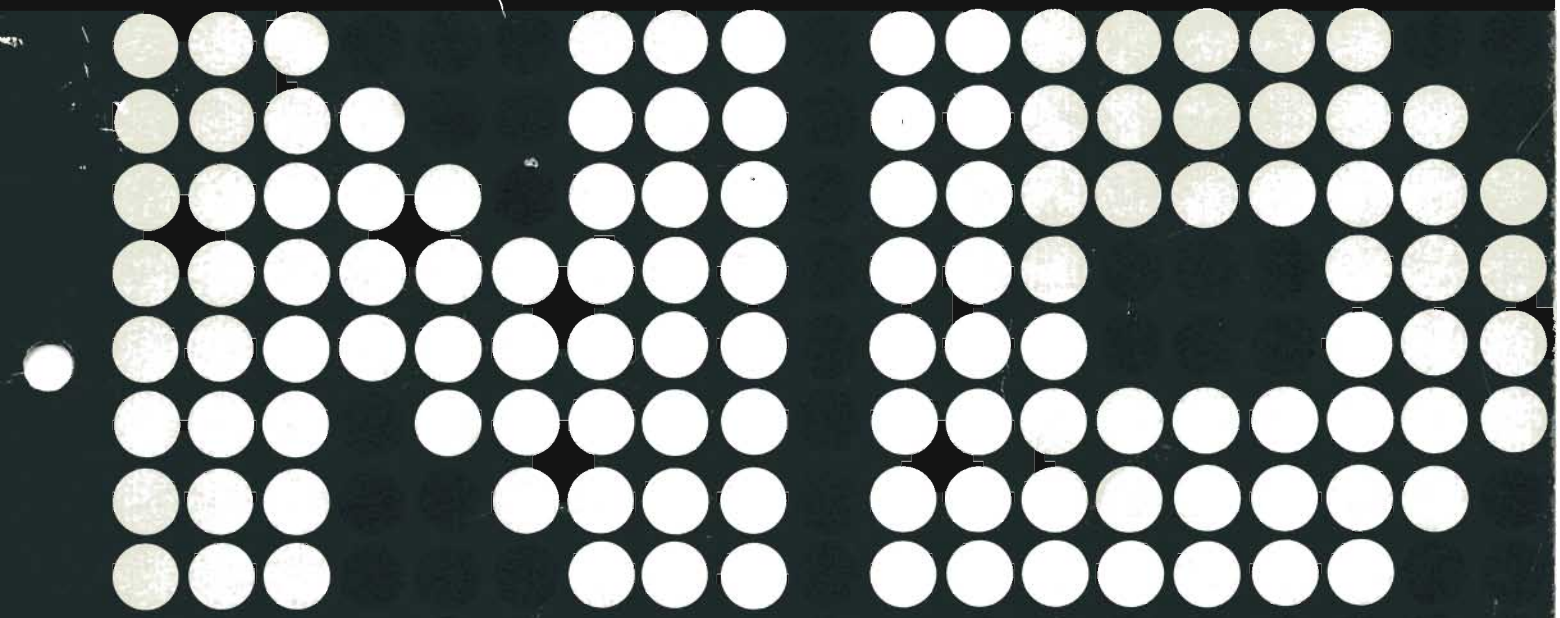
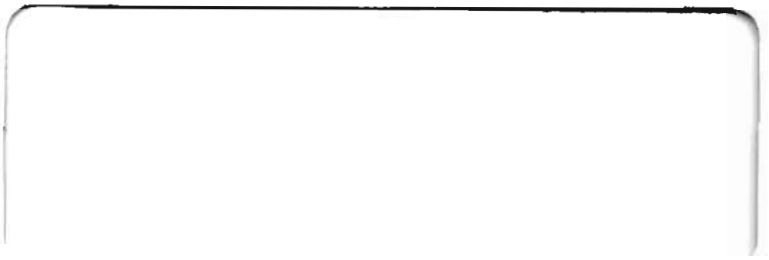
Error Correction Control (ECC) Disk Controller

NORSK DATA A.S





Error Correction Control (ECC) Disk Controller

REVISION RECORD

Revision	Notes
10/78	ORIGINAL PRINTING
06/79	Revision A
	The following pages have been updated:
	1-2, 2-3, 2-4, 5-1, 5-2, 5-3, 5-4 , 5-5, B-1, N-1, N-2

ERROR CORRECTION CONTROL (ECC) DISK CONTROLLER
Publication No. ND-11.013.01

NORSK DATA A.S
Postboks 4, Lindeberg gård
Oslo 10, Norway

COMMENTS, ERROR REPORTS AND REQUESTS

The reader's comment form at the back of this manual invites the reader's evaluation of the manual. Both general and detailed comments are welcome.

The software system field report form issued by Norsk Data may be used to report errors in documentation.

These forms, together with all other types of inquiry and requests for documentation should be addressed to:

Documentation Department
Norsk Data A.S.
Postboks 4, Lindeberg gård
Oslo 10, Norway

TABLE OF CONTENTS

+ + +

1 ECC DISK CONTROLLER — GENERAL DESCRIPTION

The ND558 ECC disk controller can handle from 1 to 4 disk units. The disk units can be ND574 (288 Mbytes disk unit), ND572 (75 Mbytes disk unit) or ND576 (37 Mbytes disk unit).

Any mixture of the above units can also be connected to the same controller.

ECC (Error Correction Control) is standard. ECC implies, for this controller, that all error bursts of up to 11 bits are detected and corrected.

All error bursts of up to 34 bits are detected but not corrected.

The controller converts the DMA data flow (data to/from memory) to a serial bit stream (to/from the selected unit).

A 64 word FIFO (temporary storage) located in the controller allows the data path band width to be exceeded for short intervals. Data flow for read and write is illustrated in Figures 1.1 and 1.2 respectively.

The ECC disk controller is located in the I/O system (Input/Output system) and occupies 9 to 12 I/O card slots which correspond to 1 to 4 units connected (respectively). Refer also to Appendix B.

For details regarding the disk units refer to Appendix M.

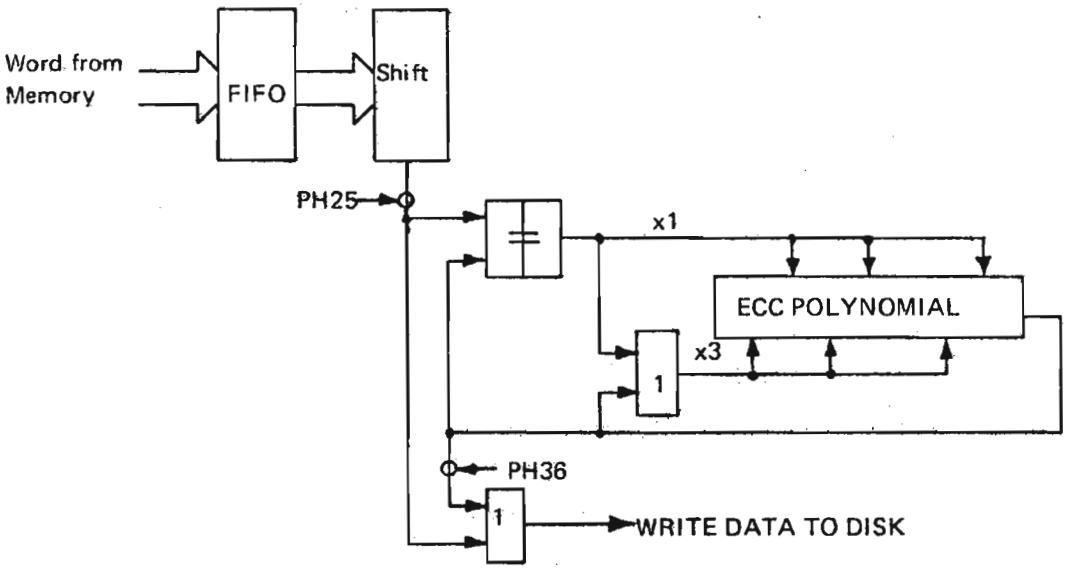


Figure 1.1: Write Data

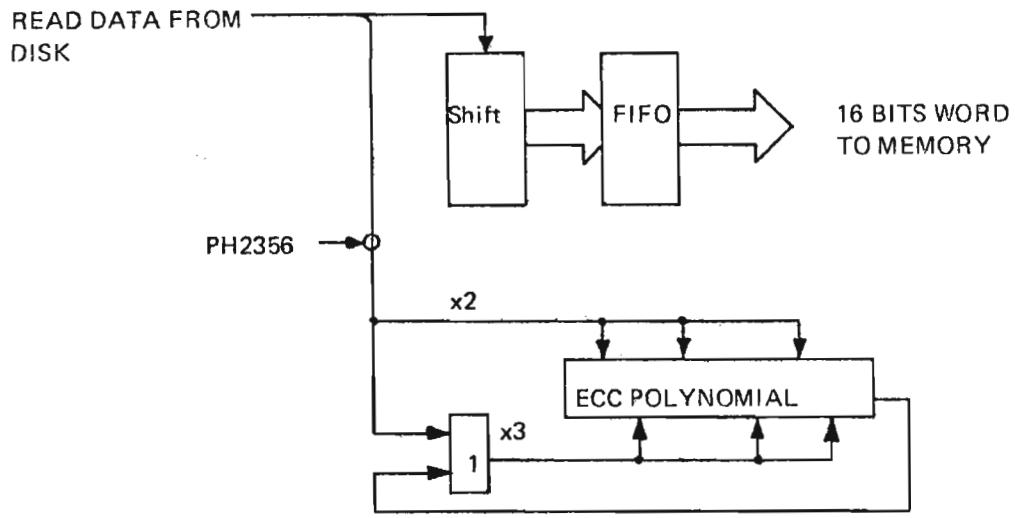


Figure 1.2: Read Data

2 ADDRESSING CONCEPT

Figure 2.1 shows the controller/units interconnection. As already mentioned, a disk system may consist of from 1 to 4 units connected to the same controller.

As illustrated in Figure 2.1, each unit is connected to the controller via two cables. The A cable is daisy-chained through all units and terminated at the end. Only one unit (the selected unit) can communicate with the controller at the same time. There is one B cable for each unit.

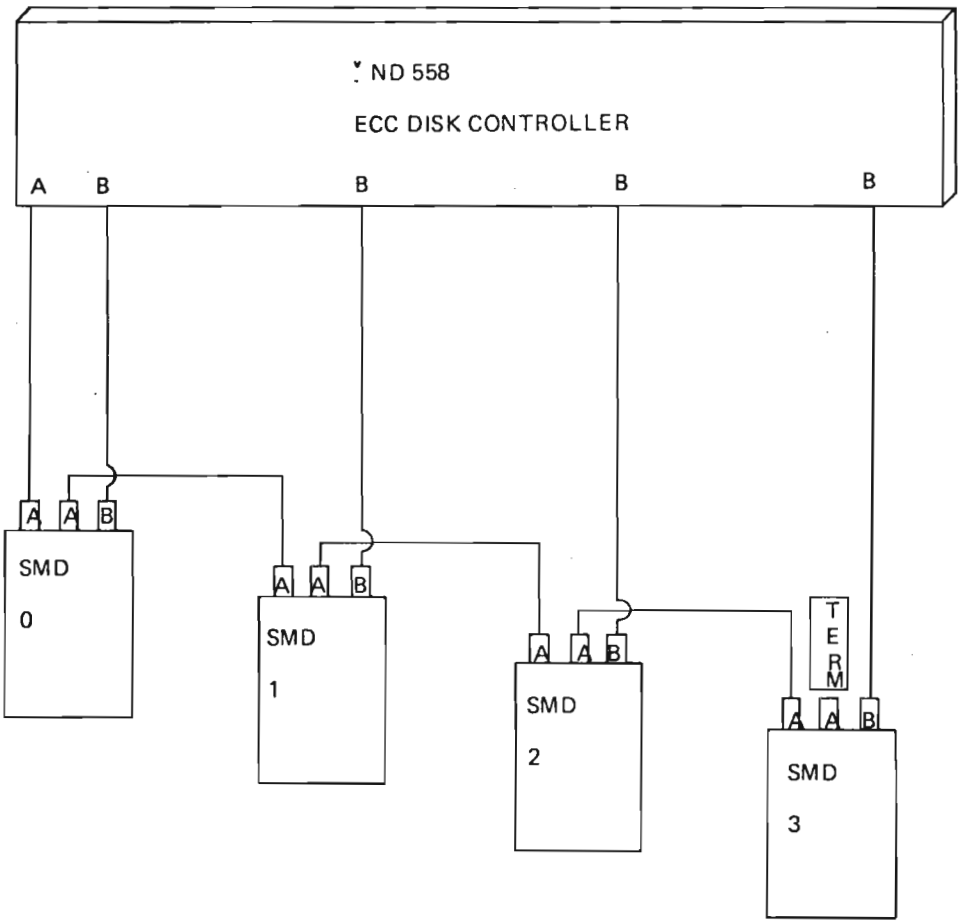


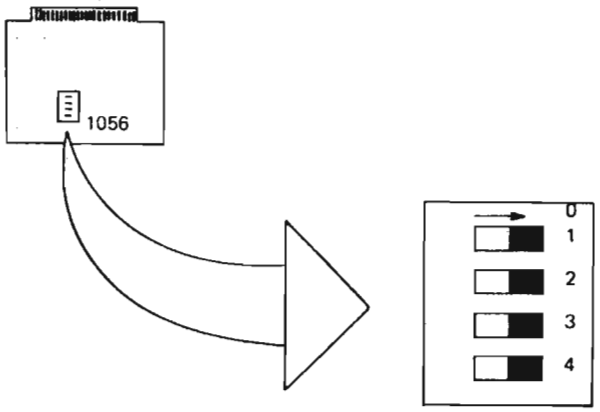
Figure 2.1: Controller/Units Interconnection

2.1 *DEFINING THE UNIT NUMBER*

On the front panel of the unit, a numbered unit select plug defines the unit number. (Each unit is shipped with a plastic bag containing 16 unit select plugs.) On the corresponding "1156 SMD UNIT CONTROL" card (connected via the B cable) in the controller, the unit define switches must be set to the corresponding value (location 9D). Refer to Figure 2.2 for switch setting.

It is imporatnt that the setting of unit number on the unit (plastic plug) and the controller (corresponding 1156 card) are equal.

Example: Illustration shows unit 0 setting on the 1156 board.



UNIT DEFINE TABLE

Switch No. / Unit No.	2^0	2^1	2^2	0 ← Value
→	1	2	3	4
0	ON	ON	ON	ON
1	OFF	ON	ON	ON
2	ON	OFF	ON	OFF
3	OFF	OFF	ON	

Figure 2.2: Unit Number Definition

2.1.1 *Unit Selection*

Bits 7 - 9 in the control word specifies the unit number that the CPU wants to access. The binary value for the unit number is transmitted on the A cable, on the unit select lines, and compared with the select plugs. The unit in which the unit codes match will be selected.

2.2 *ADDRESSING ON DISK PACK*

Once the unit is selected, a specified block of data on a disk pack can be pointed out by the block address. The block address is held by two registers in the interface; block address register I and block address register II. (Refer to Programming Specifications, Appendix N.)

The block address is logically divided into 3 fields:

- the surface (head) selection
- the track selection
- the sector selection

Figure 2.3 illustrates the block address format.

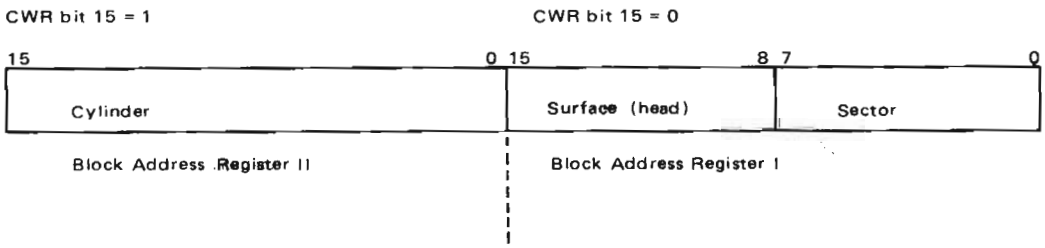


Figure 2.3: Block Address Format

2.2.1 *Surface (Head Selection)*

— The 288 Mbytes disk has 19 (0 - 18) recording surfaces (heads) and one prerecorded servo surface.

— The 38/75 Mbytes disk has 5 (0 - 4) recording surfaces (heads) and one prerecorded servo surface.

The servo head is always selected and always reading. The information from the servo surface serves a number of purposes in the unit. Figure 2.4 shows a 38/75 Mbytes disk pack and a 288 Mbytes disk pack.

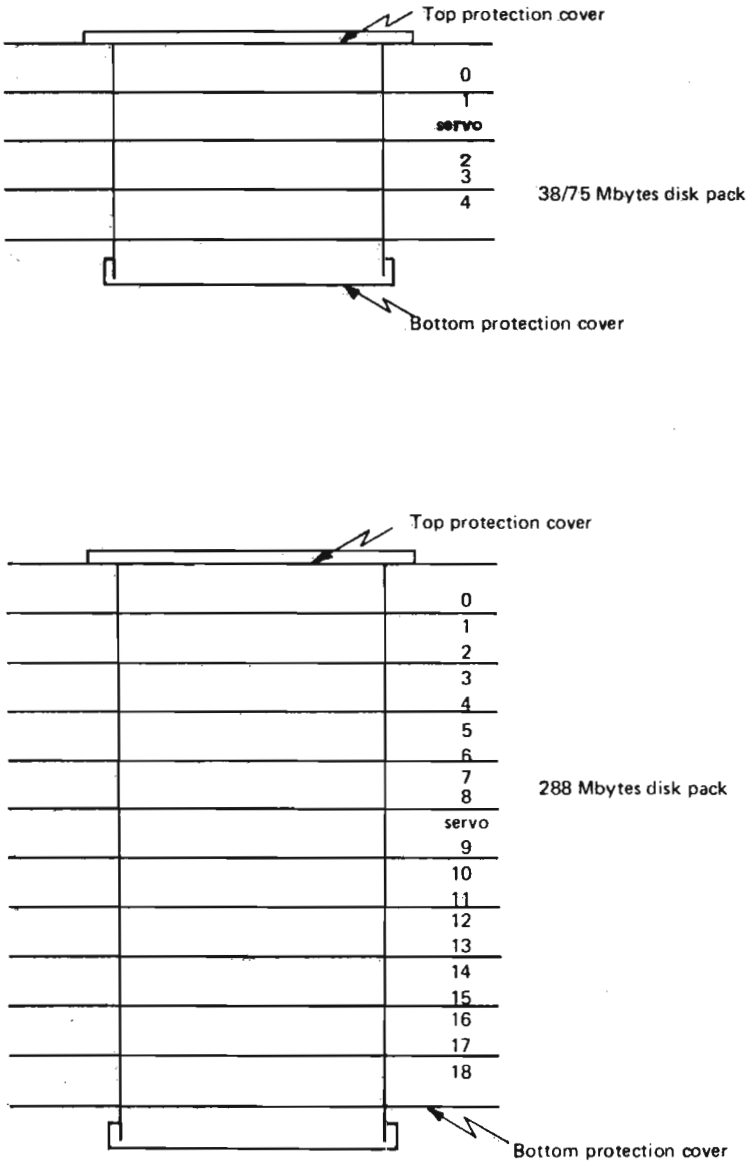


Figure 2.4: 38/75 and 288 Mbytes Disk Packs

2.2.2 *Cylinder Selection*

When control word bit 2 is activated, the content of block address register II (cylinder number) is tranferred to the servo system in the selected unit. Logic in the unit will calculate the difference between the current cylinder and the new one. The difference and direction will command the servo to seek to the new cylinder.

- The 38 Mbytes disk has 411 cylinders
- The 75/288 Mbytes disk has 823 cylinders.

Refer also to Figure 2.5.

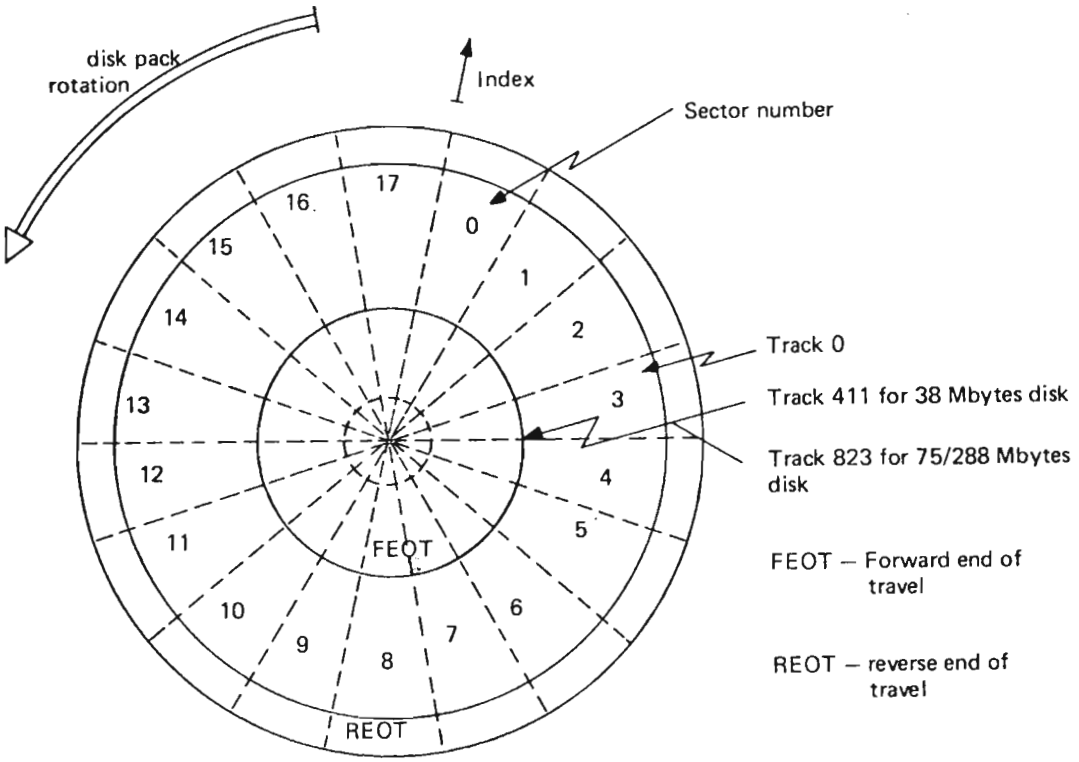


Figure 2.5: Tracks and Selectors

2.2.3 *Sector Selection*

The number of sectors accepted by the controller is 18 and applies for 38/75 and 288 Mbytes disk units.

The sector number is held by the lower byte of block address register I (refer to Figure 2.3).

On 1156 (SMD unit control) when the sector counter (syncronized to the sectors on the disk pack) matches the sector part of the block address, a read/write operation can take place.

NOTE! On the unit the sector number is defined by switches. Refer to Appendix E for details.

3 SECTOR FORMAT

As previously stated, the disk pack is divided into 18 sectors. Each sector is again divided into subfields referred to as phases. Appendix G gives a summary of the phases, etc.

3.1 *THE PHASES*

On the 1135 (SMD Timing) board, a bit counter and a phase generator are located. A sector is divided into 8 phases.

Phase 1:

This phase consists of 232 0's ending with 8 1's. The purpose of this phase is:

— to compensate for sidewise mechanical skew between the read/write heads with respect to the servo head.

— to compensate for read circuits set up time.

— to allow the data/clock separation circuits phase lock oscillator to syncronize and lock.

This phase is written during the formatting process.

Phase 2:

During formatting the block address is written onto the disk pack. Refer to Figure 2.3 for format.

Phase 3:

The block address written onto the disk in phase 2 during formatting is at the same time generating a 56 bit error correction code (ECC) which is written onto the disk in phase 3 during formatting. For more details regarding the ECC code, see Chapter 5 and Appendix J.

Phase 4:

This phase is identifical to phase 1. The purpose is to resyncronize the phase lock oscillator in the data/clock separation circuits.

Phase 4 is first written during the formatting process, but will be rewritten for each normal write operation on the sector in question.

Phase 5:

Phase 5 represents the data capasity for the sector and is equal to 512 16 bit words (1/2 K words). This data is taken from memory over a DMA channel during a write operation and reverse for a read operation.

Phase 6:

When the data is written onto the disk in phase 6 an error correction code (ECC) is generated. This code will then be written onto the disk in phase 6. Refer to Chapter 5 and Appendix J for further details.

Phase 7:

This phase consists of 8 1's indicating end of sector.

Phase 8:

This phase consists of 0's and the purpose is to compensate for sidewise mechanical skew between the read/write heads with respect to the servo head.

3.2 *THE CLOCK COUNTER*

The clock counter is also located on the 1135 (SMD timing) board. The clock counter, working as an input to the phase generator which again resets the clock counter at termination of each phase.

The clock pulses, which are counted, derive either from the write clock, read clock or an internal clock oscillator used in test mode.

The circuits which perform the clock selection are located on the 1078 (SMD receiver) module.

4 THE INTERFACE SIGNALS

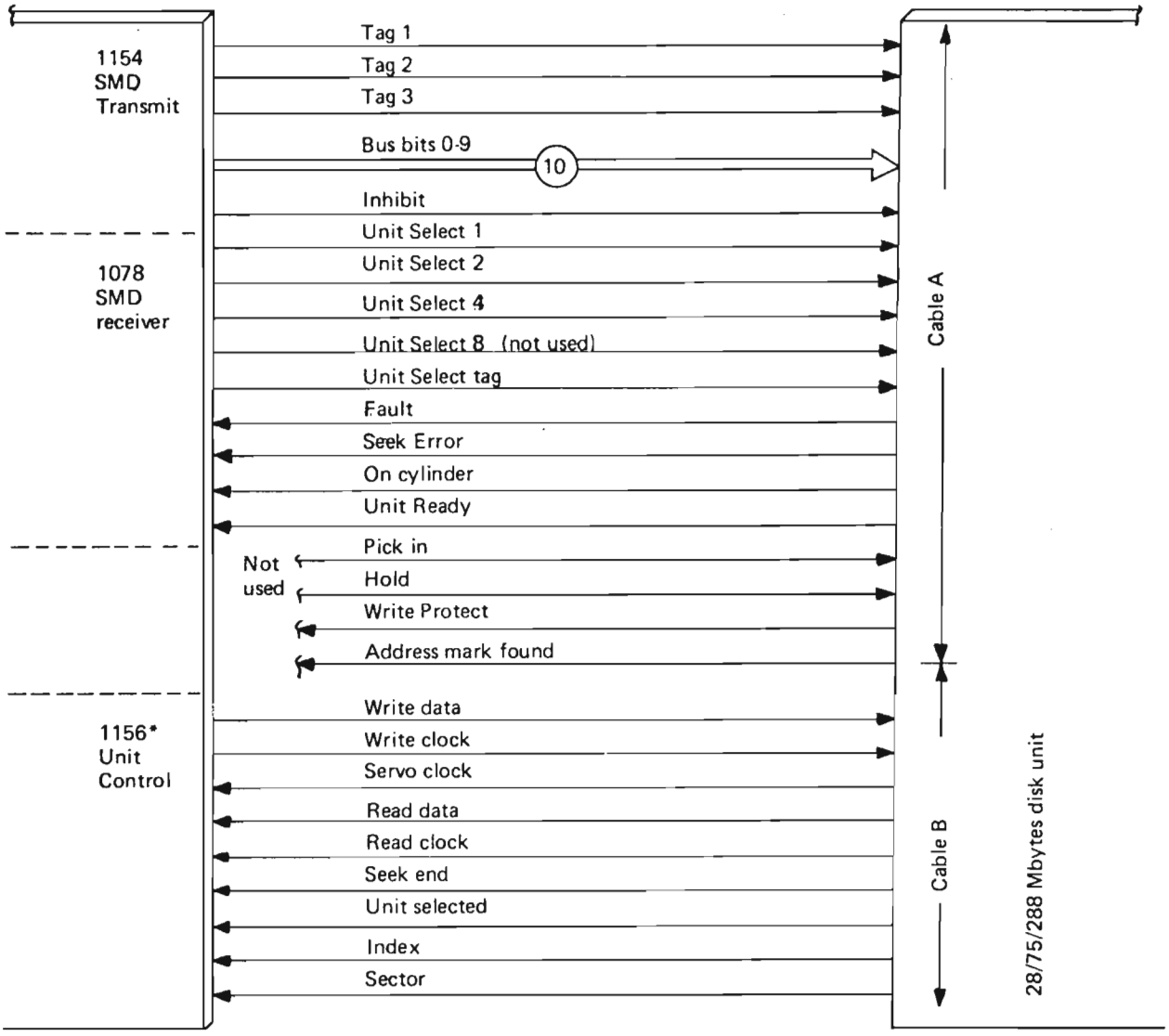
For the following discussion refer to Appendix B and Figure 2.1.

In this chapter, the signals between the unit and the controller will be listed and explained. As depicted in Figure 2.1, the signals are transferred over two cables, the A and B cables.

In order to exchange signals between the controller and a unit over the A cable, the unit must be selected. Refer to Chapter 2.

On the B cable, however, the signal exchange takes place without unit selection.

Figure 4.1 lists the interface signals on the A and B cables and the corresponding cards in the controller.



* One 1156 and one B cable per unit, max imum 4.

Figure 4.1: The Interface Lines

4.1 *SIGNAL EXPLANATION*

This section is divided into 3 parts:

- bus bit usage
- the remaining A cable lines
- the B cable lines

4.1.1 *BUS BIT USAGE*

The bus bits 0 - 9 are used for 3 purposes defined by the tag 1, tag 2 or tag 3 line.

1. Tag 1 line activated.

   Refer to Figure 2.3. The cylinder address taken from the lower part of block address register II is transferred over the bus bits and strobed into the cylinder address register in the selected unit.

2. Tag 2 line activated.

   Refer to Figure 2.3. The head select bits taken from the upper byte of block address register I is transferred over the bus bits and strobed into the head select register in the selected unit.

3. Tag 3 line activated.

   When tag 3 is activated, the various functions as given in the table below is sent from the controller to the selected unit.

Bus Bits:	*Function:*
0	Write Gate. Enable write drivers.
1	Read Gate. Enable the read circuits and data/clock separator circuits in the drive.
2	Servo Offset Plus. Offsets the actuator (heads) from the center of a track position towards the spindle.
3	Servo Offset Minus. Offsets the actuator (heads) from the center of a track position away from the spindle.
4	Fault Clear. Pulse sent to the drive to clear the fault summary latch.
5	Address Mark Enable (not used).
6	Return to Zero Seek (RTZ). Pulse sent to drive causing the actuator to seek back to track zero.
7	Data Strobe Early. Enable the data/clock separator (phased locked oscillator — PL0) to strobe the data at a time earlier than optimum.
8	Data Strobe Late. Enables the data/clock separator (phased locked oscillator - PL0) to strobe the data at a time later than optimum.
9	Not used.

4.1.1.1 Tag Timing

On the 1154 (SMD transmitter) the tag timing generator is located. For every function that should be performed on the disk tag 1, 2 and 3 will be issued in the listed sequence.

For more details refer to Figure 4.2.

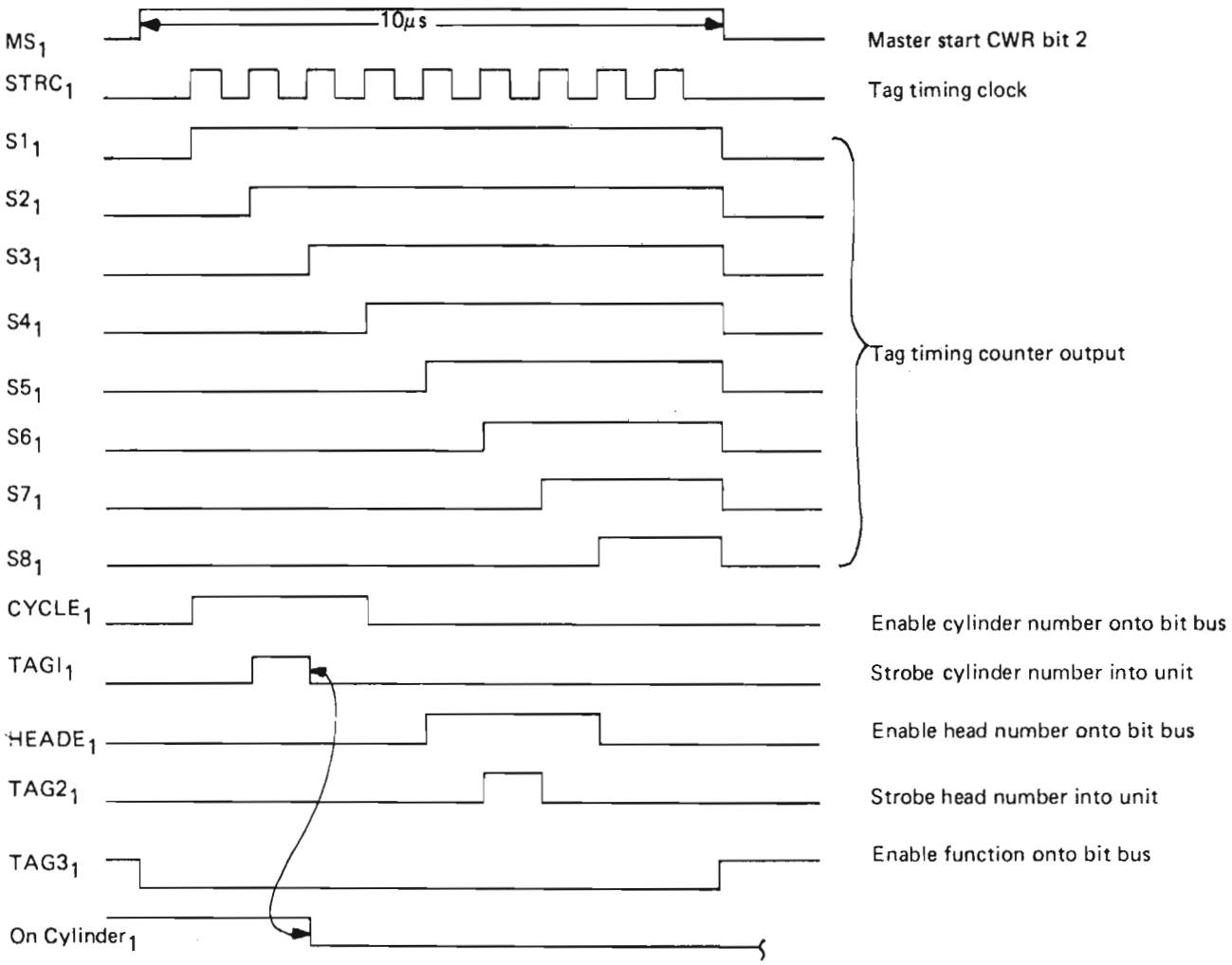


Figure 4.2: Tag Timing

4.1.2 *The Remaining A Cable Lines*

Open Cable Detect

Inhibits unit selection and any unwanted command such as Write Gate when "A" cable is disconnected or controller power is lost.

Unit Select lines 2_0 - 2_3

Used to select the drive. The binary code on these lines must match the code of the drive logical address plug for the drive to be selected. These lines are used in conjunction with the unit select tag (refer to Unit Selection).

Unit Select Tag

Starts unit select sequence (refer to discussion on Unit Selection) and is used in conjunction with Unit Select lines 2_0 - 2_3.

Fault

Indicates that one or more of these faults exist: DC power fault, head select fault, write fault, write or read while off cylinder, and Write Gate during a read operation (refer to Fault and Error Detection).

Seek Error

Indicates that the unit was unable to complete a move within 500 ms, or that carriage has moved to a position outside the recording field. A seek error interrupt also occurs if an address greater than track 822 (410) has been selected. Refer to Seek Functions for more information.

On Cylinder

Indicates drive has positioned the heads over a track (refer to Seek Functions).

Unit Ready

Indicates that drive is selected, up to speed, heads are loaded and no fault exists.

4.1.3 *The B Cable Lines*

Write Data

Carries NRZ data to be recorded on disk pack.

Write Clock

Synchronized to NRZ Write Data, it is a return of the Servo Clock. This signal is transmitted continuously.

Servo Clock

9.677 MHz clock signals derived from servo track dibits (refer to Machine Clock).

Read Data

Carries NRZ data recovered from disk pack (refer to discussions on Read/Write functions).

Read Clock

Clock signals derived from NRZ read data (refer to discussion on Read/Write functions).

Seek End

Seek End is a combination of ON CYL or SEEK ERROR indicating that a seek operation has terminated. If an address greater than 822 (410 on BJ4A1) cylinders has been selected there will be no change in Seek End status (refer to Seek Functions).

Unit Selected

Indicates that the drive is selected. This line must be active before drive will respond to any commands from the controller.

Index

Occurs once per revolution of disk pack and its leading edge is considered leading edge of sector zero.

