TRANO:                          | IND<br>RUT<br>CLR<br>IND<br>RUT<br>OUT<br>CUN<br>ANA<br>AUT<br>SUT<br>JRR<br>CUN<br>TOR<br>JZE<br>JUN<br>OUT<br>SUT<br>CUN   | A,4<br>A<br>A,5<br>5<br>2,00H<br>ERTST<br>30H<br>5<br>1,05H<br>TRANO<br>NDLAY<br>00H<br>INCSC<br>NEXTC<br>3,00H<br>C,02<br>SDLAY                             | #STORE DATA INTO 84  #LOAD THE MEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCH6/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMING #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED HAS #AN LRC CHARACTER IF ZERO #START TRANSFERRING OF THE #"O"-LRC CHARACTER  |
| 0327<br>0328<br>0329<br>032A<br>032C<br>032C<br>032F<br>0337<br>0333<br>0333<br>0334   | 5504<br>2C00<br>4000<br>5305<br>2D00<br>6200<br>4705<br>7836<br>FE85<br>0500<br>8340<br>8F31<br>6300<br>7C02   | i i i i t TRANO:                          | IND RUT CUN ANA AUT SUT CUN TOR JER CUN TOR JUN CUN CUN CUN CUN CUN ANA  | A,4<br>A,3<br>5<br>2,00H<br>ERTST<br>30H<br>TRANO<br>NDLAY<br>00H<br>TROSC<br>NEXTC<br>3,00H<br>C,02   | #STORE DATA INTO 84  #LOAD THE MEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCHG/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMING #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED HAS #AN LRC CHARACTER IF ZERO #START TRANSFERRING OF THE #"O"-LRC CHARACTER  |
| 0327<br>0328<br>0329<br>032A<br>032C<br>032C<br>032F<br>0333<br>0334<br>0335<br>0334<br>0335<br>0336   | 5504<br>2C00<br>4000<br>5305<br>2D00<br>6200<br>4705<br>4830<br>2500<br>4705<br>7B34<br>FE85<br>08340<br>8F31<br>6300<br>7C02<br>FF28  | i<br>i<br>i<br>nextc:<br>i<br>trano:      | IND RUT CUR IND CUR ANA AUT SUT JRR CUN TOR TOR OUT SUT CUN  | A,4 A,3 5 2,00H ERTST 30H 5 1,05H TRANO NDLAY 00H INCSC NEXTC 3,00H C,02 SDLAY ERTST   | #STORE DATA INTO 84  #LOAD THE WEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCHS/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMING #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED HAS #AN LRC CHARACTER IF ZERO #START TRANSFERRING OF THE #SET BYTE COUNTER  #RESET CCHS/ID, SET *CRCG  |
| 0327<br>0328<br>0329<br>0328<br>032C<br>032E<br>032F<br>0333<br>0334<br>0333<br>0333<br>0333<br>0333   | 5504<br>2C00<br>4000<br>5505<br>2D00<br>6200<br>6200<br>6705<br>9B36<br>FEB5<br>0500<br>8340<br>7C02<br>FF20<br>FEBF<br>4611   | i i i i t TRANO:                          | IND RUT CUN ANA AUT SUT CUN TOR JER CUN TOR JUN CUN CUN CUN CUN CUN ANA  | A,4<br>A,5<br>5<br>2,00H<br>ERTST<br>30H<br>5<br>1,05H<br>TRANO<br>NOLAY<br>00H<br>INCSC<br>NEXTC<br>3,00H<br>C,02<br>SDLAY<br>ERTST<br>11H                  | STORE DATA INTO 84  LOAD THE WEST OF THE DATA CHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD HALF OF "O"-LRC CHARACTER.  SET ENRST AND CCHS/ID IF NOT IN SPACE MODE SET-UP FOR 2,5 CHARACTER PERIODS GAP TIMINS "O"-LRC IF READ REQUEST EQUAL TO 1 DELAY FINISHED? THE FIRST CHARACTER RECEIVED HAS AN LRC CHARACTER IF ZERO START TRANSFERRING OF THE "O"-LRC CHARACTER SET BYTE COUNTER   |
| 0327<br>0328<br>0329<br>0326<br>032C<br>032E<br>032F<br>0330<br>0331<br>0332<br>0333<br>0334<br>0338<br>0338                                 | 5504<br>2C00<br>4000<br>5305<br>2D00<br>6200<br>4705<br>7B36<br>FE85<br>6300<br>8340<br>8F31<br>6300<br>7C02<br>FF28<br>4611<br>2500<br>FE8F<br>4611<br>2500   | i<br>i<br>i<br>nextc:<br>i<br>trano:      | RUT CUR RUT CUR ANA AUT SUT JER CUN TOR TOR JEE JUN OUT CUN ANA AUT CUN ANA AUT CUN ANA AUT CUN  | A,4 A A,5 5 2,00H ERTST 30H 5 1,05H TRANO NDLAY 00H INCSC NEXTC 3,00H C,02 SDLAY 11H 5 TRANB   | #STORE DATA INTO 84  #LOAD THE MEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCHG/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMING #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED HAS #AN LRC CHARACTER IF ZERO #START TRANSFERRING OF THE #"O"-LRC CHARACTER #SET BYTE COUNTER  #RESET CCHG/ID, SET *CRCG #FIRST TRANSFERRING OF FIRST #OATA BYTE   |
| 0327<br>0328<br>0329<br>032A<br>032C<br>032C<br>032C<br>0333<br>0334<br>0333<br>0334<br>0338<br>0338<br>0339<br>0338<br>0338                 | 5504<br>2000<br>5305<br>2000<br>6200<br>FEBF<br>4630<br>6705<br>9836<br>FE85<br>0500<br>8340<br>7C02<br>FF20<br>FE8F<br>4611<br>2500<br>FF47<br>FF20<br>6905   | ;<br>;<br>;<br>NEXTC:<br>;<br>TRANO:      | INO RUT CUN ANA AUT SUT JUN CUN CUN ANA AUT CUN  | A,4 A A,5 S 2,00H ERTST 30H S 1,05H TRANO NDLAY 00H INCSC NEXTC 3,00H C,02 SDLAY ERTST 11H S TRANB   | #STORE DATA INTO 84  #LOAD THE WEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCHS/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMING #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED HAS #AN LRC CHARACTER IF ZERO #START TRANSFERRING OF THE #"O"-LRC CHARACTER #SET BYTE COUNTER  #RESET CCHG/ID, SET *CRCG #F NOT IN SPACE MODE #START TRANSFERRING OF FIRST #DATA BYTE #SHORT DELAY #SET 2,5 PERIODS GAP TIMING   |
| 0327<br>0328<br>0329<br>0328<br>032C<br>032E<br>032F<br>0330<br>0331<br>0332<br>0333<br>0334<br>0335<br>0336<br>0338<br>0338                 | 5504<br>2C00<br>4000<br>5505<br>2D00<br>6200<br>6200<br>6705<br>9B36<br>FEB5<br>0500<br>8340<br>7C02<br>FF20<br>FEB0<br>4611<br>2500<br>FF47<br>FF20   | ;<br>;<br>;<br>NEXTC:<br>;<br>TRANO:      | IND RUT CUN ANA AUT SUT JUN CUN SUT CUN  | A,4 A,5 5 2,00H ERTST 30H TRANO NDLAY 00H INCSC NEXTC 3,00H C,02 SDLAY ERTST 11H 5 TRANB   | #STORE DATA INTO 84  #LOAD THE MEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCHG/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMING #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED HAS #AN LRC CHARACTER IF ZERO #START TRANSFERRING OF THE #"O"-LRC CHARACTER #SET BYTE COUNTER  #RESET CCHG/ID, SET *CRCG #FIRST TRANSFERRING OF FIRST #DATA BYTE #SHORT DELAY #SET 2,5 PERIODS GAP TIMING #START TRANSFERRING OF   |
| 0327<br>0328<br>0329<br>0326<br>0326<br>0327<br>0330<br>0331<br>0333<br>0334<br>0335<br>0336<br>0337<br>0338<br>0338<br>0338                 | 5504<br>2C00<br>4000<br>5305<br>2D00<br>6200<br>6200<br>6705<br>9B36<br>FE85<br>6300<br>8340<br>8F31<br>6300<br>7C02<br>FF20<br>FE8F<br>4411<br>2500<br>FF47<br>FF20<br>6905<br>BECB<br>7C01                                 | ;<br>;<br>;<br>NEXTC:<br>;<br>TRANO:      | IND RUT CUR ANA AUT SUT JERR CUN TOR TOR SUT CUN ANA AUT CUN CUN CUN ANA AUT CUN ANA AUT SUT CUN ANA AUT CUN SUT SUT SUT SUT SUT | A,4 A A,5 5 2,00H ERTST 30H 5 1,05H TRANO NDLAY 00H INCSC NEXTC 3,00H C,02 SDLAY 11H 5 TRANB SDLAY 1,05H C,01H   | #STORE DATA INTO 84  #LOAD THE MEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCHG/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMING #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED HAS #AN LRC CHARACTER IF ZERO #START TRANSFERRING OF THE #SET BYTE COUNTER  #RESET CCHG/ID, SET *CRCG #F NOT IN SPACE MODE #START TRANSFERRING OF FIRST #DATA BYTE #SHORT DELAY #SET 2,5 PERIODS GAP TIMING #FIRST TRANSFERRING OF #INEXT DATA BYTE #INEXT DATA BYT |
| 0327<br>0328<br>0329<br>0328<br>032C<br>032C<br>032E<br>0337<br>0333<br>0334<br>0338<br>0338<br>0338<br>0338<br>0338<br>0338                 | 5504<br>2000<br>4000<br>5305<br>2000<br>6200<br>FEBF<br>4630<br>2500<br>6705<br>7836<br>FE85<br>0500<br>8340<br>7C02<br>FF20<br>FE8F<br>46300<br>FF47<br>FF20<br>6905<br>BECB<br>7C01<br>FF47                                | i<br>i<br>i<br>MEXTC:<br>I<br>TRANO:<br>i | IND RUT CUR IND CUR ANA AUT  SUT JER CUN TOR JEE JUN CUN ANA AUT  CUN CUN ANA CUN  | A,4 A A,5 5 2,00H ERTST 30H TRANO NDLAY 00H INCSC 3,00H C,02 SDLAY ERTST 11H TRANB   | #STORE DATA INTO 84  #LOAD THE WEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCHS/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMING #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED WAS #AN LRC CHARACTER IF ZERO  #START TRANSFERRING OF THE #"O"-LRC CHARACTER #SET BYTE COUNTER  #RESET CCHG/ID, SET *CRCG ##FINISHED OF FIRST ##POT IN SPACE MODE ##POT IN SPACE M |
| 0327<br>0328<br>0329<br>0328<br>032C<br>032E<br>032F<br>0330<br>0331<br>0332<br>0333<br>0334<br>0338<br>0338<br>0338<br>0338<br>0338<br>0338 | 5004<br>2000<br>4000<br>5305<br>2000<br>4200<br>4705<br>7834<br>FE85<br>0500<br>8340<br>8F31<br>4300<br>7C02<br>FF20<br>4911<br>2500<br>FF47<br>FF20<br>4905<br>8BCB<br>7C01<br>FF47<br>FF20                                 | i<br>i<br>i<br>mextc:<br>i<br>trano:<br>i | IND RUT CUR ANA AUT SUT JER CUN ANA AUT SUT CUN ANA AUT CUN  | A,4 A A,5 5 2,00H ERTST 30H TRANO NOLAY 00H INCSC NEXTC 3,00H C,02 SDLAY ERTST 11H 5 TRANB SDLAY 1,05H C,04T C,04T TRANB                                     | STORE DATA INTO 84   LOAD THE WEST OF THE DATA   CHARACTER     STORE DATA INTO SS   RESET READ REQUEST. LOAD   HALF OF "O"-LRC CHARACTER.     SET ENRST AND CCHS/ID IF NOT   IN SPACE MODE   SET-UP FOR 2,5 CHARACTER PERIODS   GAP TIMINS   "O"-LRC IF READ REQUEST EQUAL TO 1     DELAY FINISHED?   THE STORE RECEIVED WAS   AN LRC CHARACTER RECEIVED WAS   AN LRC CHARACTER IF ZERO     START TRANSFERRING OF THE   "O"-LRC CHARACTER   SET BYTE COUNTER     RESET CCHS/ID, SET *CRCG   IF NOT IN SPACE MODE   START TRANSFERRING OF FIRST   SHORT DELAY   SET 2,5 PERIODS GAP TIMING   START TRANSFERRING OF   INSTANT TRANSFERRING OF   INEXT DATA BYTE   LOAD BYTE COUNTER     TRANSFER LRC CHARACTER   TRANSFER LRC CHARACTER   TRANSFER LRC CHARACTER   TRANSFER LRC CHARACTER  |
| 0327<br>0328<br>0329<br>0326<br>0326<br>0327<br>0330<br>0331<br>0332<br>0333<br>0334<br>0335<br>0336<br>0337<br>0338<br>0338<br>0338<br>0338 | 5504<br>2C00<br>4000<br>5305<br>2D00<br>6200<br>6200<br>6705<br>9B36<br>FE85<br>6300<br>8340<br>8F31<br>6300<br>7C02<br>FF20<br>FE8F<br>4411<br>2500<br>FF47<br>FF20<br>6905<br>BECB<br>7C01<br>FF47<br>FF20<br>FE85<br>FF47 | i<br>i<br>i<br>mextc:<br>i<br>trano:<br>i | RUT CUR RUT CUR ANA AUT SUT JER CUN TOR TOR SUT CUN ANA AUT CUN CUN ANA AUT CUN  | A,4 A A,5 5 2,00H ERTST 30H 5 1,05H TRANO NDLAY 00H INCSC NEXTC 3,00H C,02 SDLAY ERTST 11H 5 TRANB SDLAY L,05H TRANB SDLAY ERTST C,01H TRANB SDLAY ERTST 11H | STORE DATA INTO 84  LOAD THE WEST OF THE DATA CHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD HALF OF "O"-LRC CHARACTER.  SET ENRST AND CCHS/ID IF NOT IN SPACE MODE SET-UP FOR 2,5 CHARACTER PERIODS GAP TIMING "O"-LRC IF READ REQUEST EQUAL TO 1 DELAY FINISHED? THE FIRST CHARACTER RECEIVED HAS HAN LRC CHARACTER IF ZERO START TRANSFERRING OF THE SET BYTE COUNTER  RESET CCHS/ID, SET *CRCG IF NOT IN SPACE MODE START TRANSFERRING OF FIRST DATA BYTE SHORT DELAY SET 2,5 PERIODS GAP TIMING START TRANSFERRING OF NEXT DATA BYTE LOAD BYTE COUNTER THE LRC CHARACTER THE LRC CHARACTER SHORT DELAY   |
| 0327<br>0328<br>0329<br>0328<br>032C<br>032E<br>032F<br>0333<br>0333<br>0333<br>0335<br>0338<br>0339<br>0338<br>0338<br>0338<br>0338         | 5504<br>2C00<br>4000<br>5505<br>2D00<br>6200<br>6200<br>6705<br>9B36<br>FE85<br>0500<br>8340<br>7C02<br>FF20<br>6400<br>FF47<br>FF20<br>6905<br>BEC01<br>FF47<br>FF20<br>FF47<br>FF20<br>FF87<br>FF88                        | i<br>i<br>NEXTC:<br>i<br>TRANO:<br>i<br>i | IND RUT CUN ANA AUT SUT JUN CUN CUN CUN CUN CUN SUT CUN SUT CUN  | A,4 A A,5 5 2,00H ERTST 30H 5 1,05H TRANO NDLAY 00H INCSC SDLAY ERTST 11H 5 TRANB SDLAY 1,05H CDATT C,01H TRANB  | #STORE DATA INTO 84  #LOAD THE WEST OF THE DATA #CHARACTER #STORE DATA INTO SS #RESET READ REQUEST. LOAD #HALF OF "O"-LRC CHARACTER.  #SET ENRST AND CCHS/ID IF NOT #IN SPACE MODE #SET-UP FOR 2,5 CHARACTER PERIODS #GAP TIMINS #"O"-LRC IF READ REQUEST EQUAL TO 1 #DELAY FINISHED? #THE FIRST CHARACTER RECEIVED HAS #AN LRC CHARACTER IF ZERO #START TRANSFERRING OF THE #"O"-LRC CHARACTER #SET BYTE COUNTER  #RESET CCHS/ID, SET *CRCG #START TRANSFERRING OF FIRST #DATA BYTE #SHORT DELAY #SET 2,5 PERIODS GAP TIMING #START TRANSFERRING OF #INEXT DATA BYTE #LOAD BYTE COUNTER #THE LRC CHARACTER #THE LRC CHARACTER #THE LRC CHARACTER #RESET CCHG/ID, SET *CRCG #FRESET CCHG/ID,  |
| 0327<br>0328<br>0329<br>0326<br>0326<br>0327<br>0330<br>0331<br>0332<br>0333<br>0334<br>0335<br>0336<br>0337<br>0338<br>0338<br>0338<br>0338 | 5504<br>2C00<br>4000<br>5305<br>2D00<br>6200<br>6200<br>6705<br>9B36<br>FE85<br>6300<br>8340<br>8F31<br>6300<br>7C02<br>FF20<br>FE8F<br>4411<br>2500<br>FF47<br>FF20<br>6905<br>BECB<br>7C01<br>FF47<br>FF20<br>FE85<br>FF47 | i<br>i<br>i<br>mextc:<br>i<br>trano:<br>i | RUT CUR RUT CUR ANA AUT SUT JER CUN TOR TOR SUT CUN ANA AUT CUN CUN ANA AUT CUN  | A,4 A A,5 5 2,00H ERTST 30H 5 1,05H TRANO NDLAY 00H INCSC NEXTC 3,00H C,02 SDLAY ERTST 11H 5 TRANB SDLAY L,05H TRANB SDLAY ERTST C,01H TRANB SDLAY ERTST 11H | STORE DATA INTO 84  LOAD THE WEST OF THE DATA CHARACTER STORE DATA INTO SS RESET READ REQUEST. LOAD HALF OF "O"-LRC CHARACTER.  SET ENRST AND CCHS/ID IF NOT IN SPACE MODE SET-UP FOR 2,5 CHARACTER PERIODS GAP TIMING "O"-LRC IF READ REQUEST EQUAL TO 1 DELAY FINISHED? THE FIRST CHARACTER RECEIVED HAS HAN LRC CHARACTER IF ZERO START TRANSFERRING OF THE SET BYTE COUNTER  RESET CCHS/ID, SET *CRCG IF NOT IN SPACE MODE START TRANSFERRING OF FIRST DATA BYTE SHORT DELAY SET 2,5 PERIODS GAP TIMING START TRANSFERRING OF NEXT DATA BYTE LOAD BYTE COUNTER THE LRC CHARACTER THE LRC CHARACTER SHORT DELAY   |

0302 8F24

RREVH: J7T

REVERSE SECTION; BECAUSE THE DIFFERENCE BETWEEN THE 9-TRACK

23-28



### 23.16 DREND SEQUENCE

The DREND sequence terminates all read and write operations, first by generating the post-record delay and then, at RAMPD, by ending tape motion. While the tape is decelerating, the INREQ signal is tested for the presence of a new command, in which case the program is ready for an "on-the-fly" operation.

The post-record delay value is fetched from the SF register which was loaded during the START (write delays) and RCOMD (read delays) sequences. The timer is reprogrammed to a 20 usec clocking period.

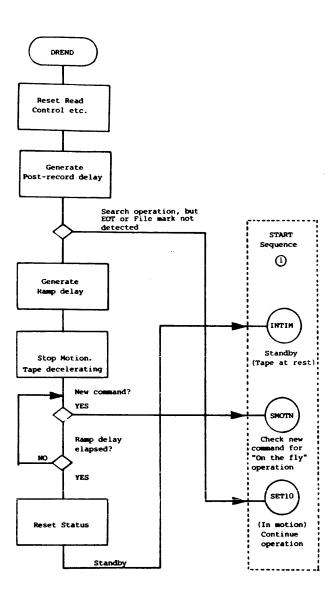
In the case of a search operation, this operation is continued by returning the program to an appropriate point in the START sequence unless a file mark or, in forward motion, an EOT marker has been detected, in which case the search operation is terminated. The detection of a file mark would have been registered in the SA register during the PREAD (PE) or NZGAP (NRZI) sequences. A watch for the EOT marker is maintained at all times, and a detection of it is registered in the SA register by the DELAY, EOTCH and NDLAY subroutines.

The NFILS entry point is provided for the GPCNT subroutine to terminate the operation if the gap detected is too long.

A ramp delay is generated after which it is assumed that the tape will have stopped. In the case of an "on-the-fly" new command, the program is returned to an appropriate point in the START sequence where decoding the command takes place.

Finally, if there is no new command, the program is returned almost to the very beginning of the START sequence, retaining only the format status information. In the START sequence, the program resets various control and indicator registers and advances to a point where it sits waiting for a new command.

