ST 157N MICROCODE CLASS

by Kumar Gajjar
11/19/87

01) Overview.

02) Microcode layout.

- a) ROM Code
- b) RAM Code
- c) SPECIAL TEST Code
- c) Misc.

03) Power-up sequence.

04) Functional code & Overlay code.

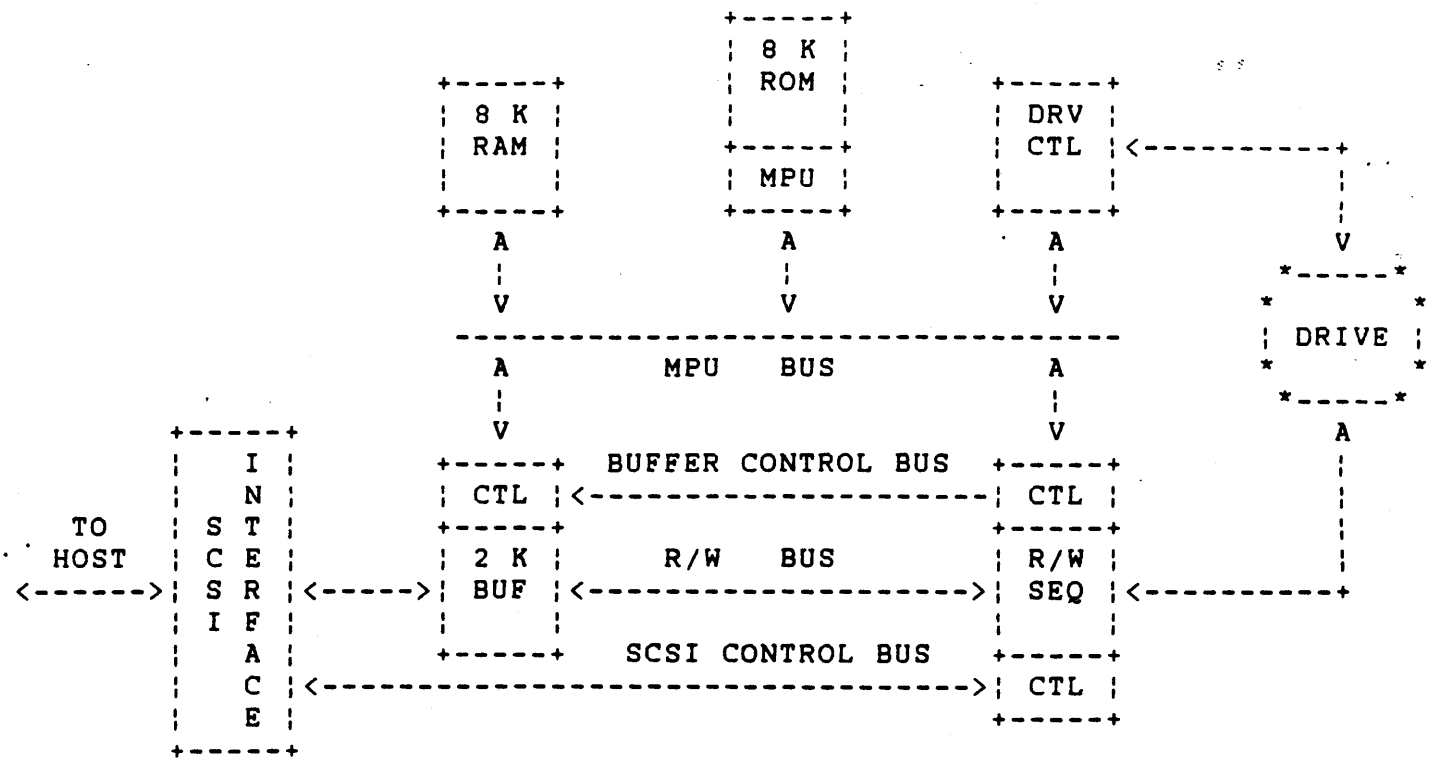
05) Typical Seek/Read operation.

06) Defect Management.