Sector

Derived from servo surface of disk pack, this signal can occur any number of times per revolution of the disk pack. The number of sector pulses occurring depends on setting of switches on the card in position A06 in logic chassis. Refer to Appendix E for Switch Setting.

5 ERROR CORRECTION CODE

For the following discussion refer to "Track/Sector Format", Appendix G and Chapter 3 where the sector phases are briefly explained.

The theoretical discussion of "Error Correcting Codes" (ECC) is given in Appendix J. From that discussion it is found that a 56 bit "Error Correcting Code" is decided on for use in the ECC disk controller.

5.1 *GENERAL ABOUT ECC*

For this discussion, refer to Figures 1.1 and 1.2. The ECC polynomial register is a shift register with several inputs, several feedbacks and one output. It is the main clock in the controller (read, write or test clock) that perform the shifts. At the same time as the address is shifted out to the disk in phase 2 during formatting, the address appears at the input of the ECC polynomial. The feedback circuits are enabled and at the end of phase 2, a 56 bit polynomial is generated.

In phase 3, the input and feedback is blocked and the content will be shifted out onto the disk.

The exact same sequence takes place during a normal write operation in phase 5 and 6, respectively. The difference, however, is the length of phase 2 and 5.

During the read operation, both address and the ECC code read from disk are shifted into the ECC polynomial register. The feedback circuits are enabled during phase 2 and 3. If the address and the ECC code now read are the same as previously written, the content of the ECC polynomial register shall be equal to zero at the end of phase 3.

The same thing takes place when reading in phase 5 and 6.

5.1.1 *Features of the ECC Polynomial*

Errors that occur to the data on the disk, often caused by bad spots, show up as error bursts. The length of the error burst is normally a few bits long.

The burst length is defined as the number of bits from the first to the last failing bit. If, for example, the first and the last bits on a sector are wrong, we regard this as one error burst of 8192 bits in length. Thus, only one error burst is possible per sector.

It is desirable that we are able to detect and correct error burst as long as possible.

As already mentioned, the ECC polynomial is generated in a 56 bit special purpose shift register according to the formula:

$$G(X) = X_{56} + X_{55} + X_{49} + X_{45} + X_{41} + X_{39} + X_{38} + X_{37} + X_{36} + X_{31} + X_{22} + X_{19} + X_{17} + X_{16} + X_{15} + X_{14} + X_{12} + X_{11} + X_9 + X_5 + X + 1.$$

The ECC polynomial is logically divided into two parts, one LO and one HI portion as indicated in Figure 5.1. With this polynomial it is possible with 100 % reliability, to detect error burst of up to 34 bits and to correct error burst of up to 11 bits. As the length of error bursts exceeds 34 bits, the chance of detecting them reduces slightly. (Additional information in Appendix J.)

It is to be noticed that the polynomial is not generated in the same fashion during read and write. During read a fixed multiplier is inserted so that the period of the polynomial is equal to the length of PH5 + PH6 = 8192 + 56 = 8248. Therefore, where running the ECC operation (M8), explained later, the shift count directly represents the displacement of the error burst. Refer to Figure 5.2.

The period of the polynomial can be found by inserting a bit pattern then close the input and enable the feedbacks, and apply shift pulses until the original bit patterns appear again. The number of shift pulses is then the period of the polynomial.

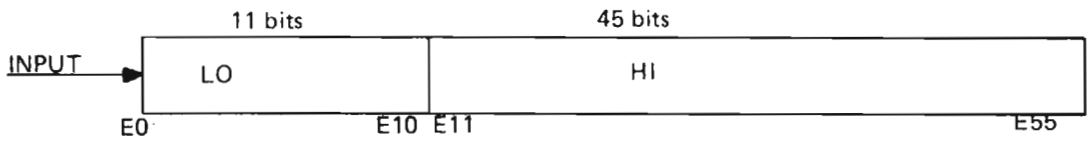


Figure 5.1: ECC Polynomial — LO/HI

5.2 *DATA ERROR (STATUS BIT 9)*

If the address/data read in phase 2/5 is not identical to the data previously written, the ECC polynomial will be nonzero at the beginning of phase 4/7. This is interpreted as Data Error and status bit 9 sets. If interrupt is enabled, an interrupt is generated to the CPU. The ECC polynomial circuits and zero detector are located on 1133 (ECC polynomials).

It is, however, of great importance to know whether the Data Error (status bit 9) means bad address (PH2) or bad data (PH5). If Data Error (status 9) is detected on the address (PH2), bit 15 of the Read Seek Condition register will also set. Refer to Programming Specifications, Appendix N.

5.3 *ANALYZING THE DATA ERROR (STATUS BIT 9)*

The status bit 9 will cause an interrupt and the system will initiate an M8 operation. (This operation is new for this controller.) After termination of the M8, the driver checks if the error has occurred in the sector address or in the data. If bit 15 in Read Seek Condition Register (SCR) is set, the controller shows that the error has occurred in phase 2, sector address. The driver decides that this is a non-correctable error. On the other hand, if SCR bit 15 is not set, the error is in the data field and the ECC polynomial has to be analyzed.

5.3.1 *Run ECC Operation, Error is Correctable*

The M8 operation is initiated to analyze the data error. This operation starts in phase 4 or phase 7. This is a shift operation where the input to the ECC polynomial is closed but the feedbacks are open. The controller clock (CL) is used as shift clock.

An ECC Count Register (ECR) keeps track of the number of shifts while the zero detector circuits look at the HI portion of the ECC polynomial.

When the HI portion is equal to zero, the operation is terminated and interrupt is generated. The content of ECR will point at the last error bit in the data and the content of the LO portion represents the error bits. (LO content is the exclusive OR representation of the actual and expected data.)

5.3.2 The Error is not Correctable

The number of shifts is for the data field maximum 8192 + 56 (ECC) = 8248.

If the number of shifts reaches this value without HI-protion equal zero, the error burst is more than 11 bits long. The error is then not correctable, RSC bit 13 is equal zero and interrupt is generated.

5.3.3 The Error is in the Error Correction Code

If the M8 operation is terminated with an ECR (ECC Count Register) value between 8203 and 8248 (max. count), the data error has occurred in the ECC itself. The data correction is then not required.

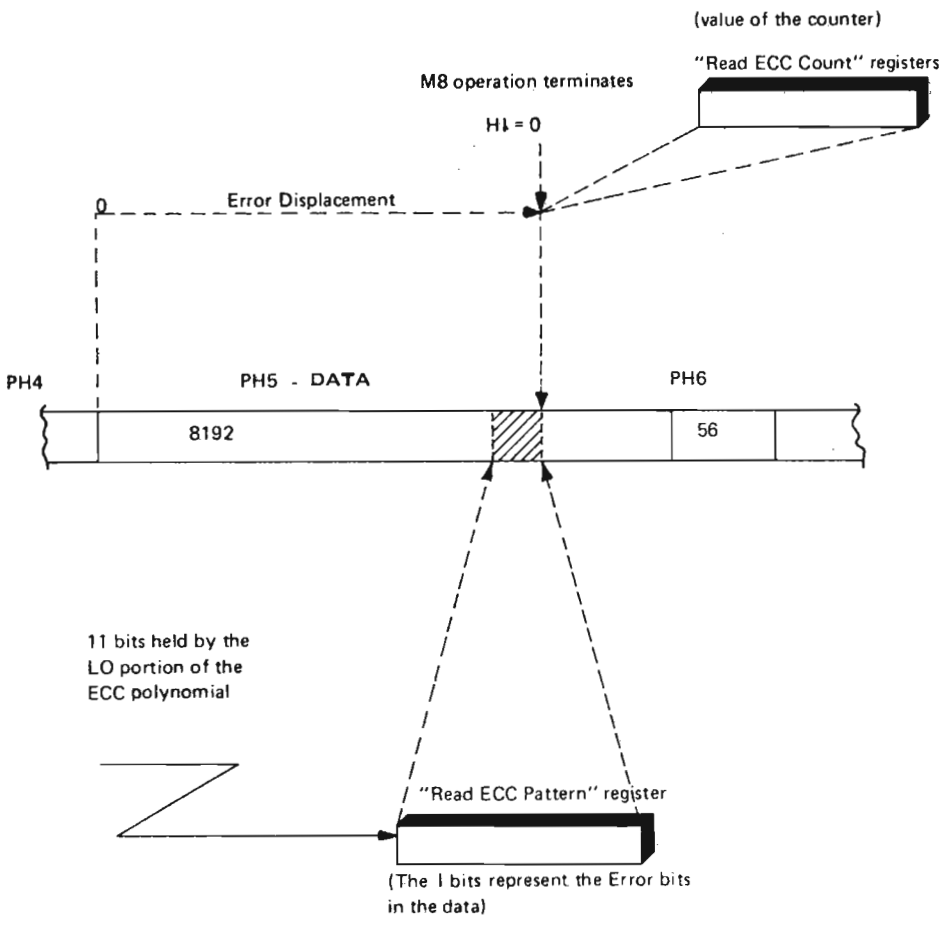


Figure 5.2: Error Detection/ Data Correction

5.3.4 Data Correction (Figure 5.2)

The data read in phase 5 is stored as one block in memory. The Error Pattern Register (EPR) holds the LO-portion of the polynomial and the Error Count Register (ECR) points at the rightmost position of the error burst. With this information the driver routine can easily perform the data correction using an exclusive OR function.

5.4 *THE RELIABILITY OF ERROR CORRECTION CONTROL (ECC)*

The ECC gives a good reliability for the data stored on the disk. It is therefore of importance to be sure of proper operation of the hardware circuits involved.

5.4.1 *The Parity Tree*

For the following discussion refer to Figure 5.3.

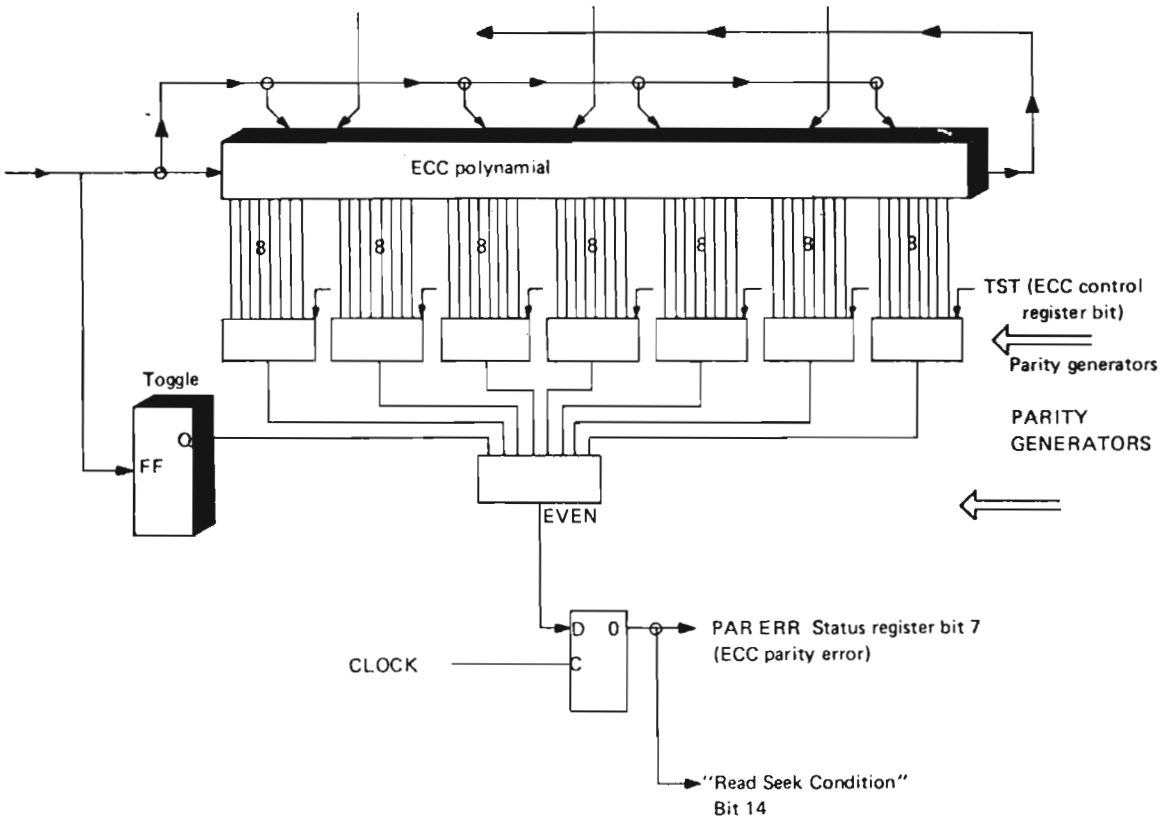
At the same time as data is shifted into the ECC polynomial shift regsiter, the 1's will toggle a flip-flop at the input. Experience shows that the number of ones in the ECC polynomial, including the state of the toggle flip-flop, is always an even number. This is true during the entire shift operation of the ECC register.

A parity tree is therefore employed to monitor the operation of the ECC polynomial. If the ECC polynomial is malfunctioning, status bit 7 (ECC parity error) will set. Bit 14 of the Read Seek Condition will also set to report this error.

5.4.2 *Checking the Parity Tree*

Refer to Figure 5.3.

It is also possible to check the parity tree for proper operations. This is done by forcing a 1 into 7 of the 8 parity generators. If all the 7 parity generators are functioning properly, status bit 7 and Read Seek Condition bit 14 will be set. However, if an even number of parity generators are not functioning it will not be detected by this test. The test bit is set by the ECC control register bit 1.



The parity circuits constantly check theECC polynomial circuits for proper operations. The TST bit checks the parity threes for proper operation.

Figure 5.3: ECC Polynomial Check Circuits

5.4.3 *A Complete Check of the Detection and Correction Capability of the ECC*

This check is performed by using the ECC control register bit 2, the "long" bit. (Refer to Figure 5.4.)

When performing a read operation (M0) with the "long" bit set, phase 5 will be extended with 64 bits so that PH5 long = PH5 + PH6 + PH7. It is thus possible to read PH5 + PH6 + PH7 as data and store it in memory. Under program control it is now possible to introduce a known failure to the data. Then a write operation is performed with the long bit set. The data with the introduced error but correct ECC is written onto the disk. Then a normal read operation is performed. During this operation Data Error should be reported through status register bit 9. An M8 operation can then be initiated and at completion the Read Seek Condition register bit 13 (ECC correctable) will indicate if the error is correctable or not. The ECC Count Register (ECR) will hold the displacement of the error (refer to Figure 5.2) and the ECC Pattern Register (EPR) will hold the ERROR.

This operation gives a complete check of the data paths and the disk controller.

Finally, the ECC polynomial must be cleared by a ECC control bit 0 (reset ECC).

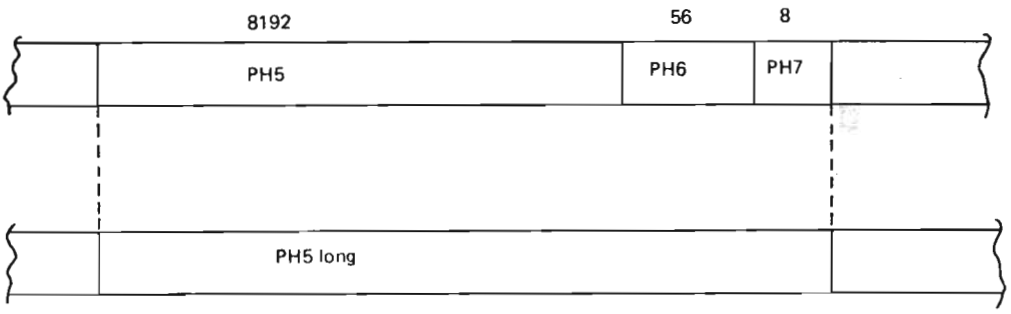


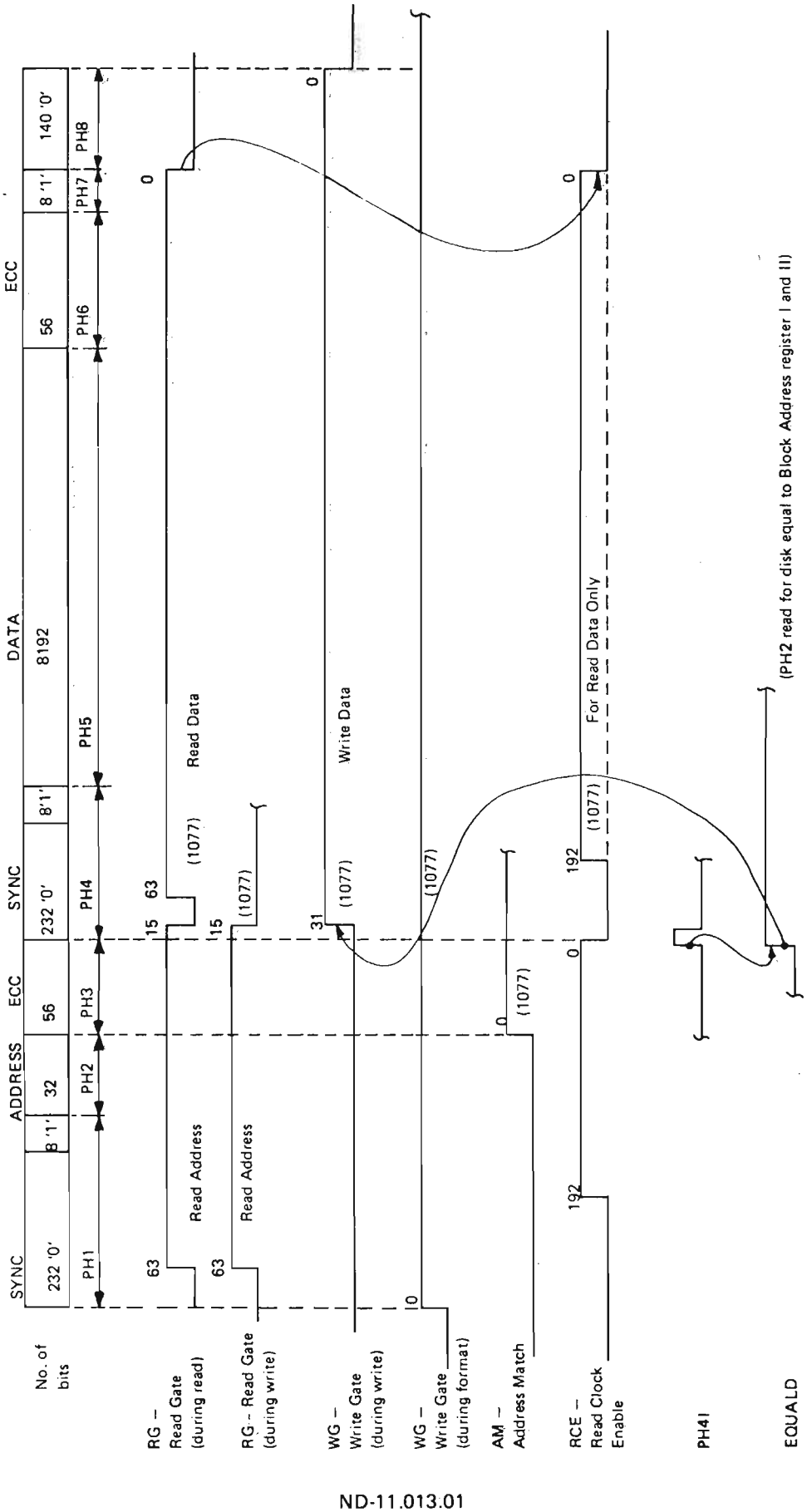
Figure 5.4: Read/Write Long

6 CONTROLLER FUNCTIONS ILLUSTRATED BY TIMING DIAGRAMS

In this chapter, a few timing diagrams are presented. Each line is labeled with the board number on which they are generated or used.

The timing diagrams do not describe the complete functions, but might be helpful when studying the logic diagrams.

The following diagrams appear:



ND-11.013.01

Figure 6.1: Control Timing

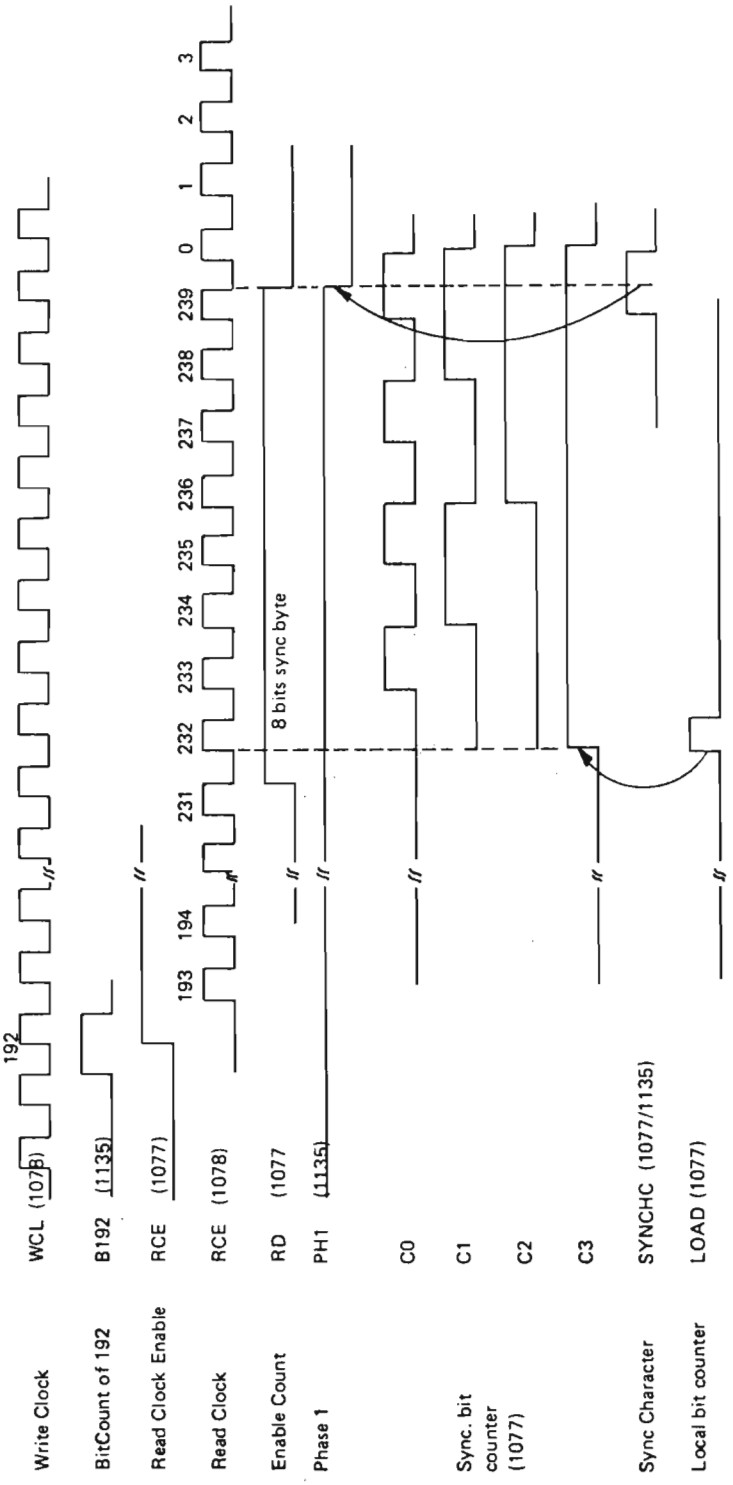


Figure 6.2: Read Sync Byte

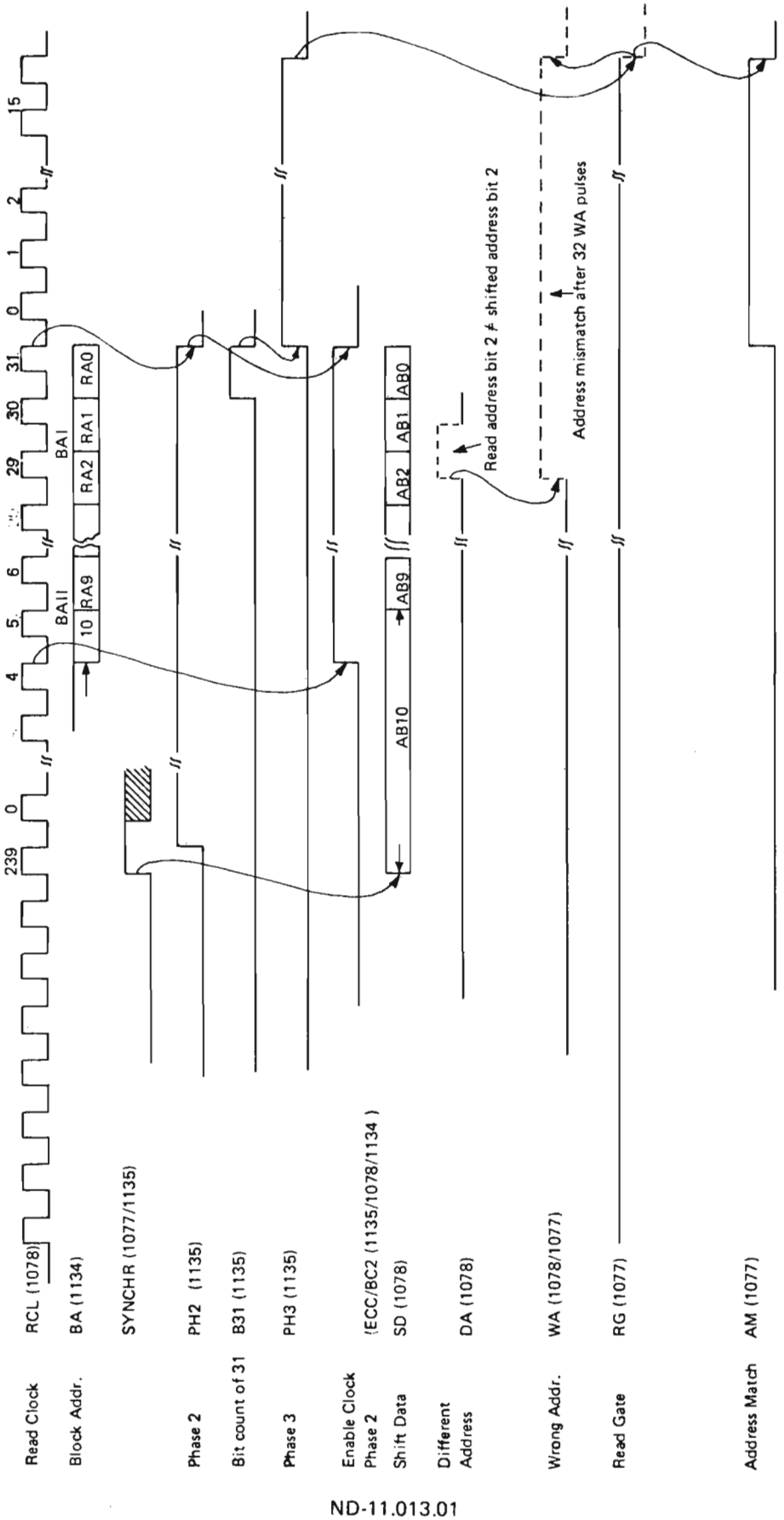


Figure 6.3: Read Address

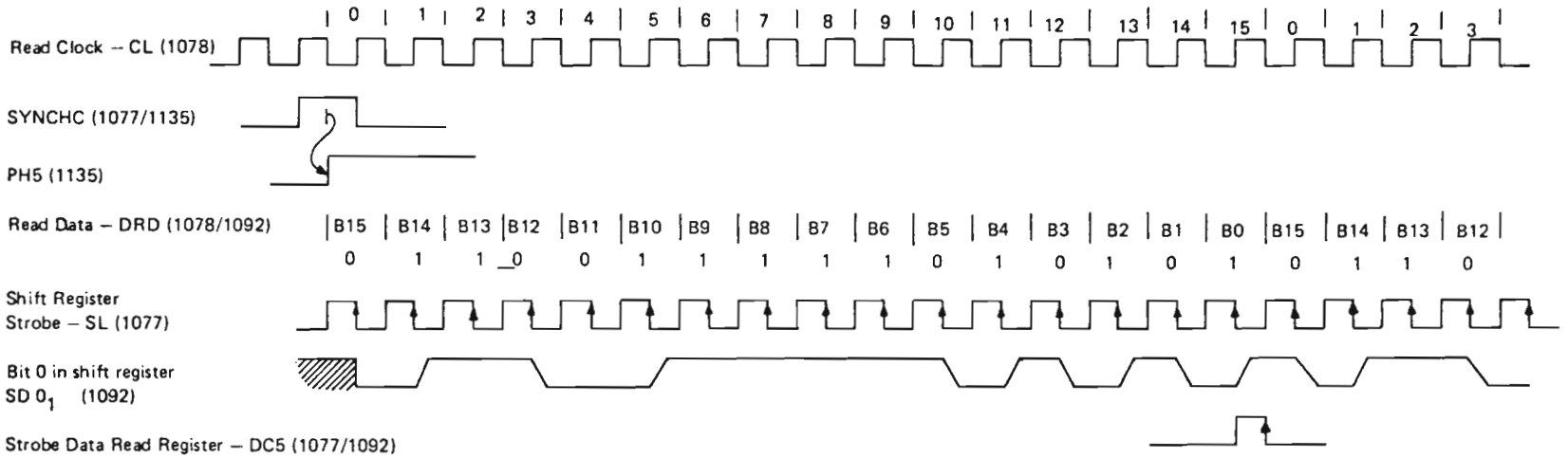


Figure 6.4: Read Data

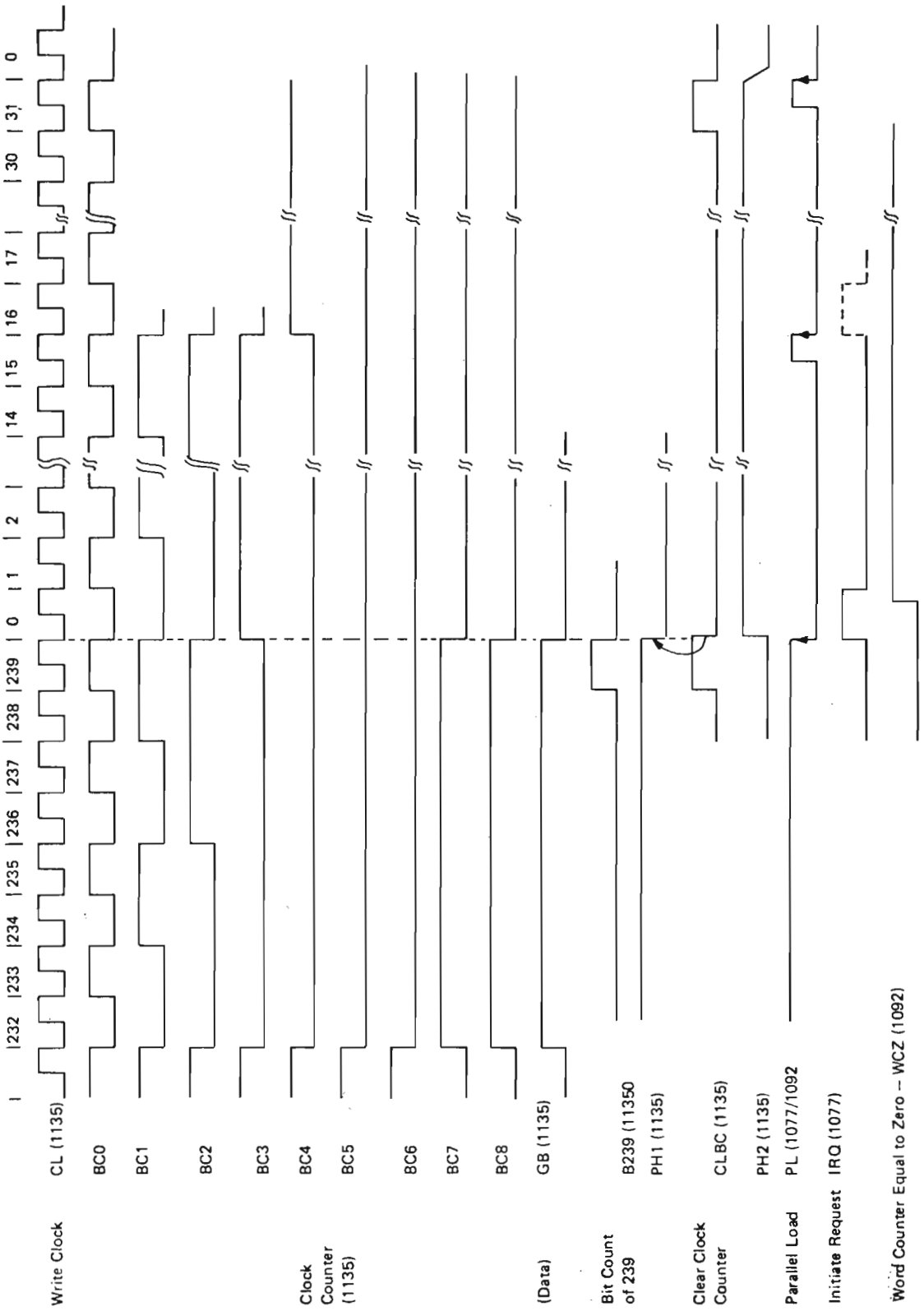


Figure 6.5: Write Sync Byte and Address

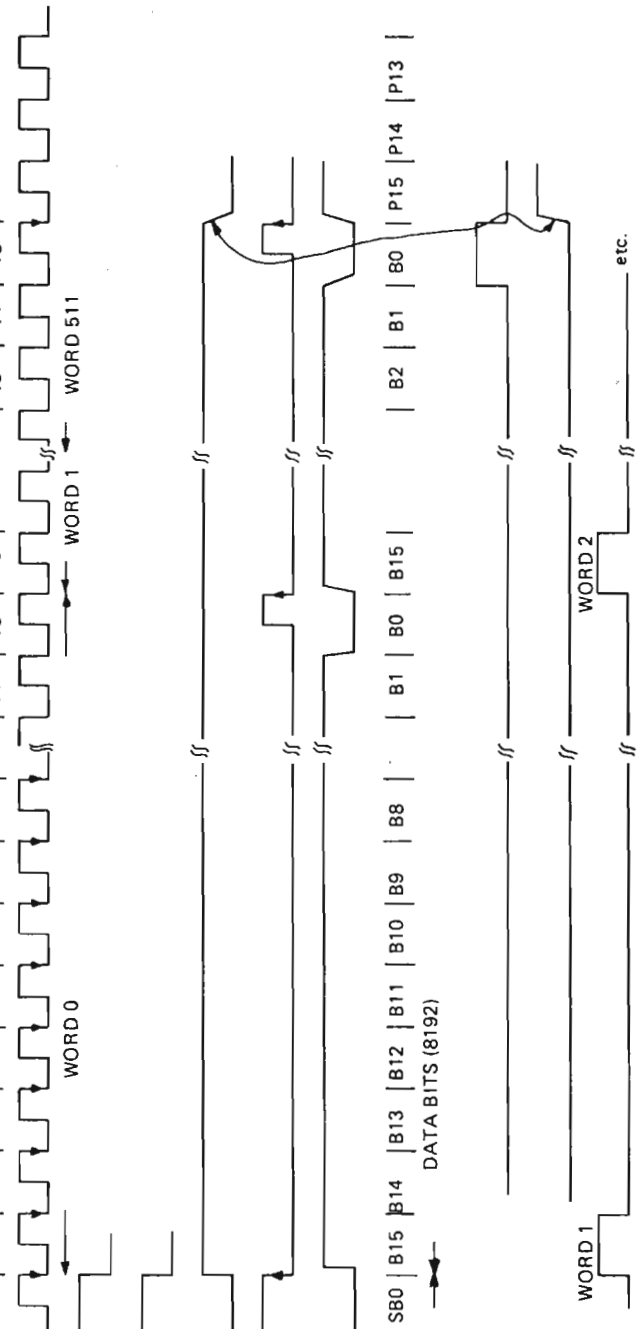


Figure 6.6: Write Data

ND-11.013.01

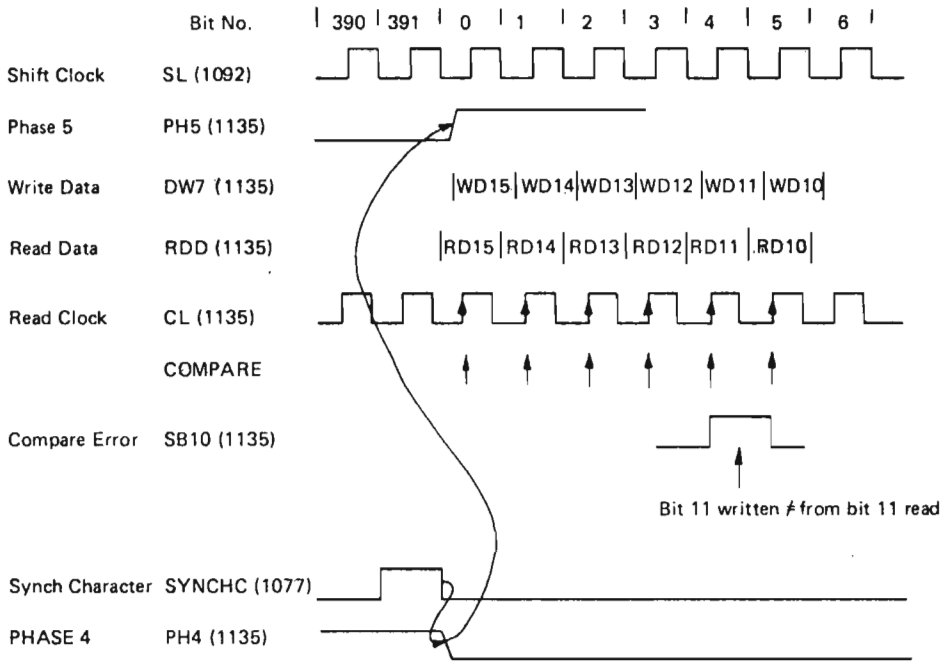


Figure 6.7: Compare Mode

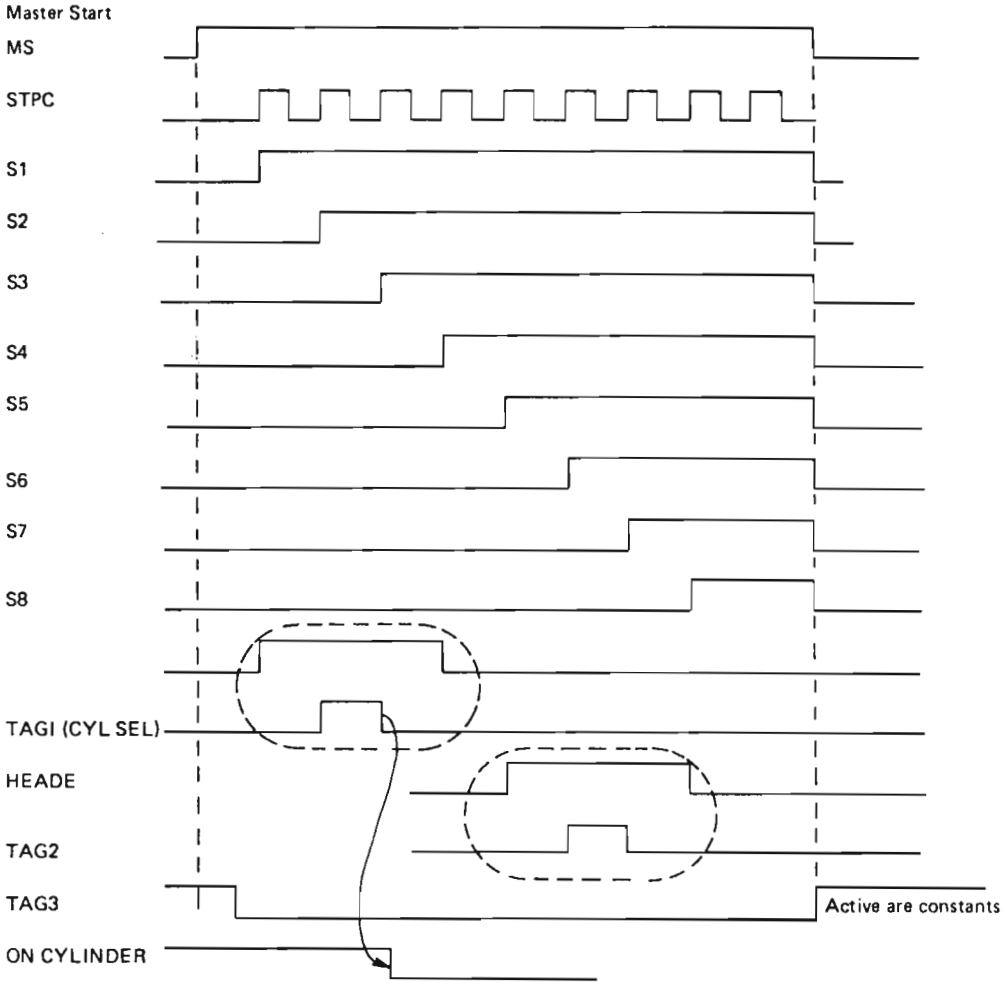
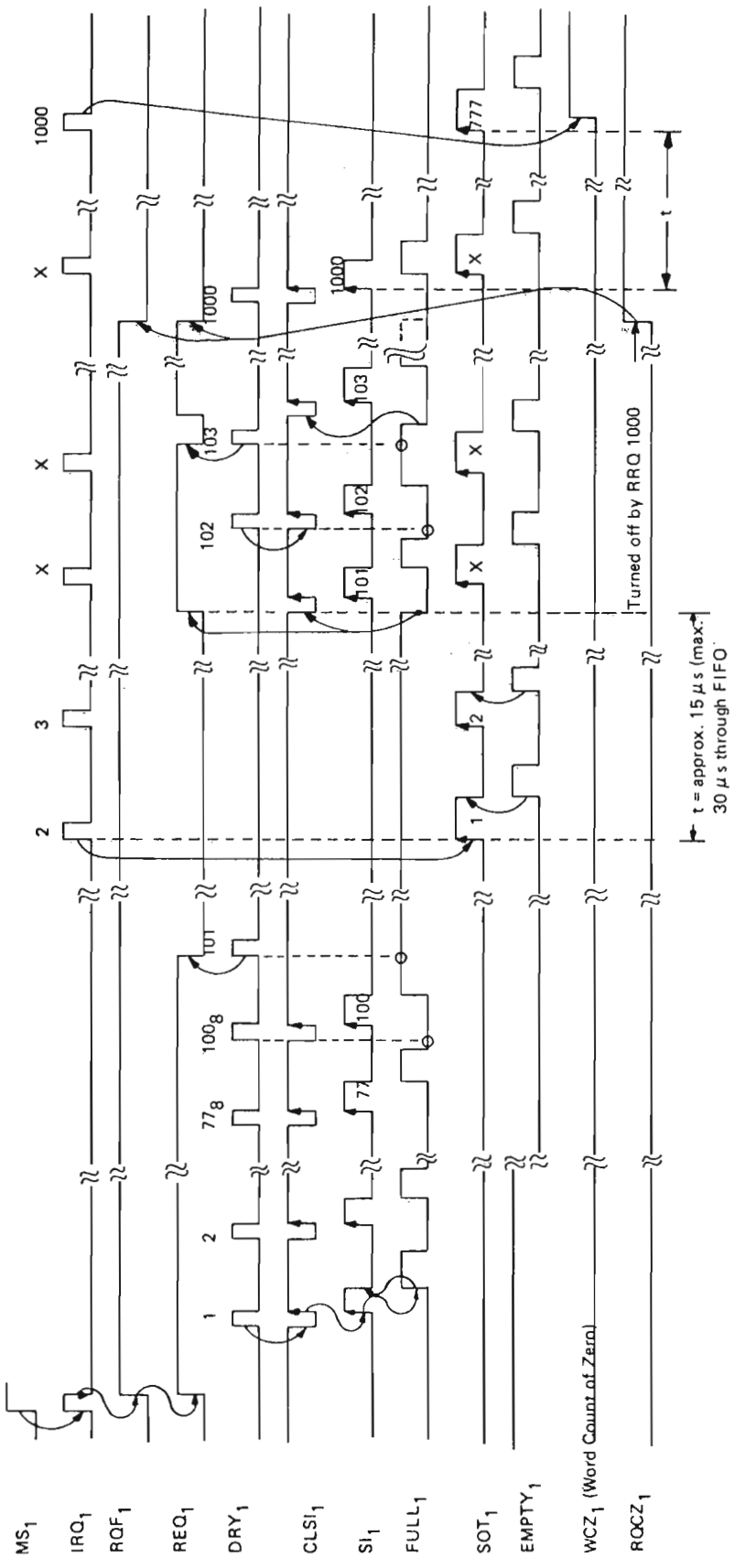


Figure 6.8: Tag Timing



ND-11.013.01

Figure 6.9: Read from Disk (Write to Core) 1085

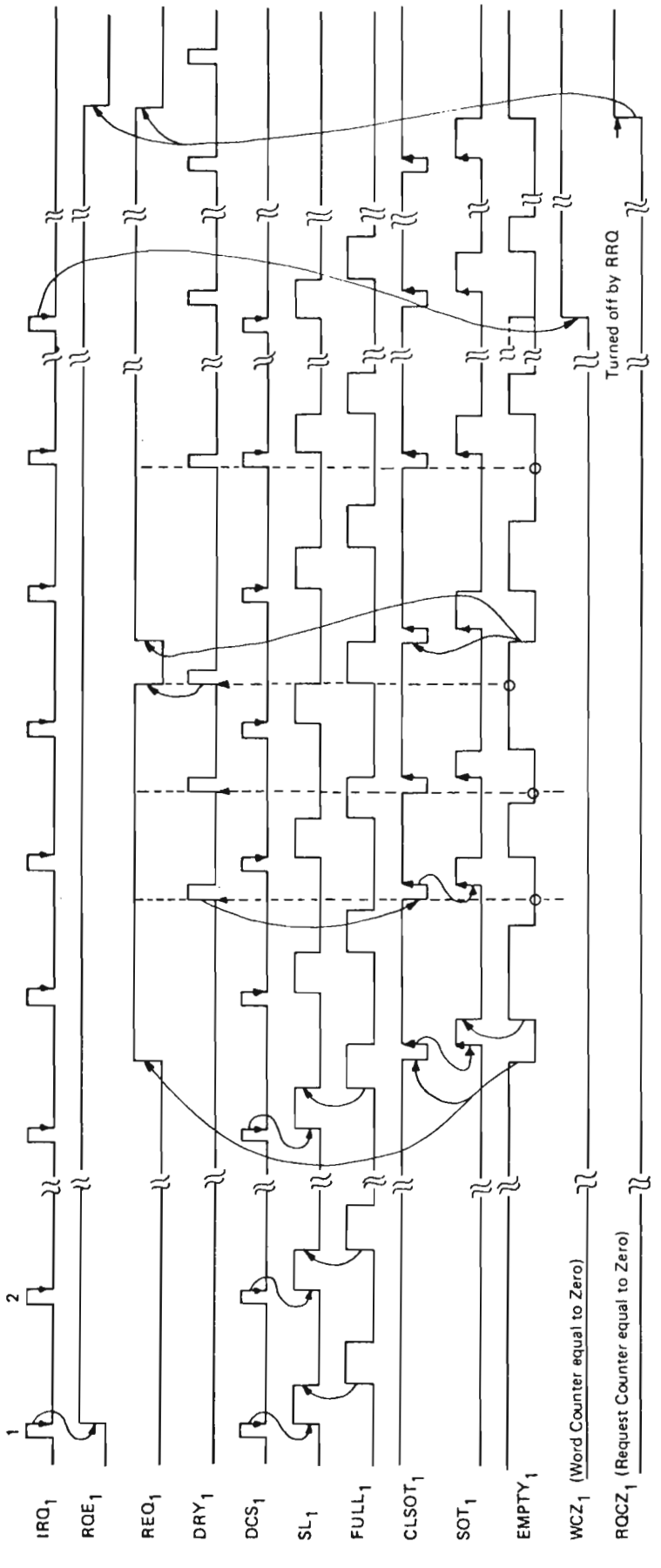


Figure 6.10: Write to Disk (Read from Core) 1095

7 INTERRUPT GENERATION AND HANDLING

For the following discussion refer to Programming Specifications in Appendix N.

The ECC disk controller is wired to interrupt level 11. The various sources for interrupt will be discussed here.

The interrupt sources can be divided into two groups: (Refer to Figure 7.1.)

- Error interrupts
- End of operation interrupts

Error interrupt is enabled by control word bit 1 and End of Operation interrupt is enabled by control word bit 0.

7.1 *ERROR INTERRUPTS*

Error interrupt will occur when status bit 4 is forced on. Status bit 4 is inclusive OR of status bits 5, 6, 7, 8, 9, 10, 11, 12 and 13.

7.2 *END OF OPERATION INTERRUPT*

End of operation interrupt is generated when the BUSY latch is reset. The Busy latch will be reset in one of the following three ways:

- normal end of specified operation (BCOMPL)
- forced clear (CLEAR)
- abnormal end of operation (BRBUSY)

7.2.1 *Normal End of Operation (BCOMPL)*

There are five different ways of generating Normal End of Operation (BCOMPL). Refer to 1077 (SMD Control) gate 17A.

1. Word counter has reached zero during a read or write operation (M_0, M_1, M_2, M_3)

   (PH8 · RA ·WCZ)

2. On cylinder is reached upon an initiate seek command

   (M4 · ON Cyl)

3. Completion of formatting one track.

   (WF · WCZ · SEC)

4. On cylinder on track 0 is reached upon completing a Return to Zero command.

   (RTZ · ON Cyl)

5. Seek completion search positive

   (M6 · SEEKC)

7.2.2 *Forced Clear (CLEAR)*

There are two ways of forcing the busy latch to reset state and thus generate an interrupt.

1. Master clear from the front panel of the CPU (MC).

2. Programmed master clear, i.e., control word bit 4 (device clear) (MDB4 · CW)

7.2.3 *Abnormal End of Operation (BRBUSY)*

This condition corresponds to status bit 12 (refer to 1078 — SMD receiver — gate 18D).

There are five conditions that can set status bit 12 and thus interrupt.

1. Loss of Ready (SB13) condition from the selected unit during an operation.
2. Address mismatch (SB8) occurs when not formatting.
3. Fault line (SB7) from the selected unit is activated during an operation.
4. Illegal load (SB5) while controller is Busy.
5. Time out (SB6).

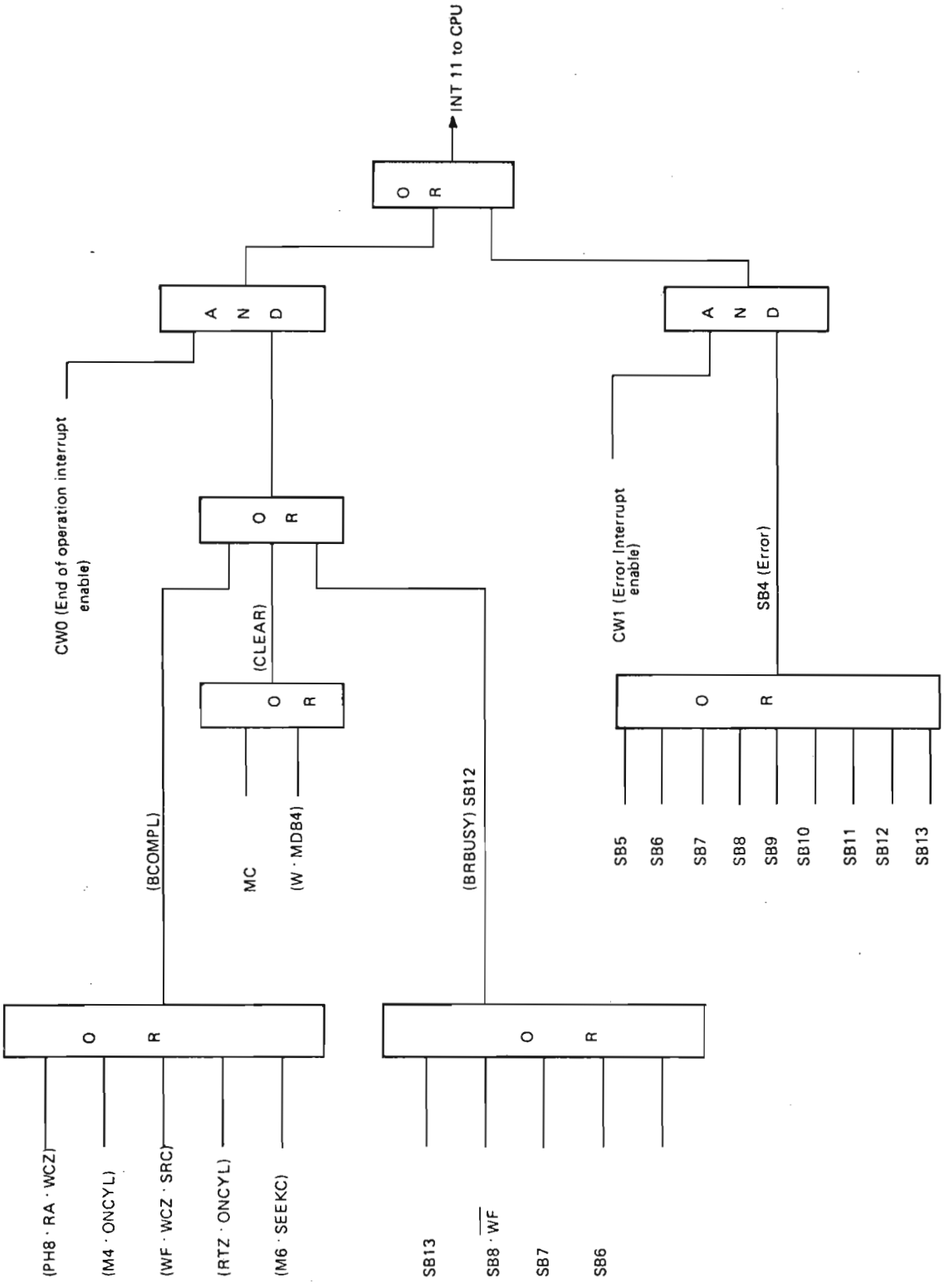


Figure 7.1: Interrupt Generation

8 DEBUGGING GUIDE

The normal procedure for checking out the ECC disk controller should be:

1. Check the operation of the IOX instructions.
2. Check data channel by writing and reading a register in the disk controller.
3. Check the operation of the disk controller by running the controller in test mode.
4. Connect a disk drive to the controller and run test programs.

8.1 *CHECK THE OPERATION OF THE IOX INSTRUCTIONS*

Refer to Programming Specifications, Appendix N.

A check should be made that all the controller registers can be accessed.

Note: Control word register bit 15 selects the two banks of registers.

8.2 *CHECK DATA CHANNEL*

The data channel can be checked by writing and reading the same register in the controller, and then compare the result. The Core Address register is a register that can be accessed during read and write.

The following program loop will also test the core address register.

```
TRA OPR      150 002   % Read the panel switches
IOX LCA      165 541   % and transfer the content to core
                       address register
SAA 0        170 400   % Reset A register
IOX RCA      165 540   % Read core address register
COPY SA DX   146 157   % Copy the value to X register
JMP * – 5    124 373   % Repeat loop
```

While running the loop, a comparison can be made between the switch setting and the result in the X register.

The following test loop will also test the block address register I.

```
SAA 10       170 410   % Set bit 3 in A register
IOX CWR      165 545   % Set controller in test mode
TRA OPR      150 002   % Read panel switches and
                       transfer the contents
IOX BARI     165 543   % to block address register I
SAA 0        170 400   % Reset A register
IOX RBARI    165 546   % Read block address register
COPY SA DX   146 157   % and copy the contents to X
                       register
JMP * – 7    124 371   % Repeat loop
```

Block address register II and the data channel will also be tested by the following test loop:

```
SAA 10       170 410   % Set bit 3 in A register
BSET ONE 170 DA 174 375 % Set bit 15 in A register
IOX CWR      165 545   % Set controller in test mode and
                       select register bank I
TRA OPR      150 002   % Read panel switches and
IOX BAR II   165 343   % transfer the contents to block
                       addr. reg. II
SAA 0        170 400   % Reset A register
IOX RBARII   165 546   % Read block addr. reg. II and
                       copy the contents to
COPY SA DX   146 157   % the X register
JMP * – 7    124 371   % Repeat loop
```

8.3 *CHECKING THE OPERATION OF THE DISK CONTROLLER (Test Mode)*

In test mode the basic parts of the disk controller operate in the same way during a normal disk transfer but the disk controller is independent of the disk unit.

Test mode is entered by specifying bit 3 in the control word (CWR 3).

The block address register I must be set to 125252 and block register II must be set to 1252 prior to a transfer in test mode.

Note: Parity error will always occur during operations in test mode.

When reading in test mode a number of words are transferred from a test data pattern generator to memory. The number of words to be transferred is specified by loading the word counter register.

The memory start address is given by loading the core address register.

After reading in test mode (control word register = 000 014) the correct status should be 041 030 where the active bits mean:

- ON cylinder
- data error
- inclusive or of errors
- operation finished

The contents of the memory buffer after transfer is complete should be:

First location:	125252
Second location:	052525
Third location:	125252
etc.	

The easiest way of checking data transfer from memory to the disk controller is by use of compare mode while the controller is in test mode.

The data buffer which was built up during read a read in test mode, will be used as output data under compare mode. The control word register is set to 014 014, to execute a compare test in test mode.

The block address registers should be set as for a normal read in test mode, the result in the status register should also be the same.

8.3.1 *ECC Test Loop*

The following loop will read data in test mode and store the data in memory, starting at address CA. The number of words to be transferred is given by WC. The loop also checks for proper operation of the core address register. If the operation is incorrect, the loop will be stopped by a wait instruction. The A register will hold the number of words not transferred.

```
Start,  LDA BSEL        044032   % Select register
        IOX CWR         165545   % bank 1.
        LDA BAR II      044031   % Load block
        IOX BAR II      165543   % address register II.
        SAA 0           170400   % Select register
        IOX CWR         165545   % bank 0.
        LDA BARI        044026   % Load block
        IOX BARI        165543   % address register I.
        LDA LCA         044025   % Load core address
        IOX LCA         165541   % register with start address
        LDA WC          044024   % Load word count register
        IOX WC          165547   % with no. of words to be
                                   transferred
        LDA CWR         044023   % Load control word
        IOX CWR         165545   % and start transfer
        SAX – 100       171700   % Delay,
        JNC * 0         132400   % (increment X and jump if
                                   negative)
        IOX STS         165544   % Read status
        BSKP ONE 30DA   175235   % Check if transfer is
                                   finished
        JMP * – 4       124374   % If no, loop again
        COPY SA DD      146151   % Copy status to D register
        IOX RCA         165540   % Read core address
                                   register
        SUB LCA         064010   % Subtract initial value of
                                   core address register
        SUB WC          064010   % Subtract no. of words to
                                   be transferred
        JAZ * 2         131002   % Jump if A register is 0.
                                   (normal condition)
        WAIT            151000   % Stop here if check is not
                                   OK
        JMP START       124347   % Do the program over
                                   again
        BSEL,           100000   % Block select constant
        BARII,          001252   % Block address register II
        BARI,           125252   % Block address register I
        LCA,            001000   % Core address register
        WC,             001000   % Word count register
        CRW,            000014   % Control word register
```

8.4 *CONNECTING A DISK DRIVE*

When the controller runs without problems in test mode, a unit can be connected. The initial start up procedure is given in the maintenance manual following the disk. The disk pack to be used must be formatted which should be done on another machine.

The test programs to be used to check out the complete disk system is described in Appendix K.

9 LOGIC BOARDS — SHORT DESCRIPTION

9.1 *1013 — DEVICE REGISTERS (POS 32)*

This module contains:

- status registers, bits 4 - 15 (6C, 6B)
- drivers for reading status, bits 4 - 15 (8C, 8B)
- block address register (14A, 14B, 12C)
- decoding of device operation, M0 - M15 (2A, 2C, 4A, 4B, 4C)
- decoder for device registers (16A)
- drivers for reading block address register (12A, 10B, 16C)

9.2 *1134 — ECC CONTROL (POS 31)*

This module contains:

- Block address register, the cylinder portion of the address (16A, 14C, 13B)
- Block address serialization circuit for address compare when address field of the sector is read (8B, 9A, 10B, 9C)
- IOX Instruction decodes (3A, 1B, 5A, etc.)
- Parts of ECC control register, bits 14, 15 (19A)
- Parts of ECC pattern register, bits 11, 12, 13, 14 (19B)
- Drivers for reading the cylinder portion of the block address register (19A, 12C, 11A)

9.3 *1092 — BUFFERRED DMA (POS 30)*

Note 1:

This module is downwards compatable with the 1014 module. The difference i a 64 word FIFO (buffer) installed in the 1092 versus a 1 word buffer in the 1014. This enables the total data throughput in the memory and/or I/O system to exceed the upper limit for short periods.

Note 2:

Refer also to the manual "NORD-10/S Input/Output System" (ND-06.012), Section 7.3 for a description of this module.

This module contains:

- Input data selector for FIFO (12D, 14D, 16D, 18D)
- 16 words data buffer (FIFO) (12E, 14E, 16E, 18E)
- Shift register input selector (4A, 6A, 8A, 10A)
- Shift register (serial to paralle, parallel to serial data conversion) (4B, 6B, 8B, 10B)
- Data bus driver (12A, 14A, 16A, 18A)
- Word counter (keeps track of the number of words to/from the disk) (12B, 14B, 16B, 18B)
- Request counter (keeps track of the number of words to/from memory) (12C, 14C, 16C, 18C)
- Generation of status bit 11, DMA channel error, overrun/underrun (2D)
- Control circuits

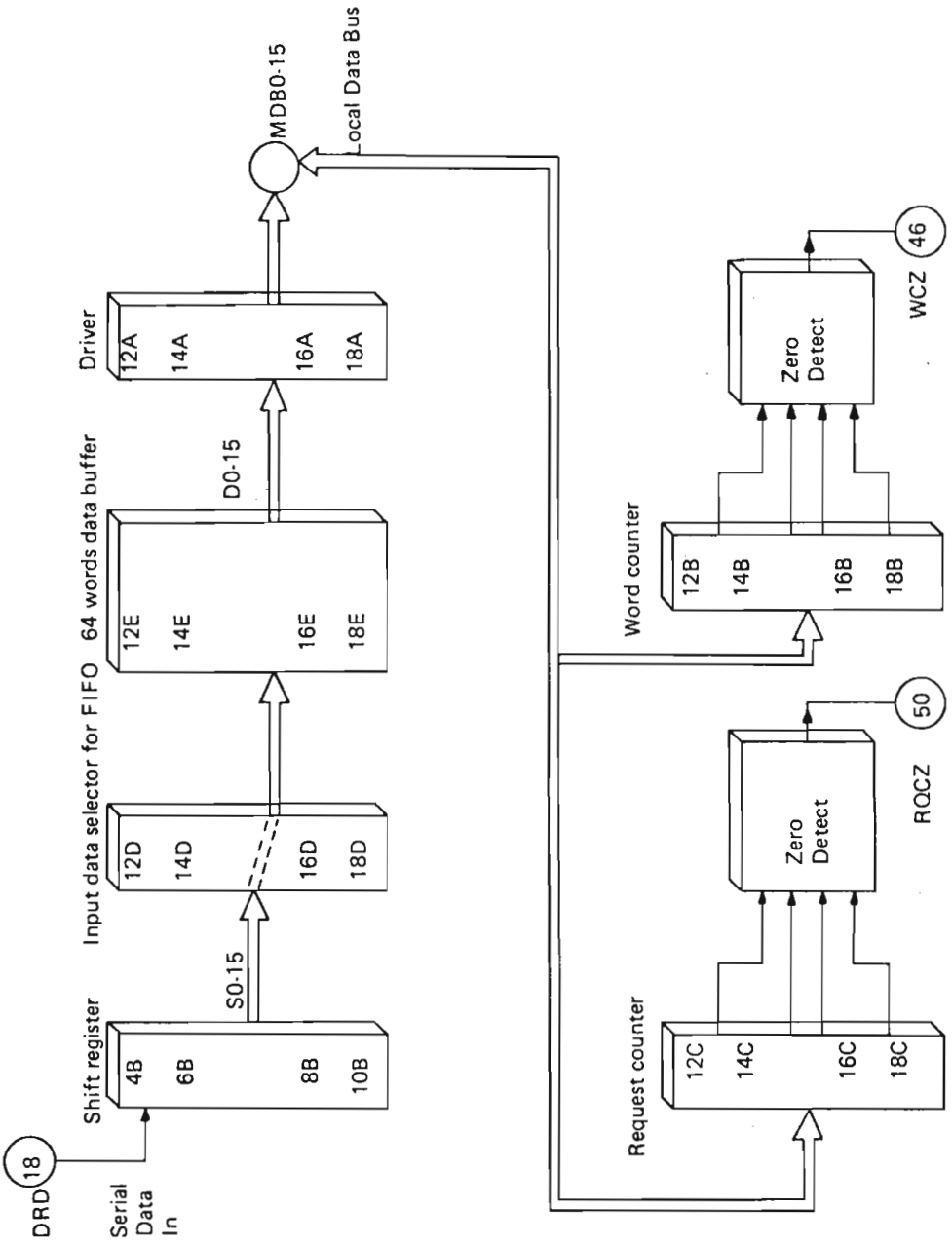


Figure 9.1: 1092 — Read Data Block Diagram

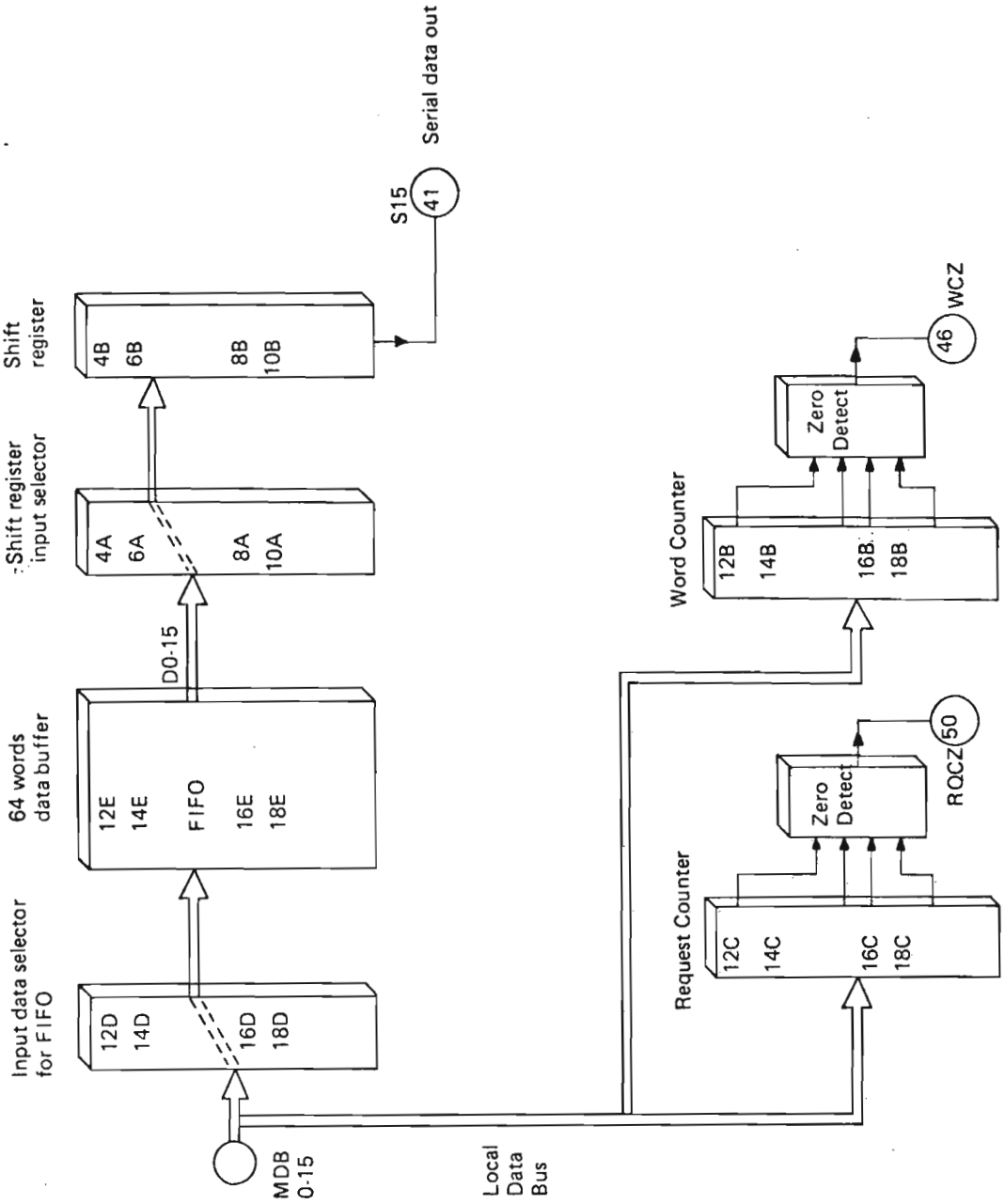


Figure 9.2: 1092 — Write Data Block Diagram

9.4 *1133 — ECC POLYNOMIALS (POS 29)*

This module contains:

- ECC polynomial (14E, 14D, 14B, 12B, 12D, 12E, 9E, 9D, 9B, 7B, 7D, 7E, 5C, 5D)
- Error displacement counter (19F, 19B, 7A, SA)
- Polynomial parity check (14F, 16B, 12F, 9F, 5B, 7F, 5F, 16F, 1C)
- Polynomial zero's detector (14, 12C, 9C, 7C, 5E)
- Bus interface buffers (9A, 19A, 16A, 14A, 12A)

9.5 *1135 — SMD TIMING (POS 28)*

This module contains:

- Phase bit counter (4D, 1D, 1E, 4E)
- Phase generator (1C, 4C, 4A, 17E)
- Miscellaneous circuits for counter decodes and controls (10C, 16C, etc.)

Note: This module is a substitute for the old 1076 module, but is NOT compatible. Circuits (13A, 16A, 19A) are used to generate inputs for the 1133 module.

9.6 *1077 — SMD CONTROL (POS 27)*

This module contains:

— Circuits for generating the following control signals used on the 1092 board.

SHTE — shift enable
PL — parallel load
DCS — data channel strobe
IRQ — initiate request
WRITE CORE — read disk
DWC — decrement word counter
SL — shift clock

(19C, 11C, 15C, 2D, etc.)

— Synchronous character detection (end of PH1 or PH4 during read) (4B)

— Generation of normal end of operation, COMPL (17A)

— Read Gate (13C)

— Write Gate (6B)

— Time out detection (generation of status bit 6) (8C)

— Read clock enable (17B)

— Format switch and format switch on indicator (edge of board)

— Detection of missing read clock (SB7) (2D)

— Detection illegal register load (SB5) (15B, 6D)

9.7 *1078 — SMD RECEIVE (POS 26)*

This module contains:

— Block address (PH2) compare network (5B, 5C, 8C, 5D)

— Cable A receivers (15A, 15B)

— Cable A transmitters (18A, 18B, 18C)

— OR for errors (generation of status bit 4) (15C)

— Selector for clock, data and sector (test mode) (5A)

— Test mode clock oscillator (8D)

— Test mode sector oscillator (2B, 2D)

— Driver for Read Seek Condition (12A, 12B)

— Control word bits 7 - 10 (unit selection and marginal recovery) (12C)

9.8 *1154 — SMD TRANSMIT (POS 25)*

Note 1:

This module incorporates the identical functions of the 1079 module, but with the additional capability of selecting more heads to the disk drive.

Note 2:

This module is downwards compatible with the 1079 module, i.e., this module can be used in the old controllers.

Note 3:

The block address serialization circuit (8B and 11B) is not used by the ECC controller).

Note 4:

The block address inputs (DA X and AX) represent different address bits for old and ECC controller due to different formats.

This module contains:

- Block address serialization circuit (8A, 11B) (not used by the ECC controller)
- Tag timing generator (4B, 5E, 2B, 8C, 2C, 2D, 4D, etc.)
- Bus transmitters (17D, 17B, 17C, 15D, 15C)
- Control transmitters (15B, 13B)
- Marginal recovery circuit (6D, 8D, etc.)
- Internal bus drivers (13C, 11C, 11D)
- Additional head selection (15E)

9.9 *1156 — UNIT CONTROL (POS 24)*

Note 1:

This module is downwards compatible with the 1080 module, i.e., this module can be used in the old controllers.

The logical function of 1156 is the same as the function of the 1080 module with exception of the sector count and the compare circuits.

For 18 sector operation (the ECC controller) the sector counter (10C) is extended by one bit (15A) which makes the circuit count module 18 rather than 16. The extra bit compare is also fed into the sector compare circuit (10B).

When the 1156 is used as a replacement of the 1080 in the old controller, the counter and compare circuits are working on module 16.

For the ECC controller the signals $CARRY_0$ and the $18S_0$ are connected via the backwiring, for the old controller these signals are not connected.

This module contains:

- B cable receivers (18C, 16B, 16D, 18D)
- Write clock transmitter (18C)
- Sector counter (10C, 15A)
- Sector compare (10B, 9A, 4D)
- Sector part of block address register (8B)
- Unit compare (8C)
- Unit decoder (4B)
- Detection of missing servo clock (14C)

9.10 *1155 — BUS CONTROL*

Note 1:

This module is downwards compatible with the 1022 module, i.e., 1155 can be used as a replacement for the 1022 in the old controllers.