FUNCTIONAL FLOW CHART:



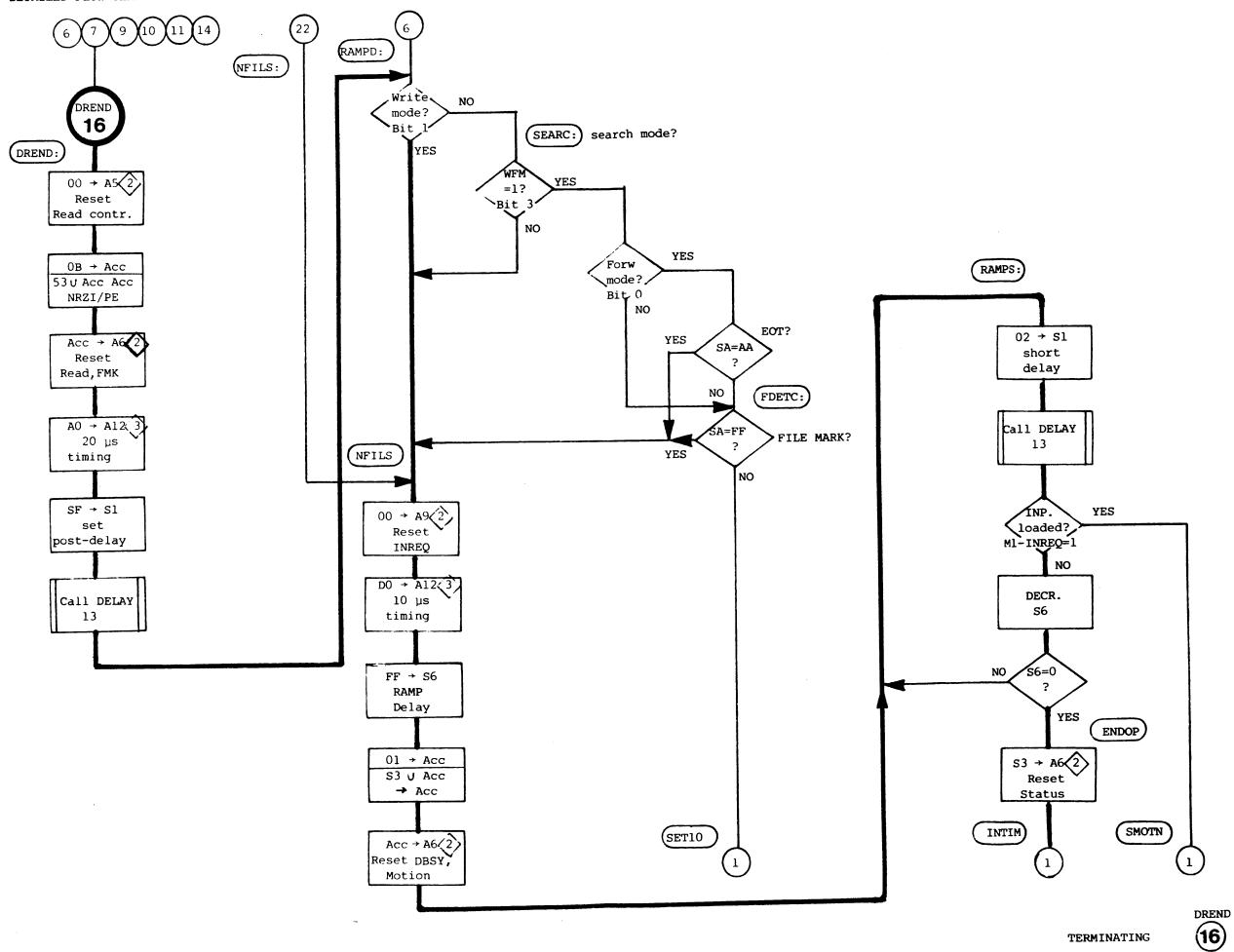
| 037C          | 4000         | DRENU:  | CLR | 5,00H<br>A | FRESET READ CONTROL REGISTER   |
|---------------|--------------|---------|-----|------------|--|
| 039D          | 4508         |         | ORA | OBH        | ILOAD RESET READ AND FINK  |
| 039E          | 5D03         |         | INO | S.3        | LOAD CORRECT HODE  |
| 039F          | 2600         |         | AUT | 3,5        | RESET READ AND FMK DETECTED  |
| 03A0          | 74A0         |         | OUT | C.OAOH     | ISET-UP FOR 20 MICROSECONDS TIMING   |
| 03A1          | 4000         |         | CLR | A          | TOTAL CONTROL OF THE PARTY OF T |
| 03A2          | 500F         |         | INO | S.OFH      | ILOAD POST-DELAY REGISTER  |
| 03A3          | 2900         |         | RUT | 1          | ISTORE POST DELAY  |
| 03A4          | FCD5         |         | CUN | DELAY      |  |
| 03A5          | BFA6         |         | JUN | RAMPD      |  |
| 03A6          | 4000         | RAMPD:  | CLR | A          |  |
| 03A7          | 5000         |         | INO | S.0        |  |
| 03A8          | 0602         |         | TAN | 0ŽH        | (TEST UPON WRITE MODE ( BIT 1 )  |
| 03A9          | 83BC         |         | JZE | SEARC      | INO WRITE MODE IF ZERO   |
|               | 7100         | NFILS:  | OUT | 7,00H      | RESET INREG.   |
| 03AB          | 74D0         |         | OUT | C.ODOH     | SET 10 MICROSECONOS TIMING   |
| 03AC          | 6EFF         |         | SUT | 6,0FFH     | SET RAMP DELAY   |
| 03AD          | 4000         |         | CLR | <b>A</b>   |  |
| 03AE<br>03AF  | 4501         |         | ORA | 01H        | ILOAD WITH RESET MOTION AND DESY   |
| 0380          | 5003<br>2400 |         | INO | 8,3        | ISET CORRECT MODE  |
| 0381          | 4902         | RAMPS:  | SUT | 6          | RESET MOTION AND DBSY  |
| 0382          | FCD5         | TOWN S. | CUN | 1,02H      | SHORT DELAY  |
| 03B3          | 8406         |         | JIL | DELAY      |  |
| 03B4          | 5004         |         | INO | SMOTH      | I "ON-THE-FLY" OPERATION?  |
| 03B5          | 4200         |         | SUB | S,4        | DECREMENT SA   |
| 03B6          | 8389         |         | JZE | ENDOP      | RAMP DOWN TIME ELAPSED IF ZERO   |
| 03B7          | 2E00         |         | RUT | 6          | TRAFF DOWN TIPE ELAPSED IF ZERO  |
| 03B8          | BFB1         |         | JUN | RAMPS      |  |
| 0389          | 5D03         | ENDOP:  | INO | S.3        |  |
| 03BA          | 2600         |         | AUT | 6          | RESET STATUS REGISTER  |
| 03BB          | BC02         |         | JUN | INTIM      | READY FOR NEW OPERATION  |
|               |              |         |     |            |  |
| 03BC          | 0408         | SEARCH: | TAN | 08H        | (TEST UPON WITH MODE ( BIT 3 )   |
| 03BD          | BJAA         |         | JZE | NFILS      | INO SEARCH IF ZERO   |
| 03BE          | 4601         |         | ANA | 01H        | TEST UPON FORMARD MODE ( BIT 0 )   |
| 03 <b>B</b> F | 83C4         |         | JZE | FDETC      | IGNORE EOT DETECTED IF REVERSE   |
|               |              | i       |     |            | MODE. REVERSE MODE IF ZERO   |
| 03C0          | 4000         |         | CLR | A          |  |
| 03C1          | 5D0^         |         | INO | S,OAH      | ILOAD SEARCH INDICATOR   |
| 03C2          | 04           |         | TXR | 0AAH       | EOT DETECTED?  |
| 03 C3         | BSAA         | EDET OF | JZE | NFILS      | EOT DETECTED IF ZERO   |
| 03C4          | 5DOA         | FDETC:  | INO | S,OAH      | LOAD SEARCH INDICATOR  |
|               |              | i       |     |            | HE KNOW ACCUMULATOR IS ZERO  |
| 03C5<br>03C4  | 04FF<br>B3AA |         | TXR | OFFH       | FILEMARK DETECTED?   |
| 03C7          | 8C13         |         | JZE | NFILS      | FILEMARK DETECTED IF ZERO  |
| V 3 C/        | BC12         |         | JUN | SET10      | IONCE MORE!  |

5.00H :RESET READ CONTROL BESTETES

037B 4500

23-30

DETAILED FLOW CHART:



# 23.20 DELAY SUBROUTINE

The DELAY subroutine will generate a delay determined by a preselected timer period generated by the timing counter (U120, 121, 3) and a multiplication factor normally stored in scratch pad registers S1 and S2. The least significant part of the number (LSB) is stored in S1, and the most significant one (MSB) in S2. The time base, or timer period, is preset by using strobe A12. See also paragraph 22.3.4-3. In connection with the latter paragraph, a detailed description will be given:

There are essentially two loops, one decrementing (SUB 0) the LSB part of the count number and the other decrementing the MSB.

In the first loop the LSB byte is decremented after every timer period generated by the timing counter U120-121. The timing counter sets timing latch U44 (see schematic 3) which is tested (M4-TIME) by instruction JNT. The LSB byte is returned (RUT 1) to location S1 every time after having been decremented. Eventually, the LSB byte is reduced to zero, and the program branches into the second loop (JZE DECRS).

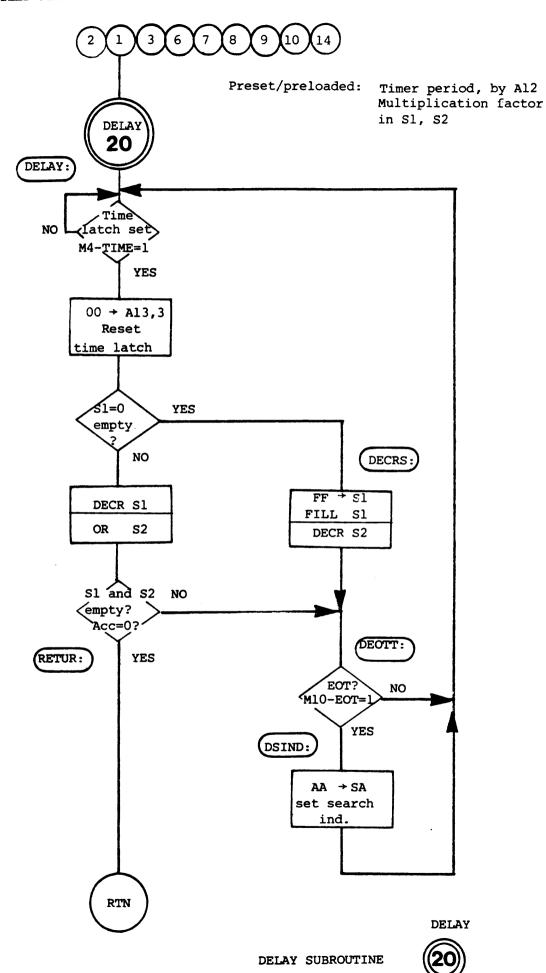
In the second loop Sl is first loaded with ones (SUT 1, FF). The MSB byte in location S2 is decremented, and the program reenters first loop after only one cycle. This first loop must now be circulated 256 times before entering the second loop again, and the just described sequence in the second loop is repeated, each time decrementing the MSB.

Every time the first loop is being circulated a test is being made in the ALU, OR'ing the MSB with the LS3 stored in accumulator (INO S,2). Only when both bytes have been reduced to zero will the program break out of the loops (JZE RETUR) and return to the main program.

On every passage through the loops a test on EOT is made (JEO). If the EOT is detected, the search indicator is set (AA loaded into loacation S10).

If there is a File Search operation, the tape must stop in the next gap after having passed the EOT mark.

```
0005 9005 DELAY! JNT DELAY HAITING FOR TIME INTERRUPT
0004 7500 OUT D,00H RESET TIME INTERRUPT
0007 4000 CLR A
0008 5D01 INO S,1
0009 8060 JZE DECRS
0004 4200 SUB 0
0008 2700 RUT 1
000C 5D02 INO S,2
00DD 80DF JZE RETUR
00DF 1000 RETUR: RTN
00DF 1000 RETUR: RTN
00E0 49FF DECRS: SUT 1,0FFH ;REMEMBER, S1 MAS ZERO
00E1 5D02 INO S,2
00E2 4200 SUB 0
00E3 2A00 RUT 2
00E4 ASE6 DEOTT: JEO DSIND
00E5 7AAA DSIND: SUT A,0AAH JUN DELAY
00E6 7AAA DSIND: SUT A,0AAH JUN DELAY
```

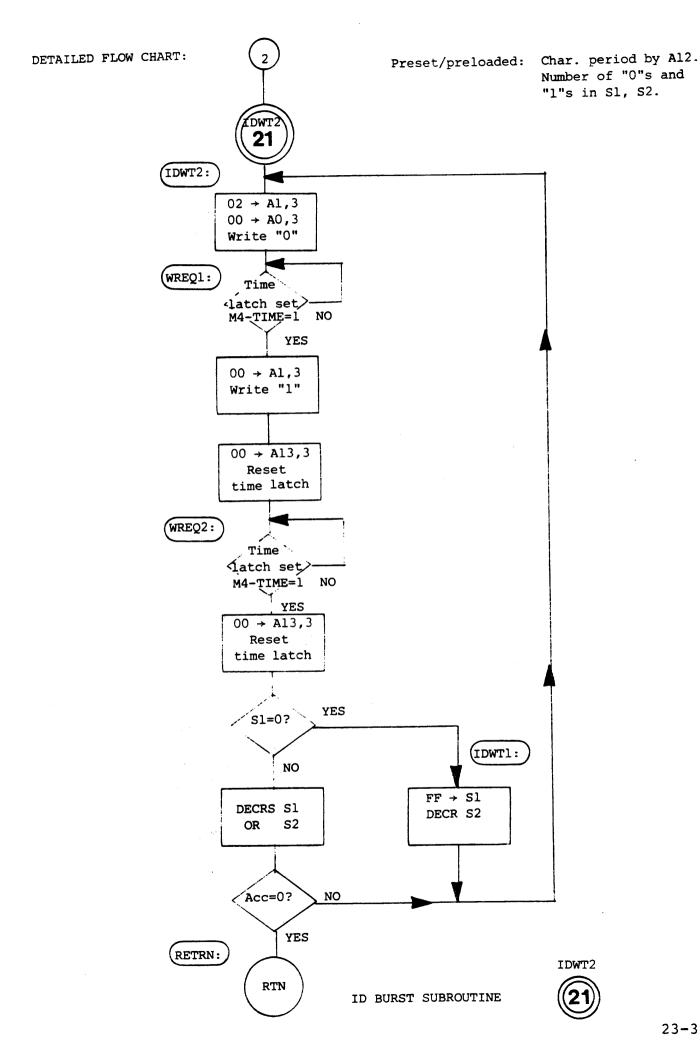


#### 23.21 IDWT2 SUBROUTINE

The IDWT2 subroutine will generate "0"s and "1"s for the number of times which is stored in the Sl and S2 scratch pad registers and is used for writing PE ID Bursts. The character period is determined by the timer counter setting (strobed by Al2). The Sl, S2 number is counted down in a way similar to the DELAY subroutine (Paragraph 23.20).

The inverse polarity is used for the "0"s and "1"s, and the subroutine starts with writing a "0" in the P-channel, then proceeds to write a "1", to decrement the S1, S2 registers and finally to repeat the action until the registers are empty.

| 0048 | 6102   | IDNT2: | OUT | 1.02H  | 1"0" DATA BIT IN P-CHANNEL        |
|------|--------|--------|-----|--------|-----------------------------------|
| 0049 | 4000   |        | OUT | 0.00H  | RESET ALL DATA CHANNELS ( NOT P ) |
| 004A | 904A   | WREG1: | JNT | WREG1  | HAIT FOR TIMING INDICATOR         |
| 004B | 6100   |        | OUT | 1,00H  | ;"1" DATA BIT IN P-CHANNEL        |
| 004C | 7500   |        | OUT | D,00H  | RESET TIME INDICATOR              |
| 004D | 904D   | WREQ2: | JNT | WREQ2  | HAITING FOR TIMING INDICATOR      |
| 004E | 7500   |        | OUT | D.00H  | RESET TIME INDICATOR              |
| 004F | 4000   |        | CLR | A      |                                   |
| 0050 | 5 DO 1 |        | INO | S, 1   |                                   |
| 0051 | 8058   |        | JZE | I DWT1 |                                   |
| 0052 | 4200   |        | SUB | 0      | DECREMENT ACC                     |
| 0053 | 2900   |        | RUT | 1      | STORE SI                          |
| 0054 | 5D02   |        | INO | 5.2    | IOR WITH 52                       |
| 0055 | 8057   |        | JZE | RETRN  |                                   |
| 0054 | BC48   |        | JUN | I DWT2 |                                   |
| 0057 | 1000   | RETRN: | RTN |        |                                   |
| 0058 | 69FF   | IDWT1: | SUT | 1,0FFH | THIS IS EQUIVALENT TO             |
|      |        | ;      |     |        | DECREMENTING THE SI BY ONE        |
| 0059 | 5 DO 2 |        | INO | 5,2    | HE KNOW ACC IS ZERO NOW           |
| 005A | 4200   |        | SUB | 0      | DECREMENT S2                      |
| 0058 | 2A00   |        | RUT | 2      |                                   |
| 0050 | RCAR   |        | JUN | IDMT2  |                                   |



#### 23.22 GPTST SUBROUTINE

The GPTST subroutine will make a halt in the program (not the tape) until a gap (absence of noise and signals) is detected which is of a certain minimum specified period (length). This period is 6 times the timer period times the gap count number stored in scratch pad register SB. The timer period is half a character period.

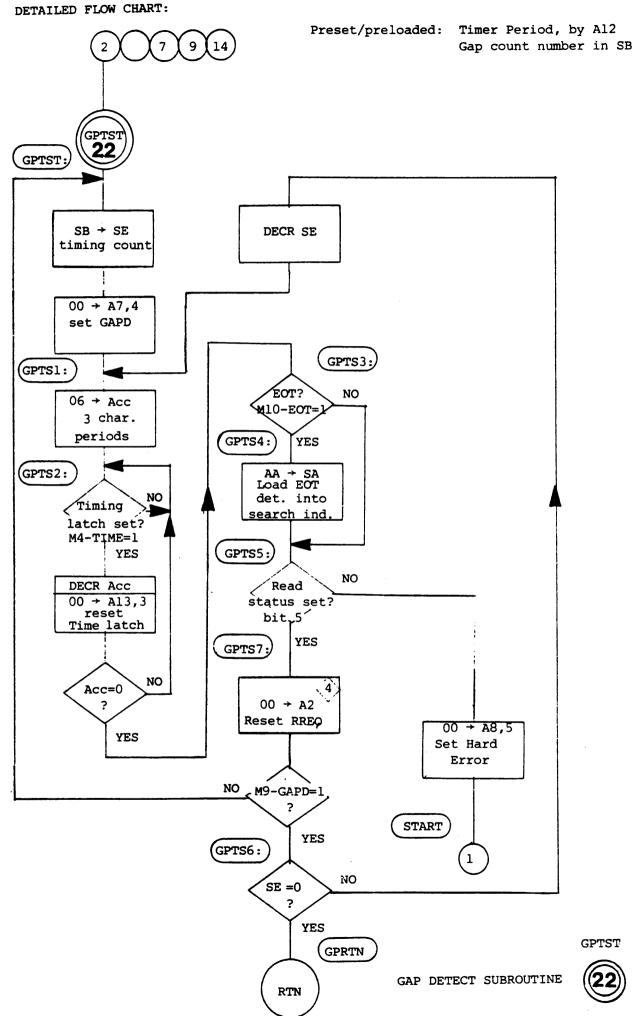
To preserve the SB value, it is transferred to the SE register for the operational countdown.

The subroutine operates by setting the GAPD flip-flop and then waits to see if signals or noise will reset it within 6 timer periods. If so, it will try again. If not, it will, unless the SE gap count number has been decremented to zero, see if the flip-flop stays set another 6 timer periods. It will try as many 6 timer period loops as the SE number allows. When the gap count is zero, it returns to the main program.

During this test, the presence of the EOT marker is being checked. There is no chance of missing the EOT marker since this check repeats within microseconds whereas the presence of the EOT marker, if passing, lasts for milliseconds. If the EOT marker is detected, search indicator SA is set which in a File Search operation makes it possible to end the operation in the next gap.

The read line is also being monitored since it would not be possible to detect a gap if this line is false. In that case the program is returned to the START sequence to avoid tape run-out.

| 0082 | 4000         | GPTST:    | CLR | A      |   |
|------|--------------|-----------|-----|--------|---|
| 0083 | 500B         | <u> </u>  | INO | S.OBH  | ILOAD GAP COUNT REGISTER  |
| 0084 | 3E00         |           | RUT | Ē.     | STORE GAP COUNT NUMBER  |
| 0085 | 4700         |           | OUT | 7.00H  | SET GAP DETECTED SIGNAL   |
| 0086 | 4000         | GPTS1 :   | CLR | A      |   |
| 0087 | 4505         |           | ORA | 05H    | SET-UP FOR 2,5 CHARACTER PERIODS  |
| 0088 | 9088         | GPTS2 :   | JNT | GPTS2  | HAIT FOR TIMING INDICATOR   |
| 0089 | 7500         |           | OUT | D.00H  | RESET TIMING INDICATOR  |
| ABOO | 420C         |           | SUB | 0      | DECREMENT ACC.  |
| 0088 | 808D         |           | JZE | GPTS3  |   |
| 008C | BC88         |           | JUN | GPTS2  |   |
| 008D | A878         | GPTS3:    | JEO | GPTS4  | ¡EOT?   |
| 008E | 5507         | GPTS5:    | INO | A.7    | ; WE KNOW THAT ACC. IS ZERO   |
| 008F | 4420         |           | ANA | 20H    | MASK OUT ALL BITS EXCEPT BIT 5  |
| 0070 | 8093         |           | JZE | GPT\$7 | TREAD LINE IS SET IF ZERO, WHICH INDICATES  |
| 0071 | 7000         |           | OUT | 8,00H  | ;OPERATION.<br>;SET HARD ERROR LINE. READ LINE<br>:IS NOT SET, WHICH MEANS THAT WE                    |
|      |              | ;         |     |        | CANNOT PERFORM A GAP TEST OPE-<br>RATION. SOMETHING IS WRONG, AND<br>TO AVOID RUN-OUT THE MOTION LINE |
|      |              |           |     |        | IAND THE MAIN REGISTERS ARE RESET.  |
| 0072 | BC00         | •         | JUN | START  | THE THE PERSON NAMED IN CO.   |
| 0072 | 6200         | GPTS7:    | OUT | 2,00H  | RESET READ REQUEST. THIS IS   |
| 0073 | 8200         | 9713/     | 001 | 2,000  | IDONE TO AVOID LOCK-UP PROBLEMS IN  |
|      |              | :         |     |        | INRZI MODE IF NOISE TRIGGERS THE READ   |
|      |              | :         |     |        | ELECTRONICS IN THE TAPE DRIVE AFTER   |
|      |              | :         |     |        | THE NORMAL READING IS FINISHED.   |
|      | A496         | •         | 1eb | GPTS4  | GAP DETECTED IF JUMP  |
| 0074 | BCB2         |           | JUN | GPTST  | TRY ONCE HORE   |
|      |              | GPTS4:    | INO | S.OEH  | ILOAD GAP TEST COUNTER  |
| 0096 | 500E<br>809D | GP   56 1 | JZE | GPRTN  | ICOND GMF 1531 COMMEN   |
| 0077 |              |           | SUB | 0      | IDECREMENT SO   |
| 0078 | 4200         |           |     | Ē      | ISTORE SB   |
| 0077 | 3E00         |           | RUT | SPTS1  | 101AUC 88   |
| 0074 |              | COTO .    | SUT | A.OAAH | ILOAD SEARCH INDICATOR  |
| 007B | 7            | GPTS4:    |     | 6PTSS  | IPAMA MPHILAL SIMPALIA.   |
| 009C | 8C8E         | ### T     | JUN | 97153  |   |
| 007D | 1000         | GPRTN:    | RTN |        |   |



### 23.23 DENST SUBROUTINE

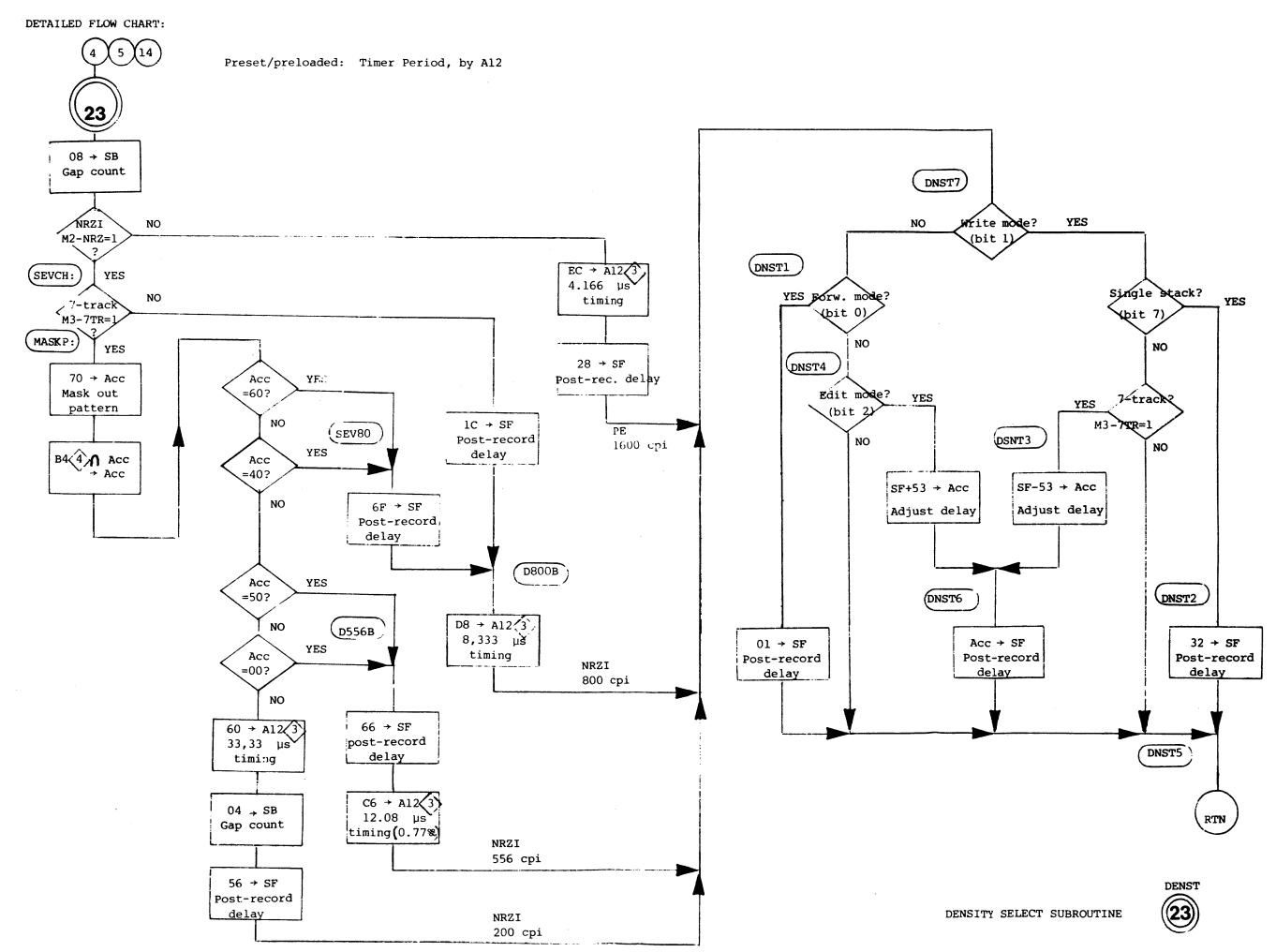
The DENST subroutine decodes the density command (TDDI) and the K, H strap configuration (Table 2, sch. 2) and selects appropriate time base for the bit rate timing counter.