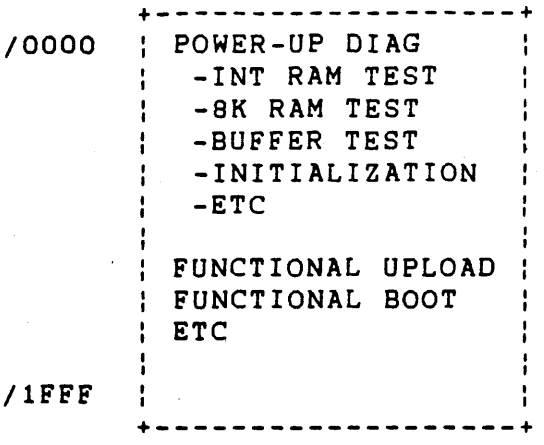
07) Extra features.

- a) Dynamic Tunning
- b) Cylinder Interleave
- c) Selftest

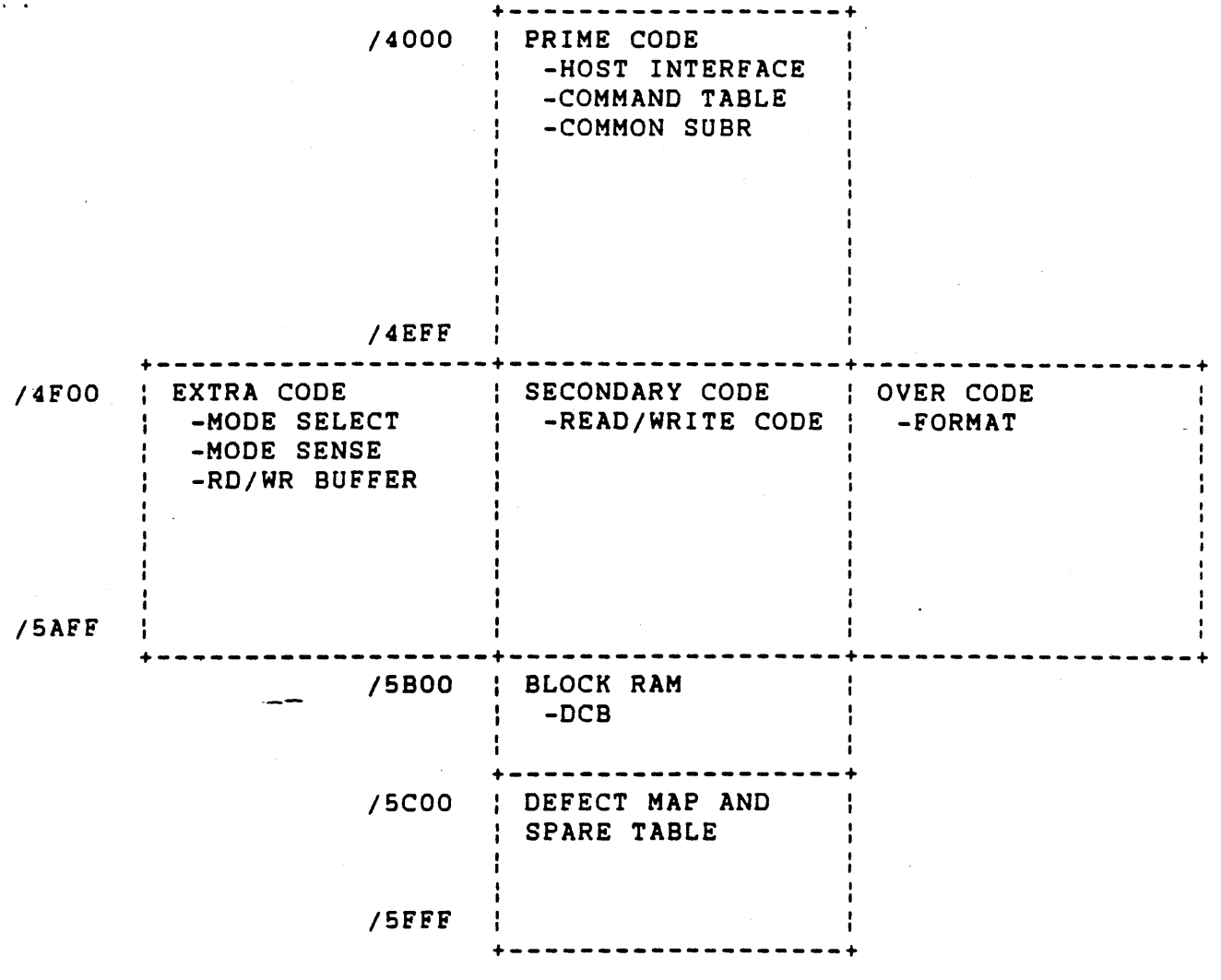
ST 157N HARDWARE BLOCK DIAGRAM



ROM MICRO-CODE LAYOUT



RAM MICRO-CODE LAYOUT



COMMANDS LAYOUT

* PRIMARY RAM CODE

  - SCSI protocol handler
  - Drive functions
  - Command kernel
  - Common subroutins
  - Commands handled: 00 Unit ready