The logical function of the 1155 module is the same as the 1022. In addition, the 1155 has circuits (15B and 11A). These circuits are added due to the redefinition of the IOX instructions and the separation into two distinct banks of instruction selected by bit 15 of the control word.

If bit 15 of the control word is zero, circuit (8A) is activated to present status information when decoded.

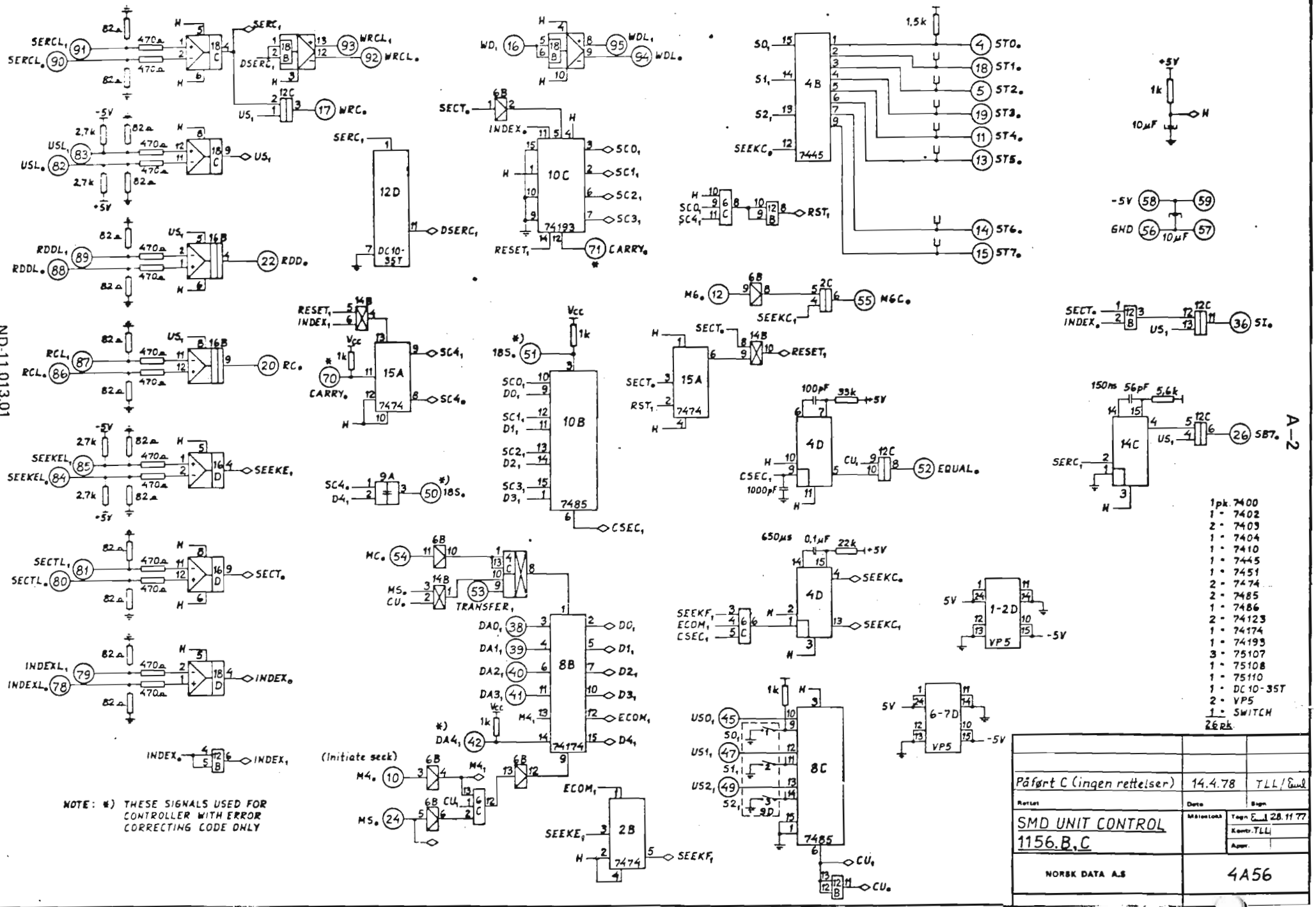
When bit 15 of the control word is one, circuit (8A) is never activated. Refer to the Programming Specifications, Appendix N.

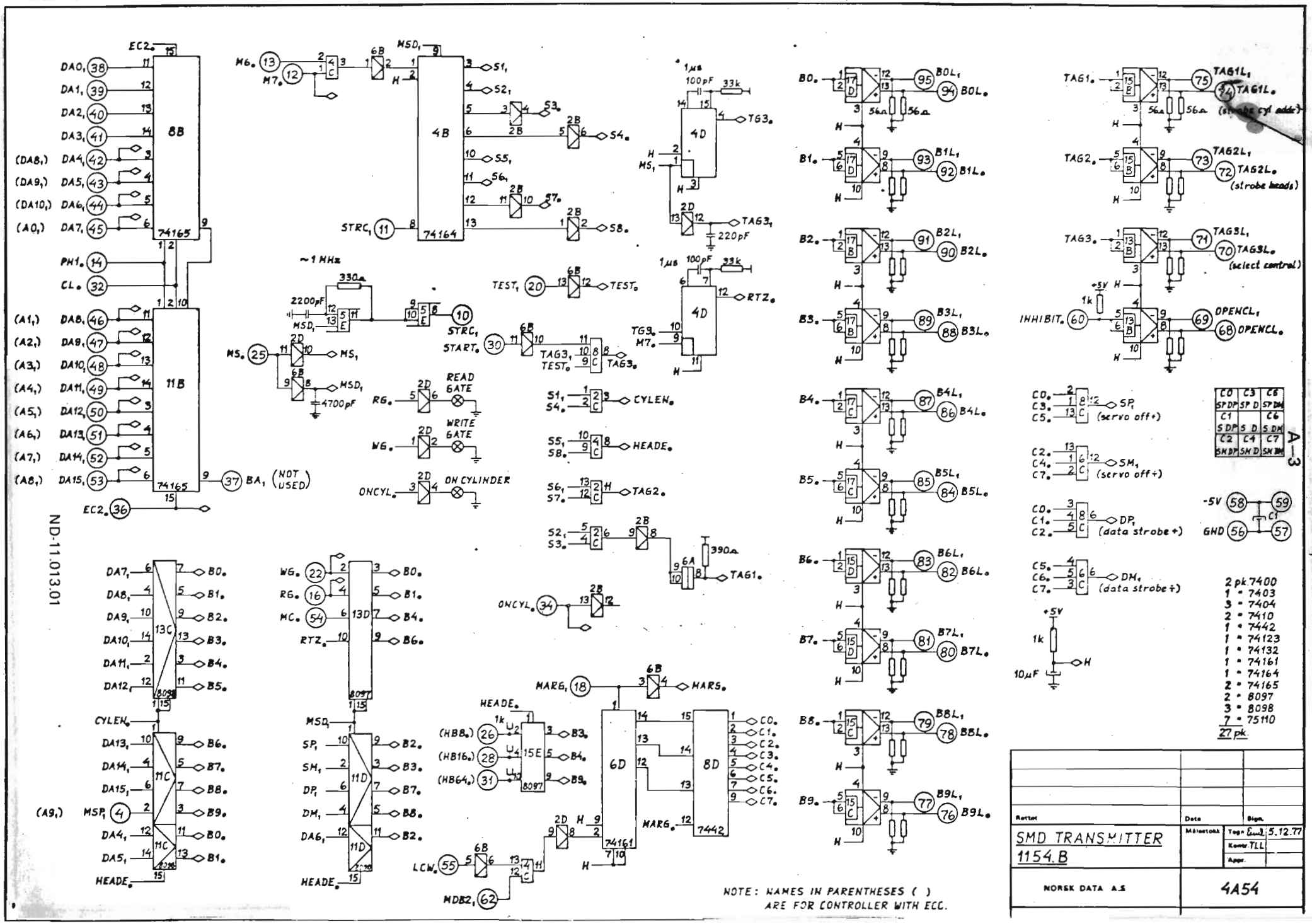
This module contains:

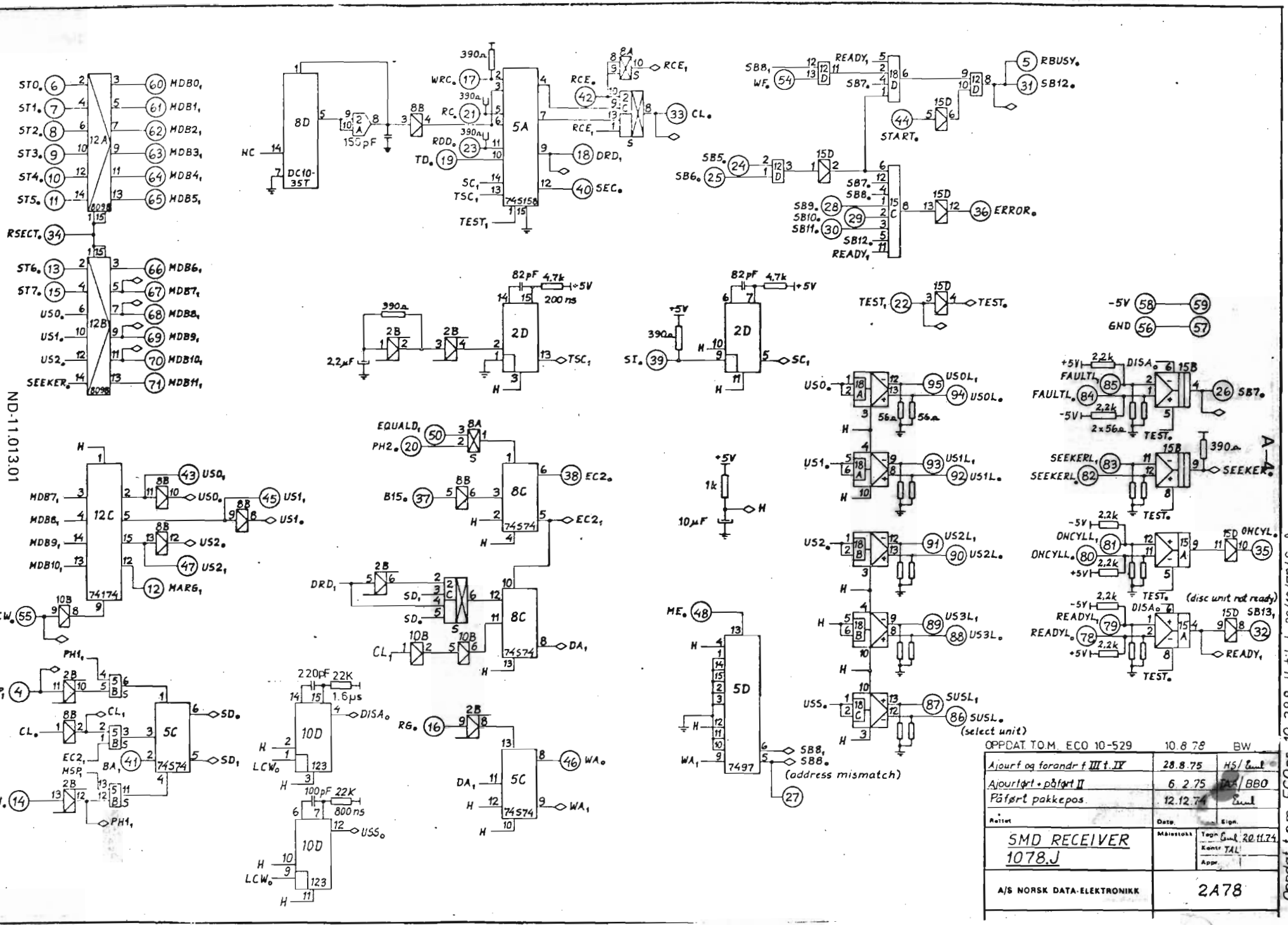
- Device select switches (11B)
- Core address select switches (7C)
- Ident code select switches (19A, 7C)
- Device equals compare circuit (7B)
- Register decoder (19B)
- Drivers for I/O data bus (BD 0-15) onto local data bus (MDB 0-15) (6A, 3B, 15C)
- Drivers for local data bus (MDB 0-15) onto I/O data bus (BD 0-15) (3A, 5B, 17C)
- Ident mechanism (1C)
- Grant mechanism (1B)
- Bit 16 and 17 of the core address register (17B)
- Driver circuit for lower 5 bits of status word (8A)
- Driver circuit for ident code (13A, 13B)
- Driver circuit for core address register selection (10C)
- Lower 4 bits of status and lower 3 bits of control word (13D, 11D)

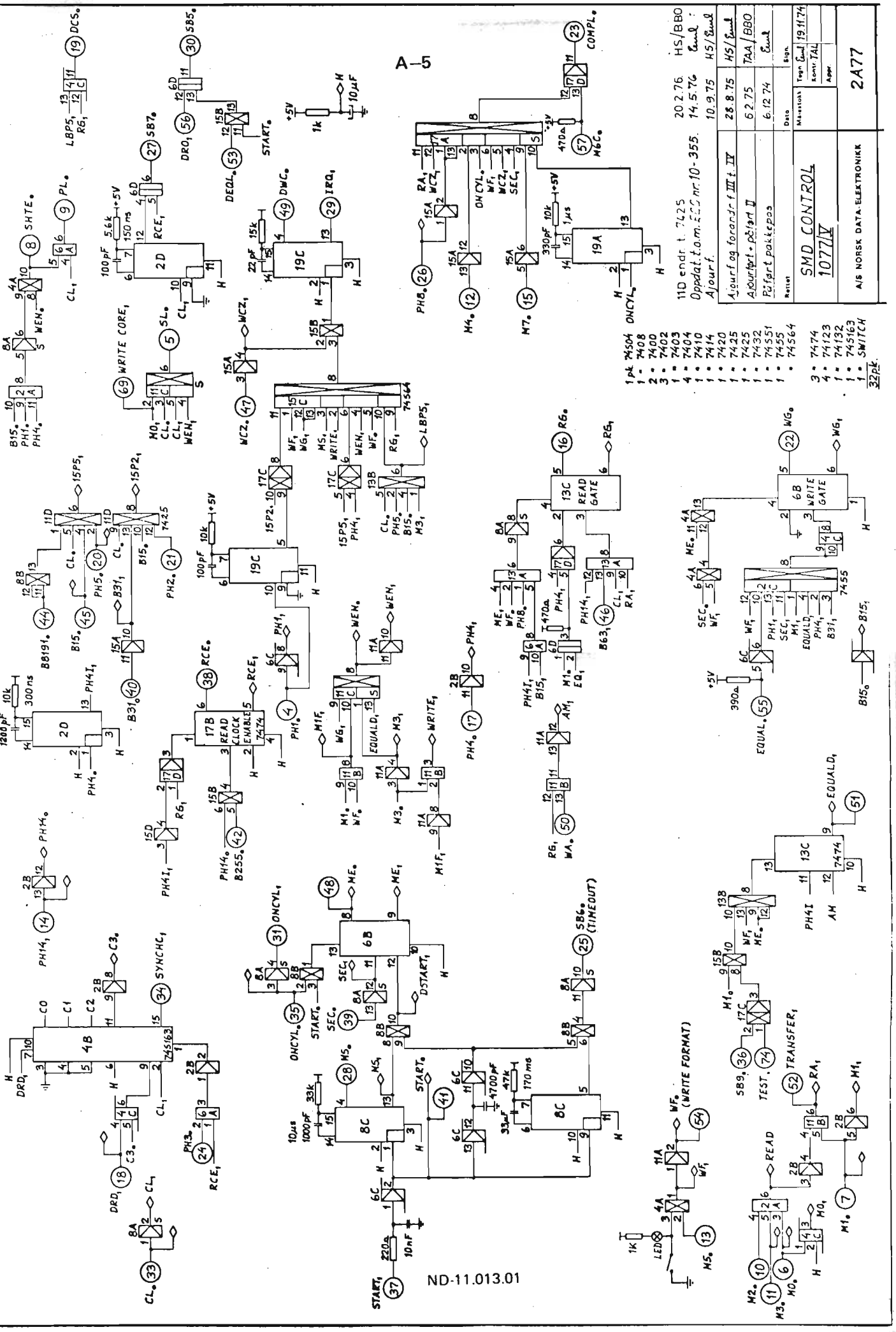
APPENDIX A

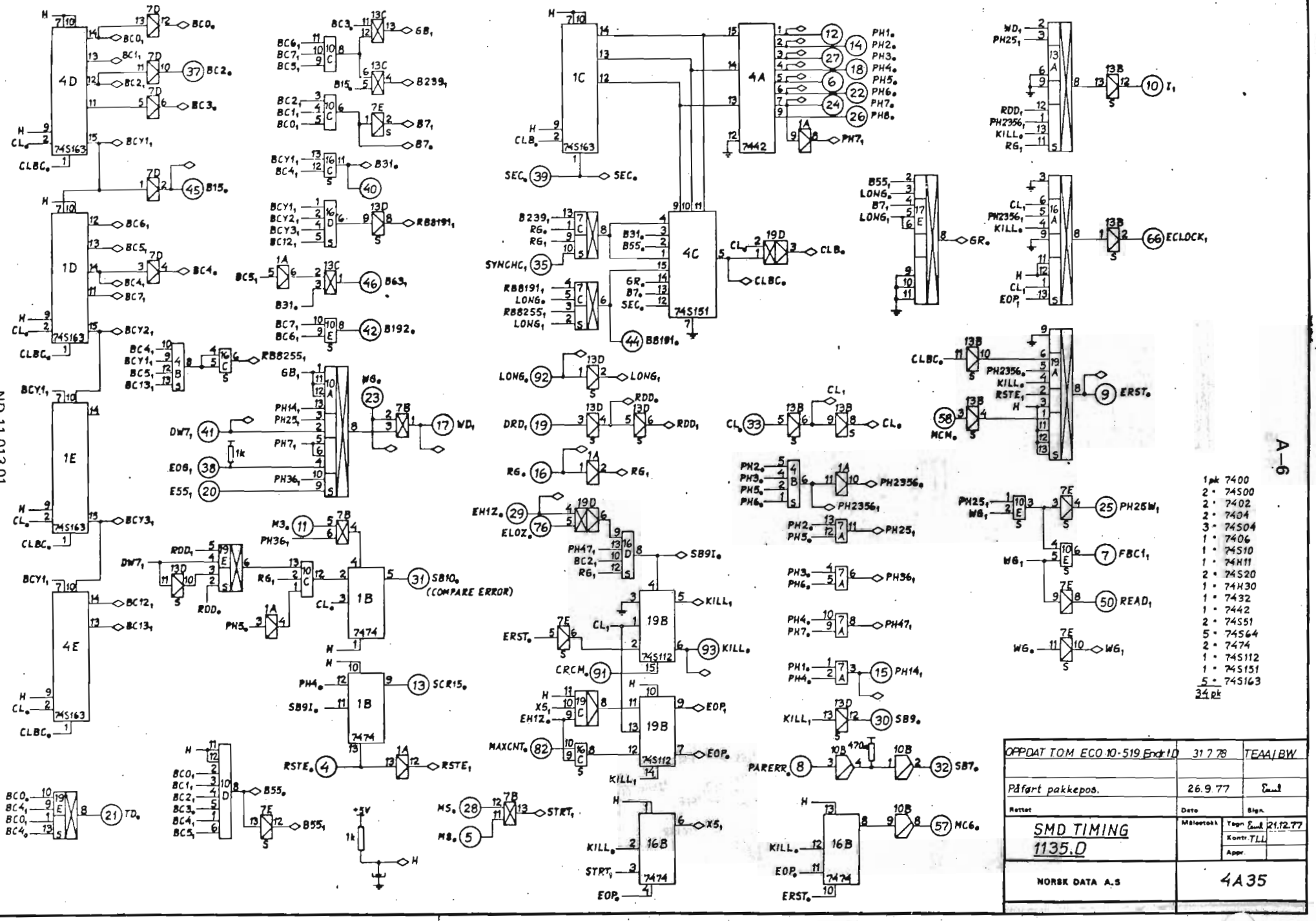
LOGIC DIAGRAMS









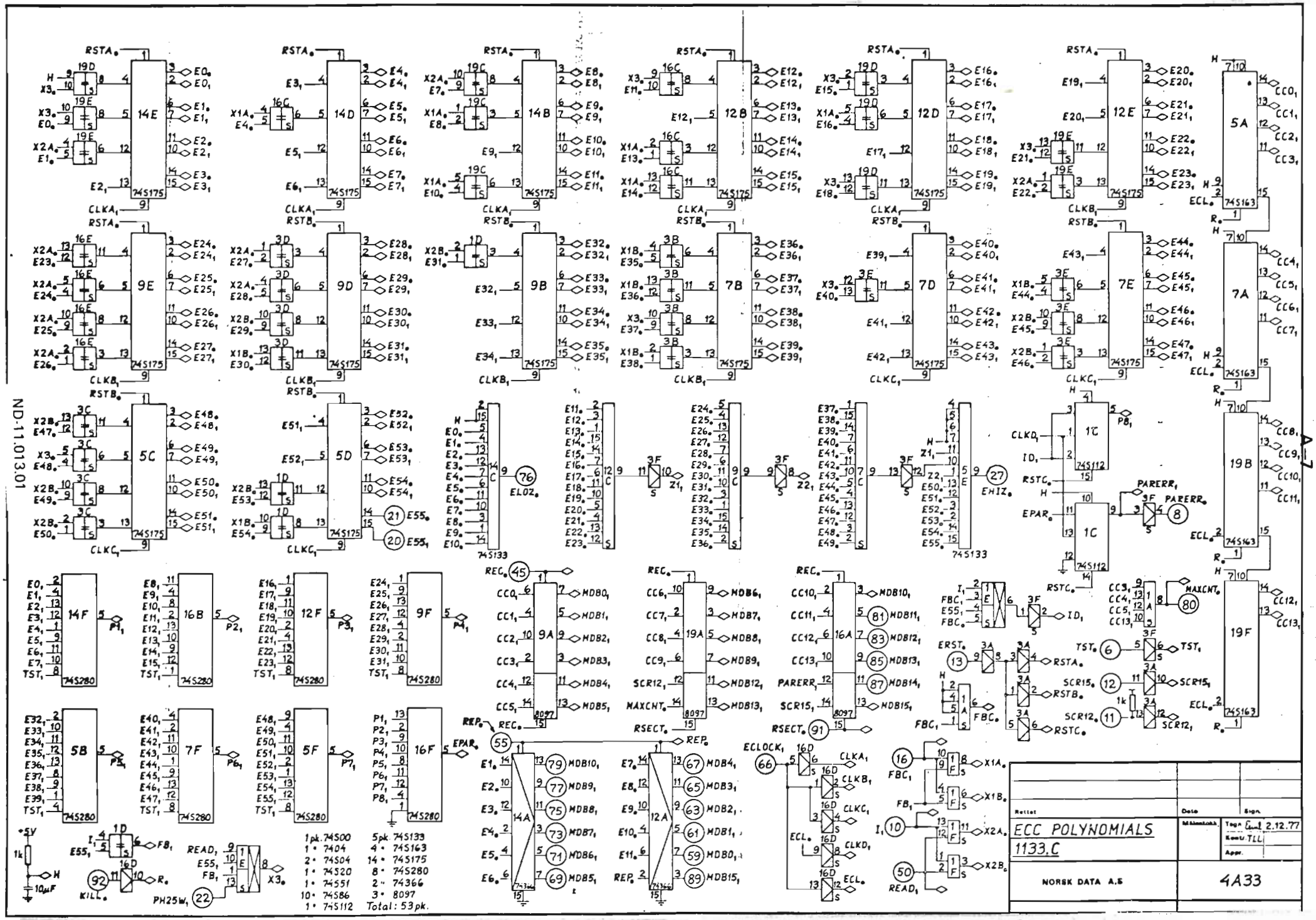


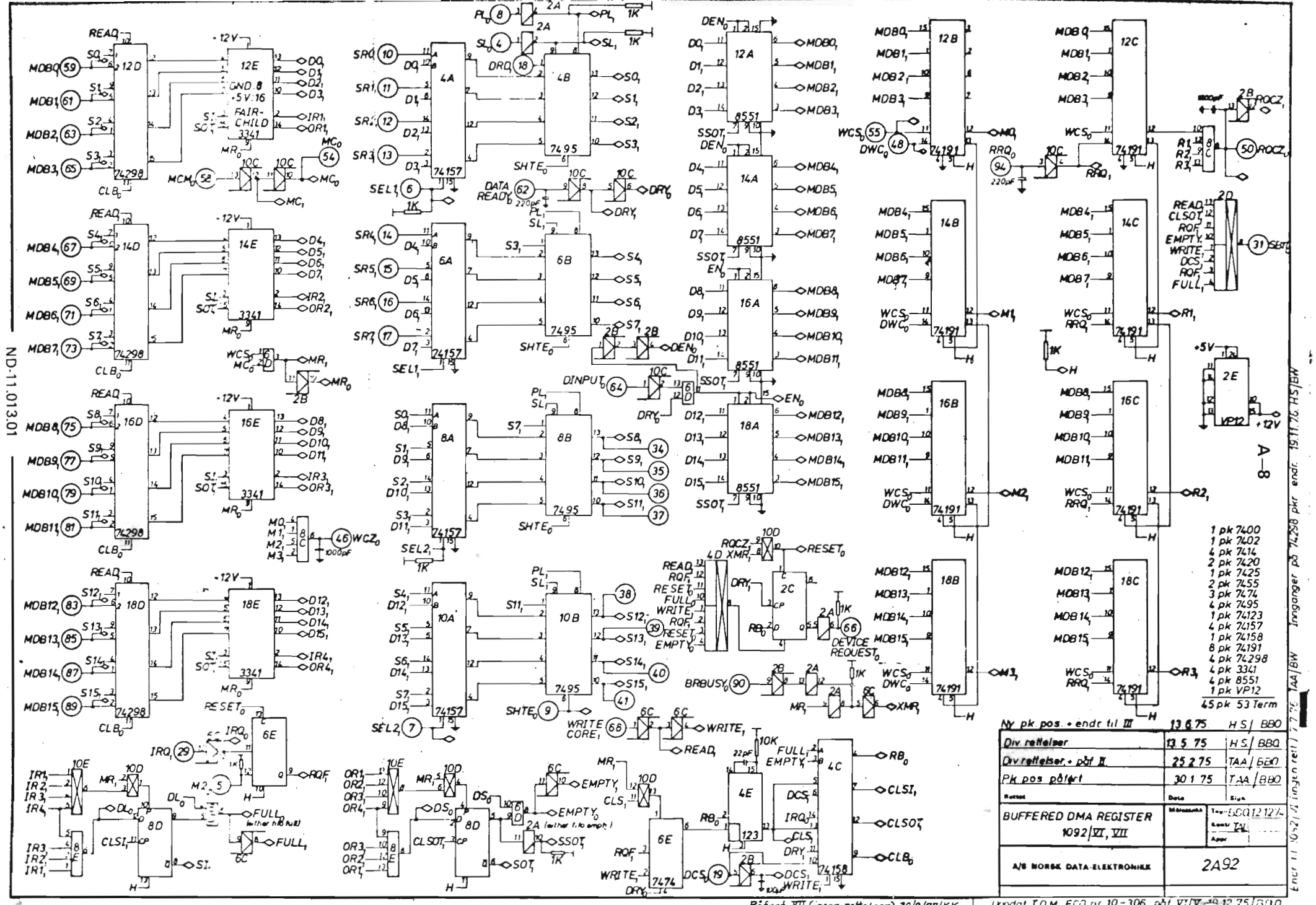
OPPDAT TOM ECO 10-519 Endr! D	31 7 78	TEAA/BW
Påført pakkepos.	26.9 77	Sign
Rettet	Dato	Sign.
SMD TIMING 1135.D	Målestokk	Tegn. 21.12.77 Kontr. TLL Appr.
NORSK DATA A.S	4A35	

1 pk 7400
2 • 74S00
2 • 7402
2 • 7404
3 • 74S04
1 • 7406
1 • 74S10
1 • 74H11
2 • 74S20
1 • 74H30
1 • 7432
1 • 7442
2 • 74S51
5 • 74S64
2 • 7474
1 • 74S112
1 • 74S151
5 • 74S163
34 pk

A-6

ND-11.013.01


ECC POLYNOMIALS
1133.C
NORSK DATA A.S
4A33
ND-11.013.01
A-7
Scanned by Jonny Oddene for Sintran Data © 2010



1 pk 7400
1 pk 7402
4 pk 7414
2 pk 7420
1 pk 7425
2 pk 7455
3 pk 7474
4 pk 7495
1 pk 74123
4 pk 74157
1 pk 74158
8 pk 74191
4 pk 74298
4 pk 3341
4 pk 8551
1 pk VP12
45 pk 53 Term

Ny pk pos. + endr til III	13 6 75	HS / BBO
Div rettelser	13 5 75	HS / BBO
Div rettelser + påf II	25 2 75	TAA / BBO
Pk pos påført	30 1 75	TAA / BBO
Rettet	Dato	Sign
BUFFERED DMA REGISTER 1092/VI, VIII	Målestokk	Tegnet EGO 12 12 74 Kontr TAA Appr
A/S NORSK DATA-ELEKTRONIKK	2A92	

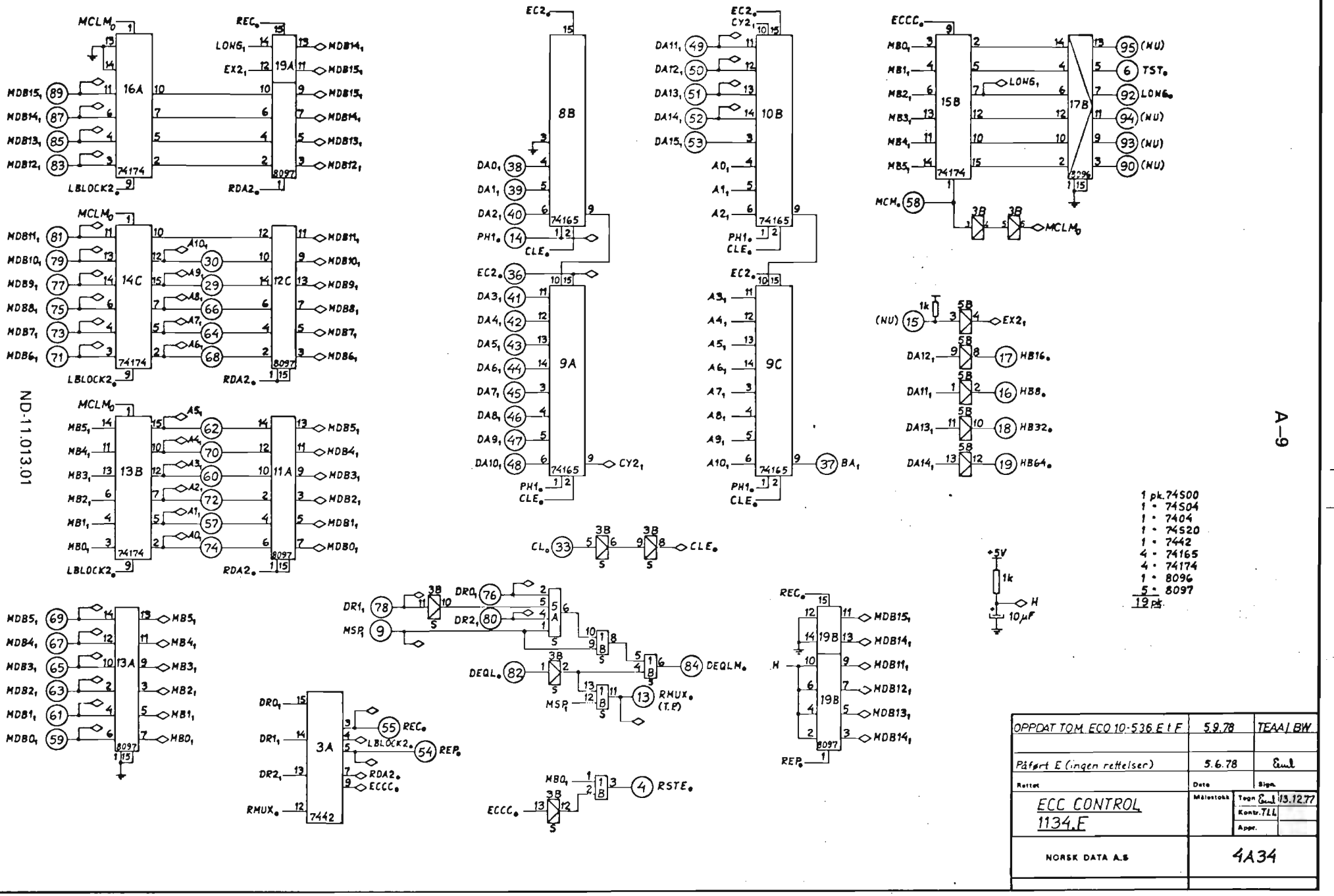
Påført III (ingen rettelser) 30/3/77/KK

Oppdat TOM ECO nr 10-306 påf VI/V 20 12 75/BBO
Oppdat TOM ECO nr 10-355 okt 1

Endr 11.10.72/AI ingen rettl 7.7.76 TAA/BW Innganger på 74298 pkr endr. 19.11.76. HS/BW

ND-11.013.01

A-8

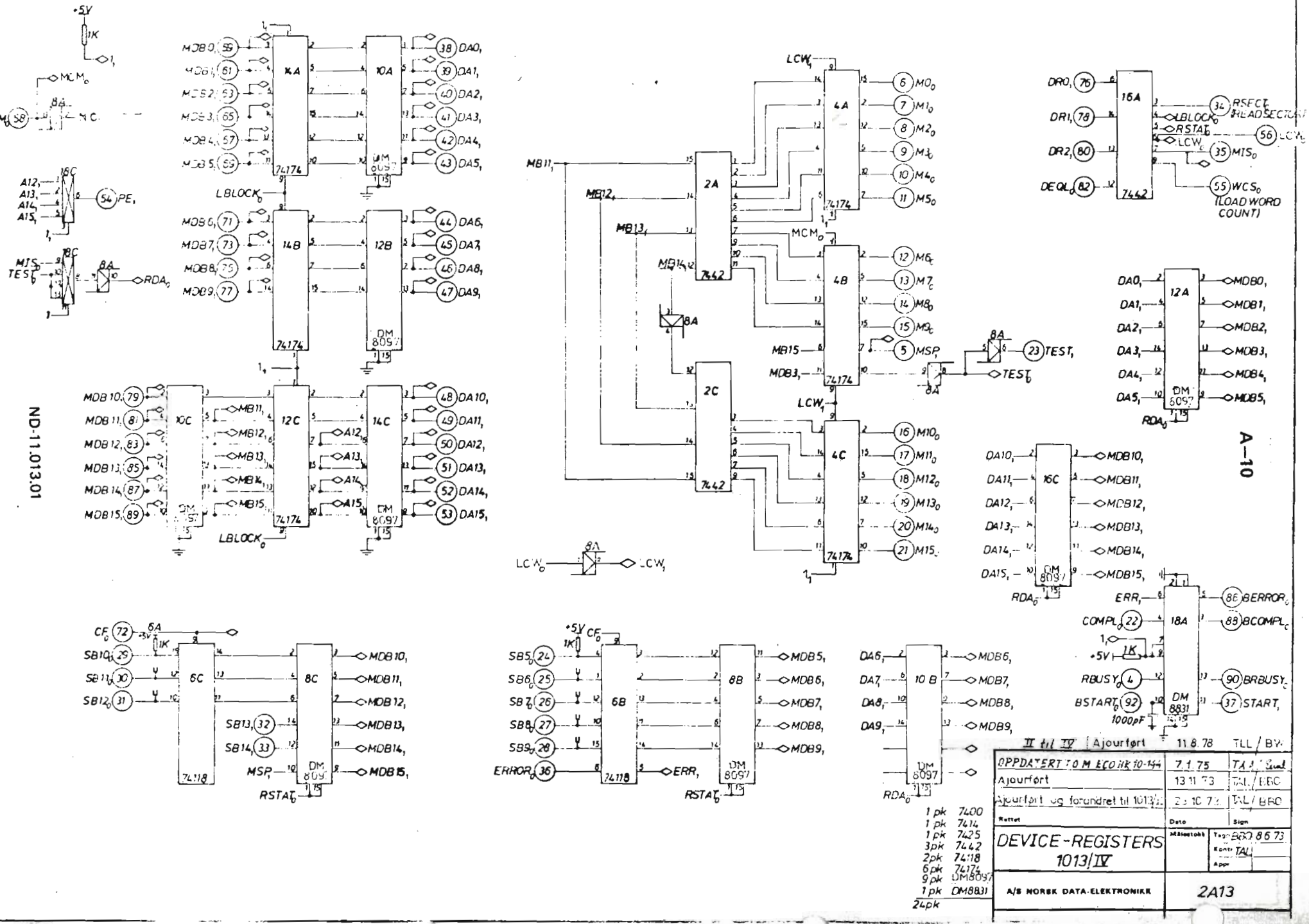


1 pk. 74S00
1 • 74S04
1 • 7404
1 • 74S20
1 • 7442
4 • 74165
4 • 74174
1 • 8096
5 • 8097
19 pk.