Whereas this time base is accurate for 200, 800 and 1600 bpi, it deviates by 0.77% for 556 bpi (to 551.8 bpi), due to the fact that the system clock frequency is not an integral multiple of the 556 bpi bit rate, subject to fractional division. The 0.77% deviation, however, is neglible compared to other sources of timing errors in the transport itself.

To provide sufficient timing resolution in NRZI and time slots for phase transitions in PE, the time base period chosen is half the character period.

The DENST subroutine also selects the proper post-record delay value according to selected mode of operation and stores this value in the SF register for use later in the DREND sequence. The values are based on a timer period of 20 usec.

| 00E8         | 7808          | DENST:  | SUT        | B,08H           | ISET GAP COUNTER   |
|--------------|---------------|---------|------------|-----------------|--|
| 00E7         | 8705          |         | JNZ        | BEVCH           | INRZI MODE?  |
| OOEA         | 74EC          |         | OUT        | C,0ECH          | ISET-UP FOR 4,047 MICROSECONOS TIMINS                    |
| OOEB         | 7F20          |         | SUT        | F,20H           | ILOAD PE POST-DELAY                                      |
| OOEC         | 4000          | DNST7 : | CLR        | <u> </u>        |  |
| OOED         | 5000          |         | INO        | 5,0             |  |
| OOEE         | 0402          |         | TAN        | 02H             | NURITE HODE ( SIT 1 )?                                   |
| OOEF         | BOF4          |         | JZE        | DNBT1           | IREAD HOOE IF ZERO                                       |
| 00F0<br>00F1 | 0480          |         | TAN        | 80H             | ISINGLE STACK HODE ( BIT 7 )?                            |
|              | BOFF          |         | JZE        | DNST2           | SINGLE STACK IF ZERO                                     |
| 00F2         | 9001<br>1000  |         | JTT        | DNBT3           | 17-TRACK?  |
| ovr 3        | 1000          |         | RTN        |                 | IPOST-DELAY OK.  |
| 00F4         | 0401          | DNST1:  |            |                 |  |
| 00F5         | BOFB          | Out 11: | JZE        | 01H<br>DNST4    | IFORMARD HODE ( BIT 0 ) ?                                |
| OOFA         | 7F01          |         | SUT        |                 | INEVERSE MODE IF ZERO                                    |
| 00F7         | 1000          |         | RTN        | F,01H           | ISET MINIMUM PORT-DELAY ( EQU. TO 0 )                    |
| OOFB         | 0404          | DNST4 I | TAN        | 04H             | SEDIT MODE ( BIT 2 )?                                    |
| OOFT         | BOFE          | 5145144 | JZE        | DNBT5           | INO EDIT IF ZERO   |
| OOFA         | 4000          |         | CLR        | . 6.00.0        | ING EDT! IN TEMO   |
| OOFB         | SDOF          |         | INO        | S.OFH           | ILOAD POST-DELAY   |
| OOFC         | 4353          |         | ADD        | 53H             | HADJUST FOR EDIT MODE                                    |
| OOFD         | 3F00          | DNST4:  | RUT        | F               | ISTORE NEW POST-DELAY NUMBER                             |
| OOFE         | 1000          | DNSTS   | RTN        | •               | 1510ME MEN POSITUELAT MURBER                             |
| OOFF         | 7F32          | DNST2 : | SUT        | F,32H           | POST-DELAY, SINGLE STACK WRITE                           |
| 0100         | 1000          |         | RTN        | . ,             | ALANIA DECILIOZIANE BINCH MILLE                          |
| 0101         | 4000          | DNBT3:  | CLR        | A               |  |
| 0102         | 5D0F          |         | INO        | S.OFH           | ILOAD POST-DELAY   |
| 0103         | 4252          |         | 9U8        | 52H             | IADJUST FOR 7-TRACK DUAL                                 |
|              |               | 3       |            |                 | ISTACK WRITE OPERATION                                   |
| 0104         | <b>BCFD</b>   |         | JUN        | DNST4           |  |
|              |               | 3       |            |                 |  |
| 0105         | 8 DO 7        | SEVCH   | J7T        | MASKP           | 17-TRACK OPERATION?                                      |
| 0106         | 7F1C          |         | SUT        | F,1CH           | INRZI 9-TRACK READ REVERSE                               |
|              |               | 3       |            |                 | FOST-DELAY   |
| 0107         | 7408          | D6008:  | QUT        | C,00 <b>0</b> H | ISET 8,333 MICROSECONDS TIMING                           |
| 0108         | BCEC          |         | JUN        | DNST7           |  |
| 0109         | 4000          | MASKP   | CLR        | <u> </u>        |  |
| 010A         | 4570          | _       | ORA        | 7011            | ITHIS IS FOR MASKING OUT                                 |
|              |               | Ŧ       |            |                 | FALL BITS EXCEPT BIT 4, 5 & 6.                           |
| 0108<br>010C | 5404          |         | INA        | 4,4<br>40H      | MASK OUT ALL BITS EXCEPT 4, 5 & 6                        |
| 010C         | 0460<br>8118  |         | TXR        |                 | THIS IS A 800 BPI INDICATION                             |
| 010E         | 0440          |         | JZE        | SEVEO           | 17-TRACK 800 BPI IF ZERO                                 |
| 010E         | 0440<br>6118  |         | TXR<br>JZE | 40H<br>8EV80    | THIS IS A 800 BPI INDICATION                             |
| 0110         | 0450          |         | TXR        | 50H             | 77-TRACK 800 BPI IF ZERO<br>THIS IS A 554 BPI INDICATION |
| 0111         | 8118          |         | JZE        | D5548           | 17-TRACK 554 BPI IF ZERO                                 |
| 0112         | 0400          |         | TXR        | 00H             | THIS IS A 554 BPI INDICATION                             |
| 0113         | 8118          |         | JZE        | 05548           | 17-TRACK 556 BPI IF ZERO                                 |
| 0114         | 7440          |         | OUT        | C. 40H          | ISET 33.33 MICROSECONDS TIMIMS                           |
| 0115         | 7804          |         | SUT        | 8.04H           | ISET GAP COUNT NUMBER                                    |
| 0116         | 7F54          |         | SUT        | F,56H           | ISET FOR 7-TRACK 200 BPI POST-                           |
|              |               |         |            | . ,             | IDELAY   |
| 0117         | BCEC          | •       | JUN        | DNST7           | TOCOM  |
|              |               |         |            |                 |  |
| 0118         | 7 <b>F</b> 66 | D5548:  | SUT        | F.66H           | ILOAD 7-TRACK 556 BPI POST-DELAY                         |
| 0117         | 74C6          |         | OUT        | C.OC6H          | ISET 12,08 MICROSECONDS TIMING.                          |
|              |               |         |            | •               | THE CORRECT VALUE SHOULD HAVE                            |
|              |               | i       |            |                 | BEEN ABOUT 11,79 MICROSECONDS.                           |
| 011A         | BCEC          |         | JUN        | DNET7           | •  |
| 0118         | 7F6F          | SEV80:  | SUT        | F,4FH           | LOAD 7-TRACK 800 BPI POSTDELAY                           |
| 011C         | <b>BD07</b>   |         | JUN        | 00008           |  |
|              |               | į       |            |                 |  |
|              |               |         |            |                 |  |



#### 23.24 GPCNT SUBROUTINE

The GPCNT subroutine will monitor the length (duration) of a gap, and if the gap exceeds a certain maximum, the tape will stop, preventing tape run-out. The program will return to the START sequence (via the NFILS sequence), ready for a new command as soon as the Formatter Busy signal FBSY goes false. Since it is actually time and not distance that is being monitored, this action will also prevent any hang-up in he case of standing at BOT after a read reverse operation, in which no data has been detected.

A test is first made to determine if the EOT has been detected and if the operation is a search-for-file mark operation. In that case the subroutine will abort the operation if data has not yet been detected.

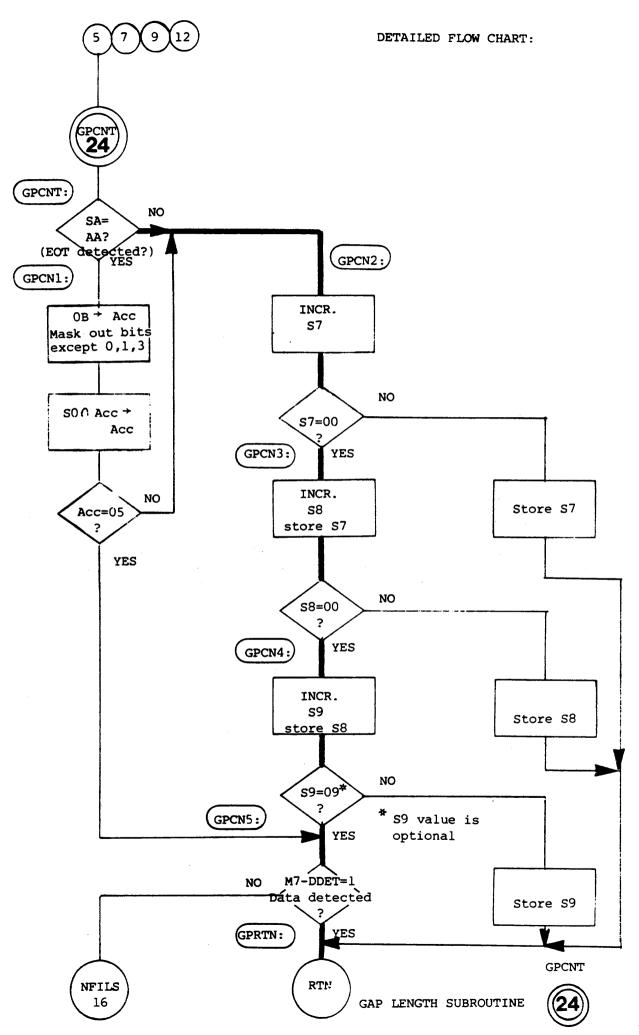
The GPCNT subroutine is normally used in a program loop and measures time by incrementing scratch pad register S7, S8 and S9 every time it is cycled through. Register S8 is incremented every time S7 exceeds 255 counts, and S9 is incremented every time S8 exceeds 255 counts. The subroutine may thus be cycled through

a maximum of  $256 \times 256 \times 256$  times, and the time elapsed is the time it takes to run through the sequence steps at the system clock frequency this many times.

This represents a maximum of about 143 meters (470 feet) if the S9 is allowed to reach a count of FF, but a maximum of 0.5 meters (1.6 feet) if the count allowed is only 01. In comparison, the maximum gap length according to international standards is 7.62 meters (25 feet).

In the TDF4050, the maximum allowable gap length is programmed to be about 4.92 meter (16 feet) by allowing a maximum S9 count of 09. This will give enough tape left, when passing EOT, to avoid a tape run-out.

| 0148 | 4000        | GPCNT:  | CLR | ۸       |   |
|------|-------------|---------|-----|---------|---|
| 014C | 5D0A        |         | INO | 5,0AH   | ILOAD SEARCH INDICATOR REGISTER                               |
| 014D | 4444        |         | XRA | DAAH    | ISA - AA?   |
| 014E | <b>0162</b> |         | JZE | GPCN1   | ISA - AA IF ZERO  |
| 014F | 4000        | GPCN2 : | CLR | A       |   |
| 0150 | 5007        |         | INO | 5.7     |   |
| 0151 | 4301        |         | ADD | оін     | INC ST  |
| 0152 | 8155        |         | JZE | 6PCN3   | 12.12   |
| 0153 | 2F00        |         | RUT | 7       |   |
| 0154 | 1900        |         | RTN |         | IMAXIMUM GAP NOT PASSED YET                                   |
| 0155 | 2F00        | GPCN3:  | RUT | 7       | ISTORE S7   |
| 0156 | 5 DO B      |         | INO | 5,8     |   |
| 0157 | 4301        |         | ADD | 01H     | INCR. SE  |
| 0158 | 8158        |         | JZE | GPCN4   | 12  |
| 0159 | 3800        |         | RUT | 8       |   |
| 015A | 1000        |         | RTN | -       | MAXIMUM GAP NOT YET PARRED                                    |
| 0158 | 3800        | GPCN4 I | RUT | 8       | ISTORE SA   |
| 015C | 5 D0 9      |         | INO | 5,9     | 75.5  |
| 015D | 4301        |         | ADD | ОÍН     |   |
| 015E | 0409        |         | TXR | 09H     | ITEST WITH MAX. GAP = 4,92 METERS                             |
| 015F | 8167        |         | JZE | GPCN5   | MAXIMUM LENGTH OF GAP HAS PARRED                              |
|      |             | i       |     | G. 0.10 | LIF ZERO  |
| 0160 | 3900        | •       | RUT | 9       | 11r 1Eno  |
| 0161 | 1000        | GAPRN:  | RTN | •       |   |
| 0162 | 450B        | GPCN1 : | ORA | 08H     | THE KNOW ACC. TO TERM WERE                                    |
| 0163 | 5E00        | G. C    | INA | S.O     | INE KNOW ACC. IS ZERO MERE<br>IMASK OUT ALL BITS EXCEPT 0 ( = |
|      |             |         |     | 3,0     |   |
|      |             | i       |     |         | FORMARD MODE ), BIT 1 ( = WRITE                               |
| 0164 | 4405        | •       | XRA | 05H     | MODE ) AND BIT 3 ( "SEARCH MODE ).                            |
|      |             |         |     | VJA     | TEST UPON FORHARD MODE, SEARCH MODE                           |
| 0165 | 8167        | •       | JZE | CDCNE   | AND READ MODE.  |
|      |             | ;       | JIE | GPCN5   | FORWARD, SEARCH AND READ MODE                                 |
| 0166 | BD4F        | •       | JUN | GPCN2   | ITF ZERO  |
| 0167 | 9 Dá 1      | GPCN5:  | JOT | GAPRN   |   |
|      |             | i i     | 301 | GAPKN   | OK IF DATA HAS BEEN DETECTED                                  |
| 0168 | BFAA        | •       | JUN | METIC   |   |
|      | J. 7411     |         | JUN | NFILS   | OPERATION ABORTED   |
|      |             |         |     |         |   |



#### 23.25 EOTCH SUBROUTINE

The EOTCH subroutine loads search indicator register SA with AA if EOT is detected.

| 0134 | A936                         | EOTCH: | JEO               | EOTDT  |      |        |           |
|------|------------------------------|--------|-------------------|--------|------|--------|-----------|
|      | 1000<br>7 <b>AAA</b><br>1000 | EOTDT: | RTN<br>SUT<br>RTN | A,OAAH | LOAD | SEARCH | INDICATOR |

# 23.26 COMWT SUBROUTINE (incl. COMMW)

The COMWT subroutine contains one part generating a bit period delay and a subsequent part loading the output write register with an all "0"s byte followed by another bit period delay. It is primarily used for writing preamble or postamble all "l"s (of the inverse polarity).

If the write commands have been activated, the subroutine will write an all "l"s data byte, and the bit period delay will serve to delay further action until the write strobe TWDS has been generated. The first bit period delay will serve the same purpose for write data which may have been loaded into the write register prior to calling the subroutine.

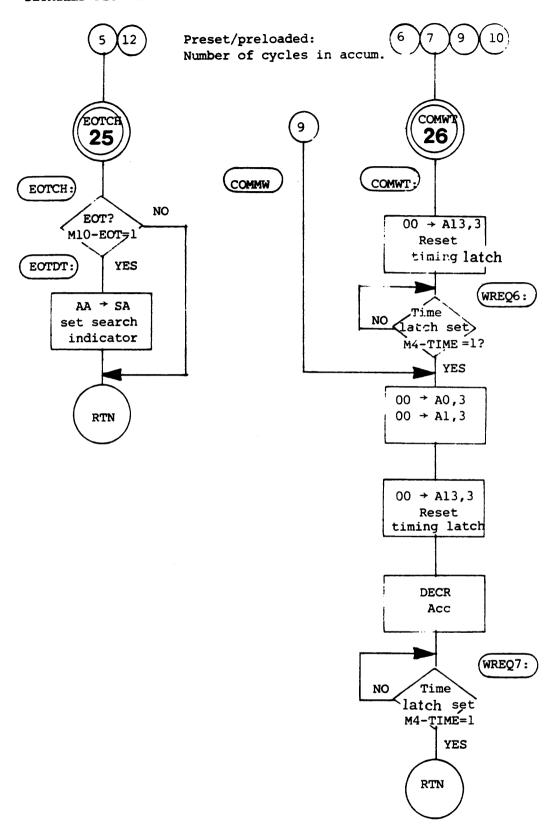
The COMWT may be used to write phase transitions as well as data transitions, and the entry point COMMW makes it possible to skip the first part if a bit period delay is not needed.

If the write commands are not activated, the subroutine can be used to generate short delays.

The decrement instruction is of use in writing file marks. A loop is included in the program, and the accumulator is used to specify the desired number of identical bits.

| 0185 | 7500 | COMMT:  | OUT | D,00H | RESET TIMING INDICATOR       |
|------|------|---------|-----|-------|------------------------------|
| 0186 | 7186 | WREG6 : | JNT | HREQ6 | HAITING FOR TIME INDICATOR   |
| 0187 | 6000 | COMMIN: | OUT | 0,00H |                              |
| 0188 | 6100 |         | OUT | 1,00H | ILOAD ALL "1""S.             |
| 0187 | 7500 |         | OUT | D,OOH | RESET TIMING INDICATOR       |
| 018A | 4200 |         | SUB | 0     | DECREMENT ACCUMULATOR        |
| 0188 | 7188 | WREQ7:  | JNT | WREQ7 | HAITING FOR TIMING INDICATOR |
| 018C | 1000 |         | RTN |       |                              |

# DETAILED FLOW CAHRTS:





**EOTCH** 



WRITE ONES SUBROUTINE

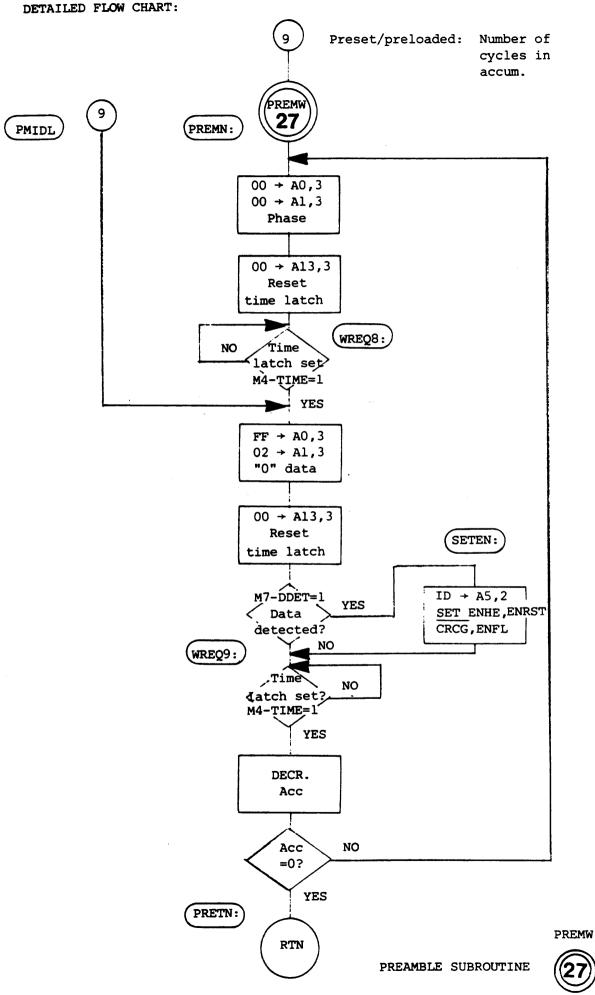
EOT DETECT SUBROUTINE

# 23.27 PREMW SUBROUTINE (incl. PMIDL)

The PREMW subroutine is similar to the COMWT subroutine and is used to write preamble and postamble all "0"s. The first part writes phase transitions and the second part, entry PMIDL, writes data transitions, using inverted polarities. Whereas COMWT for repetitive use depends on an outside program loop, such a loop has been included in the PREMW subroutine, which will write as many "0"s as the accumulator has been programmed to and then return to the main program.