* FIRST OVERLAY CODE (SECDCODE)

  - Commands handled:
    03 Request sense
    08 Read data
    0A Write data
    0B Seek
    28 Read extended
    2A Write extended
    2B Seek extended
    2F Verify
    E5 Read long
    E6 Write long
    E7 Set window margin

* SECOND OVERLAY CODE (OVERCODE)

  - Commands handled:
    01 Recalibration
    04 Format unit
    11 Read usage counter
    37 Read defect list

* THIRD OVERLAY CODE (XTRACODE)

  - Commands handled:
    07 Re-assign block
    12 Inquiry
    15 Mode select
    16 Reserve
    17 Release
    1A Mode sense
    1B Start/Stop
    1D Send diagnostics
    25 Read capacity
    3B Write buffer
    3C Read buffer

* RUN TIME PARAMETERS

  - Defect list
  - Available spare sectors map
  - Operating parameters

NEGETIVE TRACKS LAYOUT

The format of the negetive cylinders in ST157N is as follows

CYL	HD	SECTORS	OBJ FILE	CONTENTS
-2	0	00 - 15	POS1	Primary Operating System (PRIMCODE)
		16 - 32	POS2	+ Check sum bytes
-2	1	00 - 32		Back up copy #1
-1	0	00 - 32		Back up copy #2
-1	1	00 - 32		Back up copy #3
-2	2	00 - 15	SCH	Command Handler (OVERCODE)
-1	2	00 - 15		Back up copy
-2	2	16 - 31	ECH	Extra Command Handler (XTRACODE)
-1	2	16 - 31		Back up copy
-2	3	00		Drive Serial Number
-1	3	00		Back up copy
-2	3	01 - 04		Manufacture Defect map
-1	3	01 - 04		Back up copy
-2	3	05 - 15		Not used
-1	3	05 - 15		Not used
-2	3	16 - 31	STC	Special Test (SELFTEST)
-1	3	16 - 31		Back up copy

Note: Drive Parameters, Format Defect List, Reassignment Block Map & Spare Sector Map are stored in sectors 27 through 31 of Primary Operating System code, which are updated during Format.

POWER-UP SEQUENCE

01) Initialize Stack pointer and Microprocessor Ports.

02) Stiction jump and start Spindle motor.

03) Selftest.

04) Check if drive is ready.
    If not then go to step (13)

05) Check for DC erase jumper.
    If so then DC erase the Drive and go to step (13)

06) Check for LIFE test jumper.
    If so then do LIFE test and go to step (13)

07) Read ID.
    If valid then go to step (09)
        else Unpark drive

08) Read ID.
    If valid then go to step (09)
        else Softly recal to cylinder -2.

09) Seek to Cylinder 0 and then seek to -2.

10) Do Shoshine.

11) Upload RAM code from -ve tracks.
    If RAM is not valid then go to step (13)

12) Jump to RAM (functional) code.

13) Blink LED to show error status.

14) Idle in ROM operating system.

SELFTEST SEQUENCE

1) Test Microprocessors' Internal RAM.

2) Check ROM Check Sum.

3) Check and initialize Sequencer chip.

4) Check 8K RAM.

5) Check 2K Buffer.

6) Read Controller ID.

7) Initialize Buffer chip.

8) Inititalize Drive.

   a) Pre-heat Stepper Windings.
   b) Check if Index is present.
      If not then go to step (e)
   c) Check Spindle speed.
      If good then go to step (9)
   d) Stop Spindle Motor
      If first time here then wait 2 seconds and go to step (8.c) once only.
   e) Set Drive not ready and return to main program.

9) Check Parity jumper.

RAM-CODE UPLOAD PROCEDURE

1) Set retry count to 8.

2) Seek to cylinder -2.

3) Select head 0 and read Check sum sector (sector 32) and
   verify its check sum.
   If good then go to step (8).
           else decrement retry count.