OPPDAT TOM ECO 10-536 E t F	5.9.78	TEAA/BW
Påført E (ingen rettelser)	5.6.78	Eml
Rettet	Date	Sign.
ECC CONTROL 1134.E	Målestokk	Tegn Eml 13.12.77 / Kontr. TLL / Appr.
NORSK DATA A.S	4A34	

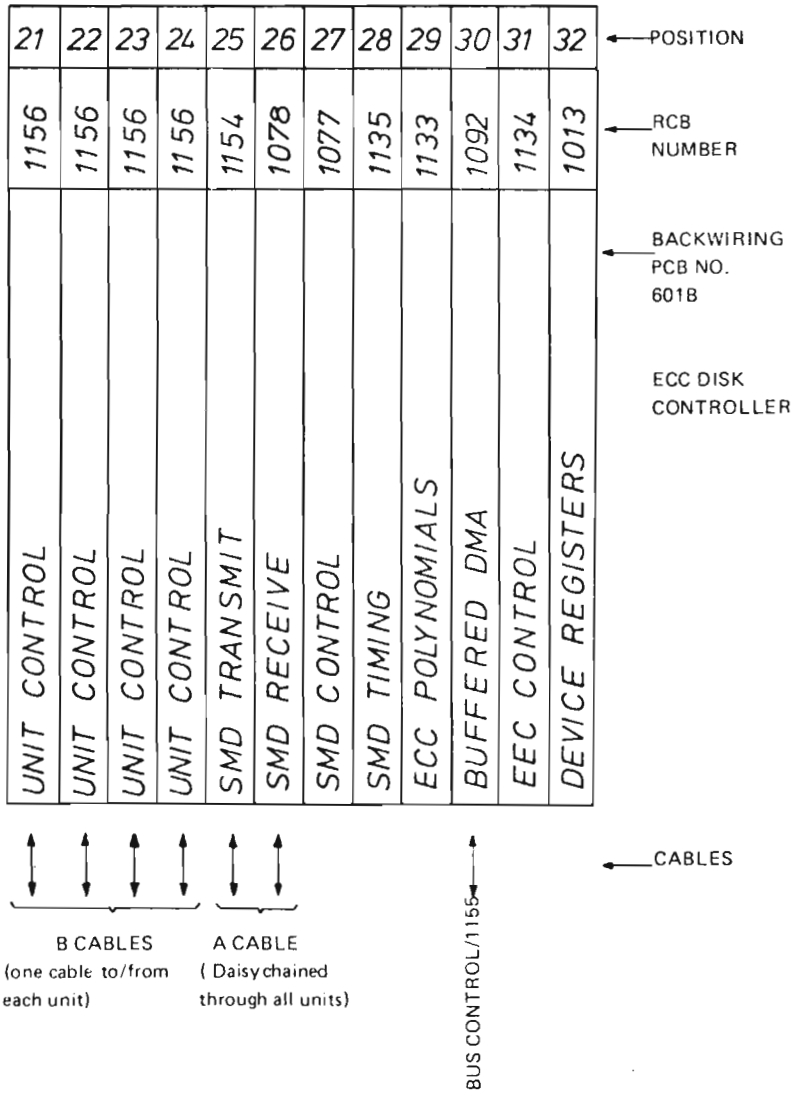
ND-11.013.01

A-9



APPENDIX B

CONTROLLER PCB LAYOUT



APPENDIX C

SIGNAL DEFINITION LIST

Signal	*Generated* POS	CARD	*Signal Explanation List:*
A0-10	B31	1134	Block address register II (cylinder portion of block address)
BA_1	B31	1134	Serial block address to be compared with the address (PH2) read from the disk
$BCOMPL_0$	B32	1013	Buffered COMPL signal
$BC2_0$	B28	1135	Phase bit counter equals 4
$BERROR_0$	B32	1013	Buffered ERROR signal
$BRBUSY_0$	B32	1013	Buffered RBUSY signal
$BSTART_0$	—	1155	Activate controller (CWR bit no. 2)
$BL0\text{-}9_{0,1}$	B25	1154	Bus lines to disk. The bus is multiplexing cylinder address bits, lead select bits and control bits to disk.
$B15_0$	B28	1135	Phase bit counter equals 15
$B31_0$	B28	1135	Phase bit counter equals 31
$B63_1$	B28	1135	Phase bit counter equals 63
$B192_0$	B28	1135	Phase bit counter equals 192
$B8191_0$	B28	1135	Phase bit counter equals 8191.
CF_0	—	1155	Clear error flags. Generated by Master Clear, Device Clear or Activate Device.
CL_0	B26	1078	Serial bit clock. The clock derives from read clock (RC) during read, servo clock (WRC) during write or an intern oscillator in test mode.
$COMPL_0$	B27	1077	Legal completion of an operation (will reset Busy and generate interrupt if enabled)
$CRCM_0$	B31	1134	

Signal	*Generated* POS	CARD	*Signal Explanation List:*
Data Ready$_0$	—	1155	DMA DATA READY to this controller. DATA READY is the termination signal for a one word transfer to/from memory. For Read (Disk to Memory) Memory has completed the store operation. For WRITE (Memory to Disk) one word is present on the information bus from memory.
DA0-15$_1$	B32	1013	Lower 16 bits of block address register
DCS$_0$	B27	1077	Data channel strobe. Strobes the 16 assembled bits (during read) into the first stage of FIFO.
DEQL$_0$	—	1155	The specified device number in an IOX instruction matches with this device number.
DEQLM$_0$	B31	1134	Enable decoding of any register with CWR bit 15 = 0 or select Control Word Register (CWR) when CWR bit 15 = 1. (Refer to programming specifications.)
DEVICE REQ$_0$	B30	1092	Request to memory for a one word read or write transfer
DINPUT$_0$	—	1155	Enables data from output stage of FIFO onto local data bus during a read transfer.
DRD$_1$	B26	1078	Serial data from disk during normal read or from test data pattern generator during read in test mode.
DR0-2$_1$	—	1155	Register select bits during an IOX instruction (3 lower bits of address bus)
DWC$_0$	B27	1077	Decrement word counter pulse. Generated for each word to be transferred to/from disk.
DW7$_1$ (SI5$_1$)	B30	1092	Serial data from shift register during write (also referred to as SI5)
ECLOCK$_1$	B28	1135	Polynomal shift clock
EC2$_0$	B26	1078	Enable compare of Block Address (PH2) from disk with block address shift register
EHIZ$_0$	B29	1133	Upper (high) 45 bits of polynomial equal to zero
ELOZ$_0$	B29	1133	Lower (low) 11 bits of polynomial equal to zero.

Signal	*Generated* POS	CARD	*Signal Explanation List:*
EOG_1			End of Gap (no source)
$EQUAL_0$	B21-24	1156	Sector match. I.e., lower part of block address (sector number) is equal to sector counter for the selected unit.
$EQUALD_1$	B27	1077	Block address (PH2) read from disk compares with block address shift register
$ERROR_0$	B26	1078	Inclusive or of error indicators. Same as STS bit number 4.
$ERST_0$	B28	1135	Reset polynomal
$E55_1$	B29	1133	Serial output from polynomial
$E55_0$	B29	1133	Serial output from polynomial
$FAULT_{0,1}$	Selected	Unit	Disk fault from selected unit
FBC_1	B28	1135	Feedback select signal for polynomial generation
$HB8_0$	B31	1134	Head select line (value = 8) (from block address bit number 11)
$HB16_0$	B31	1134	Head select line (value = 16) (from block address bit number 12)
$HB32_0$	B31	1134	Not used. Head select line. (value = 32) (from block address bit number 13)
$HB64_0$	B31	1134	Option for CMD, MMD disk units
$INDEX_{0,1}$	Wired	Unit	Start of revolution from the associated unit
IRQ_1	B27	1077	Initiate request. The controller request a new memory access (generate DEVICE REQUEST).
I_1	B28	1135	Input data to polynomial
$KILL_0$	B28	1135	Prevent clearing of polynomial when data error is discovered
LOW_0	B32	1013	Load control word register strobe
$LONG_0$	B31	1134	Error correction control bit number 2. Extends PH5 during read or write to be equal to PH5 and PH6 and PH7. Used for maintenance purposes only.
$MARG_1$	B26	1078	Marginal recovery. Control word bit number 10.

Signal	*Generated* POS	CARD	*Signal Explanation List:*
$MAXCNT_0$	B29	1133	Error displacement counter equal to 8248 (8192 (PH5) + 56 (PH6)). Used when testing whether data error is recoverable or not.
MC_0	B30	1092	Buffered MCM
MCM_0	—	1155	Master clear from CPU or device clear (control word bit number 4)
$MDB0\text{-}15_1$	—	1155	Internal data bus in the disk controller
ME_0	B27	1077	Master enable. The control is active and on cylinder.
MIS_0	B32	1013	Read block address strobe. In test mode block address register is returned to the A register in the CPU.
MS_0	B27	1077	Master start pulse. A 10μs pulse generated when controller goes active.
MSP_1	B32	1013	Control word bit number 15. Selects one of two "banks" of registers accessed by IOX instructions (refer to program specifications)
$M0_0$	B32	1013	Operation mode 0. Read transfer.
$M1_0$	B32	1013	Operation mode 1. Write transfer.
$M2_0$	B32	1013	Operation mode 2. Read parity transfer.
$M3_0$	B32	1013	Operation mode 3. Compare transfer.
$M4_0$	B32	1013	Operation mode 4. Initiate seek.
$M5_0$	B32	1013	Operation mode 5. Write format.
$M6_0$	B32	1013	Operation mode 6. Seek complete search.
$M7_0$	B32	1013	Operation mode 7. Return to zero seek.
$M8_0$	B32	1013	Operation mode 8. Run ECC operation.
$M6C_0$	B28	1135	ECC operation completed
	B21-24	1156	Seek complete search positive.
$ONCYLL_{0,1}$	Seleted	Unit	Unit has completed a seek operation and a read/write operation can be started.
$PARERR_0$	B29	1133	Hardware fault condition exists in ECC polynomial circuits. Reported in status bit number 7 and read seek condition bit number 14. (refer to program specifications).

Signal	*Generated* POS	CARD	*Signal Explanation List:*
PE_1	B32	1013	Not used
$PH1_0$	B28	1135	Sector phase 1. Refer to sector format.
$PH2_0$	B28	1135	Sector phase 2. Refer to sector format.
$PH3_0$	B28	1135	Sector phase 3. Refer to sector format.
$PH4_0$	B28	1135	Sector phase 4. Refer to sector format.
$PH5_0$	B28	1135	Sector phase 5. Refer to sector format.
$PH6_0$	B28	1135	Sector phase 6. Refer to sector format.
$PH7_0$	B28	1135	Sector phase 7. Refer to sector format.
$PH8_0$	B28	1135	Sector phase 8. Refer to sector format.
$PH14_1$	B28	1135	Sector phase 1 or 4 Refer to sector format.
$PH25W_1$	B28	1135	Write in phase 2 or 5.
PL_0	B27	1077	Parallel load of shift register with data from memory during a write operation.
$RBUSY_0$	B26	1078	Abnormal end of operation, i.e., error reset of busy FF. Same as status register bit number 12.
RC_0	B21-24	1156	Buffered RCL.
RCE_0	B27	1077	Read clock enable. Enable read clock (RCL) from disk to be the master clock. Otherwise, the servo clock is enabled. In test mode a test clock is enabled.
$RCL_{0,1}$	Seleted	Unit	Read clock from disk.
RDD_0	B21-24	1156	Buffered RDDL.
$RDDL_{0,1}$	Seleted	Unit	Read data from disk
$READYL_{0,1}$	Seleted	Unit	Disk unit ready, i.e., selected unit is up to speed, has the heads loaded and no faults exist.
REC_0	B31	1134	Read ECC count. Refer to programming specifications.
REP_0	B31	1134	Read error pattern. Refer to programming specifications.
RG_0	B27	1077	Read gate. Enables the read circuitry in the selected disk unit.

Signal	*Generated* POS	CARD	*Signal Explanation List:*
$RMUX_0$	B31	1134	Enables decoding of the second bank of registers. (Same as DEQL · CWR bit number 15 = 1) Refer to programming specifications.
RRQ_0	—	1155	Reset request and decrement request counter.
$RSECT_0$	B32	1013	Read seek condition. Refer to programming specifications.
$RSTE_0$	B31	1134	Reset ECC polynomial. (Load ECC control bit number 0.) Refer to programming specifications.
$RCL_{0,1}$	Seleted	Disk	Read clock from disk.
$READ_1$	B28	1135	Inverted WG (write gate)
$SB5_0$	B27	1007	Set when loading a register when the controller is active.
$SB5_0$	B25	1154	Load of block address while the unit is not on cylinder.
$SB6_0$	B27	1077	Timeout. An operation is not completed within 170 ms.
$SB7_0$	B21-24	1156	Detection of missing servo clock.
$SB7_0$	B26	1078	Fault from disk unit.
$SB7_0$	B27	1077	Detection of missing read clock.
$SB7_0$	B28	1135	Buffered PAR ERR. I.e., detection of hardware faults in ECC polynomial circuits.
$SB8_0$	B26	1078	Address mismatch. Generated if address match did not occur between block address register and block address read from disk (PH2) within 2 revolutions of the disk)
$SB9_0$	B28	1135	Data error on address or data.
$SB10_0$	B28	1135	Compare error. During compare mode (M3), data from disk did not match with data from memory.
$SB11_0$	B30	1092	DMA channel error. I.e., (1) during write (M1) the FIFO output stage is empty when the shift register requests a new word, or (2) during read (M0) the input stage of the FIFO is full when the shift register has assembled a 16 bits word.

Signal	*Generated* POS	CARD	*Signal Explanation List:*
$SB12_0$	B26	1078	Same as RBUSY. Abnormal end of operation.
$SB13_0$	B26	1078	Disk unit not ready.
SCR12	—	—	Not used.
SCR15	B28	1135	Data error is discovered in address field (PH2).
SEC_0	B26	1078	Sector clock from disk or from test oscillator in test mode.
$SECTL_{0,1}$	Seleted	Disk	Sector clock from disk.
$SEEKEL_{0,1}$	Seleted	Disk	End seek (on cylinder or seek error) from disk.
$SERCL_{0,1}$	Seleted	Disk	Servo (write) clock from disk.
$SHTE_0$	B27	1077	Enable the shift register to shift.
SI_0	B21-24	1156	Sector or index pulses.
$SI0\text{-}7_0$	B21-24	1156	One line for each unit, indicating which one has completed a seek.
$START_0$	B27	1077	Buffered $START_1$
$START_1$	B32	1013	Buffered $BSTART_0$
$TAG1L_{0,1}$	B25	1154	Cylinder address strobe to selected unit.
$TAG2L_{0,1}$	B25	1154	Head selection strobe to selected unit.
$TAG3L_{0,1}$	B25	1154	Control selection strobe to selected unit.
TD_0	B28	1135	Test data from test data pattern generator. Used to generate read data in test mode.
$TEST_1$	B32	1013	Controller is active in test mode.
$TRANSFER_1$	B27	1077	Controller active in modus 0, 1, 2 or 3.
TST_0	B31	1134	Force parity error. ECC control word bit number 1. Refer to programming specifications.
$US0\text{-}2_1$	B26	1078	Unit select bits. Control bit numbers 7, 8 and 9.
$USL0\text{-}2_{0,1}$	B26	1078	Unit select bits to units.

Signal	*Generated* POS	CARD	*Signal Explanation List:*
WA_0	B26	1078	Wrong address. Block address register did not match with address read from disk (PH2).
WCS_0	B32	1013	Load the word count and the request count register with the number of words to be transferred.
WCZ_0	B30	1092	Word counter equals zero. I.e., specified number of words transfer to/from disk.
WD_1	B28	1135	Write data to disk.
$WDL_{0,1}$	B21-24	1156	Write data to disk.
WF_0	B27	1077	Write format. I.e., controller in modus 5 and format switch in position.
WG_0	B27	1077	Write gate. Enables the write data to be written on the disk.
WRC_0	B21-24	1156	Write clock. Same as servo clock from selected disk unit.
$WRCL_{0,1}$	B21-24	1156	Write clock (servo clock) sent back to the selected unit with the write data.
Write $Core_1$	B27	1077	Controller in modus 0. I.e., read data from disk.
$18S_0$	B21-24	1156	18 sectors. I.e., used to indicate that format with 18 sectors is in use (ECC controller).

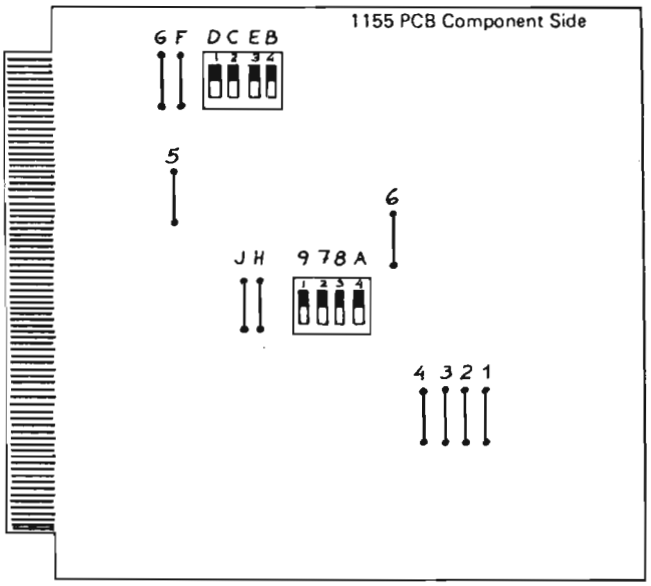
APPENDIX D

1155 — PCB SWITCHES AND JUMPERS

Refer to the assembly drawing for exact locations.

1155 PC SWITCHED & JUMPERS

(Refer to the Assembly Drawing for exact locations)

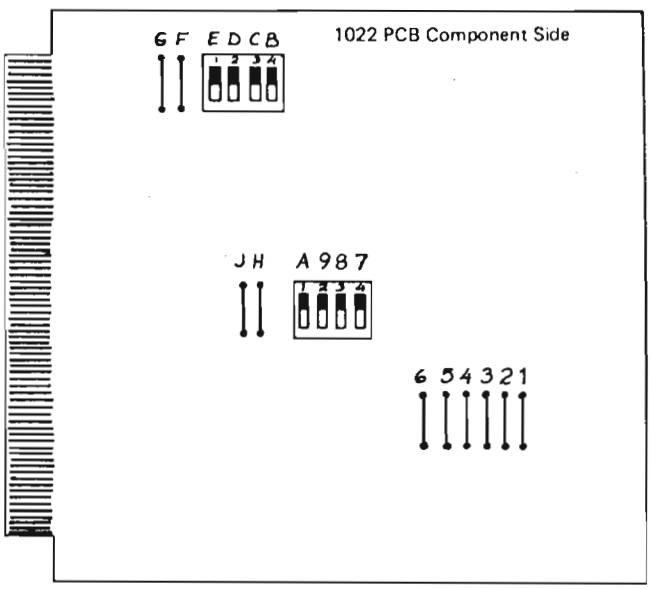


SIGNAL NAMES'

Switches and Jumpers Numbers

1	$CA3_1$	Core Address Register
2	$CA2_1$	
3	$CA1_1$	
4	$CA0_1$	
5	$I6_1$	
6	$I7_1$	
7	$D9_0$	
8	$D4_0$	
9	$D5_0$	
A	$D3_0$	
B	$I5_1$	
C	$I4_1$	
D	$I3_1$	
E	$I2_1$	
F	$I1_1$	
G	$I0_1$	
H	$D6_0$	
J	$D7_0$	

D = IOX code
I = IDENT code



I = JUMPERS
I = D IP SWITCHES

1155 Example:

IOX = 1550_8, ID = 20_8, CAR = 3

INSTALL JUMPERS: 3, 4, 5, 6, F, G, H.

"CLOSE" SWITCHES: 7, 9, A, B, D, E.

ND-11.013.01

APPENDIX E

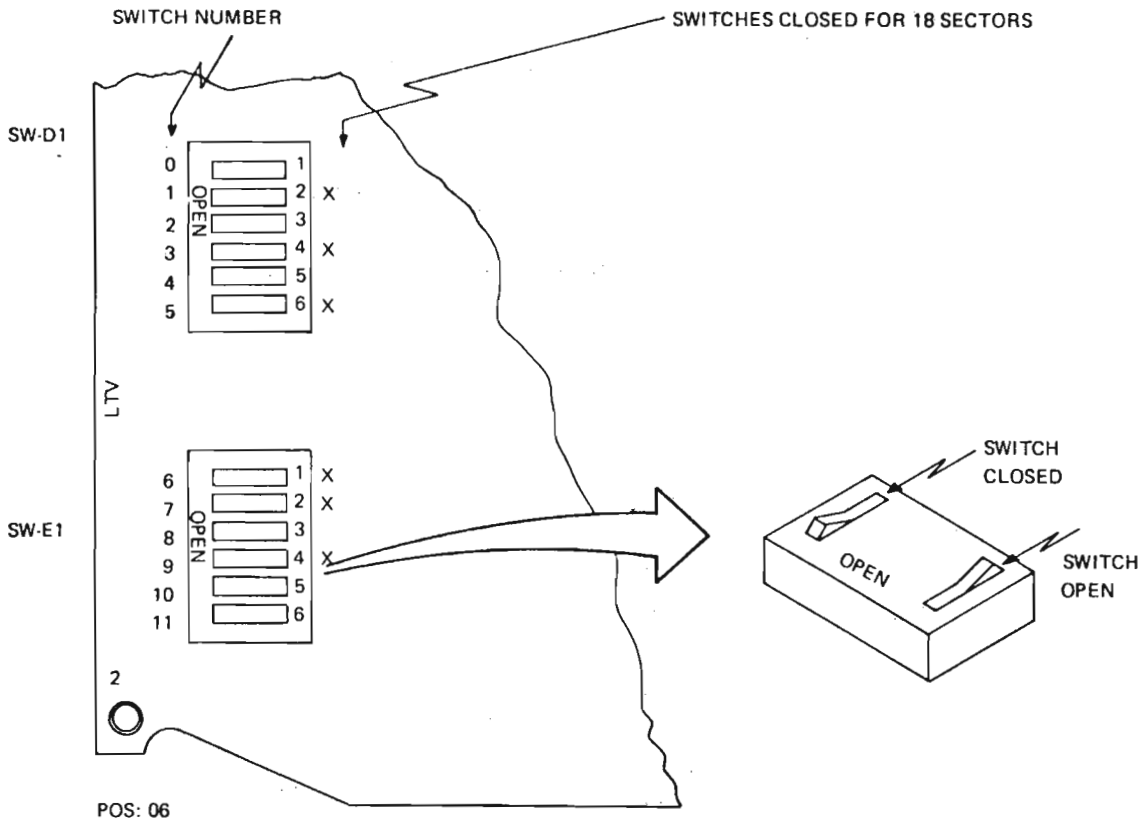
SMD SECTOR SWITCH SETTING

The number of sectors per track is selected by toggle DIP switches on the LTV card in POS 06 in the logic chassis of the disk unit.

For the ECC format 18 sectors are used. Refer to the following table and illustration for proper switch setting.

Switch Number	Open/ Closed	Switch Value	Present Value
0	open	1	
1	closed	2	2
2	open	4	
3	closed	8	8
4	open	16	
5	closed	32	32
6	closed	64	64
7	closed	128	128
8	open	256	
9	closed	512	512
10	open	1024	
11	open	2048	
		746*	

* Number of diebits per sector is 746 + 1 = 747



APPENDIX F

PCB POWER REQUIREMENT

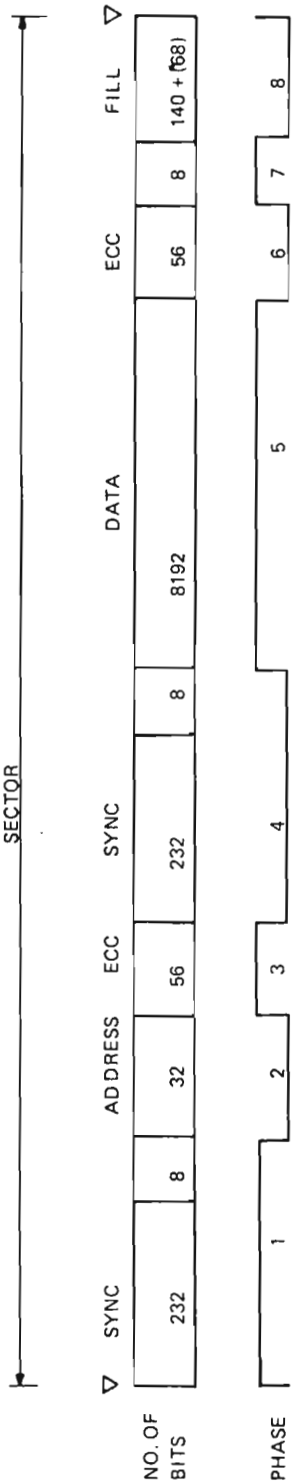
Numbers given are *nominal* @ + 5VDC (+ 10%)

1133	2.5 A
1134	0.75 A
1135	1.0 A
1154	0.75 A
1155	0.9 A
1156	1.0 A
1013	0.85 A
1077	0.6 A
1078	0.75 A
1092	1.6 A

TOTAL =	10.78 A with 1155
	9.80 A without 1155
	12.80 A with 4 x 1156 and without 1155

APPENDIX G

TRACK/SECTOR FORMAT



APPENDIX H

ECC DISK CONTROLLER BACKWIRING

A0 1	B25: 45	B31: G74		
A1 1	B25: 46	B31: 57		
A2 1	B25: 47	B31: 72		
A3 1	B25: 48	B31: 60		
A4 1	B25: 49	B31: 70		
A5 1	B25: 50	B31: 62		
A6 1	B25: 51	B31: 68		
A7 1	B25: 52	B31: 64		
A8 1	B25: 53	B31: 66		
A9 1	B25: 04	B31: 29		
A10 1		B31: G30		NO LOAD
BA 1	B26: 41	B31: G37		
BCOMPL 0	B30: 88	B32: G88		
BCZ 0	B26: 37	B28: G37		
BERROR 0	B30: 86	B32: G86		
BRBUSY 0	B30: 90	B32: G90		
BSTART 0	B30: 92	B32: 92	1155: G92	
B0L 0	B25: G94			
B1L 0	B25: 92			
B2L 0	B25: 90			
B3L 0	B25: 88			
B4L 0	B25: 86			
B5L 0	B25: 84			
B6L 0	B25: 82			
B7L 0	B25: 80			
B8L 0	B25: 80			
B9L 0	B25: G76			
B15 0	B27: 45	B28: G45		
B31 0	B27: 40	B28: G40		
B63 1	B27: 46	B28: G46		
B192 0	B27: 42	B28: G42		
B8191 0	B27: 44	B28: G44		
B0L 1	B25: G95			
B1L 1	B25: 93			
B2L 1	B25: 91			
B3L 1	B25: 89			
B4L 1	B25: 87			
B5L 1	B25: 85			
B6L 1	B25: 83			
B7L 1	B25: 81			
B8L 1	B25: 79			
B9L 1	B25: G77			
B7 0	B27: 43			
CF 0	B30: 72	B32: 72	1155: G72	
CL 0	B23: 32	B26: G33	B27: 33	B28: 33
COMPL 0	B27: G23	B32: G22		B31: 33
CARRY 0	B24: G71	B24: 70		
CRCM 0	B28: 91	B31: G91		

DATA READY 0	B30: 62	B32: 62		
DA0 1	B24: 38	B25: 38	B31: 38	B32: T38
DA1 1	B24: 39	B25: 39	B31: 39	B32: T39
DA2 1	B24: 40	B25: 40	B31: 40	B32: T40
DA3 1	B24: 41	B25: 41	B31: 41	B32: T41
DA4 1	B31: 42	B32: T42	B24: 42	
DA5 1	B31: 43	B32: T43		
DA6 1	B31: 44	B32: T44		
DA7	B31: 45	B32: T45		
DA8 1	B25: 42	B31: 46	B32: T46	
DA9 1	B25: 43	B31: 47	B32: T47	
DA10 1	B25: 44	B31: 48	B32: T48	
DA11	B31: 49	B32: T49		
DA12	B31: 50	B32: T50		
DA13	B31: 51	B32: T51		
DA14	B31: 52	B32: T52		
DA15	B31: 53	B32: T53		
DCS 0	B27: G19	B30: 19		
DEQL 0		BB30: 82	B31: 82	1155: G82
DEQLM 0	B31: G84	B32: 82		
DEVICE REQUEST 0	B30: G66	B32: 66		
DINPUT 0	B30: 64	B32: 64		1155: G64
DRD 1	B26: G18	B27: 18	B28: 19	B30: 18
DR0 1	B27: 56	B30: 76	B31: 76	B32: 76
DR1 1	B30: 78	B31: 78	B32: 78	
DR2 1	B30: 80	B31: 80	B32: 80	
DWC 0	B27: G49	B30: 48		
DW7 1	B28: 41	B30: G41		
ECLOCK 1	B28: G66	B29: 66		
EC2 0	B25: 36	B26: G38	B31: 36	
EH1Z 0	B28: 29	B29: G27		
EL0Z 0	B28: 76	B29: G76		
EOG 1	B28: 38			NO SOURCE
EQUAL 0	B24: G52	B27: 55		
EQUALD 1	B27: G51	B26: 50		
ERROR 0	B26: G36	B32: 36		
ERST 0	B28: G09	B29: 13		
E55 1	B28: 20	B29: G20		
E55 0	B29: G21			NO LOAD
FAULTL 1	B26: 85			
FAULTL 0	B26: 84			
FBC 1	B28: G07	B29: 16		
GND	B26: 04			
GND	B30: 84			
H	B30: 06	B30: 07		
HB8 0	B25: 26	B31: G16		
HB16 0	B25: 27	B31: G17		
HB32 0	B31: G18			NO LOAD
HB64 0	B25: 31	B31: G19		

INDEX L 0	B24: 78			
INDEX L 1	B24: 79			
IRQ 1	B27: G29	B30: 29		
I 1	B28: G10	B29: 10		
KILL 0	B28: G93	B29: 92		
LCW 0	B25: 55	B26: 55	B32: G56	
LONG 0	B28: 92	B31: G92		
MARG 1	B25: 18	B26: G12		
MAXCNT 0	B28: 82	B29: G80		
MC 0	B24: 54	B30: G54	B25: 54	
MCM 0	B28: 58	B30: 58	B31: 58	B32: 58
MDB0 1	B26: T60	B29: 59	B30: 59	B31,32: 59
MDB1 1	B26: T61	B29: 61	B30: 61	B31,32: 61
MDB2 1	B25: 62	B26: T62	B29: 63	B30,31,32: 63
MDB3 1	B26: T63	B29: 65	B30: 65	B31,32: 65
MDB4 1	B26: T64	B29: 67	B30: 67	B31,32: 67
MDB5 1	B26: T65	B29: 69	B30: 69	B31,32: 69
MDB6 1	B26: T66	B29: 71	B30: 71	B31,32: 71
MDB7 1	B26: T67	B29: 73	B30: 73	B31,32: 73
MDB9 1	B26: T69	B29: 77	B30: 77	B31,32: 77
MDB10 1	B26: T70	B29:79	B30: 79	B31,32: 79
MDB11 1	B26: T71	B29: 81	B30: 81	B31,32: 81
MDB12 1	B29: 83	B30: 83	B31,32: 83	
MDB13 1	B28: 85	B30: 85	B31,32: 85	
MDB14 1	B29:87	B30: 87	B31,32: 87	
MDB15	B29: 89	B30: 89	B31,32: 89	
ME 0	B26: 48	B27: G48		
MIS 0	B32: G35			NO LOAD
MS 0	B24: 24	B25: 25	B27: G28	B28: 28
MSP 1	B31: 09	B32: G05		
M0 0	B27: 06	B32: G06		
M1 0	B27: 07	B32: G07		
M2 0	B27: 10	B30: 05	B32: G08	
M3 0	B27: 11	B28: 11	B32: G09	
M4 0	B27: 10	B27: 12	B32: G10	
M5 0	B27: 13	B32: G11		
M6 0	B24: 12	B25: 13	B32: G12	
M7 0	B25: 12	B27: 15	B32: G13	
M8 0	B28: 05	B31: G14		
M6C 0	B24: T55	B27: 57	B28: G57	
ONCYL 0	B25: 34	B26: G35	B27: 35	
ONCYL 1	B27: G31	B32: 33		
OPENCL 0	B25: G68			
OPENCL 1	B25: G69			
ONCYLL 0	B26: 80			
ONCYLL 1	B26: 81			

PARERR 0	B28: 08	B29: G08		
PE	B32: G54			
PH1 0	B25: 14	B26: 14	B27: 04	B28: G12
PH2 0	B26: 20	B27: 21	B28: G14	B31: 14
PH3 0	B27: 24	B28: G27		
PH4 0	B27: 17	B28: G18		
PH5 0	B27: 20	B28: G06		
PH6 0	B28: G22			
PH7 0	B28: G24			
PH8 0	B27: 26	B28: G26		
PH14 1	B27: 14	B28: G15		
PH25W 1	B28: G25	B29: 22		
PL 0	B27: G09	B30: 08		
RBUSY 0	B26: G05	B32: 04		
RC 0	B24: G20	B26: 21		
RCE 0	B26: 42	B27: G38		
RCL 0, 1	B24: 86			
RDD 0	B24: G22	B26: 23		
RDDL 0	B24: 88			
RDDL 1	B24: 89			
READYL 0	B26: 78			
READYL 1	B26: 79			
REC 0	B29: 45	B31: G55		
REP 0	B29: 55	B31: G54		
RG 0	B25: 16	B26: 16	B27: G16	B28: 16
RMUX 0	B31: G13			
RRQ 0	B30: 94	B32: 94		
RSECT 0	B32: G34	B26: 34	B29: 91	
RSTE 0	B28: 04	B31: G04		
RCL1	B24: 87			
READ 1	B28: G50	B29: 50		
SB5 0	B25: 24	B26: 24	B27: G30	B32: 24
SB6 0	B26: 25	B27: G25	B32: 25	
SB7 0	B24: G26	B26: G26	B27: G27	B28: G32
SB8 0	B26: G27	B32: 27		B32: 36
SB9 0	B26: 28	B27: 36	B28: G30	B32: 28
SB10 0	B26: 29	B28: G31	B32: 29	
SB11 0	B26: 30	B30: G31	B32: 30	
SB12 0	B27: G31	B32: 31		
SB13	B26: G31	B32: 32		
SCR12	B29: 11			
SCR15	B28: G13	B29: 12		
SEC 0	B26: G40	B27: 39	B28: 39	
SECTL 0	B24: 80			
SECTL 1	B24: 81			
SEEKEL 0	B24: 84			
SEEKEL 1	B24: 85			
SEEKERL 0	B26: 82			
SEEKERL 1	B26: 83			
SERCL 0	B24: 90			
SERCL 1	B24: 91			

SHTE 0	B27: G08	B30: 09	
SI 0	B24: 36	B26: 39	
ST0 0	B24: 04	B26: 06	
ST1 0	B24: 18	B26: 07	
ST2 0	B24: 05	B26: :8	
ST3 0	B24: 19	B26: 09	
ST4 0	B24: 11	B26: 10	
ST5 0	B24: 13	B26: 11	
ST6 0	B24: 14	B26: 13	
ST7 0	B24: 15	B26: 15	
START 0	B26: 44	B27: G41	B25: 30
STRC 1	B25: G10	B25: 11	
SUSL 0	B26: G86		
SUSL 1	B26: G87		
SYNCAC 1	B27: G34	B28: 35	
SL 0	B27: G05	B30: 04	
START 1	B27: 37	B32: G37	
TAG1L 0	B25: 74		
TAG1L 1	B25: 75		
TAG2L 0	B25: 72		
TAG2L 1	B25: 73		
TAG3L 0	B25: 70		
TAG3L 1	B25: 71		
TD 0	B26: 19	B28: G21	
TEST 1	B25: 20	B26: 22	B32: G23
TRANSFER 1	B24: 53	B27: G52	
TST 0	B29: 06	B31: G06	
UNIT SLD 0	B25: 31		
US0 1	B24: 45	B26: G43	
US1 1	B24: 47	B26: G45	
US2 1	B24: 49	B26: G47	
USL 0	B24: 82		
USL 1	B24: 83		
US0L 0	B26: G94		
US0L 1	B26: 95		
US1L 0	B26: 92		
US1L 1	B26: 93		
US2L 0	B26: 90		
US2L 1	B26: 91		
US3L 0	B26: 88		
US3L 1	B26: G89		
WA 0	B26: G46	B27: 50	
WCS 0	B30: 55	B32: G55	
WCZ 0	B27: 47	B30: G46	
WD 1	B24: 16	B28: G17	
WDL 0	B24: 94		
WDL 1	B24: 95		
WF 0	B26: 54	B27: G54	
WG 0	B25: 22	B27: G22	B28: 23
WRC 0	B24: 17	B26: 17	
WRCL 0	B24: G92		
WRCL 1	B24: G93		
WRITE CORE 1	B27: G69	B30: 68	B32: 68
18S 0	B24: 50	B24: 51	