In addition, PREMW includes a routine, SETEN, which will activate the error detection circuits as soon as the read head begins reading data.

| 01A6  | 4000   | PREHI    | OUT | 0.00H  |                               |
|-------|--------|----------|-----|--------|-------------------------------|
|       |        | 11001101 | OUT | 1.00H  | ISET PHASE TRANSITION         |
| 01 A7 | 6100   |          |     |        | RESET TIMING INDICATOR        |
| 01 A8 | 7500   |          | OUT | D,00H  | INESE! FINIS THE GUACE        |
| 01A9  | 91A9   | WREQ8:   | JNT | WREQE  | HAITING FOR THE PHASE         |
| ***** |        |          |     |        | TRANSITION TO BE WRITTEN.     |
| 01 AA | 60FF   | PMIDL:   | OUT | 0.OFFH | •                             |
|       |        | 111200   | OUT | 1,02H  | ILOAD "O"-DATA TO BE          |
| 01 AB | 6102   |          | 001 | 1,0211 | WRITTEN                       |
|       |        | •        |     |        | WATER THE THE TANK TO THE     |
| 01AC  | 7500   |          | OUT | D,00H  | RESET TIME INDICATOR          |
| DIAD  | 9 DB 3 |          | JDT | SETEN  | CATA DETECTED?                |
| ATHO  | ,,,,,  |          |     |        | THIS WILL NORMALLY ONLY       |
|       |        | •        |     |        | HAPPEN DURING POSTAMBLE       |
|       |        | ī        |     |        | WRITE OPERATION               |
|       |        | ï        |     |        | INKTIE OLEMNITOM              |
| 01AE  | 91AE   | HREQ7:   | TNL | WREGP  | HAITING FOR THE DATA TRAN-    |
| V 2   |        |          |     |        | ISITION TO BE WRITTEN.        |
|       |        | •        | SUB | 0      | DECREMENT ACC.                |
| 01AF  | 4200   |          |     | -      | OPERATION FINISHED IF ZERO    |
| 0180  | 8182   |          | JZE | PRETN  | INFERMITOR LISTERS TO THE     |
| 0181  | BDA6   |          | JUN | PREHI  |                               |
| 0182  | 1000   | PRETN:   | RTN |        |                               |
|       |        |          |     | E 1 NU | ISET ENFL, ENRST, ENHE & +CRC |
| 0183  | 651D   | SETEN:   | OUT | 5,1 DH | 195: 5451 44461 4446 4        |
| 01B4  | BDAE   |          | JUN | WREQT  |                               |
|       |        |          |     |        |                               |

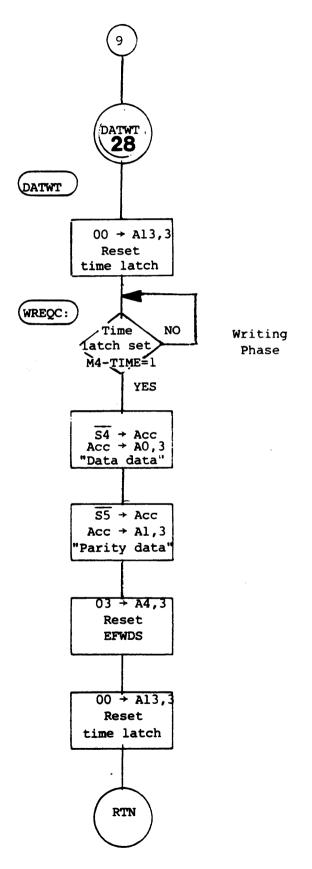


### 23.28 DATWT SUBROUTINE

The DATWT subroutine also resembles the COMWT subroutine to a certain degree, but is used to write PE data bits in general, not just "l"s. The first part of the subroutine generates the bit period delay which provides write strobe generation for the phase transition byte previously loaded into the write registers. In the second part this byte in inverted form is fetched from scratch pad register S4, S5 and transferred to the write registers to serve as the data transition byte. The bit period delay needed to secure the generation of write strobes is not present here, but in the main program.

| 0185                 | 7500           | DATUT: | OUT | D.00H | RESET TIME INDICATOR                            |
|----------------------|----------------|--------|-----|-------|---|
| 0184                 | <b>918</b> 6   | WREGC: | TNL | HREGC | HALTING FOR THE PHAGE TRANSITION TO BE WRITTEN. |
| 0187                 | 4700           |        | SET | A     | ILOAD FF TO ACC.                                |
| 018 <b>0</b><br>0187 | 5 CO 4<br>2000 |        | INX | 9,4   | ILOAD COMPLEMENT OF 84 INTO ACC.                |
|                      |                | •      |     |       | ILOADED   |
| 01BA                 | 4700           |        | SET | A     |   |
| 0188                 | 5 C 0 5        |        | INX | 5,5   | ILOAD COMPLEMENT OF 85 INTO ACC.                |
| 01BC                 | 2100           |        | AUT | 1     | CORRECT BIT FOR PARITY                          |
|                      |                | 1      |     |       | ITRANSITION LOADED.                             |
| 018D                 | 6403           |        | OUT | 4,03H | RESET EFWOS                                     |
| 018E                 | 7500           |        | OUT | D.00H | RESET TIME INDICATOR                            |
| 01BF                 | 1000           |        | RTN | •     |   |

Preset/preloaded:
Bit period delay by Al2
Data into S4, S5



DATWT



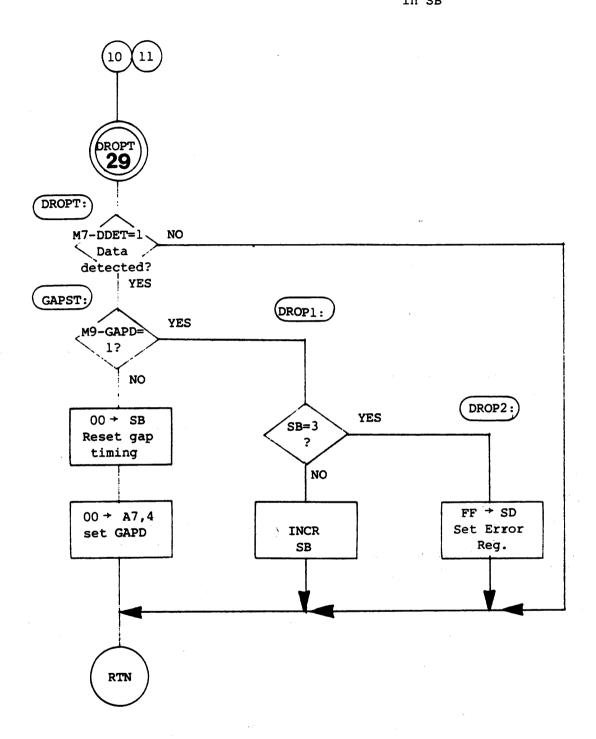
#### 23.29 DROPT SUBROUTINE

The DROPT subroutine is used to detect drop-outs during NRZI write operations. No distinction has been made in this respect for single stack write operations, since the read circuitry in this case is not active anyway.

The DROPT sequence is in effect bypassed until a data block has been
detected. Then, a gap exceeding the
time it takes to cycle through 3 DROPT
calls in the program constitutes a dropout. Since the DROPT subroutine
normally is followed by a bit delay, this
will in practice mean a gap of 3-4 bit
periods.

| 0276<br>0277 | 9E78  | DROPT:  | JDT<br>RTN | <b>GAPST</b> |                                  |
|--------------|-------|---------|------------|--------------|----------------------------------|
| 0278         | A67C  | GAPST:  | JGD        | DROP1        | IGAP DETECTED?                   |
| 02/6         |       | GATSI   |            |              |                                  |
| 0279         | 7B00  |         | SUT        | B.00H        | RESET GAP COUNT REGISTER         |
| 027A         | 6700  |         | OUT        | 7,00H        | ISET GAPD                        |
| 0278         | 1000  |         | RTN        |              |                                  |
| 027C         | 4000  | DROP1 : | CLR        | A            |                                  |
| 0270         | 5 DOB |         | INO        | S,08H        |                                  |
| 027E         | 0403  |         | TXR        | 03H          |                                  |
| 027F         | 8283  |         | JZE        | DROP2        | IGAP DETECTED SIGNAL HAS BEEN ON |
|              |       | i       |            |              | IFOR A TOO LONS TIME IF ZERO NOW |
| 0280         | 4301  |         | ADD        | 01H          | INCREMENT ACC                    |
| 0281         | 3B00  |         | RUT        | 8            | ISTORE GAP TEST COUNT REGISTER   |
| 0282         | 1000  |         | RTN        |              |                                  |
|              |       |         |            |              |                                  |
| 0283         | 7 DFF | DROP2:  | SUT        | D,OFFH       | SET ERROR REGISTER               |
| 0284         | 1000  |         | RTN        |              |                                  |
|              |       |         |            |              |                                  |

Preset/preloaded: Number of gaps in SB



The RWDAT subroutine is used to transfer read data to the host computer during a NRZI write operation. During a NRZI read-only operation the RDATA subroutine is used. Only one byte at a time is transferred.

Data is fetched from the tape transport bus (TRDx) and transferred to the read output register (FRDx) in two parts. The first part consists of channels P, 1, 2 and 3 and is brought into the accumulator by strobe B4 and from there into the read register by strobe A2. The remaining bits are brought in by strobe B5 and outputted by A3. These bits occupy lines DBO-4 on the Data Bus.

Strobe B5 also inputs bits to the DB5-7 lines. These are the same bits as were inputted by B4 except that channels P and O have been combined. The latter channels are dc-erased (i.e. "O") in NRZI file marks and shall both be false in the qualification for file mark detected. The complete byte strobed in by B5 will therefore suffice for file mark detection without the need for using the B4 in addition, thus simplifying such an operation later.

Scratch pad registers S4 and S5 are used for accruing LRC parity information for each channel. These registers are initially 0, but are exclusive-ored in the accumulator with the read data strobed in by B4, B5. The registers are then loaded with the result, every time RWDAT is cycled through, such that the bits in these registers at any moment indicate whether there has been an even or odd number of "1"s up to this moment. At the

end of the data block these bits will constitute a computed LRC character which can be used to check the recorded LRC character.

The unused bits DB4-DB7 in the B4 input operation, are filled in with "1"s in the accumulator and are being exclusive-ored with the corresponding contents of the S4 register which initially is 0. These bits will therefore alternate between 0 and 1 every time RWDAT is cycled, and thus indicate whether there is an odd (if "1"s) or even (if "0"s) number of bytes in the block.

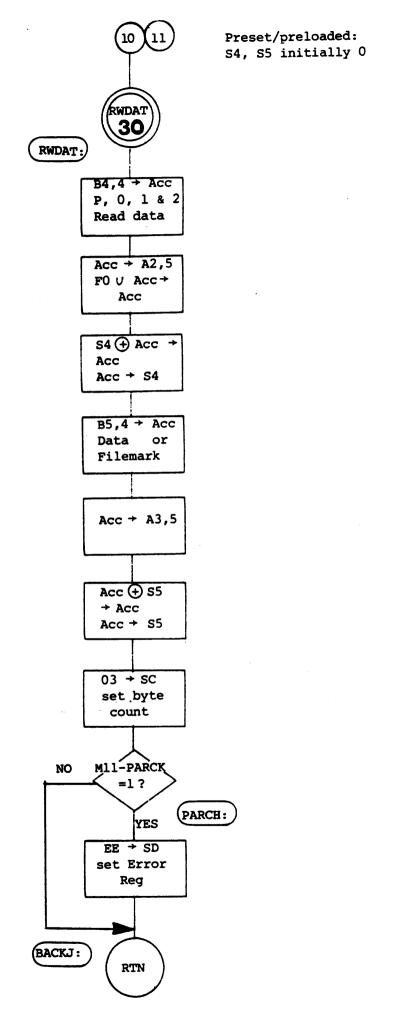
One way the RWDAT sequence differs from the RDATA sequence is in the use of the SC register. The SC register is used in the RDATA subroutine to count the number of bytes in the data block which should be a minimum of 3 in order to distinguish the block from a file mark. The SC register is then tested later. In a readafter-write operation this distinction is not important, and the SC is in the RWDAT subroutine straight off loaded with the qualifying count of 3. Consequently, the system will in a read-after-write operation accept a block consisting of one data byte and one LRC byte, but in contrast to a read-only operation not recognize this as a file mark if the byte has the same pattern as a file mark.

Vertical parity is being checked continously by Parity Checker U139 whose output M11-PARCK is being monitored by the RWDAT subroutine. If the computed parity deviates from the parity indicated by the P channel, error register SD is set.

| 022A<br>022B<br>022C<br>022D | 4000<br>5504<br>2200<br>45F0 | RHDAT:      | INO<br>AUT<br>ORA | A<br>A,4<br>2<br>0F0H |
|------------------------------|------------------------------|-------------|-------------------|-----------------------|
| 022E                         | 5 CO 4                       |             | INX               | 5,4                   |
|                              |                              | i           |                   |                       |
|                              |                              |             |                   |                       |
|                              |                              | •           |                   |                       |
|                              |                              | ;<br>;<br>; |                   |                       |
|                              |                              | ;           |                   |                       |
|                              |                              | i           |                   |                       |
| 022F                         | 2 C0 0                       | •           | RUT               | 4                     |
| 0230                         | 4000                         |             | CLR               | Ä                     |
| 0231                         | 5505                         |             | INO               | A,5                   |
|                              |                              |             |                   |                       |
| 0232                         | 2300                         |             | AUT               | 3                     |
| 0233                         | 5 CO 5                       |             | INX               | S,5                   |
|                              |                              | ī           |                   | •                     |
| 0234                         | 2 DO 0                       |             | RUT               | 5                     |
| 0235                         | 7 CO 3                       |             | SUT               | C,03H                 |
|                              |                              | •           |                   |                       |
| 0236                         | AE38                         |             | JPE               | PARCH                 |
| 0237                         | 8E37                         |             | JUN               | BACKJ                 |
| 0238                         | 7 DEE                        | PARCH:      | SUT               | D,0EEH                |
| 0239                         | 1000                         | BACKJ:      | RTN               |                       |

¡LOAD READ DATA, P 0, 1 & 2 ¡LOAD READ DATA TO READ REGISTER ¡MASK OUT BIT 4 - 7 ¡BY GENERATING THE EXCLUSIVE-OR OF ¡THIS DATA BYTE AND THE PRECEEDING ¡BYTE; ME CAN AFTERNARDS DETERMINE ¡IF IT IS AN EVEN OR AN ODD NUMBER ¡OF BYTES IN THE BLOCK ( BY ¡LOOKING AT BIT 4 - 7 ). ¡BITS 0 - 3 ARE EMPLOYED FOR CRC ¡AND LRC CHECK ROUTINES LATER ON. ¡STORE INFORMATION INTO S4

¡LOAD BITS 3-7 AND FILEMARK
¡PATTERN.
¡LOAD READ DATA REGISTER
¡THIS IS THE SAME OPERATION
¡AS WITH 54
¡STORE INFORMATION INTO SS
¡SET BYTE COUNTER TO 3 ( NO
¡FILEMARK )
¡PARITY ERROR?
¡NO PARITY ERROR
¡LOAD PARITY ERROR



R-A-W DATA TRANSFER SUBROUTINE

#### 23.31 RDATA SUBROUTINE

The RDATA subroutine is used to transfer read data to the host computer during a NRZI read-only operation. During a NRZI read-after-write operation the RWDAT subroutine is used, and the RDATA subroutine is very similar to that one, see Paragraph 23.30.

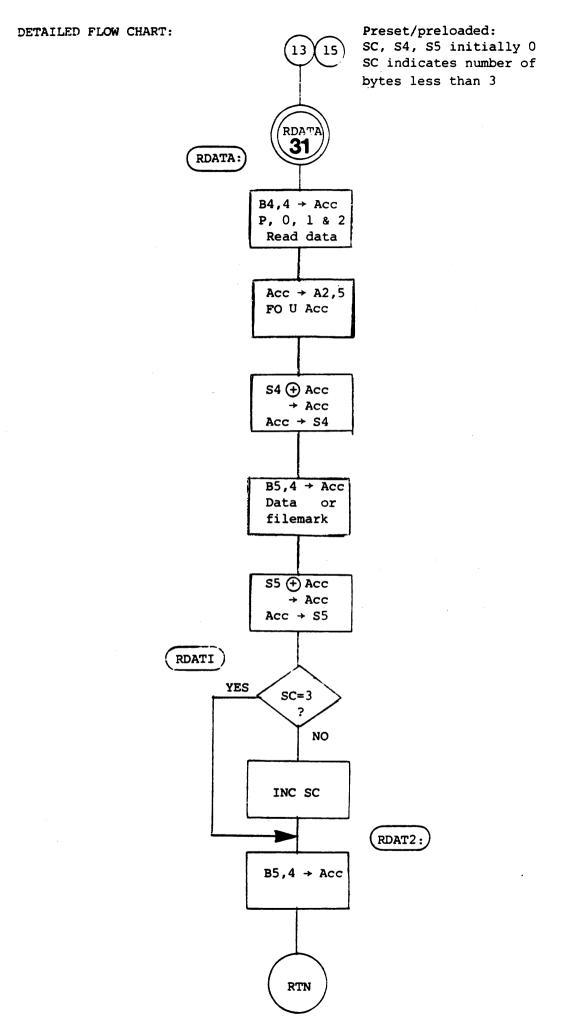
One difference is the omission of the A3 output operation for the second part of the read data byte. This operation is left to the main program in the TRANS subroutine (Paragraph 23.32) later to avoid too early a generation of the read strobe FRSTR, which starts automatically when the A3 strobe is used.

The number of bytes up to 3 is counted and stored in register SB for testing later. A one byte block is not allowed, and two bytes implies the possibility of a file mark. Three bytes or more qualifies as a data block.

The last operation is to refresh the accumulator with the B5 input read data which was destroyed by the exclusive-or operation. This data must be ready for outputting by A3 in the main program.

A vertical parity check is made in the TRANS subroutine.

| 02A7              | 4000   | ROATA    | CLR  | <b>A</b>   |                                  |
|-------------------|--------|----------|------|------------|----------------------------------|
| 02A8              | 5504   |          | INO  | A,4        | ILOAD READ DATA P.O.1 & 2        |
| 02A9              | 2200   |          | AUT  | 2          | JEGIN WEND ONLY L'A'T P S        |
| 02AA              | 45F0   |          | ORA  | ōFон       |                                  |
| 02AB              | 3 CO 4 |          |      |            | IMASK OUT SITS 4 - 7             |
|                   |        |          | INX  | 8,4        | EXCLUSIVE-OR S4 WITH ACC.        |
| 02AC              | 2 C0 0 |          | RUT  | 4          | ISTORE INFORMATION               |
| 02AD              | 4000   |          | CLR  | A .        |                                  |
| 02AE              | 5505   |          | INO  | A,5        |                                  |
|                   |        |          | 2110 | 710        | ILOAD THE REST OF THE READ DATA  |
|                   |        |          |      |            | IBYTE AND THE FILEMARK REGOGNIZE |
|                   |        | •        |      |            | PATTERN.                         |
| 02AF              | 5 CO 5 |          | INX  | 8.5        | EXCLUSIVE-OR SS WITH ACC.        |
| 02B0              | 2 D0 0 |          | RUT  | 8,5<br>5   | ATTOCATE ON SO MAIN MCC.         |
| 0281              | 4000   | RDAT1 I  | CLR  | Ă          |                                  |
| 02B2              | SDOC   |          |      |            |                                  |
| 0283              |        |          | INO  | S,OCH      |                                  |
|                   | 0403   |          | TXR  | 03H        | INO FILEMARK IF BC = 3           |
| 02 <del>8</del> 4 | 8287   |          | JZE  | RDAT2      | ISC = 3 IF ZERO                  |
| 9285              | 4301   |          | ADD  | 01H        | INCREMENT BC                     |
| 02B6              | 3 C00  |          | RUT  | Č.         | THE MENT OF                      |
| 02B7              | 4000   | RDAT2:   |      | -          |                                  |
|                   |        | HOW IS ! | CLR  | <b>A</b> ' |                                  |
| 0288              | 5505   |          | INO  | A,5        |                                  |
| 0287              | 1000   |          | RTN  | •-         |                                  |



RDATA

## 23.32 TRANS SUBROUTINE

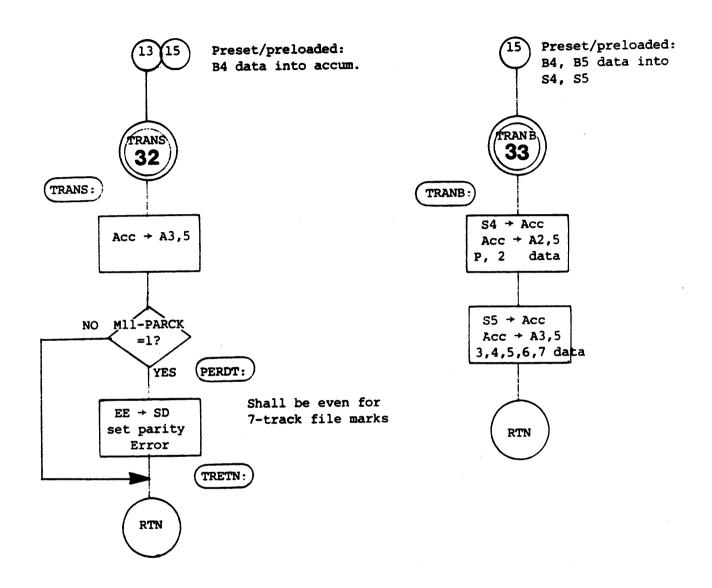
The TRANS subroutine outputs the second part of the NRZI read data loaded into the accumulator in the RDATA subroutine. It also performs a check on vertical parity identical to the one in the RWDAT subroutine.

| 02 <b>8A</b> | 2300         | TRANS  | AUT        | 3      | ISTORE READ DATA. START RDS- |
|--------------|--------------|--------|------------|--------|------------------------------|
| 0288<br>0280 | AEBD<br>1000 | TRETNI | JPE<br>RTN | PERDT  | IPARITY EVEN?                |
| 028E         | 7DEE<br>1000 | PERDT  | BUT<br>RTN | D,OEEH | LOAD ERROR REGISTER          |
|              |              | 1      |            |        |                              |

## 23.33 TRANB SUBROUTINE

The TRANB subroutine is employed during a read reverse NRZI 7-track operation to provide the transfer of the first received data byte (if the LRC byte is all "0"s) or the LRC byte (if the first byte is not all "0"s).

| 0347<br>0348<br>0349 | 4000<br>5004<br>2200 | TRANS | CLR<br>INO<br>AUT | \$,4<br>2 | ILOAD P AND 2 CHANGEL OF<br>IREAD DATA |
|----------------------|----------------------|-------|-------------------|-----------|--|
| 034A                 | 4000                 |       | CLR               | A .       |  |
| 034 <b>8</b><br>034C | 5D05<br>2300         |       | INO               | 8,5<br>3  | ILOAD REST OF READ DATA SYTE.          |
| 034D                 | 1000                 | •     | RTN               |           | ISTART RDS GENERATION                  |



TRANS

NRZI LAST PART SUBROUTINE



TRANB

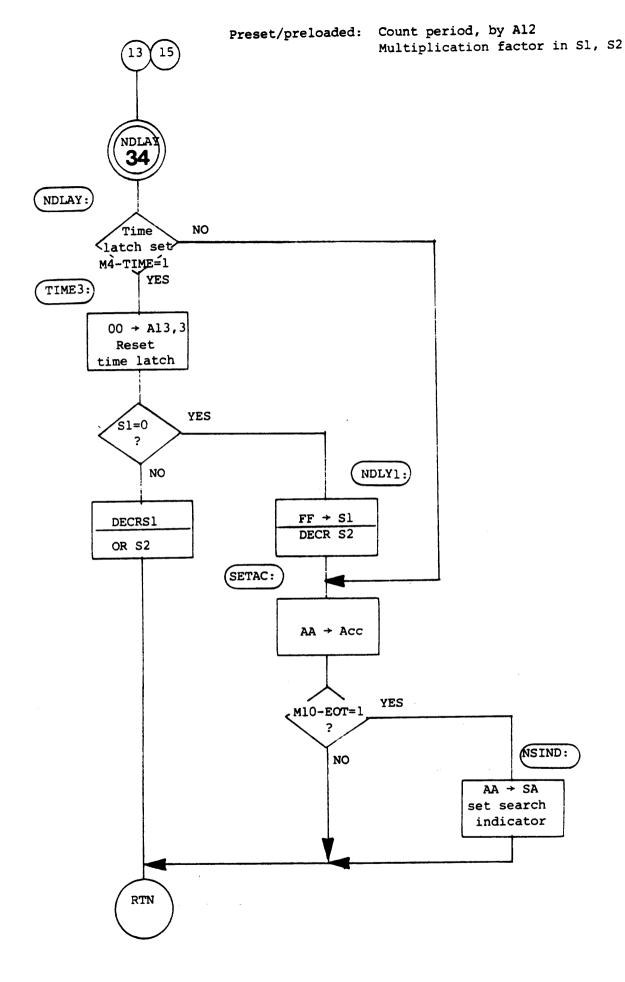


#### 23.34 NDLAY SUBROUTINE

The NDLAY subroutine will generate a delay in a manner very similar to that in the DELAY subroutine (see Paragraph 23.20), but it depends on a cycling loop in the main program outside the subroutine. In other words it must return to the main program and repeatedly be called up again in order to complete the delay.