4) Select head 1 and read Check sum sector (sector 32) and
   verify its check sum.
   If good then go to step (8).
           else decrement retry count.

5) Seek to cylinder -1.

6) Select head 0 and read Check sum sector (sector 32) and
   verify its check sum.
   If good then go to step (8)
           else decrement retry count.

7) Select head 1 and read Check sum sector (sector 32) and
   verify its check sum.
   If good then go to step (8).
            else decrement retry count.
                 if retry count = 0 then go to (16)
                                    else go to (2)

8) Save Check sum bytes in buffer.

9) Seek to cylinder -2, head 0 and set sector to 0.

10) Set retry count to 8.

11) Read sector and verify its check sum.
   If not good then go to step (13).
               else if sector = 0 then sync Index.

12) Increment sector number.
   If not last+1 then go to step (10).
                 else go to step (17).

13) Increment head number.
   If less than 2 then go to step (15).
                 else select head 0.

14) Decrement cylinder number.
   If equal to 0 then seek to cylinder -2
                 else seek to cylinder -1

15) Decrement retry count.
   If not equal to 0 then go to step (11).

16) Set RAM not valid error code and return to main program

17) Set RAM valid and return to main program

RAM CODE START SEQUENCE

01) Initialize Stack pointer.

02) Initialize Usage counter.

03) Reset sense bytes.

04) Set drive parameters.

05) Enable all interrupts.

06) Jump to wait (idle) loop.

WAIT (IDLE) LOOP SEQUENCE

01) Initialize Stack pointer.

02) Clean SCSI bus.

03) Enable all interrupts.

04) Update sense bytes.

05) Reset 10 minuite timer for Shoeshine/Dynamic tunning.

06) Check for selection.
    If selected then go to selection routine

07) Check if 10 min. timer is up
    If not then go to step (10).

08) Check Shoeshine/Dynamic tunning flag
    if Shoeshine then do Shoeshine
                 else do Dynamic tunning

09) Reset 10 minuite timer for Shoeshine/Dynamic tunning.

10) Check for command in progress.
    if so then call reselection routine and go to step (06).

11) Check for Special test jumper.
    if so then upload special test code off the disk and start test.

12) Check Spindle speed.
    if speed is good then set drive ready bit & go to step (06).
                     else set drive not ready bit.

13) If Drive ready bit is on then reset Drive ready bit,
                                  start 5 second timer &
                                  go to step (06).

14) If 5 second timer is up then Park heads,
                                 stop spindle motor,
                                 set Drive dead bit and
                                 go to step (06).

TYPICAL READ SEQUENCE

01) Initialize R/W Control registers, reset DMA pending & Buffer pointers.

02) Convert LBA to physical address. Set/reset flags if seek or head switching is required.

03) Go seek or switch heads if needed.

04) Start Read operation.

05) Check for DMA pending flag.
If DMA is pending then start DMA to Host.

06) Check if data transfer is active.
If not active then do ID error recovery.

07) Do some house keeping, i.e. increment sector number, adjust for defect of reassigned block and set up sequencer for next read operation. Set/reset flags if seek or heads switching is required.

08) Check if data transfer is still active.
If still active then wait for it to finish.

09) Check if DMA (data transfer to Host) is still active.
If still active then wait for it to finish.

09) Check for error during read operation.
If so then perform data error recovery.

10) Decrement transfer block count and set DMA pending flag.
if not last then go to step (04)

11) Start DMA of last block to Host.

READ RETRY PROCEDURE

RETRY NO.	ACTION
01 - 08	Re-read Sector with NO ECC Correction NO Micro-Stepping
	Do Shoeshine
09 - 24	Re-read Sector with NO ECC Correction 8 Micro-Stepping 2 Retries/Microstep
25 - 28	Re-read Sector with 5 Bit ECC Correction NO Micro-Stepping
29 - 32	Re-read Sector with 8 Bit ECC Correction NO Micro-Stepping
33 - 64	Re-read Sector with 8 Bit ECC Correction 8 Micro-Stepping 4 Retries/Microstep

ABSTRACT OF DEFECT MANAGEMENT FOR SEAGATE 157N

1. Four maps are used for defect management.

   1.1 Manufacturing defect list consists of Cylinder (2 bytes), head (1 byte) and bytes from index (2 bytes).

   1.2 Format defect list consists of Cylinder (2 bytes), head (1 byte) and sector (1 byte). This list is sorted in ascending order.