APPENDIX I

CONNECTOR LISTS

Connector List 1013

Connection: 232
Destination: B32
Board Name: Device Register

Pin:		*Signal Code Reference:*	*Pin:*		*Signal Code Reference:*
00		VCC	01		VCC
02		GND	03		GND
04		RBUSY 0	05	G	MSP 1
06	G	MSP 1	07	G	M0 0
08	G	M1 0	09	G	M2 0
10	G	M3 0	11	G	M4 0
12	G	M5 0	13	G	M6 0
14	G	M7 0	15		
16			17		
18			19		
20			21		
22		COMPL 0	23	G	TEST 1
24		SB5 0	25		SB6 0
26		SB7 0	27		SB8 0
28		SB9 0	29		SB10 0
30		SB11 0	31		SB12 0
32		SB13 0	33		ONCYL 1
34	G	RSECT 0	35	G	MIS 0
36		ERROR 0	37	G	START 1
38	T	DA0 1	39	T	DA1 1
40	T	DA2 1	41	T	DA3 1
42	T	DA4 1	43	T	DA5 1
44	T	DA6 1	45	T	DA7 1
46	T	DA8 1	47	T	DA9 1
48	T	DA10 1	49	T	DA11 1
50	T	DA12 1	51	T	DA13 1
52	T	DA14 1	53	T	DA15 1
54	G	PE 1	55	G	WCS 0
56	G	LCW 0	57		
58		MCM 0	59	T	MDB0 1
60			61	T	MDB1 1
62		DATA READY 0	63	T	MDB2 1
64		DINUPUT 0	65	T	MDB3 1
66		DEVICE REQ 0	67	T	MDB4 1
68		WRITE CORE 1	69	T	MDB5 1
70			71	T	MDB6 1
72		CR 0	73	T	MDB7 1
74			75	T	MDB8 1
76		DR0 1	77	T	MDB9 1
78		DP1 1	79	T	MDB10 1
80		DR2 0	81	T	MDB11 1
82		DEQLM 0	83	T	MDB12 1
84			85	T	MDR13 1
86	G	BERROR 0	87	T	MDB14 1
88	G	BCOMPL 0	89	T	MD15 1
90	G	BRBUSY 0	91		
92		BSTART 0	93		
94		RPQ 0	95		
96		GND	97		GND
98		GND	99		VCC

Connector List 1092

Connector: 230
Destination: B30
Board Name: DMA Register

Pin:		*Signal Code Reference:*	*Pin:*		*Signal Code Reference:*
00		VCC	01		VCC
02		GND	03		GND
04		SL 0	05		M2 0
06		H	07		H
08		PL 0	09		SHTE 0
10			11		
12			13		
14			15		
16			17		
18		DRD 1	19		DCS 0
20			21		
22			23		
24			25		
26			27		
28			29		IRQ 1
30			31	G	SB11 1
32			33		
34			35		
36			37		
38			39		
40			41	T	DW7 1
42			43		
44			45		
46	G	WCZ 0	47		
48		DWC 0	49		
50			51		
52			53		
54	G	MC 0	55		WCS 0
56			57		
58		MCM 0	59	T	MDB0 1
60			61	T	MDB1 1
62		DATA READY 0	63	T	MDB2 1
64		DINPUT 0	65	T	MDB3 1
66	G	DEVICE REQ 0	67	T	MDB4 1
68		WRITE CORE 1	69	T	MDB5 1
70			71	T	MDB6 1
72		CR 0	73	T	MDB7 1
74			75	T	MDB8 1
76		DR0 1	77	T	MDB9 1
78		DR1 1	79	T	MDB10 1
80		DR2 1	81	T	MDB11 1
82		DEQL 0	83	T	MDB12 1
84		ECC 0 (GROUND)	85	T	MDB13 1
86		BERROR 0	87	T	MDB14 1
88		BCOMPL 0	89	T	MDB15 1
90		RBUSY 0	91		
92		BSTART 0	93		
94		RRQ 0	95		
96		GND	97		GND
98		VCC	99		VCC

Connector List 1077

Connection: 227
Designation: B27
Board Name: SMD Control

Pin:		*Signal Code Reference:*	*Pin:*		*Signal Code Reference:*
00		VCC	01		VCC
02		GND	03		GND
04		PH1 0	05	G	SL 0
06		M0 0	07		M1 0
08	G	SHTE 0	09	G	PL 0
10		MS 0	11		M3 0
12		M4 0	13		M5 0
14		PH14 1	15		M7 0
16	G	RG 0	17		PH4 0
18		DRD 1	19	G	DCS 0
20		PH5 0	21		PH2 0
22	G	WG 0	23	G	COMPL 0
24		PH3 0	25	G	SB6 0
26		PH8 0	27	W	SB7 0
28	G	MS 0	29	G	IRQ 1
30	W	SB5 0	31	G	ONCYL 1
32			33		CL 0
34	G	SYNCHC 1	35		ONCYL 0
36		SB9 0	37		START 1
38	G	RCE 0	39		SEC 0
40		B31 0	41	G	START 0
42		B192 0	43		B7 0
44		B8191 0	45		B15 0
46		B63 1	47		WCZ 0
48	G	ME 0	49	G	DWC 0
50		WA 0	51	G	EQUALD 1
52	G	TRANSFER 1	53		DEQL 0
54	G	WF 0	55		EQUAL 0
56		DR0 1	57		MGC 0
58			59		
60			61		
62			63		
64			65		
66			67		
68			69	G	WRITE CORE 1
70			71		
72			73		
74			75		
76			77		
78			79		
80			81		
82			83		
84			85		
86			87		
88			89		
90			91		
92			93		
94			95		
96		GND	97		VCC
98		VCC	99		VCC

Connector List 1078

Connector: 226
Designation: B26
Board Name: SMD Receiver

Pin:		*Signal Code Reference:*	*Pin:*		*Signal Code Reference:*
00		VCC	01		VCC
02		GND	03		GND
04		GND	05	G	RBUSY 0
06		ST0 0	07		ST1 0
08		ST2 0	09		ST3 0
10		ST4 0	11		ST5 0
12	G	MARG 1	13		ST6 0
14		PH1 0	15		ST7 0
16		RG 0	17		WRC 0
18	G	DRD 1	19		TD 0
20		PH2 0	21		RC 0
22		TEST 1	23		ROD 0
24		SB5 0	25		SB6 0
26	W	SB7 0	27	G	SB8 0
28		SB9 0	29		SB10 0
30		SB11 0	31	G	SB12 0
32	G	SB13 0	33	G	CL 0
34		RSECT 0	35	G	ONCYL 0
36	G	ERROR 0	37		BC2 0
38	G	EC2 0	39		SI 0
40	G	SEC 0	41		BA 1
42		BCE 0	43	G	US0 1
44		START 0	45	G	US1 1
46	G	WA 0	47	G	US2 1
48		ME 0	49		
50		EQUALD 1	51		
52			53		
54		WF 0	55		LCW 0
56		GND	57		GND
58		−5V	59		−5V
60	T	MDB0 1	61	T	MDB1 1
62	T	MDB2 1	63	T	MDB3 1
64	T	MDB4 1	65	T	MDB5 1
66	T	MDB6 1	67	T	MDB7 1
68	T	MDB8 1	69	T	MDB9 1
70	T	MDB10 1	71	T	MDB11 1
72			73		
74			75		
76			77		
78		READYL 0	79		READYL 1
80		ONCYLL 0	81		ONCYLL 1
82		SEEKERL 0	83		SEEKERL 1
84		FAULTL 0	85		FAULTL 1
86	G	SUSL 0	87	G	SUSL 1
88	G	US3L 0	89	G	US3L 1
90	G	US2L 0	91	G	US2L 1
92	G	US1L 0	93	G	US1L 1
94	G	US0L 0	95	G	US0L 1
96		GND	97		GND
98		VCC	99		VCC

Connector List 1133

Connector: 229
Designation: B29
Board Name: ECC Polynoms

Pin:		*Signal Code Reference:*	*Pin:*		*Signal Code Reference:*
00		VCC	01		VCC
02		GND	03		GNDF
04			05		
06		TST 0	07		
08	G	PARERR 0	09		
10		I 1	11		SCR12 0
12		SCR15 0	13		ERST 0
14			15		
16		RBC1 1	27		
18			19		
20	G	E55 1	21	G	E55 0
22		PH25W 1	23		
24			25		
26			27	G	EHIZ 0
28			29		
30			31		
32			33		
34			35		
36			37		
38			39		
40			41		
42			43		
44			45		REC 0
46			47		
48			49		
50		READ 1	51		
52			53		
54			55		REP 0
56			57		
58			59	T	MDB0 1
60			61	T	MDB1 1
62			63	T	MDB2 1
64			65	T	MDB3 1
66		ECLOCK 1	67	T	MDB4 1
68			69	T	MDB5 1
70			71	T	MDB6 1
72			73	T	MDB7 1
74			75	T	MDB8 1
76	G	ELOZ 0	77	T	MDB9 1
78			79	T	MDB10 1
80	G	MAXCNT 0	81	T	MDB11 1
82			83	T	MDB12 1
84			85	T	MDB13 1
86			87	T	MDB14 1
88			89	T	MDB15 1
90			91		RSECT 0
92		KILL 0	93		
94			95		
96		GND	97		GND
98		VCC	99		VCC

Connector List 1134

Connector: 231
Destination: B31
Board Name: ECC Control

Pin:		*Signal Code Reference:*	*Pin:*		*Signal Code Reference:*
00		VCC	01		VCC
02		GND	03		GND
04	G	RSTE 0	05		
06	T	TST 0	07		
08			09		MSP 1
10			11		
12			13	G	RMUX 0
14		PH1 1	15		N-U
16	G	HB8 0	17	G	HB16 0
18	G	HB32 0	19	G	HB64 0
20			21		
22			23		
24			25		
26			27		
28			29	G	A9 1
30	G	A10 1	31		
32			33		CL 0
34			35		
36		EC2 0	37	G	BA 1
38		DA0 1	39		DA1 1
40		DA2 1	41		DA3 1
42		DA4 1	43		DA5 1
44		DA6 1	45		DA7 1
46		DA8 1	47		DA9 1
48		DA10 1	49		DA11 1
50		DA12 1	51		DA13 1
52		DA14 1	53		DA15 1
54	G	REP 0	55	G	REC 0
56			57	G	A1 1
58		MCM 0	59	T	MDB0 1
60	G		61	T	MDB1 1
62	G	A5 1	63	T	MDB2 1
64	G	A7 1	65	T	MDB3 1
66	G	A8 1	67	T	MDB4 1
68	G	A6 1	69	T	MDB5 1
70	G	A4 1	71	T	MDB6 1
72	G	A2 1	73	T	MDB7 1
74	G	A0 1	75	T	MDB8 1
76		DR0 1	77	T	MDB9 1
78		DR1 1	79	T	MDB10 1
80		DR2 1	81	T	MDB11 1
82		DEQL 0	83	T	MDB12 1
84	G	DEQLM 0	85	T	MDB13 1
86			87	T	MDB14 1
88			89	T	MDB15 1
90			91	G	CRCM 0
92	G	LONG 0	93		
94			95		
96		GND	97		GND
98		VCC	99		VCC

Connector List 1135

Connector: 228
Destination: B28
Board Name: SMD Terminal

Pin:		*Signal Code Reference:*	*Pin:*		*Signal Code Reference:*
00		VCC	01		VCC
02		GND	03		GND
04		RSTE 0	05		M8 0
06	G	PH5 0	07	G	FBC1 1
08		PARERR 0	09	G	ERST 0
10	G	I 1	11		M3 0
12	G	PH1 0	13	G	SCR15 0
14	G	PH2 0	15	G	PH14 1
16		RG 0	17	G	WD 1
18	G	PH4 0	19		DRD 1
20		E55 1	21	G	TD 0
22	G	PH6 0	23		
24	G	PH7 0	25	G	PH25W 1
26	G	PH8 0	27	G	PH3 0
28		MS 0	29		EHIZ 0
30	G	SB9 0	31	G	SB10 0
32	W	SB7 0	33		CL 0
34			35		SYNCHC 1
36			37	G	BC2 0
38		EOG 1	39		SEC 0
40	G	B31 0	41		DW67 1
42	G	B192 0	43		
44	G	B8191 0	45	G	B15 0
46	G	B63 1	47		
48			49		
50	G	READ 1	51		
52			53		
54			55		
56			57	G	MC6 0
58		MCM 0	59		
60			61		
62			63		
64			65		
66	G	ECLOCK 1	67		
68			69		
70			71		
72			73		
74			75		
76		ELOZ 0	77		
78			79		
80			81		
82		MAXCNT 0	83		
84			85		
86			87		
88			89		
90			91		CRCM 0
92		LONG 0	93	G	KILL 0
94			95		
96		GND	97		GND
98		VCC	99		VCC

Connector List 1154

Connector: 225
Designation: B25
Board Name: SMD Transmitter

Pin:		*Signal Code Reference:*	*Pin:*		*Signal Code Reference:*
00		VCC	01		VCC
02		GND	03		GND
04		A9 1	05		
06			07		
08			09		
10	G	STRC 1	11		STRC 1
12		M7 0	13		M6 0
14		PH1 0	15		
16		RG 0	17		
18		MARG 1	19		
20		TEST 1	21		
22		WG 0	23		
24	W	SB5 0	25		M5 0
26		HB8 0	27		HB16 0
28			29		
30		START 0	31		HB64 0
32		CL 0	33		
34		ONCYL 0	35		
36		EC2 0	37	G	BA 1
38		DA 0 1	39		DA 1 1
40		DA2 1	41		DA3 1
42		DA8 1	43		DA9 1
44		DA10 1	45		A0 1
46		A1 1	47		A2 1
48		A3 1	49		A4 1
50		A5 1	51		A6 1
52		A7 1	53		A8 1
54		MC 0	55		LCW 0
56		GND	57		GND
58		−5V	59		−5V
60			61		
62		MDB2 1	63		
64			65		
66			67		
68	G	OPENCL 0	69	G	OPENCL 1
70	G	TAG3L 0	71	G	TAG3L 1
72	G	TAG2L 0	73	G	TAG2L 1
74	G	TAG1L 0	75	G	TAG1L 1
76	G	B9L 0	77	G	B9L 1
78	G	B8L 0	79	G	B8L 1
80	G	B7L 0	81	G	B7L 1
82	G	B6L 0	83	G	B6L 1
84	G	B5L 0	85	G	B5L 1
86	G	B4L 0	87	G	B4L 1
88	G	B3L 0	89	G	B3L 1
90	G	B2L 0	91	G	B2L 1
92	G	B1L 0	93	G	B1L 1
94	G	B0L 0	95	G	B0L 1
96		GND	97		GND
98		VCC	99		VCC

Connector List 1156

Connector: 224
Designation: B24
Board Name: Unit Controller 1

Pin:		*Signal Code Reference:*	*Pin:*		*Signal Code Reference:*
00		VCC	01		VCC
02		GND	03		GND
04	T	ST0 0	05	T	ST2 0
06			07		
08			09		
10		M4 0	11	T	ST4 0
12		M6 0	13	T	ST4 0
14	T	ST6 0	15	T	ST7 0
16		WD 1	17	W	WRC 0
18	T	ST1 0	19	T	SDT3 0
20	W	RC 0	21		
22	W	RDD 0	23		
24		MS 0	25		
26	W	SB7 0	27		
28			29		
30			31		
32			33		
34			35		
36	W	SI 0	37		
38		DA0 1	39		DA1 1
40		DA2 1	41		DA3 1
42			43		
44			45		US0 1
46			47		US1 1
48			49		US2 1
50		18S 0	51		18S 0
52	W	EQUAL 0	53		TRANSFER 1
54		MC 0	55	T	M6C 0
56		GND	57		GND
58		–5V	59		–5V
60			61		
62			63		
64			65		
66			67		
68			69		
70		CARRY 0	71		CARRY 1
72			73		
74			75		
76			77		
78		INDEXL 0	79		INDEXL 1
80		SECTL 0	81		SECTL 1
82		USL 0	83		USL 1
84		SEEKEL 0	85		SEEKEL 1
86		RCL 0	87		RCL 1
88		RDDL 0	89		RDDL 1
90		SERCL 0	91		SERCL 1
92	G	WRCL 0	93	G	WRCL 1
94	G	WDL 0	95	G	WDL 1
96		GND	97		GND
98		VCC	99		VCC

APPENDIX J

A THEORETICAL INTRODUCTION TO ERROR CORRECTING CODES (ECC)

J.1 *INTRODUCTION*

Some introductory concepts regarding polynomial structures and vector representation in binary algebra.

$$A(x) = C_n \cdot x^{n-1} + \ldots + C_i \cdot x^i + \ldots + C_1 \cdot x^1 + C^0 \cdot X^0$$

where C_i are constants in modulo x arithmetic (i.e., $0 \leqslant C_i < x$) and x is the base. If $x = 2$ then $C = 0$ or 1.

As an example:

$A(x) = 5x_3 + x + 3$ base 10 $(x = 10)$

then this A(x) is actually a number

$= 5 \cdot (10)_3 + (10)_1 + 3(10)_0 = 5013_{10}$

Another Example:

$A(x) = x^5 + x^1 + x + 1$ base 2 $(x = 2)$

$= A(x) = 2^5 + 2^1 + 2 + 1 = 32 + 4 + 2 + 1 = 39_{10}$

or $= 100111^2$

In binary arithmetic, it is very convenient to represent numbers as vectors, for example:

110101^2

becomes

$1 \cdot x^0 + 0 \cdot x^1 + 1 \cdot x^2 + 0 \cdot x^3 + 1 \cdot x^4 \cdot + 1 \cdot x^5$

$= x^5 + x^4 + x^2 + 1$

or reverse

$= 2^5 + 2^4 + 2^2 + 1 = 53_{10} = 110101^2$

This means that a block of binary data (or any base for that matter) can be thought of as a number and represented by a vector A(x).

J.2 *LINEAR, CYCLIC CODES FOR BURST ERROR CORRECTION*

Without proving and demonstrating all the properties of this class of codes, the mathematics of burst error correction using these codes can be demonstrated as follows:

According to EUCLID'S ALGORITHM, we have:

For any 2 vectors f(x) and g(x):

$$f(x) = q(x) \cdot g(x) + r(x)$$

Or in other words, any number (here f(x)) is a certain multiple of another number (here g(x)) and plus a remainder (here r(x)).

This theorem is used in ECC.

Define the data as a vector = M(x)
Define the generator polynomial = G(x)

We than have:

$$M(x) = Q(x) \cdot G(x) + R(x)$$

where

M(x) = data
R(x) = ECC check bits

and we record M(x) + R(x) on the media.

It is desirable not to have to sort R(x) out from the code word to obtain M(x) when we read it back. In order to separate the two, the data vector M(x) is premultiplied by x^{n-k} where n is the length of the data + ecc bits and k is the length of the data.

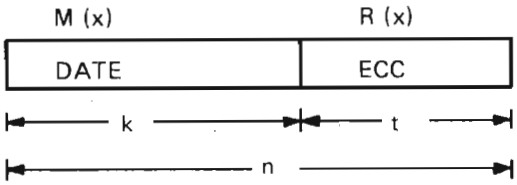
Thus,

$$x^{n-k} \cdot M(x) = Q(x) \cdot G(x) + R(x)$$

$$R(x) + x^{n-k} \cdot M(x) = Q(x) \cdot G(x) + R(x) + R(x)$$

$$x^{n-k} \cdot M(x) + R(x) = Q(x) \cdot G(x)$$

or pictorially



where t = n–k

The code word vector C(x) is recorded on the media where

$$C(x) = x^t \cdot M(x) + R(x)$$

Note that in binary Galois Fields (GF(2)), 1 + 1 = 0 which means that R(x) + R(x) = 0.

When the code word vector, C(x), is read back (module G(x)) from the media, we obtain:

$$C(x)/_{\text{mod } G(x)} = x^t \cdot M(x) + R(x)/_{\text{mod } G(x)} = 0$$

or in other words C(x) reduced by G(x) (the generator polynomial) yields a remainder = 0.

This is then used to check the data when it is read back. We know that for no errors in the data, it is necessary (but not sufficient) for $C(x)/_{\text{mod } G(x)}$ to be equal to zero.

What happens if an error is introduced into the code word?

Let the error be a burst B(x) located at position in the code word.

The error vector can then be represented by:

$$x^i \cdot B(x) = E(x) \text{ (error vector)}$$

The code word we read back (with error) will now look like:

$$C'(x) = C(x) + E(x)$$
$$C'(x) = x^t \cdot M(x) + R(x) + x^i \cdot B(x)$$

This C'(x) now gets reduced by (fed into) the generator polynomial G(x) and we get:

$$C'(x)/_{\text{mod } G(x)} = x^t \cdot M(x) + R(x) + x^i \cdot B(x)/_{\text{mod } G(x)}$$

Since we are discussing only linear codes here we get:

$$x^t \cdot M(x) + R(x) + x^i \cdot B(x)/_{\text{mod } G(x)}$$
$$= x^t \cdot M(x) + R(x)/_{\text{mod } G(x)} + x^i \cdot B(x)/_{\text{mod } G(x)}$$
$$= 0 + x^i B(x)/_{\text{mod } G(x)} = S(x) \neq 0$$

S(x) is the "syndrome" or what is left in the generator polynomial shift register when the whole code word C(x) has been read in.

The problem of error correction is now:

"Given S(x) find the error burst B(x) and its location in the data i".

The equation can be rewritten to read:

$$B(x) = x^{-i} \cdot S(x)/_{\text{mod } G(x)}$$

It might be beneficial at this point to recapitulate the events so far and also tie the theory to actual operations of logic circuits.

To generate the code vector in the first place, we needed to generate:

$$x^t \cdot M(x) + R(x)$$

M(x) = data vector
R(x) = remainder of

$$x^t \cdot M(x) / \text{mod } G(x)$$

Thus, to record the data, we need a polynomial shift register that multiplies by x^t and divides by G(x). R(x) is what is left in this shift register when all of M(x) has been fed into it.

x^t M(x) is just M(x) with t zeros after it, but instead of zeros, we write R(x) which is of length t.

To read the data back and check for errors, we need a feedback shift register that divides by G(x).

Now load a logical "1" into its least significant position and shift the register i times. The contents of the polynomial register will now be:

$$x^i = r^i(x) / \text{mod } G(x)$$

$r^i(x)$ is the reaminder of x^i reduced by G(x).

x^i is actually a binary number:

100.....0

i zeros.

Forward shifts of the polynomial then actually multiplies by x for every shift.

Likewise, if we could shift the shift register backwards i times we would get (preloaded with a "1"):

$$x^{-i} = r^{-i}(x) / \text{mod } G(x)$$

Reverse shifts of polynomial multiples by 1/x for every shift.

If we could shift the registers (with a preloaded one) sufficiently many times in each direction, we would get back to a "one" again. This is because G(x) is "cyclic" or has a "period". Let N be the number of shifts required to get back again (period = N). We then have:

$$x^N = 1 \text{ mod } G(x)$$

and $x^{-N} = 1 \text{ mod } G(x)$

The period of a polynomial can be determined by factoring it and determining the "order" of each factor. The period is then the least common multiple of the orders of the factors.

Back to the correction problem again:

From $B(x) = x^{-i} S(x) / \text{mod } G(x)$ we needed to determine B(x) and i given S(x) and of course G(x).

We could run the shift register containing S(x) backwards, thereby for every shift obtaining:

$$x^{-1}\,S(x)_{\text{mod } G(x)}$$

$$X^{-2}\,S(x)/_{\text{mod } G(x)}$$

$$\vdots$$

$$x^{-i}\,S(x)/_{\text{mod } G(x)}$$

$$\vdots$$

$$x^{-N}\,S(x)/_{\text{mod } G(x)}$$

For every shift we would now look at B(x) and see if it is small enough to be correctable. If, for some shifts, it is small enough, we have solved the problem in that we have found both B(x) and i. If we could not find a small enough B(x) until we had shifted N times, the error would be uncorrectable, i.e., minimum B(x) is too big. However, running feedback shift registers backwards is clumsy and very hardware consuming.

Now since $x^N = 1/_{\text{mod } G(x)} = x^{-N}$

We could run the shift registers forward and get the following sequence:

$$x^{-N+1}S(x)/_{\text{mod } G(x)}$$

$$x^{-N+2}\,S(x)/_{\text{mod } G(x)}$$

$$\vdots$$

$$x^{-i}S(x)/_{\text{mod } G(x)}$$

$$\vdots$$

$$x^{-1}\,S(x)/_{\text{mod } G(x)}$$

Again when B(x) is small enough we have solved the problem and found B(x) and i. And if no small B(x) is found for N shifts, the error is uncorrectable.

This correction algorithm is widely used today, and would be an ideal solution were it not for certain factors:

Cyclic codes are not well suited for error correction at longer bursts when the code word length is close to the period fo the code (N). In fact, for burst error correction is that the code word length n should be: $n << N$.

The importance of this can be seen in a later section on correction and detection capacilities. When the code is used with a maximum code word length that is less than the period, it is called a "shortened" code.

All it means is that the N–n leading digits of the code word are always equal to zero.

The fact that n << N impacts the correction algorithm where we got the following sequence for forward shifts:

$$x^{-N+1}S(x)/_{\text{mod } G(x)}$$

$$x^{-N+2}S(x)/_{\text{mod } G(x)}$$

$$X^{\&-n}S(x)/_{\text{mod } G(x)}$$

$$\vdots$$

$$x^{-i}S(x)/_{\text{mod } G(x)}$$

$$\vdots$$

$$x^{-1}S(x)/_{\text{mod } G(x)}$$

Hence, we have to do N-n shift initially while virtually "nothing is happening". If we could establish an initial condition of

$$x^{-n} \cdot S(x)$$

before shifting, we eliminate the N-n initial shifts.

We recall from reverse shifts of the polynomial preloaded with a "one", we obtained:

$$x^{-i} = r^{-i}(x)/_{\text{mod } G(x)}$$

Therefore, if we shift in reverse n times, we get:

$$x^{-n} = r^{-n}(x)/_{\text{mod } G(x)}$$

The code word with error was:

$$C(x) = x^t M(x) + R(x) + x^i B(x)$$

If we multiply C(x) by x^{-n} when at the same time we reduce C(x) by G(x), we get:

$$x^{-n} \cdot x^t M(x) + x^{-n} R(x) + x^{-n} x^i B(x)/_{\text{mod } G(x)}$$

$$= x^{-n}(M(x)x^t + R(x))/_{\text{mod } G(x)} + x^{-n} x^i B(x)/_{\text{mod } G(x)}$$

$$= 0$$

$$= x^{-n} x^i B(x)/_{\text{mod } G(x)} = r^{-n}(x) \cdot S(x)/_{\text{mod } G(x)}$$

Here we have obtained the initial condition:

$$x^{-n} \cdot S(x) = r^{-n}(X)\, S(X)/_{\text{mod } G(x)}$$

where $r^{-n}(x)$ is the remainder left in the shift register when it is reloaded by a logical one, then shifted n times in the reverse direction.

From the error equation:

$$B(x) = x^{-i} S(x)/_{\text{mod } G(x)}$$

we have now established the initial condition

$$x^{-n} S(x)/_{\text{mod } G(x)}$$

and forward shifts give us this sequence:

$$x^{-n} S(x)/_{\text{mod } G(x)}$$

$$x^{1-n} S(x)/_{\text{mod } G(x)}$$

$$x^{2-n} S(x)/_{\text{mod } G(x)}$$

$$\vdots$$

$$x^{-i} S(x)/_{\text{mod } G(x)}$$

$$\vdots$$

$$x^{-2} S(x)/_{\text{mod } G(x)}$$

$$x^{-1} S(x)/_{\text{mod } G(x)}$$

The hardware implications here are as follows:

When we read back the code word, we multiply by $x^{-n} = r^{-n}(x)/_{\text{mod } G(x)}$ and reduce $C(x) \cdot x^{-n}$ by G(x). The coefficients of $r^{-n}(x)$ must be hardwired into the polynomial shift register.

Then when the whole code word C(x) has been read, we shift the register forward one shift at a time and look for a sufficiently small B(x) to be correctable. The number of shifts is i and equals the displacement of B(x) in the code vector C(x). If no correctable, B(x) is found nor n forward shifts, the error is uncorrectable.

J.3 *SPECIFIC POLYNOMIALS*

The class of codes most frequently used for burst error detection and correction are so-called fire codes (from their "discoverer" P. Fire).

These codes are of the general format:

$$G(x) = (X^c + 1)\, \pi_i\, P(x)_i$$

where:

G(x) is the generator polynomial
$P(x)_i$ are prime, irreducible polynomials.

Fire codes are linear, cyclic codes as are their "shortened" versions.

One could construct one's own code based on the general format in this fashion:

$P(x)_i$ are available from published tables up to degree 34

$X^c + 1$ (the orders of $P(x)_i$ polynomials must not divide c)

and then thoroughly simulate the resulting code to prove its capabilities.

Or one could select among codes already used for burst error correction by manufacturers. The latter approach results in less word and increased confidence that the code is proper.

Two likely candidate codes have been selected for investigation:

1. 48 bit code (IBM 3340 and 3350)

$$G(x) = (x + 1)(x^{12} + x^{13} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)(x^{35} + x^{23} + x^8 + x^2 + 1)$$

2. 56 bit code (IBM 3330)

$$G(x) = (x^{22} + 1)(x^{11} + x^7 + x^6 + x + 1)(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)(x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$$

Both codes are Fire codes and used for burst error correction and detection.

The other parameter required is:

$$x^{-n} = r^{-n}(x)/_{\text{mod } G(x)}$$

by reverse shifting the polynomials.

Tables A and B give the value for the 48 bit and 56 bit codes respectively.

The left hand column lists $-i$ and to the right is the contents of the polynomial at this reverse shift cound $= r^{-i}(x)$.

For the 48 bit code:

n = length of data + ECC
$n = 512 \times 16 + 48 = 8240$, $N = 4.5 \times 10^{11}$

then

$$r^{-n}(x) = 1 + x + x^3 + x^5 + x^6 + x^7 + x^{10} + x^{11} + x^{13} + x^{17} + x^{20} + x^{21} + x^{22} + x^{24} + x^{26} + x^{28} + x^{29} + x^{30} + x^{33} + x^{35} + x^{36} + x^{39} + x^{40} + x^{44} + x^{45} + x^{46} + x^{47}.$$

For the 56 bit code:

$$n = 512 \times 16 + 56 = 8248, N = 585422$$

and

$$r^{-n}(x) = x^2 + x^5 + x^8 + x^9 + x^{11} + x^{15} + x^{17} + x^{23} + x^{24} + x^{25} + x^{26} + x^{28} + x^{29} + x^{30} + x^{31} + x^{32} + x^{36} + x^{37} + x^{39} + x^{45} + x^{46} + x^{47} + x^{48} + x^{50} + x^{51} + x^{54} + x^{55}.$$

- 0THIS IS REVERSE WINCHESTER CODE ROLL

8225	101111101001110100010111000110100100101000001110
8226	001[illegible]1100101[illegible]1000111[illegible]101[illegible]10100100[illegible]1101
8227	010111110010111000111000011010010100100000111010
8228	101111100101110001110000110100101001000001110100
8229	001111011011001011101011101001010001000011101001
8230	011110110110010111010111010010100010000111010010
8231	111101101100101110101110100101000100001110100100
8232	101011001001110101010111001010001011011101001001
8233	000110000011000010100100010100010101111010010011
8234	001100000110000101001000101000101011110100100110
8235	[illegible]
8236	[illegible]
8237	[illegible]01111100101011101100100111[illegible]01
8238	[illegible]0100[illegible]1010010100001010110000010011[illegible]
8239	1100101100[illegible]11100100100010010101110011010[illegible]100[illegible]1
8240	110101110011010001001110101011100101100110001111
8241	[illegible]10111010[illegible]0100101110101110010000011[illegible]111
8242	1001111111100111100100100101110010011011000111111
8243	0111111010[illegible]01000100001101110010010111000111111111
8244	1111101001001000100001101110010010111000111111110
8245	101110110[illegible]110110000011111001001010000011111101
8246	001101111011110000000101100100101011001111111011
8247	011011110111100000001011001001010110011111110110
8248	110111101111000000010110010010101100111111101100
8249	111111001110101000100110100101011010111111011001
8250	101110001101111001000111001010110110111110110011
8251	001100001011011010000010001010110111011110110011
8252	011000010110110100000100010101101110111101100111
8253	110000101101101000001000101011011101111011001110
8254	110001001011111000101000101101110100101100111001
8255	110010000111011001011011011011101010011001110011
8256	110100011110011010111100110111010111100111001111
8257	111000101100011101110011101110101100100111001111
8258	100001001000010011101101011101011010001110011111
8259	010010000000001111010000111010110111011100111111
8260	100100000000011110100001110101101110111001111110
8261	011000010000101010100100110101101111011001111101
8262	110000100000101010010011010110111101100111111010
8263	110001010001111100101100101101111000001111110101
8264	110010110011010001010011011011110011011111101011
8265	110101110110001010101001101111001011111111010111
8266	111011111100111101010011101111001000111110101111
8267	100111101010100101011010111100100101111010111111
8268	011111000010001101010000111100100110111010111111
8269	111110000100011010100001111001001101110101111110
8270	101100011000011101001001110010011000101011111101
8271	001000100000010010011001100100110010010111111011
8272	010001000000100100110011001001100100101111110110
8273	100010000001001001100110010011001001011111101100
8274	010100010010111011000110100110010001111111011001

Table J.1: 48 Bit Polynomial

```
-@THIS IS R[illegible]ERSE MERLIN   CODE ROLL
8225    00101100001011101011001000110101111110101100131100000101
8226    01011[illegible]
8227    10111[illegible]
8228    11101001110000100011010110001101110010000110100001010101
9229    01011011001100111100111100011001100001111001010011101010
9230    10110110011001111001111000911001100011110010100111010101
8231    11100100011110000100110000110010000100001000101111110101
8232    01000000010001101001010011001010010110101101011100101101
8233    10000000100011010010100110010100101101010110101110010110
8234    10001001101011011111011100101011011101011101010008110111
8235    10011011111010100100101001010100[illegible]  01010010000071110110
8238    01100100011001000010111010101001[illegible]
8239    11001000110010000101110101010011111000010111100111000010
8240    00011001001001110001111010100101110111000111011000000111
8241    00110010010011100011110101001011101110001111011000001110
8242    01100100100111000111101010010111011100011110100000011100
8243    11001001001110001111010100101110111000111101000000111000
8244    00011010110901100100111001011111110110010031110001111001
8245    00110101100311001001110010111111101100100111000111100110
8246    01101[illegible]
8247    11010[illegible]
8248    00100[illegible]
8249    01001001101001101000001111111111109011010000011110111001 10
8250    10010011010011010000001111111111000110100000011110110011 00
8251    10101110001010110101010111111111801110110101101010030110 11
8252    11010100111011001111001111111111011110111111000101011010 1
8253    00100091011011100103001111111111111111001010011011110100 1
8254    01000010110111001000011111111111111110010100110111101001 0
8255    10000101101110010000111111111111111100101001101111010010 0
8256    10000011110001011011101111111101110101001110011111001011
8257    10001111001111001101001111111001101101110100911600101 01
8258    10010110110011100000001111110000101110000000001101010 1001
8259    10100101001010111010001111100001111111101000010111010 0001
8260    11000010111000001110001111000011111000111005001100100 0001
8261    00001101011101100110001110000101110110011000111011000 0001
8262    00011010111011001100011100001011101100110001110110000 0010
8263    00110101110110011000111000010111011001190011101100000 0100
8264    01101011101100110001110000101110110011000111011000001 0000
8265    11010111011001100911100001011101100110001110110000010 0000
8266    00100110011110111110101001011100100101111010101000010 0011
8267    01001100111101111010100101110010010111101010000101000 0110
8268    10011001111011110101001011100100101111010101000101000 1100
8269    10111011011010010000000111001010110010030001101100110 11
8270    11111100110010101101001111001010011010110100100111010 1011
8271    01110100011111001110101100101011101100111010111111101 0001
8272    11101000111110011101011001010111011001110101111111010 0010
8273    01011001010010100300910001010110011010000300110111001 0111
8274    10110010100010000001000101011001101000030110111010011 10
8275    11101101101001111000011010110010101011110010101000011111
```

Table J.2: 56 Bit Polynomial

J.4 *CORRECTION AND DETECTION CAPABILITIES*

There is a small, but finite possibility that an error burst (B'(x) starting at location j and represented by the vector $x^jB'(x)$ will be instantly decoded as a different error burst B(x) starting at location i and represented by a different vector $x^jB(x)$.