This subroutine is used in NRZI operations in which read operations are completely under software control, and in which it is necessary to be able to branch out before the delay is completed, if circumstances so dictate.

| 0285 | 9292   | NDLAYI | JNT | SETAC    | TIME INDICATOR SET?                     |
|------|--------|--------|-----|----------|---|
| 0286 | 7500   |        | OUT | D.00H    | RESET TIME INDICATOR                    |
| 0287 | 4000   |        | CLR | À        | ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, |
| 0288 | 5001   |        | INO | 5,1      |   |
| 0289 | 828E   |        | JZE | NDLY1    |   |
| 028A | 4200   |        | SUB | 0        | DECREMENT SI                            |
| 0288 | 2900   |        | RUT | 1        |   |
| 028C | 5002   |        | INO | 5,2      |   |
| 028D | 1000   |        | RTN | - • -    |   |
| 028E | 69FF   | NOLY1: | SUT | 1.0FFH   |   |
| 028F | 5 DO 2 |        | INO | 5,2      |   |
| 0290 | 4200   |        | SUB | 0        |   |
| 0291 | 2A00   |        | RUT | 2        |   |
| 0292 | 4000   | SETAC: | CLR | A        |   |
| 0293 | 45AA   |        | ORA | DAAH     | ILOAD ACCUMULATOR WITH INDICATOR        |
| 0294 | AA96   |        | JE0 | NSIND    | ,                                       |
| 0295 | 1000   |        | RTN |          |   |
| 0296 | 7888   | NSIND: | SUT | A,DAAH   | LEGT DETECTED                           |
|      |        | Matur. |     | m , oram | Jea: Apipeipe                           |
| 0297 | 1000   |        | RTN |          |   |



#### 23.35 CRCET SUBROUTINE

The CRCET subroutine is used to make an extra exclusive-or operation on that part of the S4 register which keeps a tag on whether the number of bytes detected has been odd or even, cfr. Paragraph 23.30 on the RWDAT subroutine.

It is desirable to have this tag show 0 after the final LRC character has been detected, such that upon testing, a simple deviation from the expected 0 will indicate an error.

Now, if there has been an even number of bytes by the time the CRC character has been detected, the CRC character will contain odd parity and the S4 register all "1"s, that is, Fx in hexadecimal notation (The 4 most significant bits).

However, for a CRC character of even polarity and an odd number of bytes, the S4 register would contain 0x and the next exclusive-or would make it Fx. For this case the CRCET executes an extra exclusive-or to obtain the desired 0x.

For an all "0" CRC character, there is no exclusive-or operation taking place, and the 0x status in the S4 register for the last data byte is passed unchanged to the LRC character.

Thus, if there has been no drop-outs or other errors, the S4 register should after a completed read operation contain 0x.

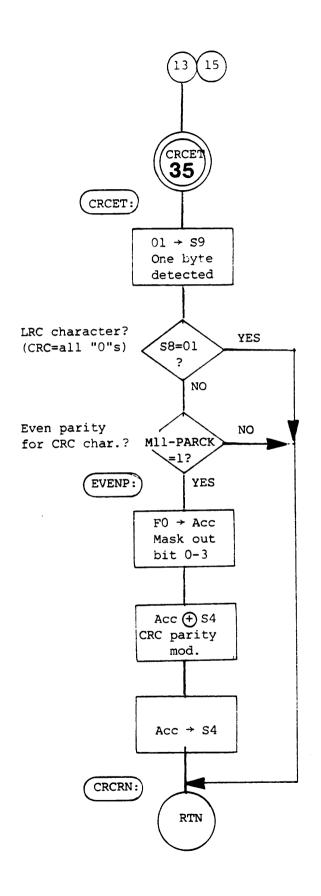
The next exclusive-or will then give the desired 0x for the LRC character.

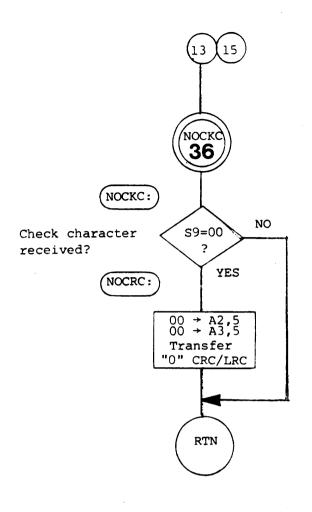
| 034E         | 7901              | CRCET:      | SUT        | 9,01H | INDICATES ONE BYTE DETECTED   |
|--------------|-------------------|-------------|------------|-------|---|
| 034F         | 500B              |             | INO        | 5.8   | TEST UPON LRC/CRC OPERATION   |
| 0350         | 0401              |             | TXR        | оін   | ILRC IF 01  |
| 0351         | 8358              |             | JZE        | CRCRN | LRC IF ZERO   |
| 0352         | AF54              |             | JPE        | EVENP | EVEN PARITY ON THE LRC CHARACTERS   |
| 0353         | 1000              |             | RTN        |       |   |
| 0354         | 4000              | EVENP:      | CLR        | Α .   |   |
| 0355         | 45F0              |             | ORA        | OFOH  | MASK OUT BIT 0 - 3.   |
| 0356         | 5C04              | ;<br>;<br>; | INX        | S,4   | EXCLUSIVE-OR ACCUMULATOR WITH<br>154. THIS MEANS THAT THE 4 MOST<br>18IGNIFICANT BITS OF THE 54<br>1REGISTER WILL ALMAYS BE "F"<br>1WHEN THE CRC CHARACTER HAS<br>1BEEN READ. |
| 0357<br>0358 | 2 CO 0<br>1 0 0 0 | CRCRN:      | RUT<br>ŘŤN | 4     |   |
|              |                   | Ŧ           |            |       |   |

#### 23.36 NOCKC SUBROUTINE

The NOCKC subroutine is used to generate an all "0" byte in the time slot for a CRC character on the assumption that if a regular CRC byte has not been detected in this time slot, the CRC character must be an all "0"s byte.

| 02F8 | 4000   | NOCKC: | CLR | <b>A</b> :                              |                               |
|------|--------|--------|-----|---|-------------------------------|
| 02FC | 5 DO 9 |        | INO | S. ?                                    |                               |
| 02FD | 82FF   |        | JZE | NOCRC                                   | INO CHECK CHARACTER IF ZERO   |
| 02FE | 1000   |        | RTN |   | THE GREEK GENERALIER IF ZERO  |
| 02FF | 6200   | NOCRC: | OUT | 2,00H                                   |                               |
| 0300 | 6300   |        | OUT | 3.00H                                   | TRANSMIT ALL ZEROES CHARACTER |
| 0301 | 1000   |        | RTN | • | THE PERIODS GRANDICK          |





CRCET

CRC TEST SUBROUTINE



NOCKC



#### 23.37 SDLAY SUBROUTINE

The SDLAY subroutine is used to generate a short delay equal to about 18 system clock periods.

The subroutine is employed during NRZI read reverse operations to ensure a minimum delay between the read strobes at the beginning of the block.

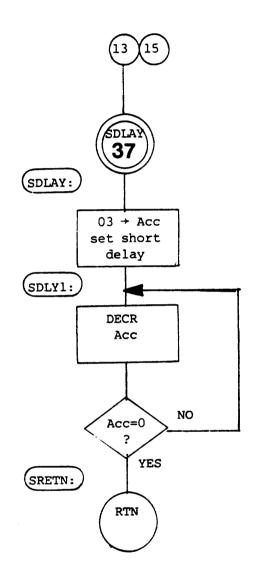
It is also employed during NRZI read operations when an all "0"s byte is being transferred, to avoid a false parity error indication, since it takes 12 clock periods after the all "0"s byte is loaded before the parity check on it is finished.

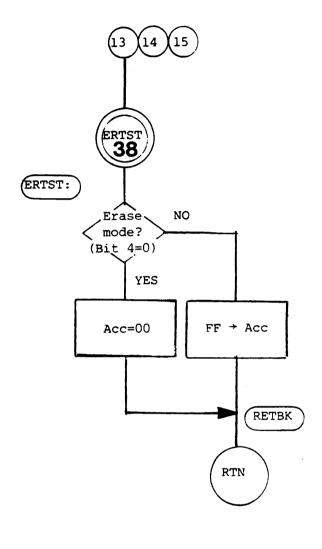
| 0320<br>0321 | 4000<br>4503 | SDLAY: | CLR        | A<br>HEO     | LOAD DELAY TIME |
|--------------|--------------|--------|------------|--------------|-----------------|
| 0322<br>0323 | 4200<br>8325 | SDLY1: | SUB        | 00H<br>SRETN | DECR. ACC.      |
| 0324<br>0325 | BF22<br>1000 | SRETN: | JUN<br>RTN | SDLY1        |                 |

#### 23.38 ERTST SUBROUTINE

The ERTST subroutine is used to load the accumulator with FF unless we have a space operation or a search/ignore data operation. In the latter case (when bit 4 is zero) the accumulator ends up with a 00.

| 028F<br>02C0<br>02C1<br>02C2<br>02C3 | 4000<br>5000<br>4610<br>82C4<br>4700 | ERTST: | INO<br>ANA<br>JZE<br>SET | A<br>S,0<br>10H<br>RETBK<br>A | ITEST UPON ERASE MODE ( BIT e4 )<br>IERASE ( SPACE ) IF ZERO<br>IFF TO ACC. |
|--------------------------------------|--------------------------------------|--------|--------------------------|-------------------------------|---|
| 02C4                                 | 1000                                 | RETBK: | RTN                      |                               |   |



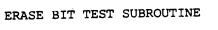


SDLAY

SHORT DELAY SUBROUTINE



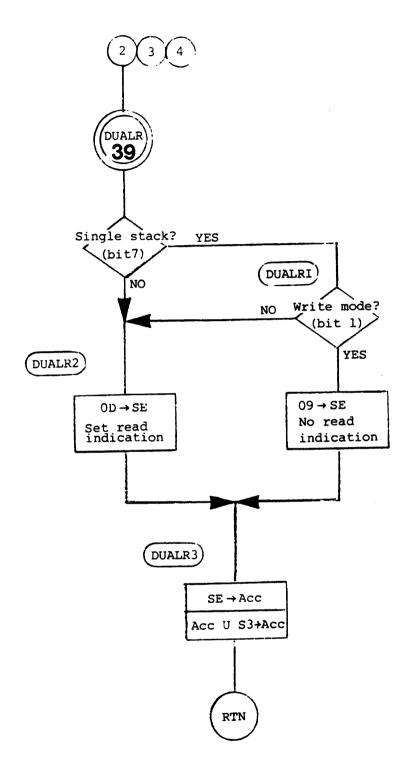
ERTST



#### 23.39 DUALR SUBROUTINE

The DUALR subroutine is used to enable read operations unless a single stack head is being used in a write operation and no reading is possible. The resultant bit pattern is placed in register SE, then mixed with the contents of S3, which contains the selected format bit, and finally placed in the accumulator. The main program may now mix in more control bits before strobing the complete byte into the A6 control register.

| 00AE | 4000 | QUALR: | CLR | A      |                             |
|------|------|--------|-----|--------|-----------------------------|
| 00AF | 5000 |        | INO | 5,0    |                             |
| 0080 | 0480 |        | TAN | 80H    | TEST UPON SINGLE STACK MODE |
| 00B1 | 8087 |        | JZE | DULRI  | SINGLE STACK IF ZERO        |
| 0082 | 7E0D | DULR2: | SUT | E,ODH  | ;SET READ LINE              |
| 0083 | 4000 | DULR3: | CLR | A      |                             |
| 00B4 | 5D0E |        | INO | S,0EH  |                             |
| 0085 | 5003 |        | INO | . S.3H |                             |
| 0086 | 1000 |        | RTN |        |                             |
| 0087 | 0602 | DULR1: | TAN | 02H    | TEST UPON WRITE MODE        |
| 0088 | 8082 |        | JZE | DULR2  | NO WRITE IF ZERO            |
| 0087 | 7E09 |        | SUT | E,09H  | ;NO READ                    |
| 00BA | BCB3 |        | JUN | DUL R3 |                             |
|      |      |        |     |        |                             |



# SECTION 24 FORMATTER CIRCUITRY

## CONTENTS 24.0 FORMATTER CIRCUITRY 24.1 SCOPE 24.2 SCHEMATIC 24.2.1 System Clock 24.2.2 Input/Output Circuits SCHEMATIC 3 24.3 24.3.1 Write Data In/Out 24.3.2 Timer Circuit SCHEMATIC 4 24.4 24.4.1 NRŽI Read Data In 24.4.2 PE Read Channels 24.4.3 Scan Generator 24.4.4 Data Block and Gap Generator PE READ CIRCUITRY, GENERAL 24.5 DESCRIPTION, SCHEMATIC \$ 24.5.1 Timing Decoder Prom 24.5.2 Timing Selection Circuits 24.5.3 PE System Decoding Circuitry 24.5.4 Envelope Circuits 24.5.5 Skeq Buffer 24.5.6 DIAC Detector 24.5.7 Postamble Detector 24.5.8 Read Data Output Circuits 24.5.9 Error Detector Circuits

## 24.0 FORMATTER CIRCUITRY

#### 24.1 SCOPE

This section contains the schematics for the formatter system exclusive of the microprocessor described in section 22. The accompanying text provides details and supplementary information to each schematic, with reference to particular circuit areas on the schematic. Paragraph 24.5 gives a general description of the PE read hardware.

#### 24.2.1 SYSTEM CLOCK

The basic clock frequency is 14.4 MHz. This is divided down by U76 and U9 to obtain the final system clock frequency, see Table 1. The system clock pulse is about 70 nanosec wide, irrespective of the pulse periods listed in Table 1.

U76 and U9 is preset by strapping to obtain the proper divisor. U9 can in addition be controlled remotely from the tape transport by the TLSP signal if strap E is connected to ground, permitting two tape speeds in a multiple tape transport system, one speed being half the normal speed.

The basic oscillator may be disabled by grounding the DISCL input, and an external frequency source may be introduced at input EXCLK.

SUP is a pulse immediately following the system clock pulse. It is used to suppress effects of transients on the Data Bus. See wave forms Fig. 22.1. The SUP signal goes true at the trailing edge of the system clock signal, and false at the next basic clock pulse through the action of U10.

#### 24.2.2 INPUT/OUTPUT CIRCUITS

Signals Fzzz come from the CPU and signals Tzzz come from the tape transport. See respective signal descriptions in Paragraph 6.2, 6.3 and Paragraph 7.1, 7.2 in Part 1 of the manual. One more formatter can be connected to the CPU Bus. These are selected by the FADx signals, and strap S7 is used to assign either station number 0 or 1 (as shown) to the formatter.

When addressed and enabled (FFEN true), the SEL gate signal will enable tape transport input signals and formatter output signals. The tape transport signals are synchronized to the system clock as need be.

Tape Transport Selection Truth Table

| FTADl | TSLTx |
|-------|-------|
| 0     | 3     |
| 1     | 2     |
| 0     |       |
| 1     | Ø     |
|       | 0     |

The more important CPU command signals are strobed into suitable registers when FGO goes true and the tape transport is ready (TRDY true). At the same time, a latch is set, Ml-INREQ, which can be tested by the microprocessor program and reset by strobe A9. Thus informed about the presence of a new command, the microprocessor may then proceed to input the command signals via the Data Bus by using strobe B7.

Density information is introduced using strobe B4. The transport TDDI signal selectshigh or low density out of four possible combinations. The formatter is strapped for one of these combinations using the H and K straps, see Table 2.

Output formatter status signals to the CPU, and microprocessor control signals to the read and write circuitry, are strobed from the Data Bus into suitable registers by using strobes A5 and A6, among others.

All registers are cleared initially by the power-on clear pulse POCP through master clear pulse CLEAR. They are also cleared when the formatter enable signal FFEN goes false.

#### 24.3.1 WRITE DATA IN/OUT

The write data is strobed into the Data Bus using strobes BO and Bl. The parity bit is generated internally, but the formatter can accept an external parity bit by restrapping Sll.

Since the data byte is 9-bit and the Data Bus is 8-bit, the data byte is handled in two parts. The first part consists of all the data bits. These bits are simultaneously available to the parity generator as long as the BO strobe is active, but will have left the Data Bus by the time Bl goes active, although still present on the FWDx hus. The parity generator output parity bit is therefore strobed into a flip-flop (U50) by BO and stored there until Bl goes active.

The write input data goes from the Data Bus into the microprocessor accumulator and then from there right out onto the Data Bus again and into the Write output registers. First come the data bits, using strobe BO for inputting, as explained, and AO for loading the AO Write output register, and then comes the parity bit, using Bl, respectively Al.

The complete byte is now present on the 9-bit TWDx bus. The actual writing, however, takes place when the Timer Circuit generates the TWDS write strobe. This occurs on the first clocking pulse out of the timer after the A4 strobe has passed the ETWDS Write enable signal. In NRZI mode, the ETWDS signal is used to disable every other timer clocking pulse, since the clocking rate is twice the bit rate.

In PE mode, writing is complicated by the need to write phase transitions in between the data transitions, and the ETWDS signal is active all the time during the write operation. The input write data is processed in the accumulator and is also going into locations S4, S5 in the microprocessor scratch pad memory. Phase transition write data are loaded into the A0, Al output registers in between the data transition data. See PWBGN Sequence, Paragraph 23.9. (NRZI writing is handled by the NZBGN sequence, Paragraph 23.10.)

File mark and ID Burst Write data are generated internally and not appearing on the FWDx bus.

In writing data, the FDWDS strobe is used to demand the next write data from the CPU. The EFWDS enable signal is used to select in the Write cycle the opportune clocking pulse for the FDWDS signal.

The ENLRC (enable LRC) signal selects the final clocking pulse which results in the TWRS signal to the tape transport, resetting all write flip-flops and thereby generating the LRC character.

The CRC character computed in the Ull7 CRC Generator is strobed into the microprocessor by Bl, B3 and processed further by software before eventually being loaded into the AO, Al registers (see Paragraph 23.11). Strobe Bl was also used for inputting the parity bit, but the CRC parity bit is fed onto a Data Bus line (DBO) different from that of the FWDP bit (DB1).

#### 24.3.2 TIMER CIRCUIT

The Timer Circuit provides clocking and time base signals for the various strobing, timing and delay requirements of the formatter and aids in synchronizing various software controlled operations.

The timer circuit consists of a timing counter (U120, U121) clocked by the system clock, and a time base register (U142) which determines the clocking period by presetting the counter on every counting cycle. The frequency divisor is loaded into the time base register from the Data Bus by strobe Al2. Normally, the clocking period is half the character period. For generation of delays the clocking period is at 75 ips 10 or 20 usec.

An output flip-flop U43 is used to generate a strobe signal which is five times the width of the system clock period (1.04 usec at 75 ips). The flip-flop is set when the A and C outputs of the first counter section cause a high at the J input and is reset five system clock cycles later when the ripple count output goes low at the K input.

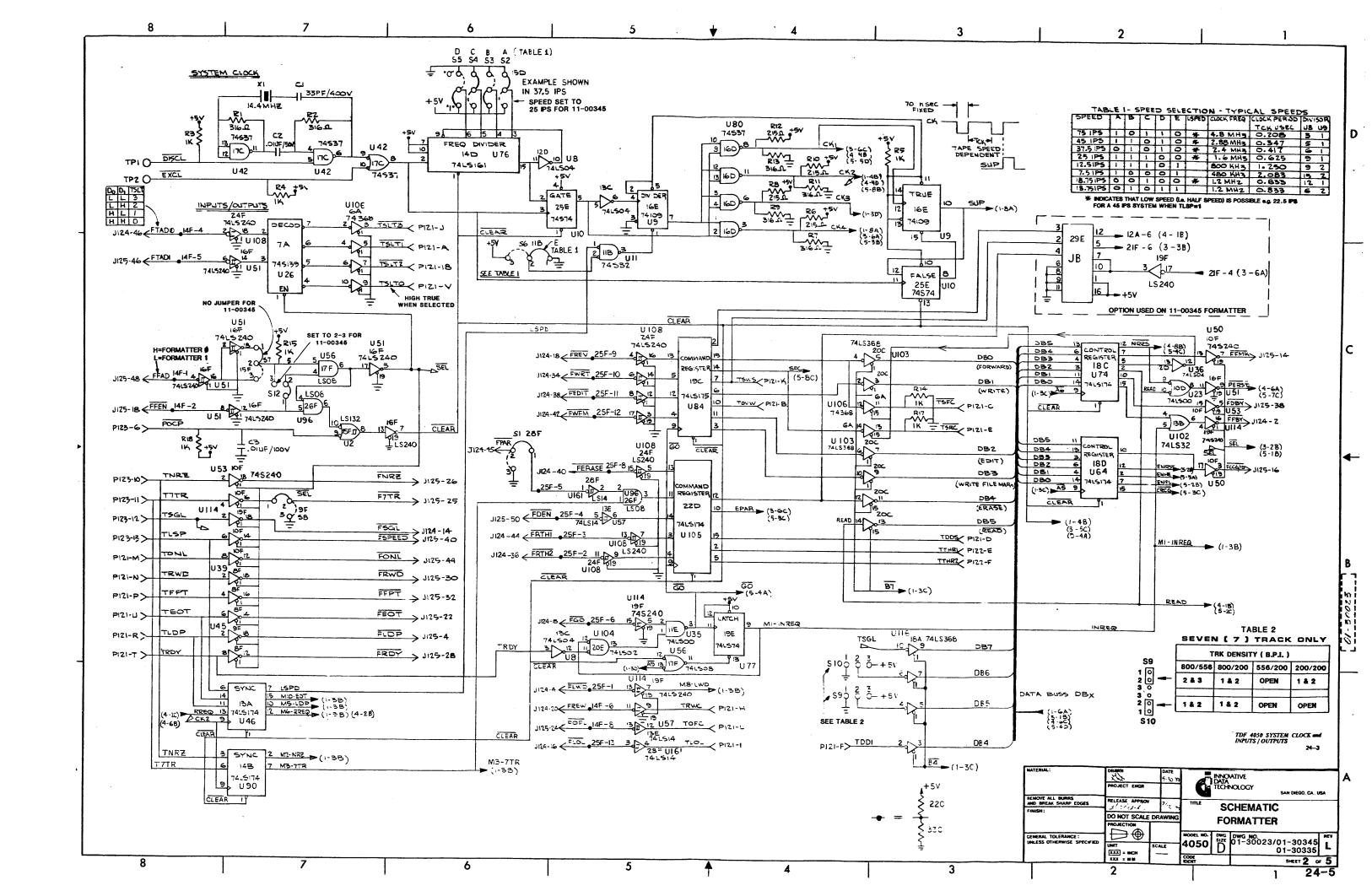
Another flip-flop (U44) is set at the trailing edge of the generated strobe, but depends on an external strobe Al3 for being reset. The output, M4-TIME, is tested extensively by the program to check the completion of a strobe or to detect the end of a clocking period.

In the description on the Write Data In/Out circuitry, the TWDS, FDWDS and TWRS strobes were mentioned and how these strobes were generated by enabling the timer circuit clocking signals at about the right moment.

The program will in these cases wait for the M4-TIME signal to indicate the completion of the associated instruction before proceeding to the next instruction in the program. For this reason the Q output is used, giving as it does, a "1" as long as it remains in the reset state. A "1" is needed by the multiplexer test circuitry to qualify a jump. Upon detecting the "1" the program will be able to jump back one step and thereby repeat the test instruction. Thus, a waiting loop is obtained with only one instruction.

When the clocking period is over and the M4-TIME output goes "0", the program will upon the first detection of this "0" advance to the next instruction.

The flip-flop will remain in the set state until it is time to use it again. An instruction to reset the flip-flop, by generating the Al3 strobe, must always precede the M4-TIME test instruction, but need not be immediately preceding.



In a NRZI read operation, the 9-bit read byte TRDx from the transport is strobed in two portions into the accumulator via the 8-bit Data Bus, using strobes B4 and B5. The read signals are thus handled exclusively by software and are loaded into the read output register A2, A3 (Schematic no. 5) almost as soon as they arrive, using output strobes A2 and A3. The latter strobe simultaneously starts the generation of an output read strobe FRSTR (Schematic no. 5).