   1.3 Reassign block list consists of Old logical block address (LBA) (3 bytes) and new physical ID (2 bytes).

   1.4 Unavailable spare sector list. Each bit indicates the availability of one spare sector.

   1.5 Max no. of defects allowed can be calculated as follows.

   1022 bytes = 4 * manufacturing defects + 5 * reassigned blocks

   Ex. for drives with 60 manufacturing defects, additional 156 reassigned blocks are allowed.

2. The last sector of each cylinder is designed as spare for future reassignment. When reassignment occurs, the closet spare sector available is used.

3. ID header format (4 bytes) for vaious sectors.

Normal sector	LBA(H),	LBA(M),	LBA(L),	flag bytes
Defective sector	FF ,	FF ,	FF ,	FF
Spare sector	CYL(H),	CYL(M),	CYL(L),	FF

4. The drive is formatted skipping the last sector of each cylinder and defective sectors in format defect list. 2 additional cylinders are formatted to compensate the lost defective sectors.

5. The capacity of the drive is guaranteed even with maximum number of defects. The capacity is equal to

   (# Sectors/Track * # Tracks/Cyl - 1 Spare Sector) * # Cylinder

   for 157N (26 * 6 - 1) * 530 = 155 * 530 = 82150 Blocks

6. Reassign a block:

The physical location (cylinder) for the bad LBA is calculated. Then the unavailable spare sector list is searched. If available then an entry is created in the reassign block list and the spare is added to the unavailable spare sector list. If not available then try the next closest cylinder. The searching sequence always starts at current cylnder, then current cyl + 1, current cyl - 1 and so on.

7. Sequence of searching for target ID

7.1 Read LBA from ID header.

7.2 Translate LBA to physical cylinder, head and sector.

7.3 Do adjustment for manufacturing defects.

7.4 Check if the block is reassigned.

8. Format of Defect list as stored on -ve tracks.

Byte	Content	
00	B E	
01	E D	
02	# DEFECTS	HEADER
03	RSVD	
04	RSVD	
05	CYL #	
06	CYL #	
07	HD #	FIRST DEFECT
08	BYTES	
09	FROM IDX	
/	/	/
N	OFFH	

Example: ST 157NB has 530 cylinders, 6 heads and 26 sectors/track.

Assume there are 2 manufacturing defects. These are

cylinder 0, head 0, sector 1
cylinder 0, head 0, sector 3

and 5 sectors in cylinder 1 have been reassigned, the sequence is as follows.

LBA	00	01	00	is reassigned to	cyl	1,	head	5,	sector 25
LBA	00	01	05	"	"	"	2	"	"
LBA	00	01	09	"	"	"	0	"	"
LBA	00	01	03	"	"	"	3	"	"
LBA	00	01	07	"	"	"	4	"	"

1. Format defect list:

CYLH	CYLL	HD	SECT	CYLH	CYLL	HD	SECT	EOL
00	00	00	01	00	00	00	03	FF

Note: 4 Bytes are needed for each defective sector.

Assume the drive is formatted with 1 : 1 interleave, the relationship between LBA and physical address is as follows.

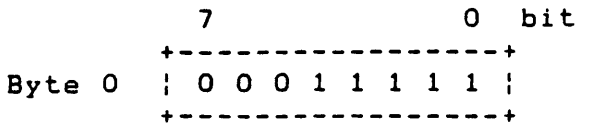
	Physical sector		LBA
Cylinder 0 head 0	0	--------	0
	1	--------	defect
	2	--------	1
	3	--------	defect
	4	--------	2

2. Reassigned block list:

LBA			SP	CYL
00	01	00	00	01
00	01	03	00	03
00	01	05	00	02
00	01	07	00	04
00	01	09	00	00

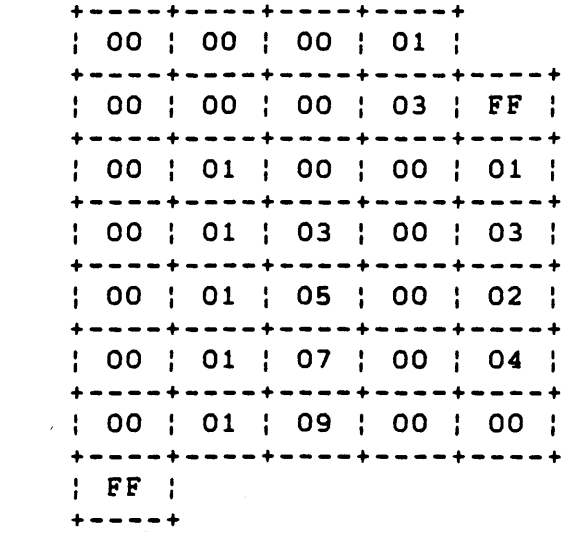
Note: 5 Bytes are needed for each reassigned sector.