This phenomena can be expressed as:

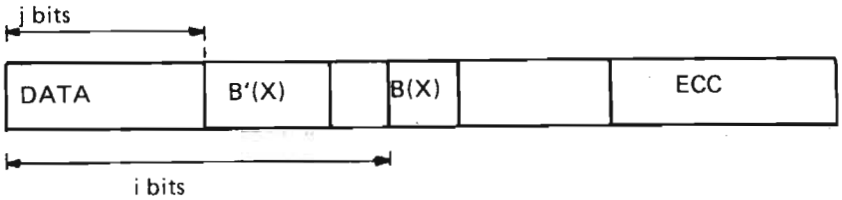
$$x^iB(x) = x^jB'(x)_{\text{mod } G(x)}$$

or, in other words, when the error vectors are reduced, mod G(x) they are identical.

Of greatest concern are situations where B(x) is a burst of sufficiently small length to be correctable and B'(x) can be short (correctable) or long (uncorrectable).

When B'(x) is long, we get the undesirable situation that an uncorrectable error will be decoded as a correctable error. The system might then correct the data based on the erroneous information and we are left with data with errors but little or no indication that such a condition exists.

Pictorially:



There is no analytical method that can provide detailed insights into this situation as far as magnitude is concerned. However, empirically, the condition can be chartered.

From the error vector equation

$$x^iB(x) = x^jB'(x)/_{\text{mod } G(x)}$$

we get, after dividing by x^i

$$B'(x) = x^{i-j}B(x)/_{\text{mod } G(x)}$$

B(x) is the short (correctable) error burst that the long (uncorrectable) B'(x) turns into when it is reduced by G(x).

$x^{i-j}B(x)/_{\text{mod } G(x)}$ is generated by shifting B(x) i–j times n the generator polynomial G(x). (i.e., the same as multiplying B(x) by x^{i-j} module G(x).)

Since n is the length of the code word (data + ecc), we get the polynomial forward (multiply by x) and for i–j = negative, we shift in reverse (divide by x).

Therefore, the mechanics of the sitation are:

1. Load the polynomial with B(x). The length of B(x) is b which now is the maximum number of correctable bits desired.

2. Shift the polynomial n–1 shifts in each direction ($-n \leqslant i-j \leqslant +n$).

3. For each shift, look for a burst pattern B'(x) which is now the pattern that gets misinterpreted as B(x). The length of B'(x) is d which is the maximum number of detectable bits desired.

A NORD Standard FORTRAN program was developed to perform this task and Table J.3 gives the results for the 48 bit polynomial, Table J.4 for the 56 bit polynomial.

The first column indicates the sign of the second column (P = positive, N = negative), the second column is the value i–j, the third column is B(X) and the fourth column is B'(x).

These tables give just a sample of all the patterns (for 48: maximum d = 30, for 56 maximum d = 38).

Tables J.5 and J.6 (J.5 for 48 and J.6 for 56) tabulates the total number of misinterpreted bursts for the polynomials. Vertical, on the left side are the values of d, horizontal across the top are the values of b and the table entries give the number of bursts for given b's and d's.

The approximate numbers can also be derived statistically.

For each pattern B(x), we need n–1 shifts in each direction, i.e., 2(n–1) shifts total. There are 2^b-1 different patterns of B(x) but half of these have a logical 0 in the bottom position, thus their b is actually b–1. We then consider the number of B(x)'s to be $1/2\,2^b-1 = 2^{b-1}-1$.

From this argument, the number of misinterpreted patterns should be:

$$2(n-1)\,(2^{b-1}-1)$$

Now about half of these patterns of B'(x) will have a logical 0 in the bottom position, thus the number of patterns we should expect is half this number:

$$1/2\,(n-1)\,(2^{b-1}-1) = (n-1)\,(2^{b-1}-1)$$

Keep in mind that this is valid for cases where the number of detectable bits (d) is equal to the length of the generator polynomial G(x).

Now let us define the length of G(x) to be t, thus if d is less than t we get the number reduced by one factor:

$$_s5-d$$

such that

$$(n-1)\,(2^{b-1}-1)\,(2^{t-d})-1 \approx n2^{b+d-t-1}$$

for $n >> 1$ and $b >> 1$

For NORD-10 systems, n = 8192 + t

thus if $t << 8192$

we get

$$\approx 2^{b+d-t-1}\,(8192 + t) \approx 2^{13} \cdot 2^{b+d-t-1}$$

$$= 2^{b+d-t+12}$$

This is the number for patterns B'(x) of length d or less.

Exactly of length d will be about half

$$\rightarrow \#\approx \tfrac{1}{2}2^{b+d-t+12} = 2^{b+d+11-t}$$

Now for the 48 polynomial t = 48.

$$\# \approx 2^{b+d+11-48} = 2^{b+d-37}$$

which is in agreement with Table J.5.

For polynomial t = 56, $\# = 2^{b+d-45}$ corresponding to Table J.6.

One interesting aspect here is that statistically, b+d = constant for a given polynomial, such that we can trade off correctability and detectability on a one for one basis.
It can also be see from the equation (and Tables J.5 and J.6) that b has to be very large if we are concerned about the possibility of one correctable pattern being misinterpreted as a different correctable pattern.

For 56 polynomial:

$$2^{b+d-45} = 1$$

$$\rightarrow b + d - 45 = 0 \text{ or } b = 45 - d$$

since by definition, $d \geqslant b$, it means that $b \approx 23$ before correctable patterns are misinterpreted.

One other aspect of correction and detection is when an error (correctable or uncorrectable) will be misinterpreted as a no error condition.

Of cours, if the hardware is malfunctioning this can happen, but in an operable system this means:

$$[x^j B'(x)]_{\text{mod } G(x)} = 0$$

or "Syndrome equal to zero".

The original data vector was

$$M_1(x)$$

and the recorded code word vector was

$$M_1(x) \cdot x^t + R_1(x) + x^j \cdot B^1(x)]_{\text{mod } G(x)}$$

This can only happen if:

$$M_1(x) \cdot x^t + R_1(x) + x^j \cdot B^1(x) = M_2(x) \cdot x^t + R_2(x)$$

and

$$[M_2(x) \cdot x^t + R_2(x)]_{\text{mod } G(x)} = 0$$

That is, the original code word (1) was through introduction of errors transformed into another legal code word (2).

The probability of this happening depends on the "minimum distance" between any 2 code words. The minimum distance is equal to the minimum number of bits that are different between any two code words.

Since I have yet to figure this out for the 48 and 56 polyunomials, this will be addressed later.

N	6017	111011	1
P	3230	100	1
P	3297	110	1
N	3547	111	1
P	5041	100	1
N	909	111	1
P	6418	110	1
P	5041	110	1
P	323	100	1
P	4355	100	1
N	119	100	1
N	5775	100	1
P	5041	111	1
N	477	101	1
P	7084	110	1
P	323	110	1
N	1680	101	1
N	909	100	1
N	607	111	1
P	5041	101	1
P	3063	101	1
P	6975	101	1
P	323	111	1
N	607	100	1
P	5041	111	1
N	5515	101	1
P	6964	111	1
P	7398	110	1
P	5041	101	1
N	909	110	1
N	2038	110	1
P	5041	100	1
N	5775	110	1
N	6601	101	1
P	323	101	1
P	7084	101	1
P	7756	101	1
P	2795	100	1
P	5041	110	1
N	477	111	1
P	6368	101	1
N	909	101	1

Table J.3: 48 Bit Polynomial

N	3206	111100000000	111000111000011100000011100000001000111
P	4864	100110100000	111110010111100100000011011111111111001
P	5744	101001100000	1111111111111100000000001101011001111
P	5744	111101010000	10000000000000100000000101111101010001
N	4760	101000001000	1011111011001110100000000111110110011
P	5744	110111001000	101111111111111010000000100010001101101
P	5744	100011111000	11000000000000110000000111000111110011
P	5744	110010000100	1010000000000010100000001001001000010101
F	6454	100110100100	11100100011111000100000111001001110011
P	3376	111100010100	1001010010010110010100100101001010101
P	5744	100110110100	11011111111111001000000111110010101001
N	4760	111100001100	111000001101010011100000001000011010101
N	1235	110001001100	110101100011110000010010101011000111101
F	5338	101011001100	10001000011011001011000100100000110111
P	5744	101100101100	111000000000001110000001100110010110111

N	5635	110110101100	11100101111010000110111011110111101
P	5099	111001101100	1000100110110100011100010100001101101
P	5744	111000011100	100111111111110110000001010011110001011
N	1133	110010000010	110111111110000000000011101111111111001
P	5918	100100010010	10010011001100010000000000000001000010001
P	8228	100000110010	10010110001001100000000000101010000011
P	5453	101011110010	111101111111110000010000111101101010011
P	1889	100000001010	11000011100111110001001100011010011111
N	4760	100010001010	100100010111110100100000110001011111111
P	3619	100011001010	11010110000110001001011101011000001001
P	5338	111110101010	110011000101101011101001101100001011001
P	4810	101001101010	11110110001110011001100101011000111011
N	1285	101001101010	10111101001000100011011111110100100111
P	495	111100011010	100011011101011100111010100010111110101
P	5099	100101011010	110011010111011100100100111110010110111
N	5635	101101111010	10010111100011100010110011000110000111
P	2100	100011111010	11110010010000011111010111100100110001
N	2344	110011100110	111110001110110010011011111100011011111
P	5987	110000110110	10011110000100001000000010000110001111101
F	6454	110101110110	1001011001000010011000100101101001 0101
P	3005	110010001110	11001111010110000010001000011110101111
N	4760	110110001110	11001110000110100011000000101110000011001
N	2682	111001001110	101010010100111100000010010000001110111
P	7663	101101001110	111100101110110111000001111001110000 1
P	4410	111101101110	11111011000101101011101111101100101011
P	3376	100010011110	11011110110111010111101101111011111111

Table J.4: 56 Bit Polynomial

VALUES ARE AS FOLLOWS:

MAX=11 T=48 HICNT= 8250

S	1	2	3	4	5	6	7	8	9	10	11	12
0												
1	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	0	
21	0	0	0	0	0	0	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	0	1	1	
23	0	0	0	0	0	0	0	0	0	0	1	
24	0	0	0	0	0	0	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	0	0	1	
26	0	0	0	0	0	0	0	0	0	0	1	
27	0	0	0	0	0	0	0	1	2	2	2	
28	0	0	0	0	0	0	0	0	2	4	6	
29	0	0	0	0	0	0	0	0	0	5	10	
30	0	0	0	0	0	1	3	4	7	9	20	
31	0	0	0	0	1	1	2	4	8	24	33	
32	0	0	0	0	0	2	2	4	18	30	70	
33	0	0	0	0	1	4	12	18	30	73	122	
34	0	0	0	0	1	4	12	28	52	115	259	
35	0	0	1	2	6	11	25	63	132	253	489	
36	2	2	2	6	9	26	64	132	259	521	1036	
37	1	4	4	9	20	44	114	247	492	1030	2082	
38	3	7	19	28	59	111	235	504	1030	2029	4035	
39	8	14	29	73	119	236	459	939	2006	4025	8100	
40	13	28	56	114	264	493	997	2005	3999	8234	16501	
41	30	58	118	235	490	1019	2012	4091	8103	16155	32814	
42	58	121	246	511	1025	2054	4177	8178	16350	32610	65434	
43	137	263	532	1024	2051	4187	8305	16671	33058	65947	131512	
44	253	501	1022	2000	4145	8175	16515	33069	65556	131955	263838	
45	553	1036	2053	4107	8308	16467	32991	65785	131695	264400	527072	
46	1065	2140	4049	8235	16524	33100	65898	131840	263505	526817	1055106	
47	2019	4109	8415	16510	32978	65937	131735	263391	526849	1053552	2107519	
48	4105	8193	16581	37207	65756	132004	263437	527066	1053622	2108289	4215552	

Table J.5: 48 Bit Polynomial

VALUES ARE AS FOLLOWS:

MAX=11 T=56 HICNT= 9250

S	1	2	3	4	5	6	7	8	9	10	11
0											
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	1	1	1	2	2
36	0	0	0	0	0	0	0	1	2	3	5
37	0	0	0	0	0	0	0	0	2	7	13
38	0	0	0	1	1	1	2	2	2	6	21
39	0	0	0	1	2	2	2	4	8	13	35
40	0	0	0	0	2	4	4	8	16	27	57
41	0	0	0	0	1	5	10	14	29	57	116
42	1	1	1	1	1	2	17	34	61	127	260
43	0	1	1	2	4	11	20	76	133	254	517
44	1	1	3	4	7	19	51	110	272	551	1020
45	0	2	7	23	31	50	110	219	441	1023	2115
46	7	12	18	36	74	131	258	492	959	1934	4106
47	9	17	31	70	137	293	533	1050	2016	4038	8090
48	15	36	72	130	270	540	1099	2054	4139	8119	16242
49	28	62	130	251	400	1031	2039	4153	8103	16403	32708
50	62	135	249	520	1003	1995	4123	8224	16651	32936	65837
51	144	264	529	1032	2092	4180	8200	16634	33007	66212	131540
52	250	516	1034	2123	4126	8246	16426	32746	65770	131	26
53	478	977	2027	4063	8243	16305	32858	65595	131151	263	526
54	1074	2088	4101	8238	16476	33041	65919	132216	264	529	1053
55	2034	4101	8195	16273	32784	65	131704	263055	526	1	2107
56	4125	8233	16571	32822	65983	132	263023	526	1053	2107	4214

Table J.6: 56 Bit Polynomial

J.5 *SUMMARY*

The question arises: How many bits should I be able to correct and detect?

Correction might be the simplest question. Factors here are based on the storage and recording technology of data utilized such as flying height, tpi, bpi, media thickness and homogenity factors, etc. for disk files. But perhaps the most important factors are the industry standard and the qualification procedures for the particular recording technology in question. In other words, the vendor or equipment manufactorer should not be asked what he recommends, as much as "what do you use to qualify the equipment you are shipping us?".

For the CDC 40, 80 and 300 Mbytes disk drives in question, the anser is simple: CDC uses a QA procedure where the correctability limit is set at one burst per record of maximum 11 bits in error. The technology used in these disk drives are derivates of the IBM 3330 (MERLIN) technology which surfaced in 1971. IBM uses an 11 bit burs per record correction scheme here (maximum code word = 104304 bits), hence both industry standard and QA criteria are set at 11 bits correctable, and this number should be used.

The detectable side in the question is more diffuse. Much of the same arguments as for correctables could be used.

Ideally, one should concentrate more on multiple burst detection; or translated to sea level — one should prepare more for two 100 year waves in rapid succession rather than one 10,000 year wave.

Multiple burst detection is a very complex issue and little understood and most equipment manufacturers settle for the second best approach; that is to make the detectables as long as possible within reason dictated by the hardware required.

Then given that correctables (b) should be 11, the Tables J.3 and J.4 will give:

for 48 polynomial, b = 11
for correctable (d) = 21 maximum

for 56 polynomial, b = 11
for correctables d = 34 maximum

For the 56 polynomial, IBM advertises b = 1 d = 22 for n = 104304.

The wisest choice here is for the 56 polynomial for NORD systems. For little additional hardware over the 48 polynomial (∿ IC's) the detection capabilities are quite improved:

b = 11, d = 34 for n = 8248.

Since, statistically, d and n are related by $n \cdot 2^d$ = constant, the record length could be increased many times before detection capabilities would become seriously impacted.

This seems like a very long-winded and time consuming report on ECC. There are some factors that I think makes it important to know in detail what one is up against when applying ECC to large data bases.

- ECC has an element of risk associated with it. The "holes" of the code should be tabulated and enumerated for the given record length for the cases where the circuitry turns bad data into worse data and signals that "all is well".

- Sophisticated customers concerned about integrity of their data might demand information at this detailed level to be able to decide upon back-up systems, duplications and redundancies of their particular installation.

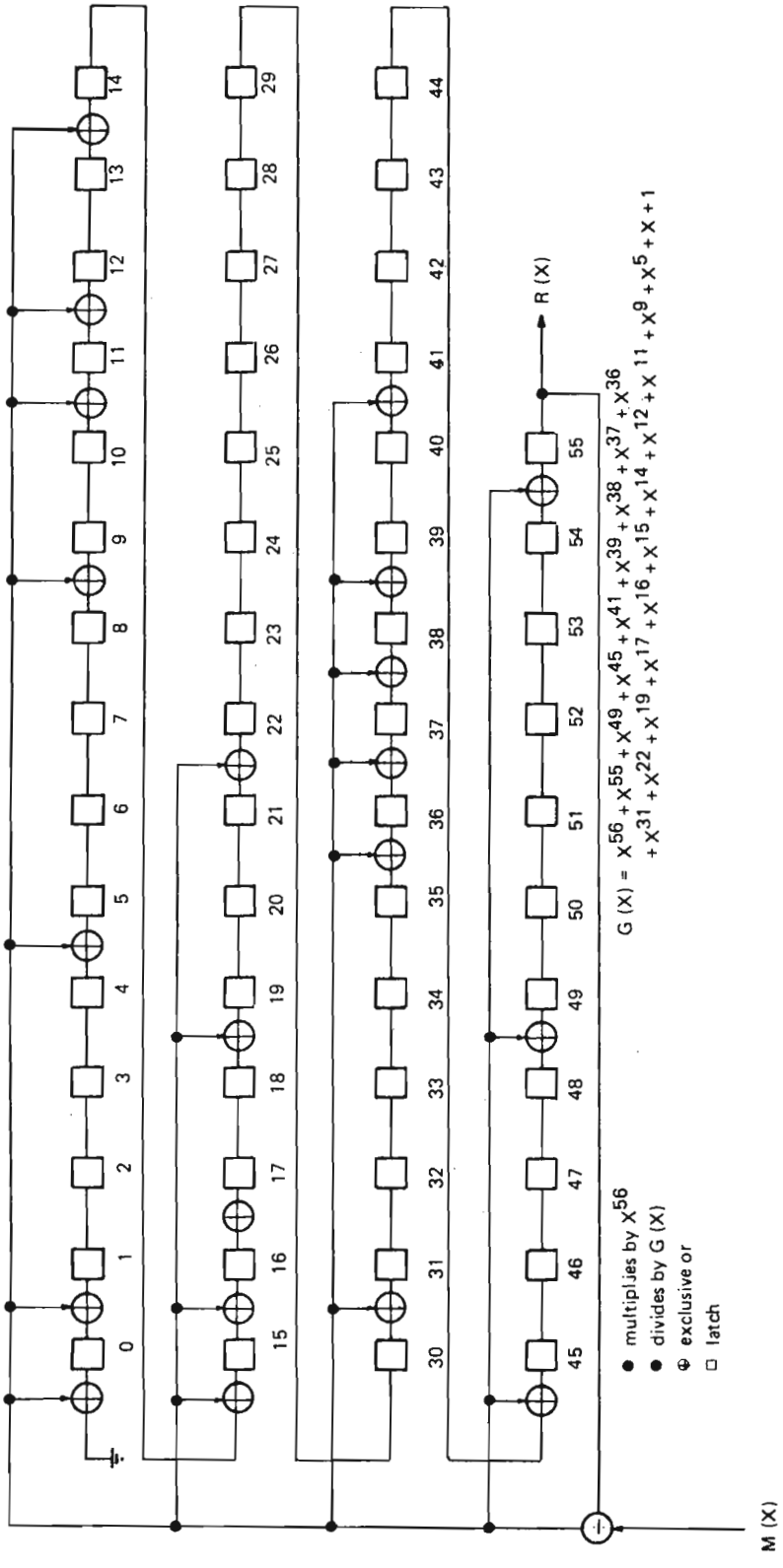


Figure J.1: 56 Bit Encode (Write)

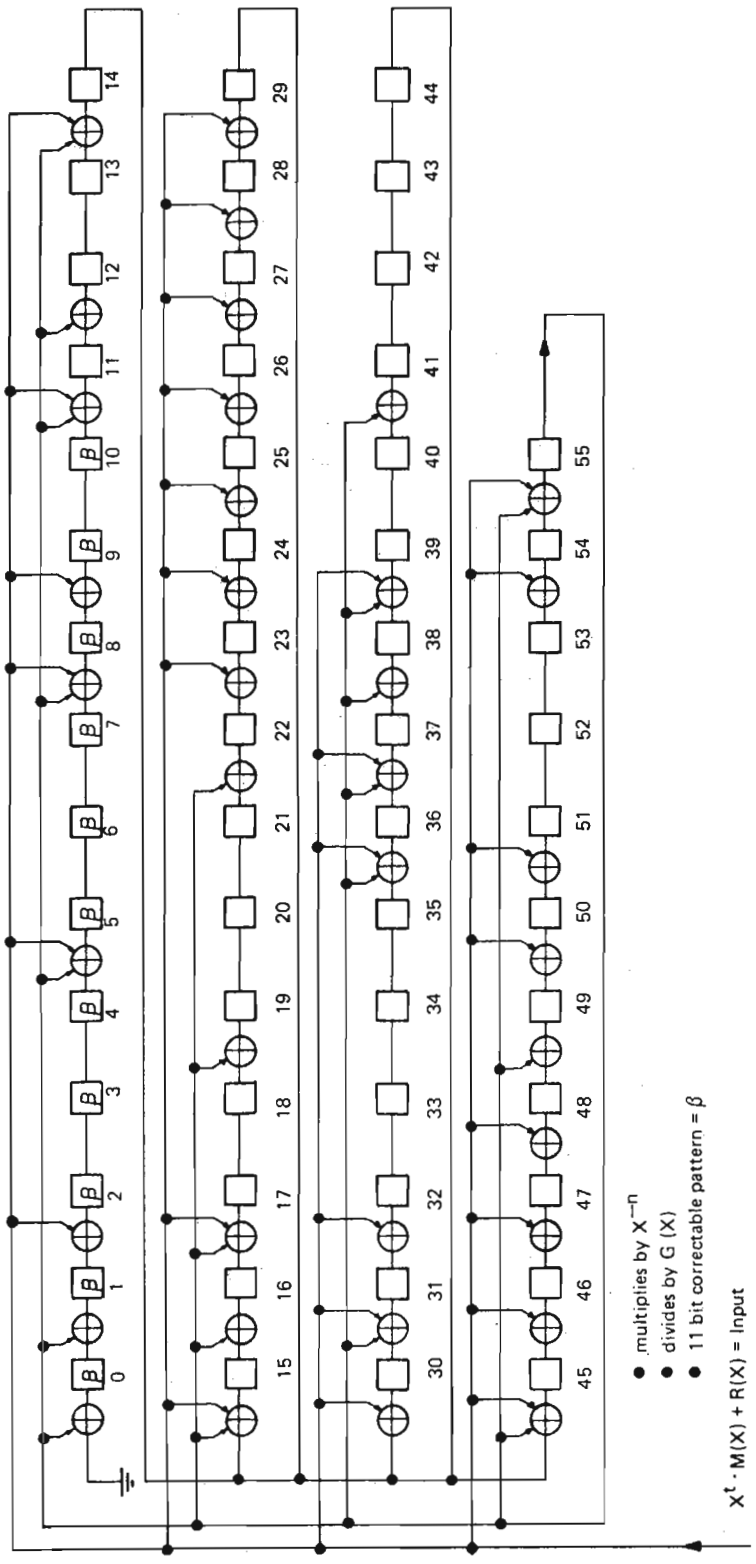


Figure J.2: 56 Bit Decode (Read)

APPENDIX K

TEST PROGRAMS

The following test programs are available:

- PASCAN
- Super-Rand
- ECC Test
- Bigfunc

K.1 *PASCAN TEST PROGRAM — 2226*

This is a stand-alone "PACK-SCAN" program for the disk controller with ECC. The program will sequentially read through the entire pack and report "hard" and "soft" errors.

"Hard" errors are defined as any errors reported in the status word of the controller *except* for a correctable error in the read data which is termed "soft" error (i.e., recoverable through the use of the ECC system).

The intended primary use of the program is for Pack surface analysis.

There is an option to be specified prior to running of the program:

> "Address and Data" means that *all* address fields and data fields are read and verified.

"Data only" means that all data fields, but *not* all address fields within each track are read and verified.

Error Reporting:

Hard Error Displays:

1. the current logical (octal) sector address
2. the controller status register

Soft Error Displays:

1. the current logical (octal) sector address
2. the address in main memory where error correction was applied
3. the two error correction pattern words to be exclusively or'ed with data at the main memory address to correct the data

When an error has been encountered and reported, the test will continue the scan until the entire pack is read and "Scan Completed" will then be reported.

K.2 *SUPER-RAND TEST PROGRAM — 2222*

This is a stand-alone, random address, random data, controller and disk read/write test. The disk addresses and write data are generated by a pseudo-random number generator.

Example of Operation:

1. At disk address A, a random data pattern (i) is written from main memory.
2. At disk address B, a random data pattern (j) is written from main memory.
3. The written data at disk address A is read back and verified (against i).
4. The written data at disk address B is read back and verified (against j).

and the process continues in the outlined fashion.

Errors and data mismatches are reported as specified by the test at run-time.

Options to be specified at run-time:

Retries active? when retries are specified, errors are not reported if they are recoverable by the retry and recovery procedure of the test.

ECC active? If active, *correctable* data errors are not reported.

RT clock? will print out the time of day value associated with each error report.

The test runs continuously and two disk addresses can be specified and the test will then run simultaneously against both addresses.

K.3 *ECC TEST PROGRAM — 2224*

This is a diagnostic program for the disk controller with ECC. The test will do read and write functions to test the functions of the error correction circuitry and it is therefore required that an on-line drive with a disk pack is attached to the controller. The reading and writing is done on a track that is not used by the SINTRAN III Operating System, hence the test does not require a special scratch pack mounted.

The test will completely diagnose, the 1133 pcb of the controller and associated control circuitry on other pcb's. Data records with no errors, correctable errors and uncorrectable errors are written and read-back-verified. The process is repeated many times varying the error-pattern and its displacement within the data record.

RUN Control and Error Reporting:

The test can be run once, or looped. When in loop mode and errors occur, the test will abort and start from the beginning of the test again.

Errors are reported as they occur and a brief description is displayed with status word information.

K.4 *BIGFUNC TEST PROGRAM — 1824*

This is a stand-alone test program, intended to test the disk status word and some of the disk operatoins. Reading and writing are done on a track not used by the SINTRAN III operating system but it might still be wise to use a scratch pack when running BIGFUNC.

The program is supposed to be self-documenting. The following tests are done.

1. The core address register is written and read 131072 times.
2. Each of the block address registers are written and read 131072 times.
3. Data is read from the interface in test mode. The status word, the data read and the core address register are checked. Word count is in the range 1-2000B.

The status bits are then checked 8 times, in different sequences:

4. Status bit 0 is loaded and read twice.
5. Status bit 1 is loaded and read twice.
6. Status bit 2 is check after device clear and read.
7. Status bit 5 is checked by loading the core address register, the block address register I, the word count register and the control word when a parity check operation is active, and by loading block address register I when the disk arm is not on-cylinder.
8. Status bit 6 is checked by doing a short parity check and a very long formatting operation (the track is formatted 48 times).
9. Status bit 7 is checked by reading from a non-specified unit (see below). The reading is done at most 8 times.
10. Status bit 8 is checked by formatting a track with incorrect format data, and reading it back.
11. Status bit 9 is checked by reading with word count 1004B and bit 2 in ECC control loaded (long bit).
12. Status bit 10 is checked by doing read, compare, change one bit in the disk buffer, compare.
13. Status bit 11 is checked by doing read with the instructions IOX 0 three times in the waiting loop, and by doing write with the instråuction IOX 0 twice in the waiting loop.
14. Status bit 13 is checked by selecting specified units and by reading from non specified units (see below).
15. Status bit 14 is checked by doing return-to-zero-seek and initiate-seek.
16. Status bit 15 is loaded and read twice. Status bits 3, 4 and 12 are not checked separately. All units not specified to be tested should be turned off (stop the disk pack, turn off power in the back of the disk unit).

17. Read and write are checked by reading to and writing from different buffers. Seek-complete-search is checked with no previous seek.

18. Read-seek-condition is checked by doing return-to-zero-seek, initiate-seek and by reading from an illegal block address.

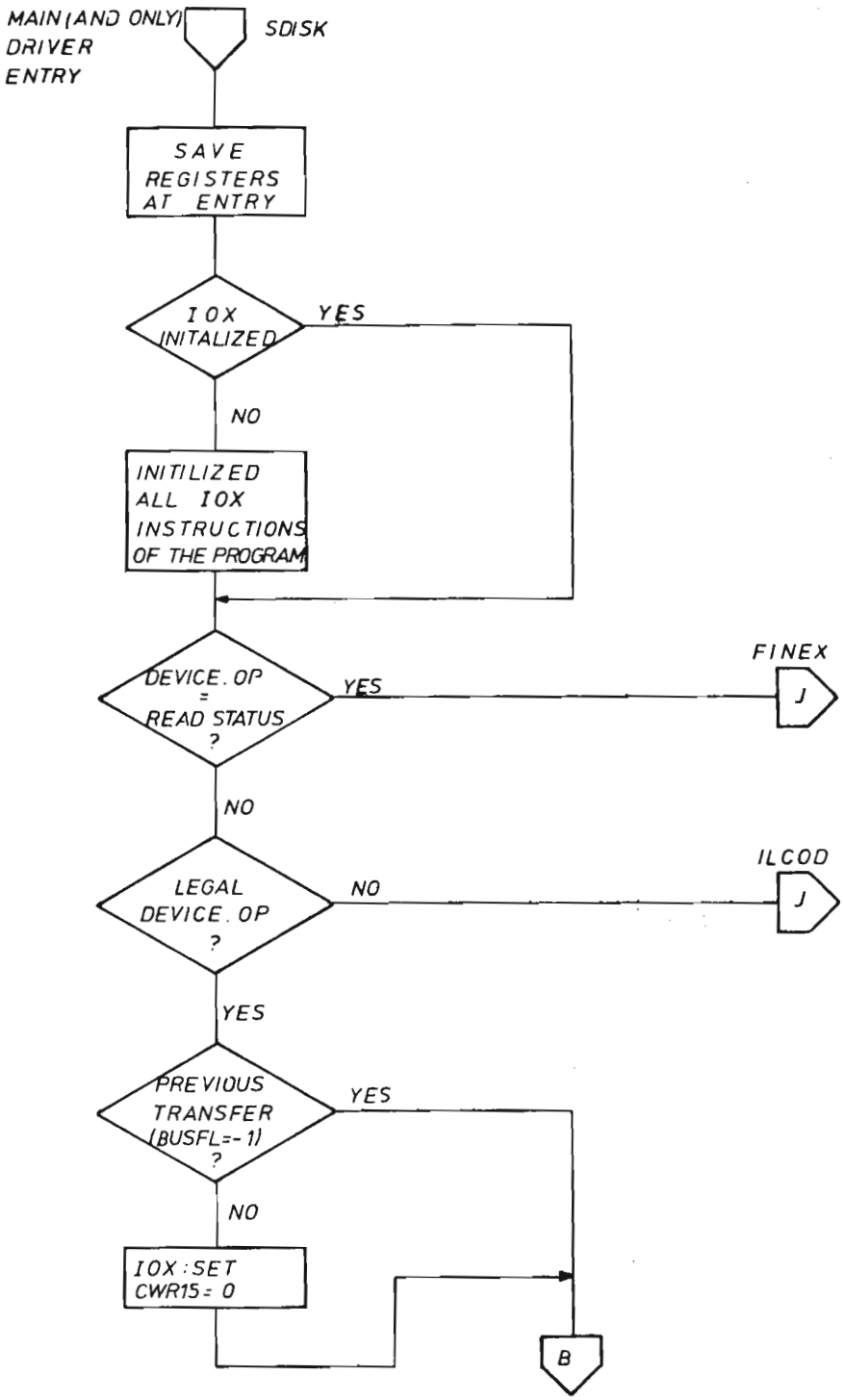
The test repeats itself indefinitely.

All errors will be reported by error messages on the terminal.