The inputting strobes B4, B5 are not generated before the program has been advised of a new read data byte by the accompanying input read strobe TRDS. This strobe will set a flip-flop U25 whose output M6-RREQ will be tested by the program. The flip-flop is reset when the first portion of the read data byte is strobed A2 into the output read register, actually by the same strobe A2.

The NRZI read data transfer operations are handled by the RWDAT (in read-afterwrite) and RDATA (read-only) subroutines (Paragraphs 23.30, 23.31).

#### 24.4.2 PE READ CHANNELS

In a PE read operation, the handling of the PE read data bytes is done in hardware. While this process is going on, the program is almost at a standstill, from the moment it receives a "data block detected" signal (M7-DDET=1) until the moment it receives a "postamble detected" signal (M14-POSTD=1). The program does little but check for the presence of a file mark in the PE readonly sequence PREAD (Paragraph 23.8), and even less in the reading-after-write activity of the PE write sequence PWBGN (Paragraph 23.9).

It is mainly the microprocessor speed limitations and the necessity of deskewing the PE read data which preclude the handling of the PE read operation in software. Each of the 9 read channels requires individual and independent processing which in practice means multiplexed, sequential handling. Furthermore, the bit rate is higher than for the NRZI format. This calls for high operating speeds, and the

operations are in the TDF 4050 at 75 ips carried out at the system clock rate of 4.8 MHz.

The PE read hardware consists of 9 read circuits and common processing circuitry which is time-shared between the 9 channels. The time-sharing is controlled by a scan generator which multiplexes the read data information produced by the read channels and presents this information sequentially to the processing circuitry. The resultant sequential read data is shifted serially into the read output register A2, A3 (Schematic no. 5), and is on every completed scan cycle strobed out in parallell to the read output bus FRDx. In the NRZI mode, the read data is loaded in parallel into the same read output registers.

The information obtained from each input PE read channel consists of the input read LEVEL signal and a 7-bit binary number CNTx which indicates the time elapsed between the last two level transitions. A new number will be generated, and the level signal will change polarity, every time there has been an input level transition, but not necessarily on every scan, since each channel will be scanned several times in a character period (about 4 times per character period, see Fig. 24.1). The read

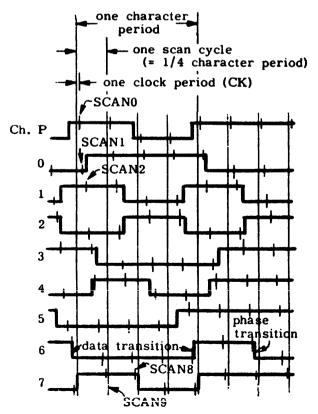


Fig. 24.1 SCANx signal slots relative to read signal transitions.

channel output signals are synchronized to the system clock, i.e. flux transitions actually occurring in between clock pulses, will appear to the system to occur at the clock pulse transitions.

The CNTx number will be used later to check the correct spacing between flux transitions and to detect phase transitions. Phase transitions can be distinguished from data transitions by the fact that the spacing between data and phase transitions is half of what it is between two data transitions.

A PE channel consists essentially of a spike-free edge detector circuit, e.g. U100 in channel TRDP, and a counter e.g. U13. The edge detector is synchronized to the system clock. The counter keeps counting at the system clock rate and is reset whenever there is a signal level reversal at the input as detected by the edge detector. The final count, before being reset to zero, is loaded into an output register, U143.

The edge detector consists of two flipflops. The first one, U100-19, follows the input read signal synchronized to the clock, see Fig. 24.2. The LEVEL signal is derived from this flip-flop. The other one, Ul00-12, follows one clock period later. By exclusive-oring the two outputs, a strobe pulse STRBP is obtained indicating the occurrence of a level transition. This signal is used for data block detection and interblock gap detection. In addition, it suppresses the clock input to the counter Ul3, just before the counter is reset, permitting the counter to settle down before the final count is transferred to the output register U143.

The strobe signal also triggers a third flip-flop U125, generating a pulse at U125-7, on the trailing edge of the STRBP pulse. This is the fast strobe signals which transfers the count to output register U143 and simultaneously resets counter U13.

Counter U13 contains two counter units. The second unit is clocked (pin 13) by the last stage (pin 6) of the first counter. A phase transition will occur nominally at count 20 from the last data transition, whereas the next data transition in the absence of a phase transition will occur nominally at a count of 40. (In addition to effects of skew, the actual count will be 2 less due to

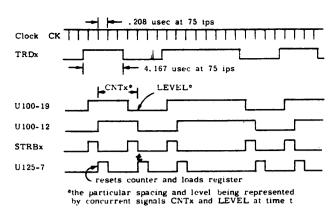


Fig. 24.2 PE Read Channel Waveforms

clock pulses lost while the counter is being suppressed by the STRBx signal.)

The counter itself is capable of running to 128. However, in the absence of any transitions, that is, in the case of a drop-out, the counter will reset itself at count 64 through the feedback of the 7th bit (pin 9) to the reset flip-flop U125. The 7th bit, in the appearance of CNT6, is used later to indicate a drop-out. CTN0 to CTN5 is fed to the PE Timing Decoder PROM (Schematic 5) to represent the interval between two LEVEL transitions.

The output register is edge-triggered, meaning that it is being set and latched already on the rising edge of the loading pulse. This property makes it possible to use the same pulse to reset the counter since there is here a finite delay before the clearing action takes effect. This ensures the safe transfer of the counter output to the register before the output count goes to zero.

#### 24.4.3 SCAN GENERATOR

The scan generator consists of a decade counter U126 and a decoder U127,U26. Notice that there are two sets of outputs, the SCAx, which are in a 4-bit binary form, and the SCANx, which in a decoded form are 10 separate, sequential strobe signals. SCANO and SCAN9 mark respectively the beginning and the end of the scanning cycle and are used also as enabling signals. The strobe signals are at 75 ips about 0.2 usec wide, and a complete scan cycle takes about 2 usec. See Fig. 24.3.

The occurrence of scan strobes relative to read channel level transitions is illustrated in Fig. 24.1. The scan strobes occur relatively often, and there is not necessarily a LEVEL transition in one channel from one scan to

the next. Also, because of skew, the LEVEL transitions in different channels will not necessarily occur at the same time.

Note also that the PE read data strobed into the common processing circuitry in one scan cycle is processed further in the next scan cycle. At one stage there is a shift in strobe position relative to respective read channel such that channels P to 9, being initially represented by SCAN 0 to 8, later are represented by SCAN 1 to 9.

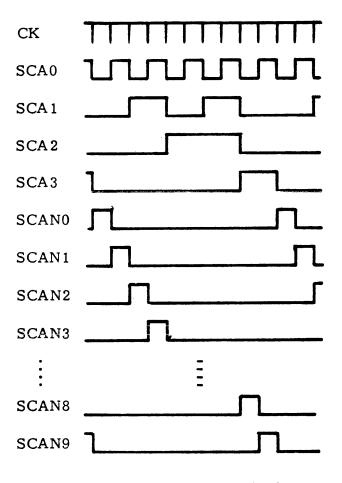


Fig. 24.3 Scan Generator Signals

#### 24.4.4 DATA BLOCK AND GAP DETECTOR

The Data Detector is used to indicate the presence of a data block. This is manifested by the output flip-flop U6 being set and the M7-DDET signal going true.

In the NRZI mode, the flip-flop is set by the first arriving read strobe, via the RREQ flip-flop U25.

In the PE mode, in which there are no read strobes, the STREx signals from 24-8

the PE Read Channels are used. These are generated whenever there is a level transition, and in the preamble, each of the initial "0" bits will be accompanied by a phase transition as well as a data transition, and will produce two STRBx strobes.

Two channels, STRBO and STRB2, are used, in case one of them should have a drop-out. More than these two channels need not be sampled since more than one drop-out would invalidate the readings anyway.

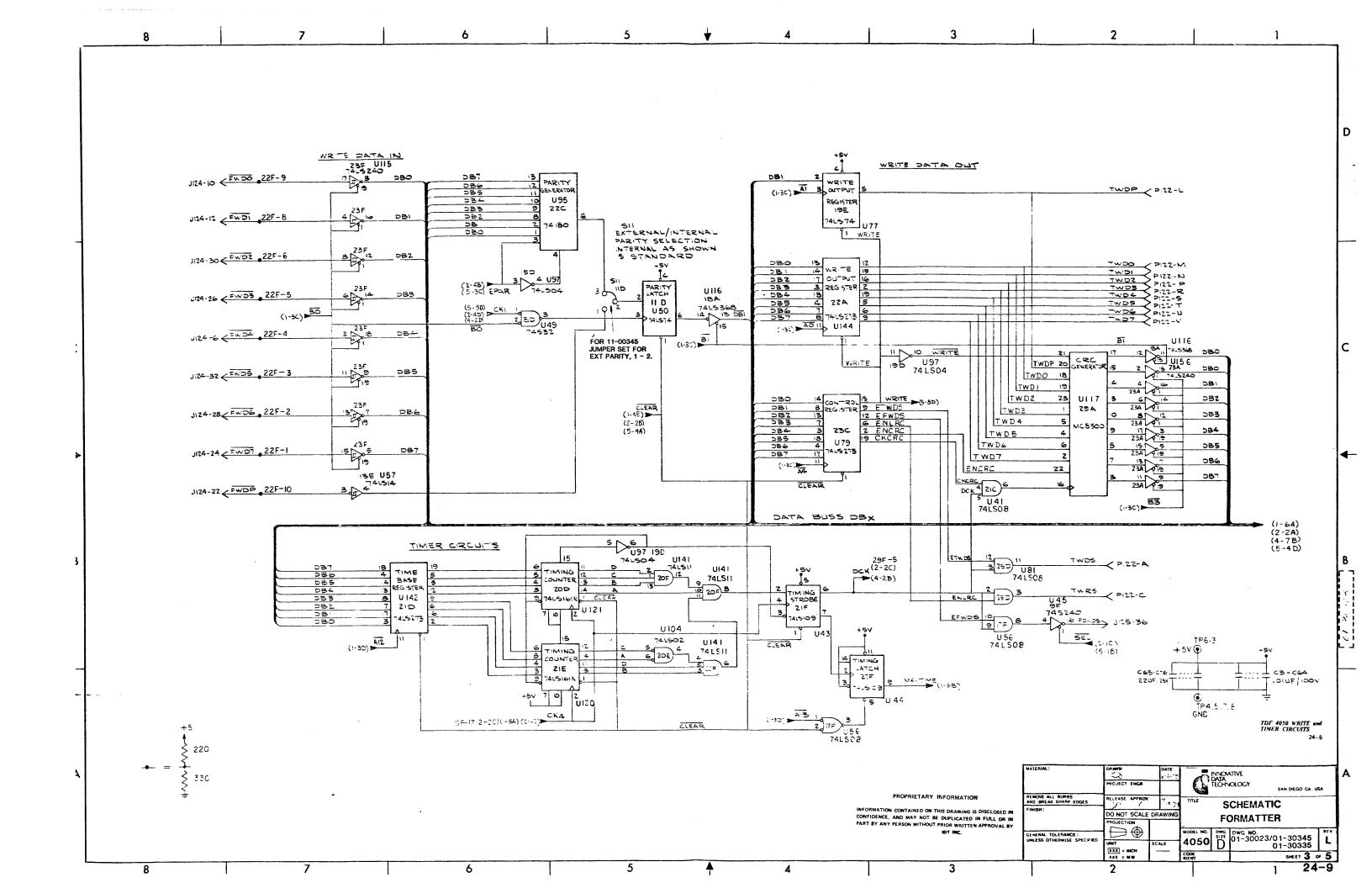
To prevent an incidental noise spike from triggering the M7-DDET flip-flop, a count by counter U32 of 16 strobes is required to qualify a data block. This means that, if the STRBO and STRB2 strobes are overlapping, the flip-flop will not be set until the 8th byte of the preamble have appeared. However, if these strobes appear separately, which will be the normal case, this will occur after 4 bytes.

It is also required that the gap between successive strobes not exceed the bit period. If so, counter U32 will be reset by a strobe gap detector circuit centered around counter U38.

The input DCK signal clocks this counter at the double bit rate, and the counter is enabled if all the strobe signals STRBx are absent. If this absence lasts for two clock periods, that is, one bit period, a reset pulse will be generated and prevent the setting of the M7-DDET flip-flop. The strobe gap detector will have no effect, however, once the flip-flop is set, since U6-7 will keep clear-input U38-2 high.

The M7-DDET flip-flop may be set by the program, using strobe AlO. This is done in the ID Burst sequence IDWRT when the absence of signals in channels other than the P channel otherwise would have failed to set the flip-flop, which in turn is necessary in order to activate the PE read circuitry for ID Burst detection.

Interblock Gap Detection is afforded by flip-flop U25. To perform a gap detection test, the program simply sets the flip-flop using strobe A7, generates a delay, and then, at the end of the delay, tests the M9-GAPD output to see if the flip-flop is still set. It will not remain in the set state if, in the delay period, there has been any signal present in any of the read channels capable of generating a STRBx strobe.



#### 24.5 PE READ CIRCUITRY, GENERAL DESCRIPTION

The program initiates the PE <u>reading</u> operation when the NRZS status signal goes false and the READ signal goes . true (strobe A6 register on schematic no. 2). The read data <u>processing</u> starts automatically when a data block is detected (M7-DDET = 1, on schematic no. 4) and the enable signal PEDET goes true.

PE read signals from the tape transport are applied to special PE Read Circuits, one for each channel (Schematic no. 4). The outputs of these circuits are being sampled sequentially by the Scan Generator. They supply read data parameters on bit spacing (CNTx) and bit logical level (LEVEL).

The bit spacing parameters are fed to the PE Timing Decoder PROMs (Schematic no. 5), where they are examined for conformity to standards on bit spacing tolerances. The PROMs contain several look-up tables, and the Timing Selection Circuits aid in selecting the appropriate table.

The resultant bit spacing qualification data and the LEVEL signal are fed to the PE System Decoder Circuits. These circuits separate phase transition bits from data bits and forward the latter to the Skew Buffer. Drop out information is forwarded to the Envelope Circuit.

The Skew Buffer deskews the data bits by feeding them serially into suitable registers (RAMs). Skew Buffer circuits look out for preamble "l"s and feed them to a Data-In-All-Channels Detector. As soon as the presence of a preamble "l" bit in all 9 channels have been established, notes are taken of the relative register positions to which the preamble "1" bits have reached. Read data is now being read out from the positions next to the noted positions, and new data is being shifted into these positions as they arrive. The process ensures that bits from the same byte are collectively being read out during the same scan cycle.

After the deskewing process, the data bits are shifted serially into shift registers, and transferred in parallel as a complete byte to the FRDx Output Registers. Upon this transferral, the Envelope Circuit will point to the drop-out channel if any, and help reconstruct the bits of the disqualified track.

Error Detection Circuits will disqualify the whole block of data if there are drop-outs in more than one channel.

The Postamble Detector will generate an output signal (M14-POSTD) when there is a postamble "10" sequence in all channels. This will enable the return of the reading operation to microprocessor program coatrol.

#### 24.5.1 TIMING DECODER PROMS

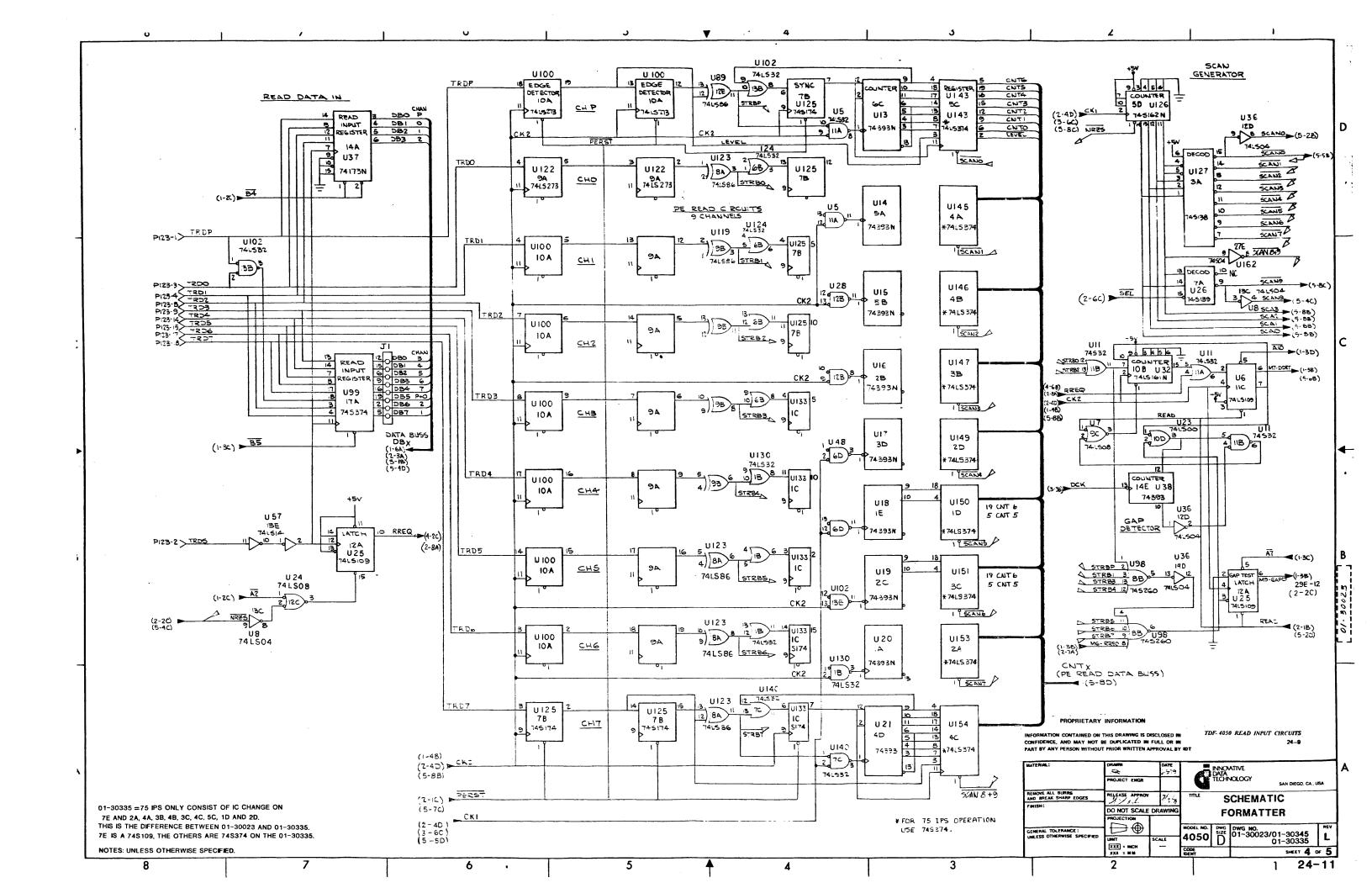
The PE read data processing starts out with interpretation of the bit spacing signals CNTx obtained in the PE Read Circuits. These signals are applied to two PROMs, U128 and U107, which contains eight timing tables, four each for read-only and read-after-write operations. Only one table of the eight is in use at any time, and a 3-bit selection signal consists of WRITE, LOWR (low range) and SEL1 (select PROM1):

| Selection Signals |      |      | Table Selected |          |             |  |
|-------------------|------|------|----------------|----------|-------------|--|
| WRITE             | SEL1 | LOWR | . Mode         | PROM no. | Table Range |  |
| O                 | Ò    | 1    | Read-only      | 1        | Lower       |  |
| 0                 | o    | 0    | •              | ì        | Lou         |  |
| 0                 | 1    | 1    | -              | 2        | High        |  |
| 0                 | 1    | 0    | -              | 2        | Higher      |  |
| 1                 | 0    | 1    | R-A-W          | 1        | Lower       |  |
| 1                 | 0    | 0    | -              | 1        | Low         |  |
| 1                 | ı    | ı    | •              | 2        | High        |  |
| 1                 | 1    | 0    | •              | 2        | Higher      |  |

The PROMs put out two sets of signals, one set  $S_i$  reflecting on the quality of the signal, and the other set  $T_i$  reflecting on the appropriateness of the table selected:

| so | Sl | Spacing is                                     |
|----|----|--|
| 0  | 0  | Too short. Drop-in assumed                     |
| 0  | 1  | Satisfactory, data-to-phase and phase-to-data. |
| 1  | 0  | Satisfactory, data-to-data                     |
| 1  | 1  | Too long. Drop-out assumed                     |

| 23-17 | EFWDS   | EFDWS  |
|-------|---|--|
| 23-21 | Set write only  | (Set write only) Reset ETWDS   |
| 23-26 | CALRC   | CHLRC  |
| 23-26 | 00011101 (19)   | 00011001 (19)  |
| 23-26 | The respective longitudinal parity bits shall be "0". | The respective longitudinal parity bits in S4, S5 shall be "0". (Because the calculated LRC, when it is exclusive-ored with the recorded LRC, will result in 0 if equal) |
| 23-27 | NRNDZ   | NRND2  |
| 23-31 | Call DELAY  | Call DELAY<br>20   |
| 23-40 | 0164 05н  | 0164 09н   |
| 23-41 | Acc = 05?   | Acc = 09?  |
| 23-47 | EFWDS   | EFDWS  |
| 23-48 | 3-4 bits periods                                      | 3 timer periods, i.e. 1.5 character periods.   |
| 23-3  | B7  | B7 —   |
| 24-11 | ENV0 ENV7 (8 praces)                                  | ENVO  ENV7   |



| To | T <sub>1</sub> | Table range is                             |
|----|----------------|--|
| 0  | 0              | Too high, select a lower range             |
| 0  | 1              | Satisfactory, somewhat on<br>the high side |
| 1  | 0              | Satisfactory, somewhat on the low side     |
| 1  | 1              | Too low, select a higher range             |

The PROMs will in effect examine the bit spacing data, look up one table and determine whether the spacing falls within the tolerances of that required for a phase-to-data, data-to-phase or a data-to-data transition. Such tolerances are specified for written records by international standards and must be rigidly controlled in the PE write operation.

There are no such particular standard for the reading operation, since reading quality depends on the transport, for which there are no standards. However, in the read-only operation it is necessary to establish certain tolerances, however wide, to be able to distinguish the categories listed in the Si table from one another within safe margins. Naturally, a read-only table will be more lax than a corresponding R-A-W table.

Spacing is measured by actually measuring time, assuming a nominal, constant tape speed. A table of spacing tolerances based on a particular tape speed will therefore be less satisfactory if the actual tape speed deviates from the assumed speed. In the TDF 4050 formatter this contingency has been met by providing several overlapping tolerance tables to cover the range of tape speeds allowed within the standards.

A table range is defined as being low when the bit spacing is on the short side, as it would be for higher tape speeds.

The selection of appropriate table is performed by the circuitry centered around the Timing Select PROM U101.

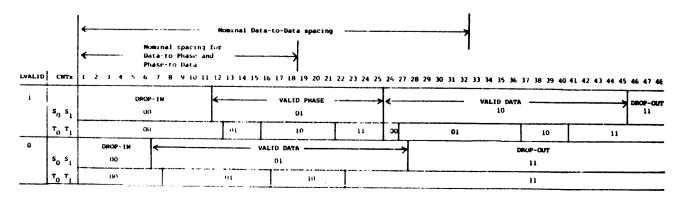
In looking up a table in the PE Timing Decoder PROMs, it is necessary to know whether the last transition was a data transition or a phase transition. This is provided by the LVALID (last bit valid data) signal. In the former case, the expected spacing would be short for a data-to-phase transition and long for a data-to-data transition. In the latter case, the spacing would be expected to be short (within tolerances different from the other short spacing) for a phase-to-data transition, whereas a long spacing would be disqualifying and indicative of a drop-out.