3. The unavailable spare sectors list



"Spare" sectors in cylinder 0 thru 4 are unavailable.

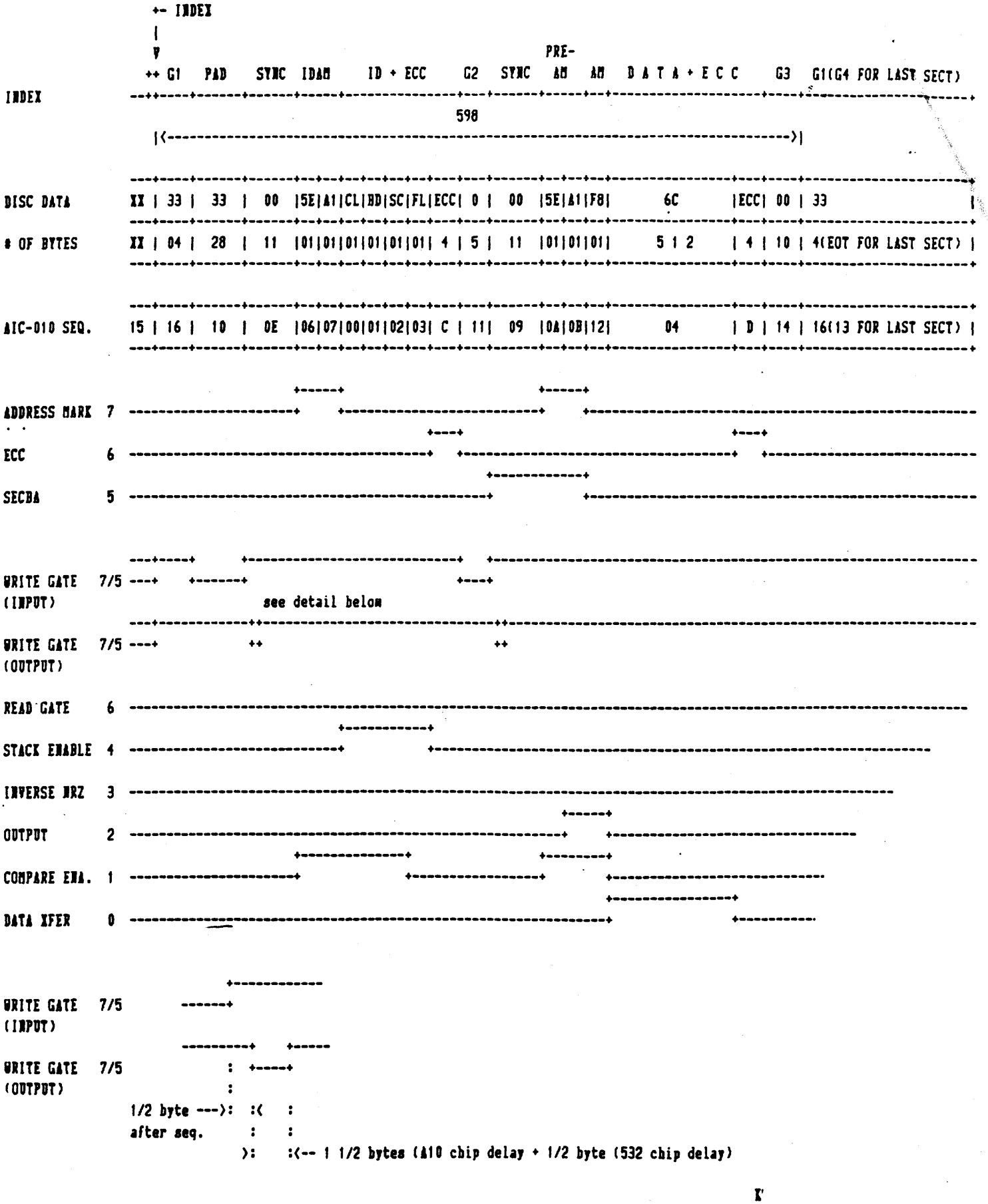
4. Overall defect list:



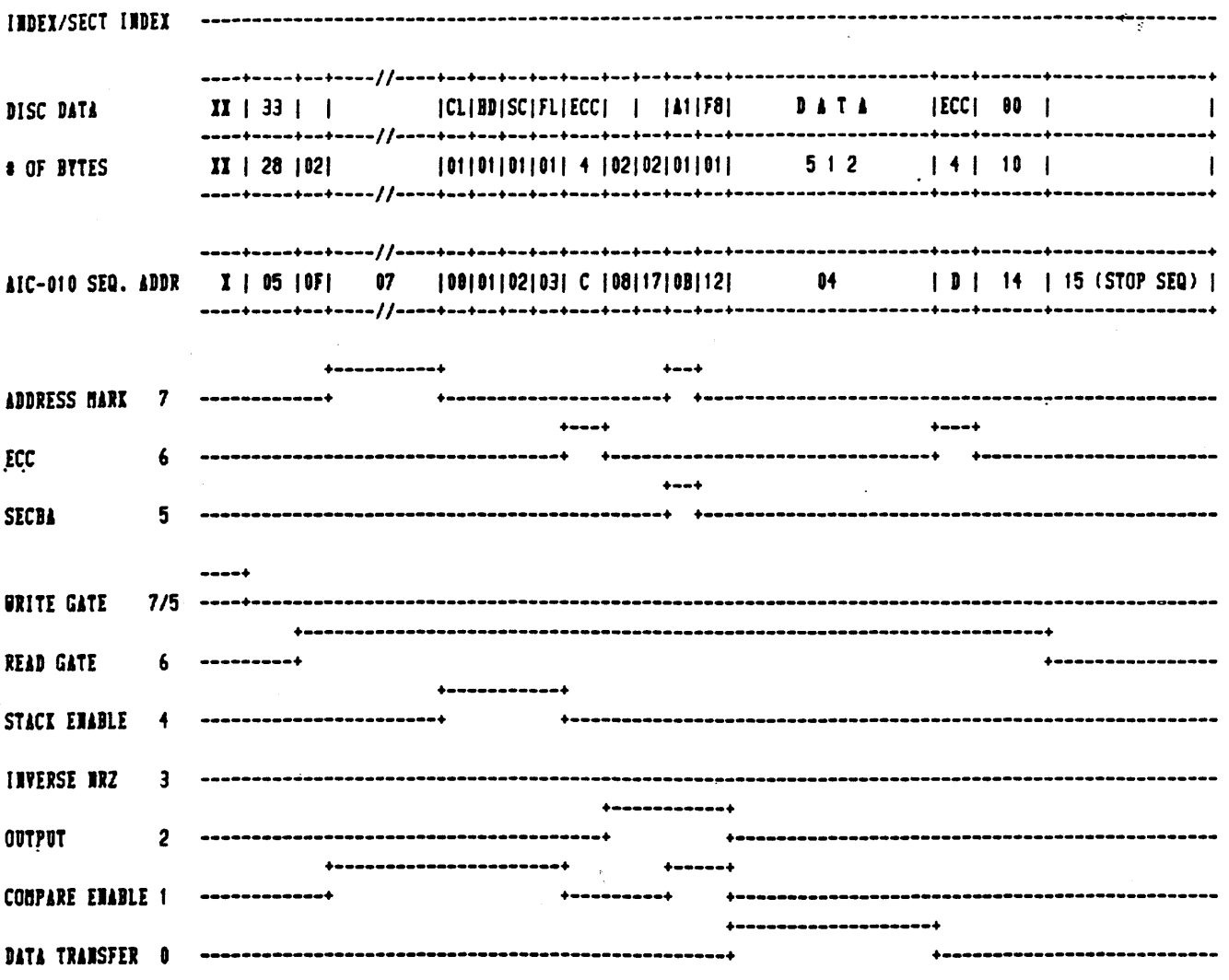
5. The drive capacity is unchanged. However, the last LBA is in cyl 530, head 0, sector 1 instead of cyl 529, head 5, sector 24.

```
Capacity = (26 sectors/trk * 6 trk/cyl -1) * 530
         = 82150 sectors
```