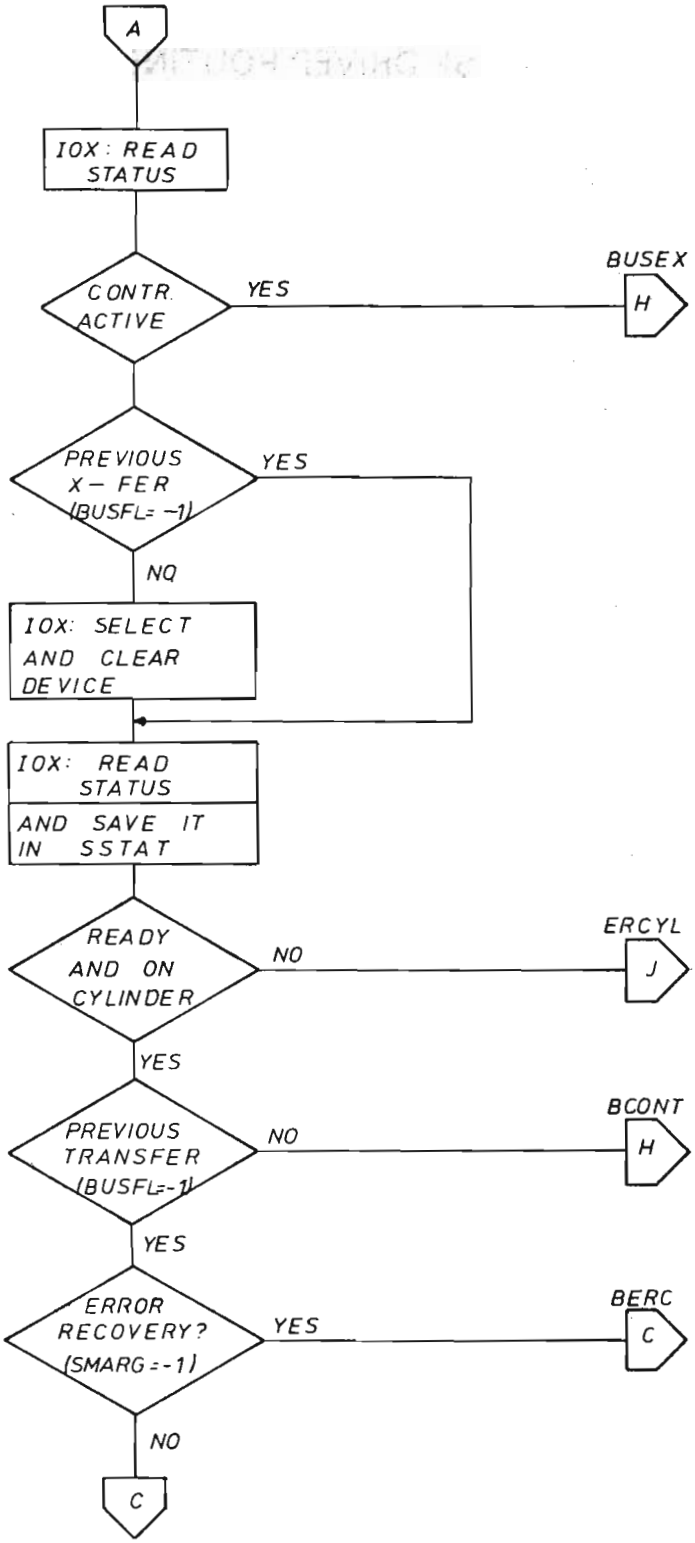
APPENDIX L

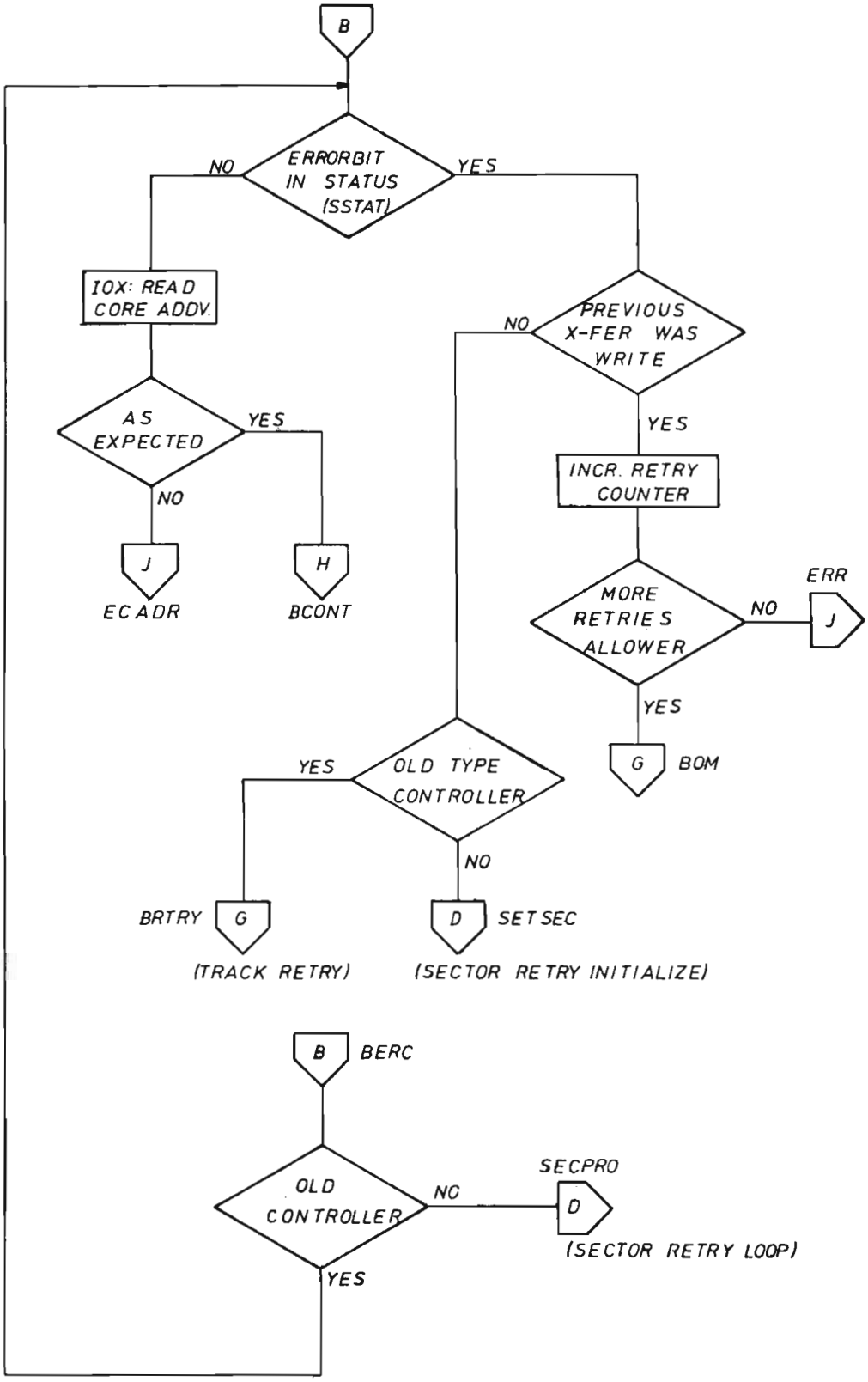
SINTRAN III — SMD DISK DRIVER ROUTINE

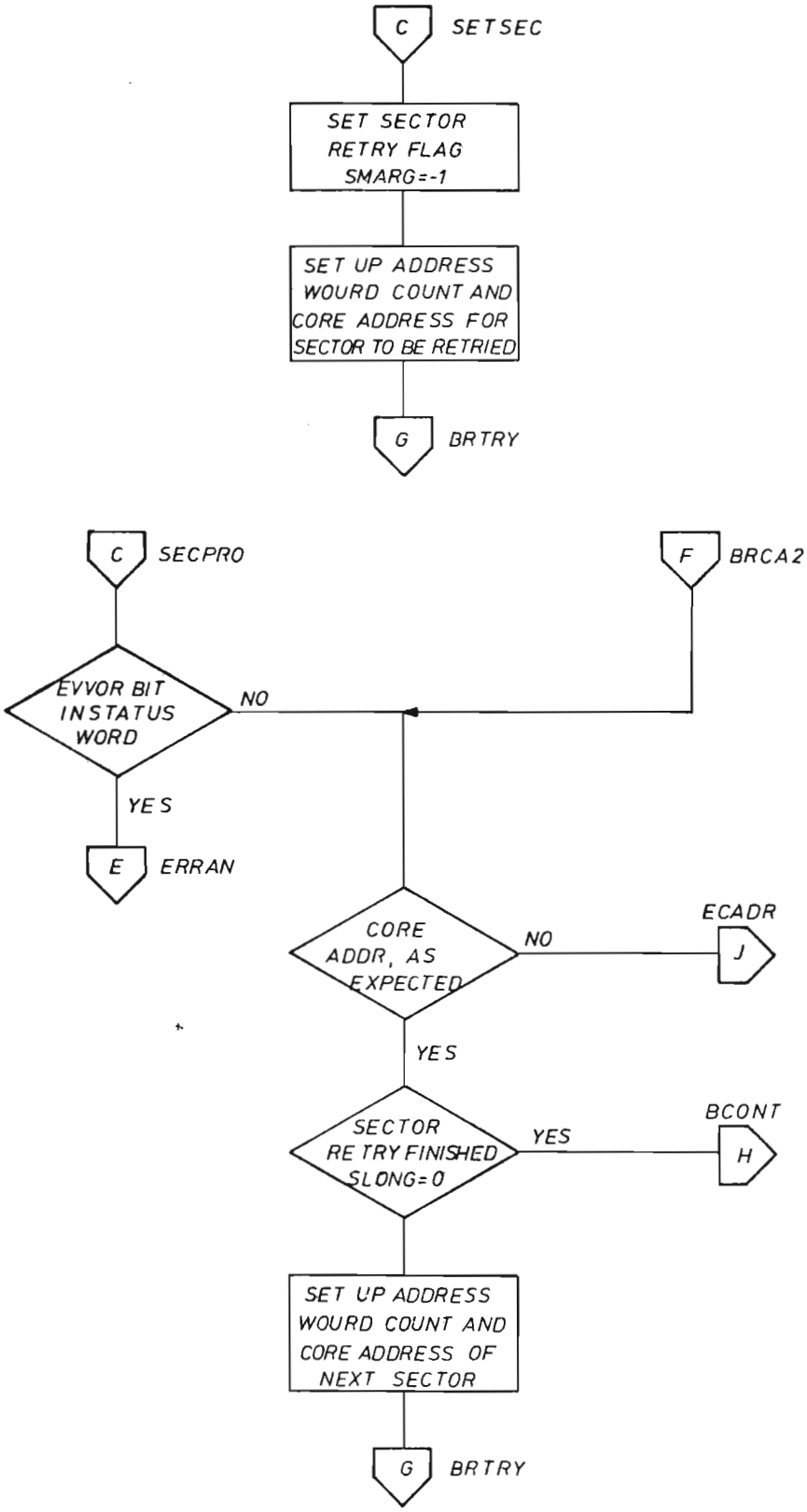
Main (and only) Driver Entry:

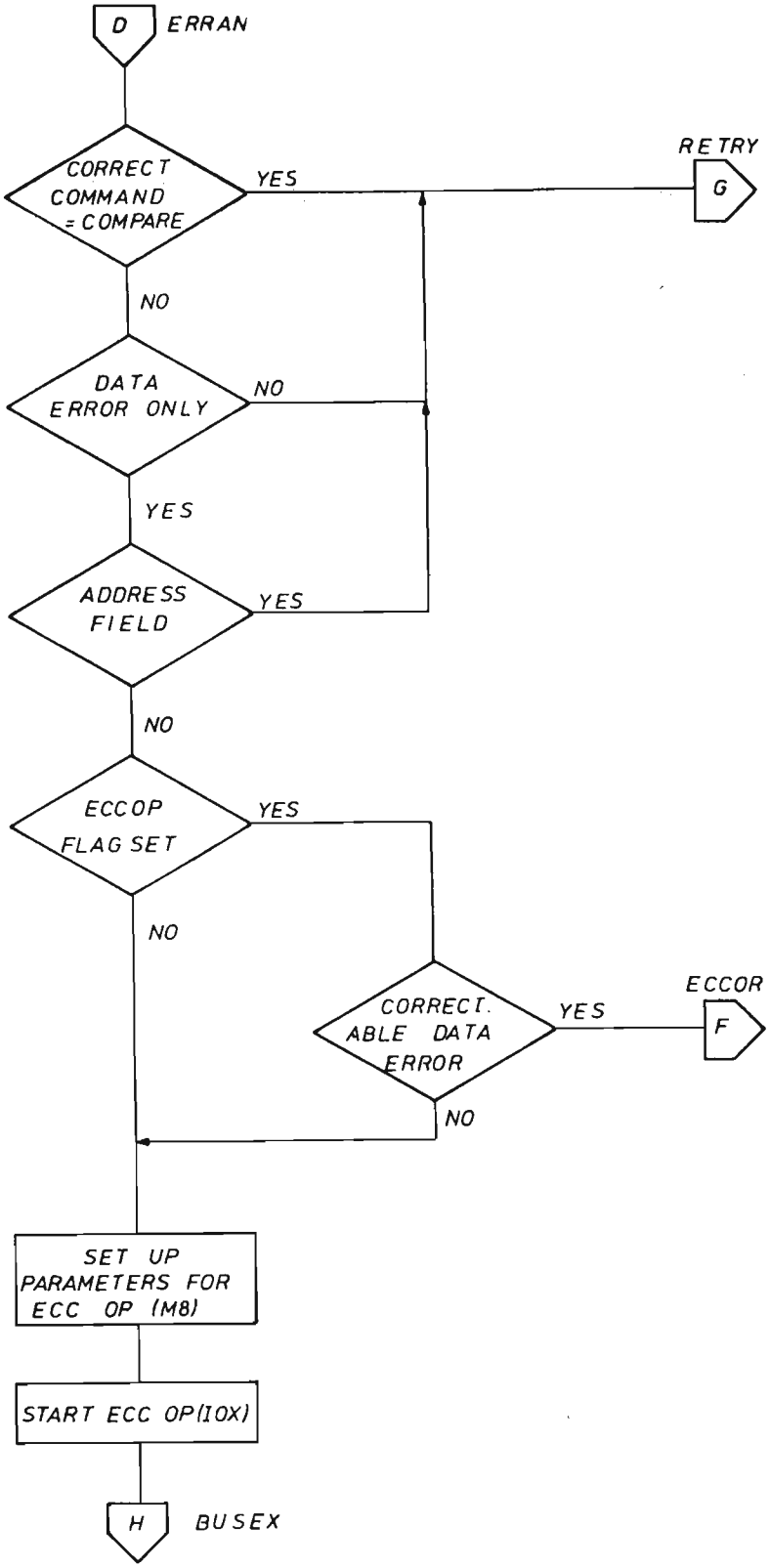


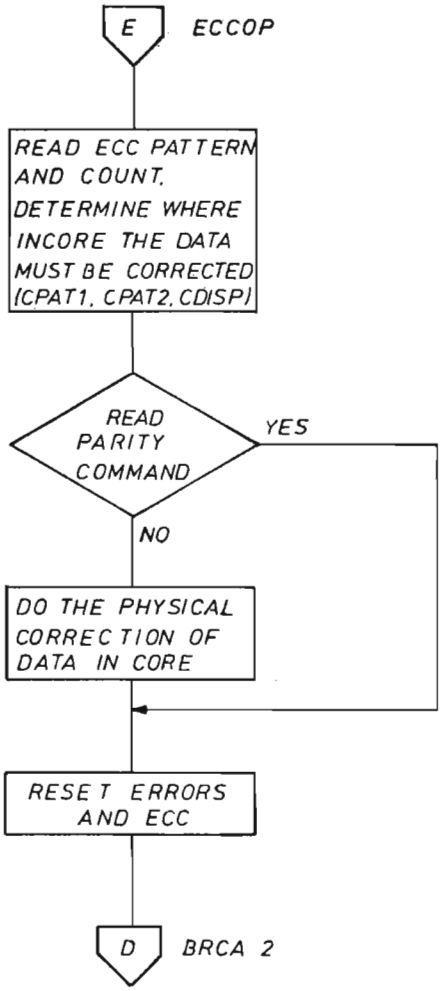
ND-11.013.01

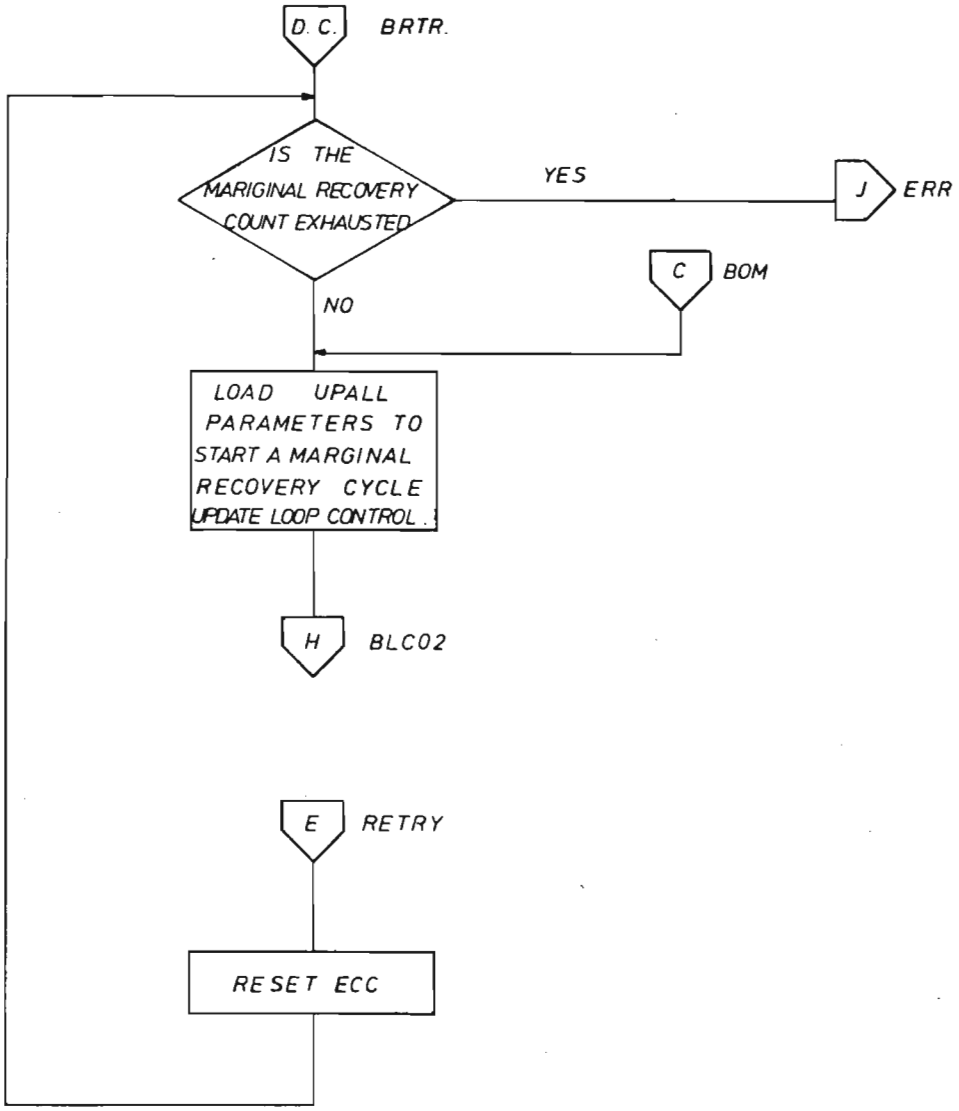


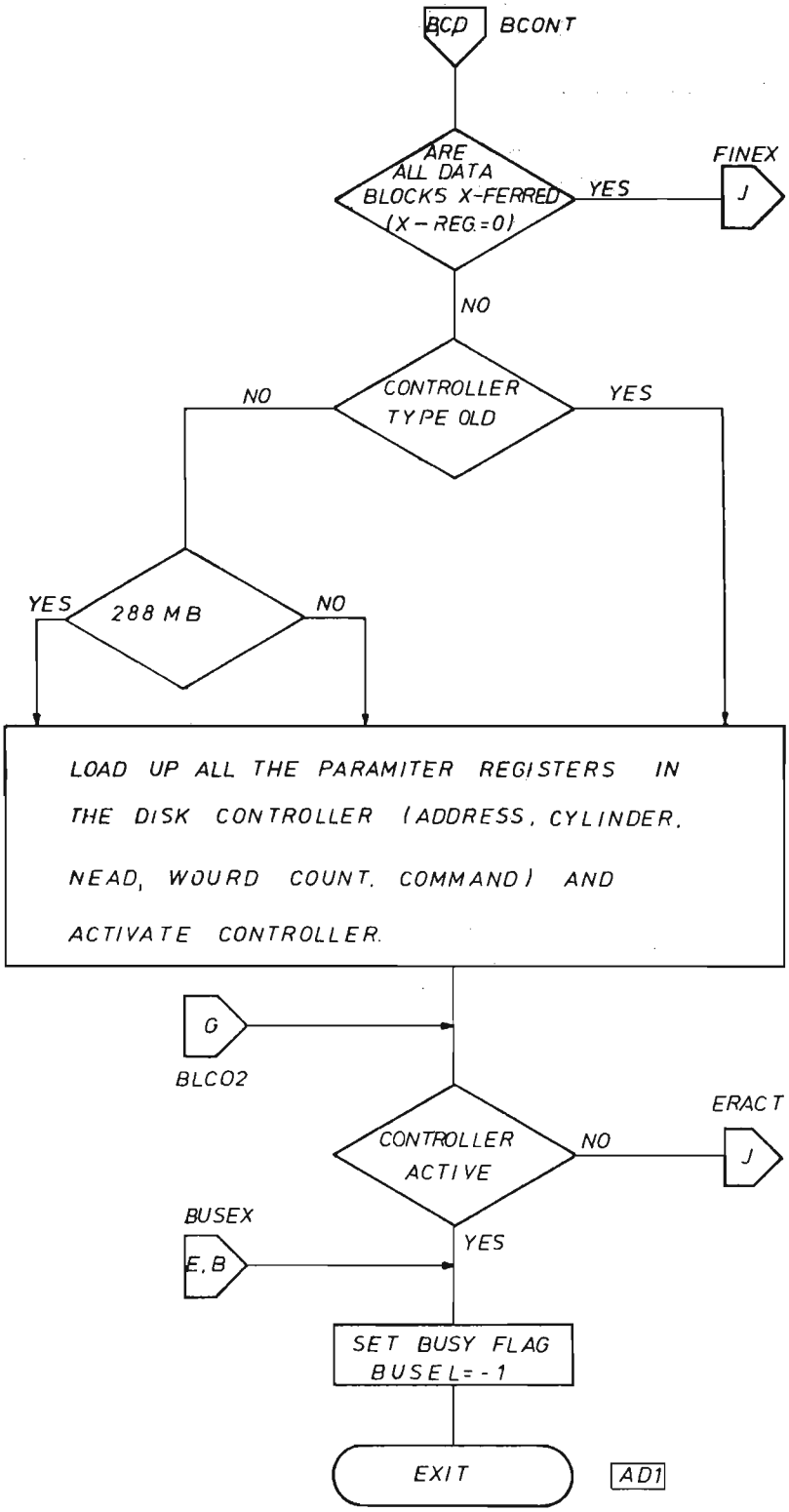


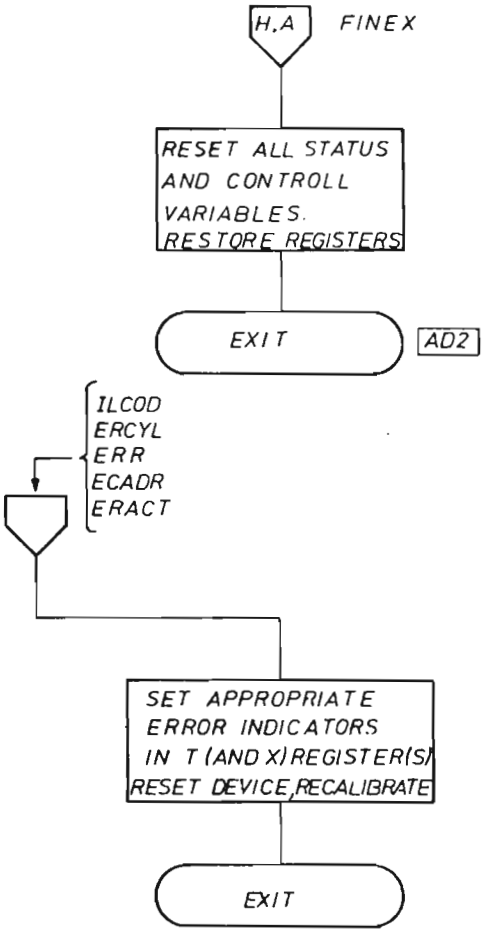












APPENDIX M

DISK SPECIFICATIONS

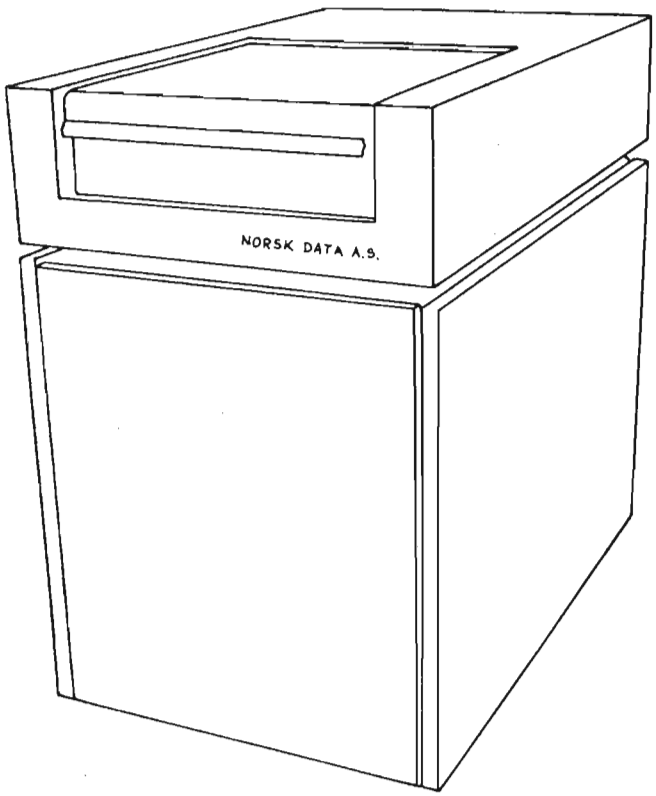
ND 574 — 288 Mbytes Disk Unit:

Maximum number of units (ND574)/controller (ND558): 4.

Size:	Height: 91.4 cm Width: 58.4 cm Depth: 91.4 cm
Weight:	252 kg.
Temperature:	Operating: 155.5°C to 32°C. Gradient: 6.6°C/hour.
Humidity:	Operating: 20% to 80%. No condensation.
Altitude:	Operating: —305 m to 2000 m.
Power Requirement:	Voltage: 220 VAC +7 —14. Effect: Operating — 1300w; Standby — 500w. Frequency: 50Hz + 0.5 — 1.0.
Spindle Speed:	3600 r/minute.
Seek Characteristics:	Mechanism: voice coil, driver by servo loop. Maximum seek time: 55 ms. One track seek time: 6 ms. Average seek time: 30 ms.
Latency:*	Average: 8.3 ms. Maximum: 16.6 ms.

*Values given for 3600 r/minute spindle speed. Latency time is defined to be the time required to reach a specified sector after drive is on-cylinder (seek completed).

Disk Pack:	Type: ND575 or equivalent. Disks/pack: 12 (top and bottom for protection only) Data surfaces: 19. Servo surfaces 1. Data tracks/surface: 823. Tracks/cm: 151 Sectors/track: 18
Data Capacity (formatted):	Sector: 512 words (1/2 K words) Track: 9.216 words (9 K words) Cylinder: 175.104 words (175 K words) Disk Pack: 144.110.592 words (144 M words) = 288 M bytes.
Transfer Rate:	Bit rate: 9.677 M bits/s Word rate: 605 K words/s
Recording:	Mode: Modified Frequency Modulation (MFM) Bit Density: 1590 bits/cm for outer track Bit Density: 2377 bits/cm for inner track



ND574 288 MBYTES DISK UNIT

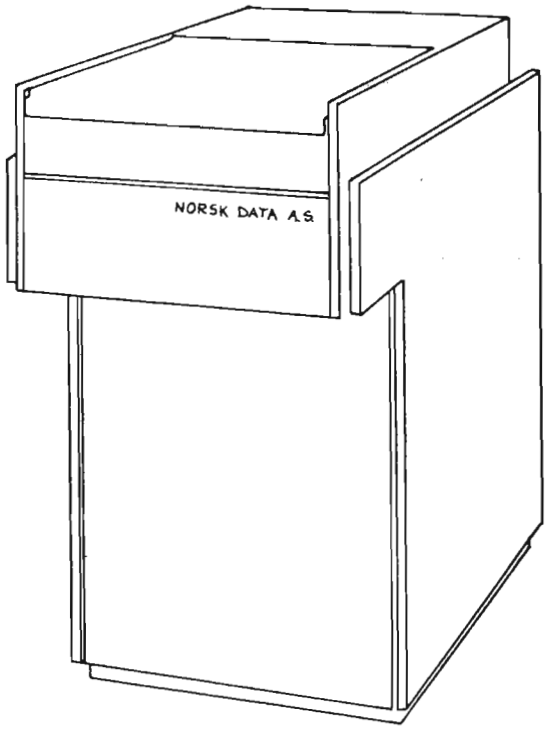
ND 576 — 37 Mbytes Disk Unit and ND 572 — 75 Mbytes Disk Unit

Maximum number of units (ND576) or (ND572)/controller (ND558): 4.

Size:	Height: 86.4 cm Width: 48.9 cm Depth: 85.1 cm
Weight:	110.1 kg.
Temperature:	Operating: 15.5°C to 32°C. Gradient: 6.6°C/hour.
Humidity:	Operating: 20% to 80%. No condensation.
Altitude:	Operating: —305 m to 3050 m.
Power Requirements:	Voltage: 220 VAC + 15 —25. Effect: Operating — 700w; Standby — 300w. Frequency: 50 Hz + 0.5 — 1.0.
Spindle Speed:	3600 r/minute.
Seek Characteristics:	Mechanism: voice coil, driven by servo loop. Maximum seek time: 55 ms. One track seek time: 7 ms. Average seek time: 30 ms.
Latency:*	Average: 8.3 ms Maximum: 16.6 ms.

*Values given for 3600 r/minute spindle speed. Latency time is defined to be the time required to reach a specified sector after drive is on-cylinder (seek completed).

Disk Pack:	Type: 37 Mbytes: ND577. Type: 75 Mbytes: ND573 or equivalent. Disks/pack: 5 (top and bottom for protection only) Data surfaces: 5. Servo surface: 1. Data tracks/surface: 37 Mbytes: 411 — 75 Mbytes — 823. Tracks/inch: 37 Mbytes — 75.6; 75 M bytes — 151. Sectors/track: 18.
Data Capacity (formatted):	Sector: 512 words (1/2 K) Track: 9.216 words (9 K) Cylinder: 46.080 words (46 K) Disk pack: ND576; 18.938.880 words — 37.877.760 bytes. Disk pack: ND572; 37.923.840 words — 75.847.680 bytes.
Transfer Rate:	Bit rate: 9.677 M bits/s. Word rate: 605 K words/s.
Recording:	Mode: Modified Frequency Modulation (MFM) Bit density: 1590 bits/cm for outer track. Bit density: 2377 bits/cm for inner track.



ND576 37 MBYTE DISK UNIT
ND572 75 MBYTE DISK UNIT

APPENDIX N

ECC DISK CONTROLLER PROGRAMMING SPECIFICATIONS

N.1 *DISK DEVICE REGISTER ADDRESS*

The IOX instruction can address two banks of registers. Which bank is being addressed is controlled by bit 15 of the Control Word Register (CWR).

The codes below are relevant for Disk System I. Each disk system may consist of 4 disk units. For Disk System II, add 10_8 to the specified codes.

	CWR bit 15 = 0	CWR bit 15 = 1
IOX 1540:	READ CORE ADDRESS	READ CORE ADDRESS
IOX 1541:	LOAD CORE ADDRESS	LOAD CORE ADDRESS
IOX 1542:	READ SEEK CONDITION	READ ECC COUNT
IOX 1543:	LOAD BLOCK ADDR I	LOAD BLOCK ADDR II
IOX 1544:	READ STATUS REGISTER	READ ECC PATTERN
IOX 1545:	LOAD CONTROL WORD	LOAD CONTROL WORD
IOX 1547:	LOAD WORD COUNT	LOAD ECC CONTROL

Each transfer is limited to one track (18 sectors) of data.

IOX 1546:	READ BLOCK ADDRESS I	READ BLOCK ADDRESS II

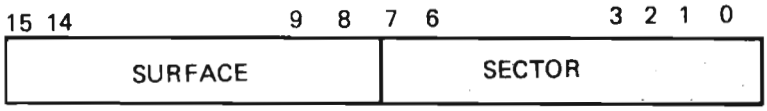
This instruction is implemented for maintenance purposes only. By first loading, a control word with bit 3 (Test Mode), this instruction will return the previously loaded block address to the A register.

N.2 *DISK FORMAT*

N.2.1 *Disk Address*

There are two block address registers that both have to be loaded to completely specify a disk address. The formats are:

Block Address Register I:



Bits 0-7: Sector number, 18 per track (0-17)
Bits 8-15: Surface number;
— for 38/75 Mbytes disk, 5 maximum (0-4)
— for 288 Mbytes disk, 19 maximum (0-18)

Block Address Register II:

Bits 0-15: Cylinder number;
— for 38 Mbytes disk — 411 maximum.
— for 75/288 Mbytes disk — 823 maximum.

N.3 *CONTROL WORD*

N.3.1 *Control Word Content*

Bit:

0	Enable interrupt on device not active
1	Enable interrupt on errors
3	Test mode
4	Device clear (clear the active flip-flop) and controller error bits
5	Address bit 16 } Extension of core address register
6	Address bit 17
7-9	Unit select (maximum 4 units)
10	Marginal recovery cycle
11-14	Device operation code
15	Register multiplex bit

N.3.2 *Select Unit*

When a control word is loaded, the disk unit number (0-7) has to be set up in bits 7-9.

N.3.3 *Marginal Recovery Cycle*

The marginal recovery cycle (control word bit 10) may be used in connection with read operation codes M0, M2 and M3 as defined in Section N.3.4. These control bits are included to be an aid in recovering marginal data. For consecutive read transfers with this bit set the controller will cycle through the following conditions:

1 marginal read: Servo offset positive, data strobe early
2 marginal read: No servo offset, data strobe early
3 marginal read: Servo offset negative, data strobe early
4 marginal read: Servo offset positive, nominal data strobe
5 marginal read: Servo offset negative, nominal data strobe
6 marginal read: Servo offset positive, data strobe late
7 marginal read: No servo offset, data strobe late
8 marginal read: Servo offset negative, data strobe late
9 = 1, etc.

N.3.4 *Device Operation*

All device operation codes will be activated when the code is given together with bit 2 (activate device). For all codes except M6, the correct unit number must also be selected.

Bit 14	13	12	11		
0	0	0	0	M0	Read Transfer
0	0	0	1	M1	Write Transfer
0	0	1	0	M2	Read Parity Transfer
0	0	1	1	M3	Compare transfer
0	1	0	0	M4	Initiate Seek
0	1	0	1	M5	Write Format
0	1	1	0	M6	Seek Complete Search
0	1	1	1	M7	Return to Zero Seek
1	0	0	0	M8	Run ECC Operation

M0 Read Transfer

This operation causes the controller to transfer data from the disk to the computer memory. The number of blocks transferred depends upon the word count as defined by the word count register.

M1 Write Transfer

Transfer of data from the computer memory to the disk.

M2 Read Parity Transfer

The controller will check the parity on the *address* and *data* of the sectors specified. Data is transferred to the controller and the check word for both the address field and the data field of a sector is compared with the correct check word as generated by the controller. No data transfer to the computer memory is performed.

M3 Compare Transfer

This function is included to positively check the data written on the disk. During compare transfer the controller compares the data read from the disk and data from the computer memory is compared bit by bit. Mismatch causes compare error to be set.

M4 Initiate Seek

This function is included to enable a unit to position the heads prior to a data transfer. The heads will be positioned according to the contents of the Block Address register. As soon as this function is accepted by the disk, the operation will be completed.

M5 Write Format

Together with a switch on a card in the interface set, this function will cause the controller to write the address field within each sector.

M6 Seek Complete Search

This function will enable the controller to go in a waiting state until any unit has completed a seek. This function is independent of the unit select code in the control word.

M7 Return to Zero Seek

This will cause the selected disk to perform a seek to cylinder 0 and will also clear the seek error bit in the unit.

M8 Run ECC Operation

This function will, when a data error has occurred, initiate the hardware operation that determines if the error is correctable or uncorrectable. If the error is correctable, the error pattern and its displacement within the data field is computed.

N.4 *READ SEEK CONDITION*

Bits 0-7: Seek Complete

Seek complete status for untis 0-7. True if the unit has moved the heads to the correct cylinder or a seek error has occurred and the heads are under the sector number prior to the one specified by the block address loaded before the initiate seek commands for that unit has first been issued.

Thus, after an initiate seek command is given, the Seek Complete bit for that unit will appear once per revolution after the unit is positioned on the correct cylinder, or a seek error has occurred. The condition will last until a transfer command is given.

Bits 8-10: Unit Select

The unit number as loaded by the last control word.

Bit 11: Seek Error

Seek error for the selected unit. This signal indicates that the unit was unable to complete a move within 500 ms, or that the heads have moved to a position outside the recording field, or that an address greater than the maximum number of tracks has been selected.

This signal will only be cleared by performing a Return to Zero command on the unit.

Bit 12: Not defined.

Bit 13: ECC Correctable

After the hardware ECC operation M8 has been performed after a data error, this bit signals that the error is correctable and that the ECC Count and ECC Pattern Registers contain valid information for correction of the data. The bit is reset by Reset ECC (ECC Control register bit 0) or Device Clear.

Bit 14: ECC Parity Error (STS bit no. 7)

This bit signals that a hardware fault condition exists in the ECC polynomials. This condition will also set bit 7 of the status word register and hence trigger an error interrupt if this is enabled. The error is reset by the Reset ECC signal (ECC Control register bit 0) or by Device Clear Signal (CWR bit 4). The error is forced set when ECC Control Register bit 1 is active (Force Parity Error).

Bit 15: Address Field

This bit indicates that the last field read from the disk was the address field within a sector (used for ECC processing after a data check only).

N.5 *READ STATUS*

Status Word:

Bit 0:	Controller not active interrupt enabled
Bit 1:	Error interrupt enabled
Bit 2:	Controller active
Bit 3:	Controller finished with a device operation
Bit 4:	Inclusive OR of errors (bits 5-13)
Bit 5:	Illegal load, i.e., load while status bit 2 is true or load of block address while the unit is not on cylinder
Bit 6:	Timeout
Bit 7:	Hardware error (disk fault + missing clocks + missing servo clocks + ECC parity error)
Bit 8:	Address mismatch
Bit 9:	Data error
Bit 10:	Compare error
Bit 11:	DMA channel error
Bit 12:	Abnormal completion
Bit 13:	Disk unit not ready
Bit 14:	On cylinder
Bit 15:	Register multiplex bit (from CWR bit 15)

N.6 *ECC COUNT REGISTER (ECR)*

When a correctable data error has been detected, this register will contain the bit displacement from the beginning of the data field to the last bit in error of the error burst.

N.7 *ECC PATTERN REGISTER (EPR)*

Bits 0-10:	Contain the RIGHT justified error pattern, such that the last bit in error always occupies bit position 0 of this register. This pattern (the contents of this register bits 0-10) should be exclusively OR'ed with the data in the CPU memory at the proper location.
Bits 11-14:	Set to logical "one".
Bit 15:	Register Multiplex bit (from CWR bit 15)

N.8 *ECC CONTROL*

Bit 0: Reset ECC

This bit will cause the ECC polynomials to reset to the zero initial state. This function is only used when a data error has occurred, otherwise the polynomials automatically go to the zero state upon completion of a Read or Write. Device Clear function will also reset ECC.

Bit 1: TST — Force Parity Error

Used for maintenance purposes only. This bit will force ECC parity error to be set.

Bit 2: Long

Used for maintenance purposes only. When a sector is read or written, the data field of the sector is extended by 64 bits (the length of the ECC appendage plus "end of record" byte). The data and the extra bits are read into or written from the memory of the CPU. This function is used to diagnose the operation of the ECC circuits and can be used with the following Device Operations: M0, M1, M2 and M3.

This bit is "echoed" in ECR bit 14.

NORSK DATA A.S
Postboks 4, Lindeberg gård
Oslo 10, Norway

COMMENT AND EVALUATION SHEET

ERROR CORRECTION CONTROL (ECC) DISK CONTROLLER
OCTOBER 1978 ND-11.013.01

In order for this manual to develop to the point where it best suits your needs, we must have your comments, corrections, suggestions for additions, etc. Please write down your comments on this pre-addressed form and post it. Please be specific wherever possible.

FROM _____

_____

_____

– we make bits for the future

NORSK DATA A.S BOX 4 LINDEBERG GARD OSLO 10 NORWAY PHONE: 39 16 01 TELEX: 18661