A typical table is indicated below.

# 24.5.2 TIMING SELECTION CIRCUITS

The Timing Selection Circuits are centered around the Timing Select PROM Ul01, and generate the signals LOWR and SEL1 which help select the appropriate timing table in the PE Timing Decoder PROMS. The inputs to the circuitry are the  $T_0$ ,  $T_1$  signals from the latter PROM which indicate the suitability of the presently used timing table.

Cumulative effects of bit crowding, tape sway, transient speed variations and other causes may result in occasional large deviations in bit spacing which are within tolerances, but cause T<sub>0</sub> T<sub>1</sub>



to indicate that the presently used table is not suitable. Previously generated  $T_0$   $T_1$  signals are therefore stored in a RAM and used to establish whether the disqualification was due to an isolated, incidental case or to the culmination of a longer term trend. Thus both the current and the stored, previous states of the  $T_0$ ,  $T_1$  and the LOWR, SEL1 signals are used as inputs to the Timing Select PROM. The previous state signals are LT0, LT1, LLOWR and LSEL1.

The finally resulting LOWR and SELl signals are stored in another RAM for actual use in the selection of timing table in the next scan cycle. At the same time a TREND signal is developed, expressing a longer term upward (TREND = 0) or downward trend (TREND = 1). The TREND signal is stored and also used as an input (LTREND) to the PROM, providing a still broader decision base.

For each combination of the 9 inputs to the PROM, a 4-bit output configuration has been specified. Some input combinations are necessarily self-contradictory and should never occur in practice, but have been specified anyway. In such cases the fourth output bit, CHECK, is set true. This bit is not used in the formatter system, but is available for test purposes.

In specifying the outputs, that is, programming the PROM, the following rules have been followed:

- A change of tables is always confined to the next higher or the next lower table.
- 2. If the present  $T_0$   $T_1$  indication is 01 or 10 (satisfactory), there is no change unless the previous indication  $LT_0$   $LT_1$  is identical to the present one, and preferably the TREND signal points in the same direction.
- 3. If the present T<sub>0</sub> T<sub>1</sub> indication is 00 or 11, a move to the next lower, respectively the next higher table may be made.
- 4. If the presently and previously selected tables instead of being next to each other are one table apart, 85% importance is placed on the present selection, and a change is made only if the present and previous T<sub>0</sub> T<sub>1</sub> indications point in the same direction. This situation is one of

those not anticipated, and the CHECK output is true in this case.

5. If the presently and previously selected tables are two tables apart, that is, on the opposite extremes, no importance is placed on the present selection, and a change is made in the direction pointed to by the T<sub>0</sub> T<sub>1</sub> indication. The CHECK output goes true.

#### Example:

Present and previous selection indicate that the table used is too low. This is a strong indication, and the TREND indication is ignored if it points in the opposite direction:

|      | PROM1 |                     |                         | PROM2       |  |
|------|-------|---------------------|-------------------------|-------------|--|
|      | LOWER | LOW                 | HIGH                    | HIGHER      |  |
| SELl | 1     | 1                   | 0                       | 0           |  |
| LOWR | 1     | 0                   | 1                       | 0           |  |
|      |       | = 11,               | ent selecti<br>too low. | <b>.</b> .  |  |
|      |       | ious sel<br>too low | ection, T <sub>O</sub>  | $T_1 = 11,$ |  |

Trend is to higher tables, TREND = 0. This gives a PROM input 10 11 0 11 11, which, with correct polarities, should be 01 11 1 11 00 or in hexadecimal notation, 0FC. The PROM output is 4, which gives selection signal SEL1 = 0 and LOWR = 1, the next higher table. Now, if the trend had been reverse, TREND = 1, the input would have been OEC. The PROM output is still 4, however, meaning that this indication must be ignored in face of the others.

An example of the trend indication being decisive is the case in which there has been no change of tables, but the present  $T_0$   $T_1$  signals indicate that the table is not satisfactory and should move in a particular direction. If the trend is in the same direction, there will be a change of tables, e.g. D8 = 4. But not if the trend has been in the opposite direction, e.g. C8 = 2.

The Timing Selection Circuits, as they stand, generate table selection signals independently for each channel. This, however, is a matter of convenience and carries no special significance, since the operating environment to which these circuits adapt the operation, is more or less common to all channels.

# 25.5.3 PE SYSTEM DECODING CIRCUITRY

The PE System Decoding circuitry centers around the PE SYSTEM PROM Ul29, and carries further the operation performed by the PE Timing Decoder PROMs in decoding the PE Read Circuit output signals.

The objective of the decoding operation is to separate data bits from phase bits and to develop signals (e.g. VALID) which indicate the occurrence and presence of valid data bits.

To this end the following input signals are needed:

- S<sub>0</sub> S<sub>1</sub> from the Timing Decoder gives information on bit spacing:
  - 00 Drop-in
  - 01 Bit Spacing is half a character period
  - 10 Bit Spacing is a full character period
  - 11 Drop-out

indicates present logical level LEVEL of read input. If, in the forward mode, the level goes from low to high, it must, in reverse mode, go from high to low, due to the nature of magnetic recording. Since in PE the bits are represented, not by the level, but by the direction of level change or transition, it becomes necessary to reverse the LEVEL polarity when going in the reverse mode. This is accomplished by exclusive-oring the LEVEL signal with the SFC signal (Synchronous Forward Command).

CNT6 from the PE Read Circuits indicates a drop-out.

PEDET indicates that a block of data has been detected in the PE mode. It goes true on the 4-8th preamble byte.

LLEVEL indicates the previous LEVEL signal, which was stored in a RAM on the preceding scan cycle. If the present LEVEL signal is different from the previous LLEVEL signal, then obviously there must have occurred a level transition since last being scanned. This represents one of the conditions for generating a new VALID bit signal.

LVALID indicates whether the last transition was a data transition, and this signal was stored in the RAM on preceding scan cycle. If LVALID is true and the  $S_0$   $S_1$ signal indicates a short bit spacing, then obviously the new transition detected by the LEVEL/ LLFVEL comparison must be a phase transition, and the VALID signal goes false. A long bit spacing makes VALID true. Conversely, if LVALID is false, then a short bit spacing makes VALID true. A long bit spacing (signifying a drop-out) is in this case not possible since this possibility has already been excluded by the presence of the LVALID signal in the PE Timing Decoder PROMs, which supply the  $S_0$   $S_1$  signal. Instead the  $S_0$   $S_1$  signal indicates a drop-out in the first place.

SCAN9 is used to make all outputs false in the 10th time slot when no read channel is being sampled.

ENV if false, indicates that a dropout has been detected in the present channel in previous scan cycles. It is used to stop further clocking of signal parameters for that channel.

In this description of the input signals an idea has been given as to the line of reasoning which has been followed in specifying the PE System PROM outputs and how the desired signals can be obtained. These signals include:

- PIN 12, VALID. This signal goes true when a transition occuring since the channel was last scanned, proves to be a valid data tran-It is stored in the sition. U159 RAM and used as LVALID in the next scan cycle. It is also synchronized to the system clock and emerges as the LOAD signal which is used to strobe the new data bit into the Skew Buffer. The logical level of the data bit is then given by the polarity of the DLEVEL signal, which is the LEVEL signal synchronized to the system clock and delayed one clock period. More specifically, VALID is true (Pin 12 = 0) when
  - Data follows a phase transition after half a character period:

 $s_0 s_1 = 01$ 

LVALID = 0

LEVEL = LLEVEL

(There has been a transition)

CNT6 = 0

PEDET = 1

SCAN9 = 0

2) Data follows a data transition after a full character period:

 $s_0 s_1 = 10$ 

LVALID = 1

LEVEL = LLEVEL

CNT6 = 0

PEDET = 1

SCAN9 = 0

3) In the very early phase of the preamble, the first transitions have produced a bit spacing count less than 64 (one-and-a-half character period), and we know that the first data transition bits are 0:

CNT6 = 0

(Count of less than 64)

PEDET = 0

LEVEL = LLEVEL = 0

(Preamble "0")

SCAN9 = 0

 $S_0$   $S_1$  is ignored to prevent disqualification because of early bit spacing deviations, which are assumed to be of a transient nature. Since this will be the first VALID signal, there is, of course, no previous valid LVALID signal to take into account.

Pin 11, DROP-OUT. This signal goes true when a drop-out or a drop-in has been detected. It is used to derive signals SENV and ENV (Envelope) which when true indicate the presence of a valid data bit, that is, the last data bit. The ENV signal is used as an input to the PE System PROM to allow clocking of valid signal parameters in the PE decoding circuits (see page 24 - 14). The SENV signal is used for detection of errors in the Error Detection circuits, and for the identification of ID Bursts and file marks via the Envelope Circuit.

More specifically, DROP-OUT is true (Pin ll = 1) when

1) Bit spacing exceeds the tolerances:

 $S_0 S_1 = 00 \text{ or } 11$ 

CNT6 = 0

(There has been a transition)

PEDET = 1

LEVEL = LLEVEL

(There has been a transition)

SCAN9 = 0

 No transition have been detected within one-and-a-half character period.

CNT6 = 1

(Count of more than 64)

PEDET = 1

SCAN9 = 0

Pin 10, CLOCK ENABLE. This signal is used to enable the clocking (ECK) of RAMs U86, U112 and U159 in the Decoding and Timing Select circuits and to up-date the stored parameter signals.

Such clocking shall normally take place, not on every scan strobe, but on every detected transition.

More specifically, the signal goes true (Pin 10 = 0) when

1) There has as yet been no transition, and it is desirable to define the initial logical level of the LLEVEL signal in the RAM:

PEDET = 0

CNT6 = 0

= 0SCAN9

2) There has been a level transition in the early phase of the preamble:

LEVEL = LLEVEL

= 0PEDET

SCAN9 = 0

3) There has been a normal level transition:

LEVEL = LLEVEL

ENV

CNT6 = 0

PEDET = 1

SCAN9 = 0

Note that as soon as a drop-out occurs, ENV goes false and inhibits further clocking of channel parameter signals, and Thus, ENV will stay false and maintain this frozen state in the following character periods, also in the presence of subsequently generated valid transition signals.

4) In case of a drop-out in the absence of a detected transition, it is desirable to perform clocking in order to register the drop-out and stop further clocking:

CNT6 = 1

PEDET = 1

SCAN9 = 0

These are necessarily the same conditions as for the situation which produced the false DROP-OUT indication, see case 2 for Pin 11.

#### 24.5.4 ENVELOPE CIRCUIT

The ENV signals from the PE System Decoder circuits indicate the presence of valid signals in the various channels being scanned. The ENV signals are shifted serially into a shift register (U152, U78) and are available as a 9bit ENVx byte on every scan cycle when SCAN 0 goes true.

Through a gate U94 the byte can be strobed onto the Data Bus by the B6 strobe and be examined for ID Burst or file mark identification in the microprocessor accumulator. For this purpose there need be no distinction between ENV3 and ENV4 (either one being true will disqualify identification), and they are combined to obtain an 8-bit Data Bus word.

The ENVx signals are also used in the PE correction circuits in the Read Data Output circuitry. By indicating the presence of a drop-out, an ENVx signal will enable the reconstruction of the data bit for that particular channel by passing the appropriate vertical parity check signal.

#### 24.5.5 SKEW BUFFER

On the tape, bits on the various that includes the ENV signal itself. recording tracks, belonging to the same byte, cannot be expected to lie on a straight line across the tracks, or, upon reading, to be received by the formatter at the same time. This is due to head and tape path imperfections. The data is said to have skew, and for PE data the formatter must deskew the data such that the bits belonging to the same byte upon being transferred to the CPU, are all transmitted at the same time.

> This is accomplished in the formatter by storing and delaying the first arrived bits until all the bits belonging to the same byte have arrived, and then transferring them collectively to the CPU.

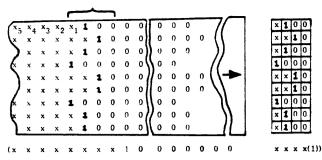


Fig. 24.4 Skewed PE Read Data

An illustration of PE data with skew is shown at left on Fig. 24.4. At right is shown the skew registers or skew buffer into which the data enters. The moment chosen for the illustration is the moment when all the preamble "l"s have just arrived. It is from this moment when there is "data in all channels" that data transfer can take place. A 4-bit skew buffer will normally suffice. (The track shown below the tape indicates what happens if there is exessive skew: The preamble "1" now lands outside the skew buffer, and we have a case of buffer overflow).

The skew pattern, exemplified by the configuration of the preamble "l"s, stays essentially fixed throughout the block, and the skew buffer circuitry, by initially taking note of the relative position of the preamble "l"s, can assemble the bytes by successively reading out bits from these same positions.

In the TDF 4050, the skew buffer consists of a RAM, a selector circuit and a PROM. The RAM is used to store the data while they are being deskewed, and to store the preamble "1"s. The latter are used as flags to indicate to the selector circuit which locations in the RAM it should select bits from. The PROM is used to control the bit and flag positions and to shift these positions as appropriate.

In the RAM there are six positions for each of the 9 channels. When one channel is being sampled, the six positions are appearing at the output of the RAM and form, together with the three input signals LOAD, DIAC and SLEVEL the input address to the PROM. The output of the PROM is then, at the end of the same sampling period, being clocked into the six positions as new data.

As explained before, the read channels

are sampled in succession using suitable scan strobes, and there are about four such samplings per bit period. Hence, on only one sampling in four will there have been a data transition which requires action and clocking of the skew buffer. For this purpose the LOAD signal is used, whereas the SLEVEL signal addicates the bit value.

Each time LOAD goes active, the PROM will shift the flag and the data, and place the newly arrived bit in the position occupied by the previous bit before the shift. Eventually, the flag enters an advanced position in which it is detectable by the selector circuit and signals the presence of a valid data bit. This process goes on for each channel until a flag has been detected in each channel.

This condition is recognized by a special detector circuit, the Data-In-All-Channels detector, whose output, DIAC, then goes true. From this moment on, the PROM will advance the data bit positions as before, but not the flag. The flag will remain in essentially the same position and serve to indicate to the selector circuit the position where the next bit is to be read out. It will do so for the remainder of the block while the incoming data bits are moving up next to it. The relative flag positions, incidentally, will reflect the tape skew pattern.

The bits which are read out will disappear on the next shift and be replaced by the bits next in line. Obviously, it cannot be known until the end of a scan cycle, after all the channels have been scanned, whether there are flags in all channels qualifying a true DIAC signal. Hence, it is on the following scan cycle that DIAC goes true and that the data bits on the selected RAM lines are read out and transferred.

These data bits, however, should only be read out once. Since it is the DIAC signal which enables the read-out, this signal must be made to go false on subsequent scan cycles until there is a set of new valid data bits. This is accomplished by having the PROM backspace all the flags one position. This will result in at least one flag becoming non-detectable by the selector circuit. That will suffice to making

DIAC go false on the subsequent scan cycle.

DIAC will go true again when, upon the entry of new bits, the flags for the channels in question again become detectable. Thus, once the preamble all "l"s have been detected, the flags will in effect rock back and forth one position while the data bits are shifting through all positions up next to the flag. The data bits are being read out when the flag is in the forward position, but making a halt when the flag is in the find position. This position is at such time vacant by virtue of being occupied by the bit just read out, which is no longer of any interest.

In case of a dropout the logic will normally substitute for the missing flag (with SONE, explained later) and maintain normal operation of the skew buffer. Otherwise, new valid bits not being read out will cause overflow by eventually advancing the flag beyond the extreme position.

Fig. 24.5 shows the skew buffer circuit together with details on the function and action going on. Only one channel is considered, and the RAM output lines represent the various positions in the skew register for this channel. The outputs are inverted, and the "1"s shown, representing moment t<sub>1</sub>, are all preamble "0"s. At time t<sub>2</sub>, the preamble "1" appears in the bottom position as a "0". Then with each LOAD clocking, it advances upwards, with data bits in its wake.

In between  $t_1$  and  $t_2$ , the other channels are being scanned, (see Fig. 24.1) but for the present description, only this one channel is being considered.

The four upper position outputs are connected to a priority encoder. The purpose of this encoder is to generate an address for a selector circuit which selects the position in the RAM in which the data bit is to be read out of. The priority encoder is a circuit which will generate a 2-bit binary number according to which of its increasingly higher ranking inputs receives a "0". Thus, the output signal is 00 if the top input is 0, regardless of what the other inputs are. If this input is "1", and the next highest input is "0", then the

address is 01, regardless of what the remaining inputs are. And so on.

Since now the preamble all "0"s appear as 1, and the preamble all "1"s appear as 0, and since the data bit appears next to the preamble "1", the encoder will generate an address signal according to the position next to the preamble all "1" flag.

In addition, the priority encoder is used to detect flags and to generate the desired detection signal to the Data-In-All-Channels Detector. This signal, FLAG, goes true whenever a flag, that is, a "0", appears on any one of the four input lines.

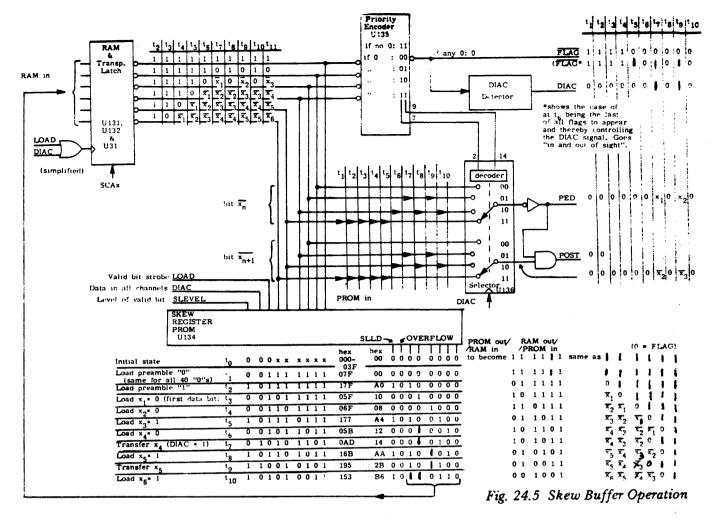
The selector circuit is for illustrative purposes shown in Fig. 24.5 as a 2-pole 4-position selector switch. The encoder output address determines which input position is connected to the selector (pole) output. Thus, a preamble "1" (now actually "0") on the top input to the encoder will connect the selector outputs (poles) to the top inputs of the selector circuits, and so on.

One set of selector input lines is connected to the position for the current bit  $\mathbf{x}_1$  next to the preamble "1" flag. The other set of selector input lines (used for postamble detection), is connected to the bit positions  $\mathbf{x}_2$ , for the bit  $\mathbf{x}_2$  following the current bit  $\mathbf{x}_1$ .

In this manner, the PED line, representing the PE data bits being read out, is effectively being connected to the position in the RAM where the desired bits are to be found, wherever they are.

This is shown in Fig. 24.5 by arrows on the selector circuit input lines. They indicate selected input line for each instant of time  $t_i$ . If there are no "0"s out of the RAM, the address is 11, selecting the bottom input line.

This is also the address when the "0" flag appears on the fourth RAM output line at time  $t_4$ . The flag has become detectable, FLAG goes true, and the Selector circuit looks at  $\mathbf{x}_1$ . At time  $t_5$ , as the "0" move upwards, the address and the select line change, until FLAG has gone true also for all the other channels (not shown) and the DIAC signal goes true, e.g. at time  $t_6$ . The address



is now 01, and the selector is looking at the next highest position, but still looking at  $x_1$ , which has followed the flag to this position. When DIAC goes true, the selector is enabled and the  $x_1$  bit is transferred via the output PED line.

Simultaneously, the PROM reshuffles the RAM positions, the result of which is apparent at time  $t_7$ . The data bit positions are unchanged, but the flag is moved back one position and has replaced  $x_1$  which has now been dropped. A "1" is put in the previous flag position. The clocking of the RAM into the  $t_7$  configuration was performed by the DIAC signal, not the LOAD signal this time.

From now on, the flag appears on the same line on every LOAD cycle (and on the line below on every DIAC cycle), but the data bits continue to advance, and appear on the output PED line in the proper sequence. New bits appear at the bottom RAM output line, current bits appear on the third line just below the flag bit, and old bits are removed as soon as they have been read out.

#### 24.5.5-1 SKEW REGISTER PROM

As previously mentioned, it is the skew register PROM which controls the location of bits and flags in the RAM. The PROM receives as input the LOAD clocking signal, the SLEVEL data bit signal and the DIAC signal. In addition, it looks at the RAM positions being sampled. On this basis is generated the new RAM contents and the Overflow and SLLD signals. The latter signal is the SLEVEL signal AND'ed with the LOAD signal in the PROM and can be ignored in this connection.

The conversion of PROM input signals into appropriate RAM input signals is just a matter of programming the PROM. The general principles will be shown, referring to bottom table of Fig. 24.5.

As long as there is, in the absence of flux transitions on the tape, no LOAD signal, the output of the PROM will be 000 (in hexadecimal notation), regardless of what the other PROM inputs are.

This will also be the case during reading of a data block for those 2 out of 4 scanning cycles in which no tran-

sition (LOAD false) or data transfer (DIAC false) take place. The RAM input being 00, the inverted Ram output will be 3F.

When the first preamble "0" is detected, the LOAD signal will go active, and the SLEVEL will be "0", of course. Since the PROM inputs from the RAM is 3F the total PROM input will be 07F. And since we want the RAM outputs to be all "1" for any preamble "0", the PROM output is programmed to be 000. At t2, when the preamble "1" has been detected, the SLEVEL which represents the value of detected bits, goes true. The PROM input becomes 17F, the RAM output still being 3F. Now we want the zero flag to appear in the bottom position of the RAM, and hence the corresponding PROM output is made to be AO.

In general, in order to write the PROM program, we take the previous RAM outputs and the new LOAD, DIAC and DLEVEL inputs as the new PROM address input, and program the PROM to produce an output which will give the desired next RAM output in inverted form.

The next input bit (at  $t_3$ ) is a data bit which can be either a "0" or a "1". The PROM inputs are respectively 05F and 15F, and for each of these inputs, we must specify a PROM output, in this case 10 and 80 respectively. The bit is chosen to appear in inverted form  $(x_1)$  in the bottom position of the RAM output, whereas the flag has been planted in the next higher position.

At  $t_4$  there is another data bit which can be either "0" or "1". For each of the 05F and 15F cases above, we must specify two PROM output for the  $x_2$  bit. Thus for inputs 06F, 16F, 04F and 14F, we specify outputs 08, A8, 18 and B8, accomplishing the desired shift in data and flag positions. In Fig. 24.5 only the 06F case is shown.

For the next bit, at  $t_5$ , we must now specify 8 PROM outputs, and so on, until we have covered all combinations of the four bit positions.

At time  $t_6$ , we assume that  $x_4=0$ , that a flag has appeared in all 9 channels and that DIAC consequently goes true on the subsequent scan cyle at  $t_7$ . At  $t_7$  then, SLEVEL is "0", DIAC is true, and LOAD is false. The desired RAM output 24-20

shall contain a "0" in the previous  $x_1$  position, a "1" in the previous flag position and  $x_2$   $x_3$   $x_4$  in the same positions. In other words, the PROM shall give a 14 output for an OAD input.

If, by the 6th bit, DIAC is still false, we must specify in the seventh digit of the PROM output a "1" which signifies buffer overflow. This digit is the OVERFLOW signal.

At tg, a new bit  $x_5 = 1$  is clocked in by LOAD. DIAC is false, since at least for one channel a step backwards brings the flag for that channel "out of sight". For example, if in the present illustration DIAC went true already at t4, then no flag would appear on the input lines to the priority encoder, and DIAC would go false again even though the flag is present on these lines for the other channels, such as it is in Fig. 24.5 at  $t_7$ .