AIC-010 CHIP FUNCTIONS DURING FORMAT COMMAND (157H)

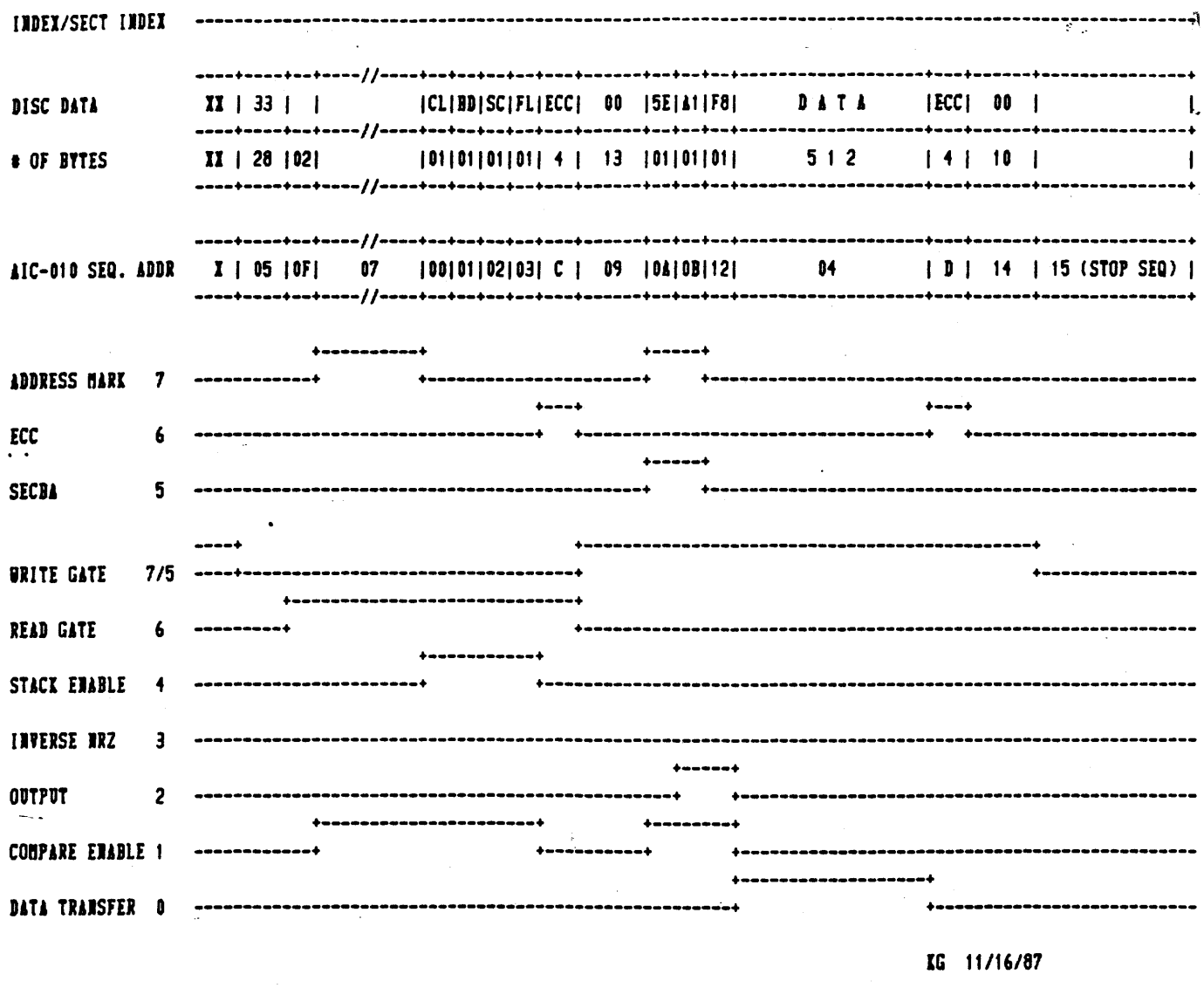


AIC-010 CHIP FUNCTIONS DURING READ COMMAND (157H)



KG 11/18/87

AIC-010 CHIP FUNCTIONS DURING WRITE COMMAND (157H)



KG 11/16/87