At  $t_8$  we want all data bits to move up one position, making room for  $x_5$  at the bottom. The flag is placed in its forward position, leaving its recent position to the advancing  $x_2$ . Input 16B must therefore eventuate as AA.

The Skew Buffer has an output POST which is used for postamble detection. It is obtained by and ing the current bit PED and the following bit in inverted form such that POST will go true when a "l" bit is followed by a "0" bit, such as is expected at the start of the postamble. The POST signal is fed to the Postamble Detector circuit, which checks all the channels.

# 24.5.6 DATA-IN-ALL-CHANNELS (DIAC) DETECTOR

The DIAC Detector will produce an active output signal for one scan cycle following a preceding scan cycle in which the skew buffer FLAG signal has been found to be true for all the 9 channels being scanned. In other words, a string of 9 true FLAG signals precedes the active DIAC signal.

The detector consists essentially of two flip-flops. The first one, U34, is reset by SCANO at the beginning of each scan cycle, unless it is clamped in the set state through its preset input U34-11. Being clocked, the U34 is dependent on a continuous true FLAG signal to stay reset.

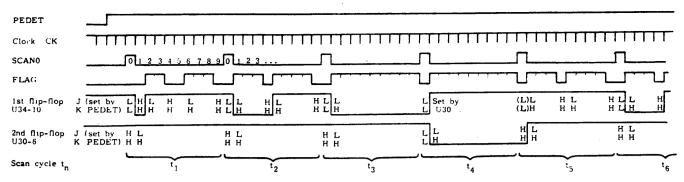


Fig. 24.6 DIAC Detector Signals

The second flip-flop, U30, stays set (DIAC false) as long as the first flip-flop, in the presence of false FLAG pulses, is incapable of staying reset. However, once the first flip-flop, in the absence of any false FLAG pulses, has managed to stay reset during a complete scan cycle, the second flip-flop becomes reset and locks the first flip-flop in the set state, thereby assuring that the second flip-flop will be set on the subsequent scan cycle. In other words, assuring that the DIAC signal is active for no more than one scan cycle at a time. See Fig. 24.6.

Note that in the example of Fig. 24.6, channels represented by SCAN 2,3,6,7,8 are among the first to receive signals  $(t_1)$ , SCAN 4 and 9 the last ones  $(t_2)$ . Later, when the Skew Buffer PROM backspaces the flags, it is affected channels 4 and 9 (at  $t_6$ ) which keeps DIAC false until next LOAD pulse.

The first DIAC signal sets the PREMD (preamble detected) flip-flop U22. Subsequent DIAC signals transfer read data out of the Skew Buffer and strobe them into the read output registers U137, U138 by enabling the CK clock signal.

In the case of a drop-out in one channel, a replacement for the FLAG signal, SONE, is generated in the Logic PROM U158. This will maintain the operation of the circuit and enable the completion of the reading operation which will include corrective action for the drop-out channel. SONE is true whenever a drop-out is indicated (SENV = 0) after the preamble (PREMD = 1).

# 24.5.7 POSTAMBLE DETECTOR

The Postamble Detector will produce an active output signal if, for one complete scan cycle, the POST signal from

the Skew Buffer has been found to be true for all of the 9 channels being scanned.

The detector consists essentially of two flip-flops which operate in a manner similar to the DIAC detector. The input logic involving signals POST, SCANO, etc. have been incorporated into the Logic PROM U158. More specifically, U158-10 =  $\frac{\text{SCANO}}{\text{SCANO}}$  and U158-11 = 1 only when all  $\frac{\text{SCANO}}{\text{SCANO}}$  =  $\frac{\text{SENV}}{\text{SCANO}}$  =  $\frac{\text{POST}}{\text{SCANO}}$  = 1.

The first flip-flop, U30, initially set by the PEDET signal is reset by the SCANO signal (U158-10), and depends on the POST signal continuously being true during a scan cycle in order to stay reset. If it manages to do so, the second flip-flop U22, which was initially reset by PEDET, will be set on the next SCANO strobe, and the M14-POSTD output signal will go true.

If a drop-out occurs in one channel, the SENV signal will in effect replace the POST signal such as SONE did for the FLAG signal in the DIAC Detector. (If SENV = 0, U158-11 = 0 independent of POST.)

# 24.5.8 READ DATA OUTPUT CIRCUITS

In the NRZI mode, read data DBx from the Data Bus are strobed in parallel into shift registers U137, U138 by strobes A2 and A3. Strobe A3 will also, through U6, provide an input to the U148 shift register (down below). The shift register will on the next clock pulse generate a clock pulse which will transfer the read data to the FRDx Output Registers U83, U50. The read data will pass unchanged through the passive PE correction circuits, but will be checked for vertical parity errors by the Parity Checker U139. Four clock pulses later the U6 flip-flop will be

reset, and after still three more clock pulses, the read strobe FRSTR will be generated. Finally, the REGCL signal will clear the shift registers U137, U138.

In the PE mode, the read data represented by PED, is strobed serially into shift registers U137, U138 by the clock, when SCANO is false and the read data is qualified by the DIAC signal. At the end of the scan cycle, when SCANO goes true, the U6 flip-flop enters a "1" into shift register U148 and causes a read strobe, FRSTR, to be generated in the same manner as for the NRZI mode.

This time, however, the PE correction circuits will be active, and if the associated ENVx signal from the Envelope Circuit is false, indicating a dropout, the Parity Checker output will, on the basis of the signal already present on the affected read data line, provide a corrective signal, if necessary, to the Exclusive-or gate of that channel, which will appropriately change the polarity of the read data bit fed into the FRDx Output Register.

## 24.5.9 ERROR DETECTION CIRCUITS

The Parity Checker Ul39 will compute the vertical parity signals of the data part of the read data and compare the result with the parity bit of the read data. Any discrepances will constitute parity error and result in the PARE signal going true. This in turn will give a hard error indication, FHER = 1.

In the 7-track mode, adjustments are made for the options of even or odd parity by exclusive-oring the Parity Checker output with the EPAR signal.

Parity errors are ignored when the program sets CRCG true in expectation of reading the CRC character. They are also ignored when a drop-out has been detected and the parity bit is being used to reconstruct the remaining bytes. In this case the Soft-Error flip-flop FCER (U43) is set and the Q output CORR is active.

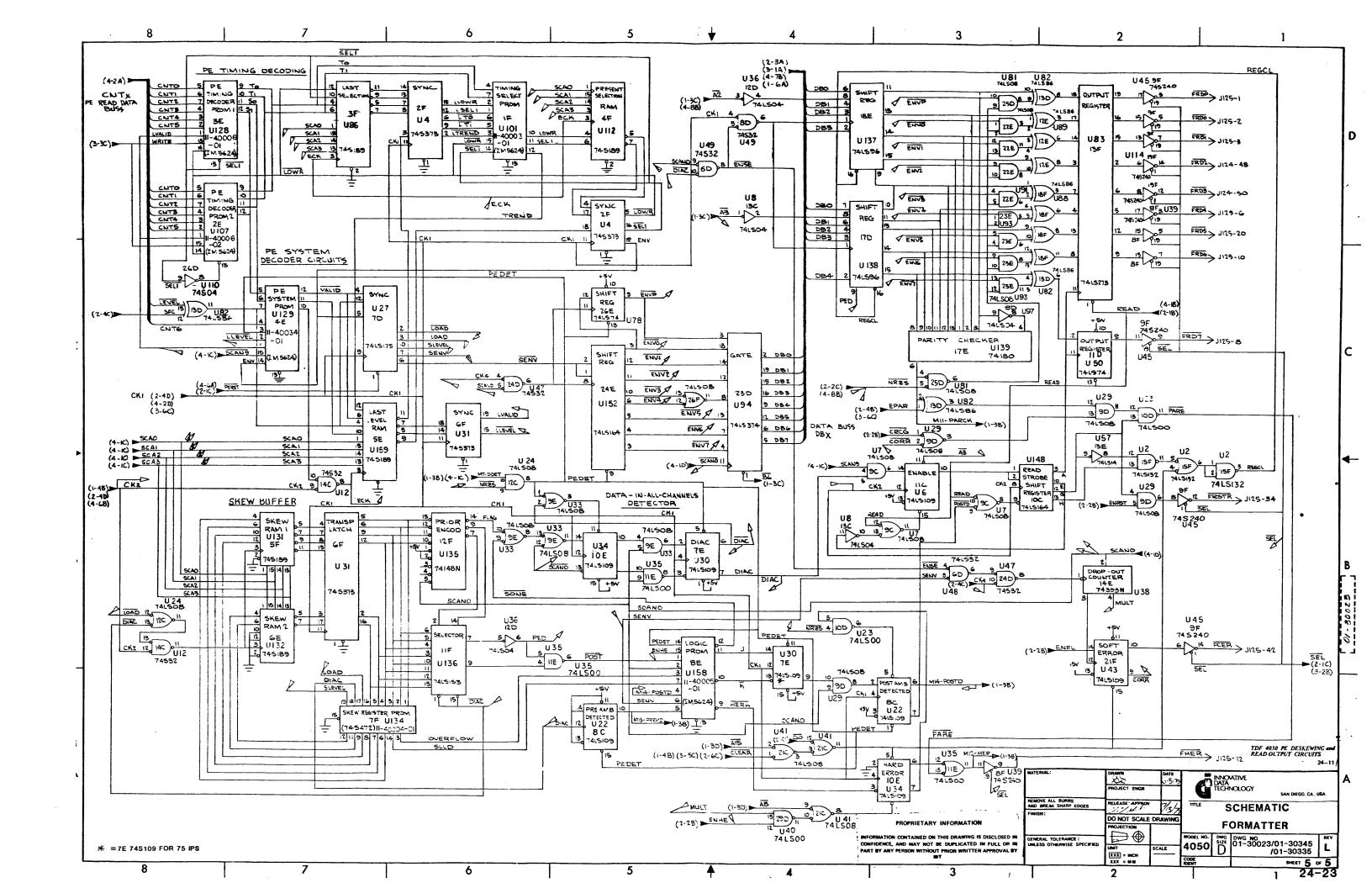
The latter flip-flop is set by drop-out counter U38 which counts the number of drop-outs in each scan cycle. If there is one drop-out the flip-flop is set. 24-22

If there are more than one drop-out, the MULT output goes true and resets the Hard-Error flip-flop U34, making hard-error signal FHER go true.

The Hard-Error flip-flop can also be set by the Logic PROM (U158) which monitors various hard-error conditions. Some of these conditions must hold true for all cases of hardware error.

Thus, the program must have enabled hard-error indication (ENHE = 1), and the hard-error must occur in a data block (PEDET = 1) and in the active part of the scan cycle (SCANO = 0). Then, a hard-error indication will ensue (U158-9 = 0) if there either is a

- Skew Buffer overflow: OVERFLOW=1
- 2) A drop-out in the preamble: SENV = PREMD = 0
- 3) Or an extraneous "1" bit in the
  postamble:
  ENV = SLLD = POSTD = 1



# **SECTION 25** SERVICE

## CONTENTS

- 25.0 SERVICE
- 25.1 CORRECTIVE MAINTENANCE
  - 25.1.1 Required Servicing Equipment 25.1.2 Fault Diagnostics

  - 25.1.2-1 Microprocessor

  - 25.1.2-2 Faulty PE Operation 25.1.2-3 Faulty NRZI Operation
- 25.2 PARTS LIST TDF 4050 FORMATTER
- 25.3 PROM LISTINGS

#### 25.0 SERVICE

### 25.1 CORRECTIVE MAINTENANCE

#### 25.1.1 REQUIRED SERVICING EQUIPMENT

Effective servicing of the complex circuitry of the TDF 4050 Formatter infers the use of advanced test equipment. This should include

- 1. A two-channel 100 MHz oscilloscope (Tektronix 465 or equivalent).
- 2. A logic state analyzer with a 16-bit bus and a memory of minimum 16 words (Hewlett-Packard 1600A, 1670A, 1602A or equivalent.)
- 3. A volt-ohm meter.

A trigger probe at least 4 bits wide (Hewlett-Packard 10250A or equivalent) will give increased flexibility in the use of the logic state analyzer.

Furthermore, a logic pulser (Hew-lett-Packard 546A or equivalent) and current tracer (Hewlett Packard 547A or equivalent) will be useful for locating faults on an internal bus where two or more outputs are connected to the same line.

It is assumed that the formatter is mounted in a TDI 1050 tape drive (from which it draws power and can receive read/status signals) and is connected to a computer, on which it is possible to write simple test programs and run various tape drive operations.

### 25.1.2 FAULT DIAGNOSTICS

Check that the +5V supply voltage is between 4,75 and 5,25 volts.

Enable the formatter (FFEN = 0) and test all four clock outputs from U80-16D. Check actual waveform and clock period against supposed value in the Speed Selection Table on the circuit schematic, 2.

### 25.1.2-1 Formatter System

| ERROR TYPE   | CHECK THAT   |  |  |  |  |  |  |
|--|--|--|--|--|--|--|--|
| <pre>l. The tape drive starts running whenever it is placed on-line, even when the formatter is not enabled.</pre> | A. CLEAR signal not high  B. Motion signals TSFC or TSFR (2)  not high  C. MOTION not low.   |  |  |  |  |  |  |
| 2. When the formatter is enabled the<br>tape drive starts running as soon as<br>it is placed on-line.              | A. Reset the formatter for a moment, (momentarily disable), keeping the tape drive on-line. If the tape drive starts running again when the formatter is enabled, check that the INREQ line is not high, 2.  B. MOTION line not low. If low, go to 25.1.2-2.  C. Reset the formatter and perform checks lA, lB above.  |  |  |  |  |  |  |
| 3. The tape drive will not start running.  | A. Formatter is enabled, FFEN not high, CLEAR not low, 2. Formatter is addressed, SEL not high. Tape drive is on-line, TONL not low.  B. Tape drive is ready, TRDY not low.  C. INREQ not high (but high after GO has been transmitted).  D. MOTION not high. If high, go to 25.1.2-2.  E. Motion line TSFC or TSRC is high, but not both high. If high, check tape drive. |  |  |  |  |  |  |

| ERROR TYPE   | CHECK THAT  |
|--|---|
| 4. The tape drive starts running in a normal way, but won't stop until the formatter is reset. | A. Same error occurs in both PE and NRZI. If in PE only, go to 25.1.2-3. If in NRZI only, go to 25.1.2-4.  B. Same error occurs in both read and write mode. If in read only, go to 25.1.2-2. If in write only, go to 25.1.2-3.1 or 25.1.2-4.1.  C. Go to 25.1.2-2 and follow operation byte to byte. |
| 5. When reading PE tapes, errors occur frequently or constantly.                               | A. Tape drive is not in NRZI mode,  NRZS not low 2.  B. Tape being read is 1600 cpi PE tape.  C. Go to 25.1.2-3.2.  |
| 6. When writing a PE tape, errors occur frequently.  | A. Go to 25.1.2-3.1.  |
| 7. When reading NRZI tapes, errors occur frequently or constantly.                             | A. Tape drive is not in PE mode, NRZS is low. Tape drive has correct head, T7TR, Formatter density is correct, K, H etc.  B. Tape being read is NRZI, proper density and number of tracks.  C. Go to 25.1.2-4.2.  |
| 8. When writing a NRZI tape, errors occur frequently.  | A. Go to 25.1.2-4.1.  |

#### 25.1.2-2 MICROPROCESSOR

Connect the 10 outputs of the program counter, U71-73 1, to the logic state analyzer. The other inputs to the analyzer are connected to the signal lines to be checked. The clock input line is connected to system clock signal CK<sub>3</sub>. Ascertain that the program listing at hand actually represents the program ROM in use.

Disable the formatter for a moment and set up the analyzer to trigger on number 001 (hex notation). With the formatter enabled the analyzer should display the following pattern.

If the operation is normal, the program should now be waiting for a command. Note that because the program counter always indicates current instruction plus 1, the address displayed on the analyzer points to the next instruction and not to the one being currently executed. In this case the highest address actually being used is OOA, not OOB.

If this sequence operates satisfactorily, the analyzer is next set to trigger at some suitable address which will lay in the middle of the memory field of interest, such that operations both immidiately before and after the trigger address can be tested, e.g. at OOC. To break out of the above waiting loop, a command is now made to be transferred from the host computer (or test box). The analyzer should then present the following program steps:

| A00 |         |          |
|-----|---------|----------|
| 00B |         |          |
| 00A | command | detected |
| 00B |         |          |
| 00C | trigger | word     |
| 00D |         |          |
| 00E |         |          |
| 00F |         |          |
| 010 | •       |          |
|     |         |          |

Proceed in this way through the complete program until deviant operation is detected and corrected.

During this process the other analyzer input lines are used to test upon relevant signal lines.

#### 25.1.2-3 FAULTY PE OPERATION

### 25.1.2-3.1 PE Writing

Errors during writing would be detected by the read circuitry. Therefore, the read circuitry must first be excluded as the source of errors. This can be accomplished by forcing through a write operation without having the write ring mounted on the tape reel in the tape drive and using an already written tape known to be error-free.

Program sequence errors can be tracked down by methods suggested in 25.1.2-2. Check at Ull5 3 that proper data are being received from the computer. Use a binary count bit pattern.

Check that proper write data, including phase transition, appear at the output of U144. Check the parity bit, both at U95 and U77, and check the TWDS, FDWDS and TWRS strobe signals.

#### 25.1.2-3.2 PE Reading

In the TDF 4050 the PE reading operation is largely handled by hardware, which for fault detection places a corresponding emphasis on checking hareware.

# Signal presence in the PE Read Circuit

Trigger scope on a data input line, e.g. the P-channel. Observe the STRBx signal at the OR-gate and also the other input to the OR gate, that is, the CNT6-equivalent "reset" signal fed back to the OR-gate. In the absence of data signals, the "reset" signal shall occur every 64 clock counts. In the presence of data signals, there shall be no "reset" pulses. Check all 9 channels. (See Fig. 25.1).

# Bit spacing check 4

Monitor the PE Read Circuits output CNTx bus lines, using the SCANx signals to enable the analyzer one read channel at a time. To examine selected parts of the data block, trigger the analyzer on the respective PEDET, PREMD and POSTD signals, 5.

The PEDET signal shall go high for the duration of the data block. The M13-PREMD and the M14-POSTD signal go high after respectively passing the preamble and postamble "1" and stay high until PEDET goes low.

Ahead of PEDET, the CNTx signals signals shall indicate the count of 64 or 65. After PEDET, in the preamble, and after POSTD, in the postamble, the count shall range from about 10 to 25, around the nominal count of 18. After PREMD, for a bit pattern of alternative "C"s and "l"s, the count shall range from about 25 to 50 around a nominal of 38.

The LEVEL signal, which will shift between "0" and "1" according to read data level, will change every 2 to 4 scan cycles. The data-to-phase transition spacing may deviate somewhat from the phase-to-data transition spacing, but the respective counts are expected to be fairly constant for short sequences of data. However, the counts shall not change from one scan cycle to the next as long as there has been no LEVEL shift. Typical signals are shown on Fig. 25.2 next page.



Fig. 25.1 Typical Read Circuit signals, P-channel (not to scale).

|                              | LLEVEL                | PREMD                 | PEDET            | LVALID                | LEVEL            | CNT 6 | CNT 5   | CNT 4            | CNT 3            | CNT 2            | CNT 1            | CNT 0            |   |
|------------------------------|-----------------------|-----------------------|------------------|-----------------------|------------------|-------|---------|------------------|------------------|------------------|------------------|------------------|---|
| "0" Data<br>Phase            | 0<br>1<br>1<br>0<br>0 | 0 0 0 0               | 0 0 0            | )<br>1<br>1<br>0<br>0 | 0<br>0<br>1<br>1 | 00000 | 0 0 0 0 | 1<br>1<br>1<br>1 | 0 0 0 0          | 0 0 0 0          | 0<br>0<br>1<br>1 | 0<br>0<br>1<br>1 |   |
| "0" Cata<br>PEDET=1<br>Phase | 0<br>1<br>1<br>0      | 0 0 0                 | 0<br>1<br>1<br>1 | 0<br>1<br>1<br>0      | 0<br>0<br>1<br>1 | 0000  | 0 0 0   | 1<br>0<br>0      | 0<br>0<br>1<br>1 | 0<br>0<br>1<br>1 | 0 0 0            | 1 1 1            |   |
| "0" Data                     | 0 1 1 1               | 0 0 0 0               | 1 1 1            | 0 1 1 1               | 0000             | 0000  | 0 0 0 0 | 1 1 1 1          | 0 0 0 0          | 1<br>1<br>1      | 1 1 1            | 1<br>1<br>1      |   |
| "l" Data Phase "l" Data      | 1<br>0<br>0<br>1<br>1 | 0<br>0<br>0<br>1<br>1 | 1<br>1<br>1<br>1 | 1<br>1<br>0<br>0      | 1 0 0 1          | 00000 | 1 0 0 0 | 0 0 1 1 1        | 0 0 0 0          | 1 0 0 0          | 0 0 1 1 0        | 0<br>0<br>1<br>1 |   |
|                              | 0                     | 1                     | 1                | 1                     | 11               | 10    | 0       | 1                | 0                | 0                | 0                | 1,               | J |

Fig. 25.2 Typical signals

Preamble, PEDET = 0 (It takes a few preamble "0"s before a data block is recognized as such and PEDET goes true).

- Tringer word

Preamble, PEDET = 1
(Only a few preamble bytes are included
in this example).

Last 2 bytes of Preamble.

1st data byte

# Skew Buffer 5 :

Reading filemarks does not involve the Skew Buffer and should be checked first to exclude circuitry external to the Skew Buffer as a source of errors.

Use the SCANx signal to enable the analyzer and to observe individual read channels.Monitor inputs and outputs of the U134 Skew Register PROM and check against respective PROM listing in Paragraph 25.3. Use a binary count read data bit pattern and sync on selected counts at the read input.

# Analyze Error Indications (5).

Observe the M12-HER signal. If pulsed, check for parity errors. If set high, check the HERK output of the U158 Logic PROM.

Monitor the PROM inputs on the analyzer and trigger on HERK going low. For conditions of hard error, see Paragraph 24.4.9 and trace error accordingly.

(See also Paragraph 2.13.)

Also, check the various resetting inputs to the HARD ERROR PROM U34.

In general, trace error to suspect chip and monitor inputs and outputs of that chip using the analyzer. Use SCANx to check individual read channels and clock on system clock CKx. Use strategic bit patterns in the data block.

#### 25.1.2-4 FAULTY NRZI OPERATION

### 25.1.2-4.1 NRZI Writing

Use those error-checking procedures in Paragraph 25.1.2-3.1 which are applicable to NRZI operations.

In addition, check the CRC generator circuitry Ull7 3. Use data blocks having the following bit pattern.

#### Channel P 7 6 5 4 3 2 1 0

lst byte 1 0 0 0 0 0 0 0 0 0 0 2nd byte 0 0 0 0 0 0 0 0 0 1 3rd byte 0 0 0 0 0 0 0 0 1 1 etc.

binary count up to and including

17th byte 0 0 0 0 1 0 0 0 0

Expected check characters are then

CRC 0 1 0 1 0 1 0 1 0 LRC 0 1 0 1 1 1 0 1 0

#### 25.1.2-4.2 NRZI Reading

NRZI reading is primarily handled in software, and program sequence errors can be found by methods suggested in Paragraph 25.1.2-2. Check at U37 and U99 4 that proper read data are being received from the tape drive, using e.g. a binary count bit pattern.

Check at the FRDx outputs 5 that the read data are properly transferred, including the FRSTR strobe.

Analyze error indications 5 Observe the M12-HER signal. If pulsed, check for parity errors. If set high, check HARD ERROR PROM U34 clear inputs. See Paragraph 2.6.

