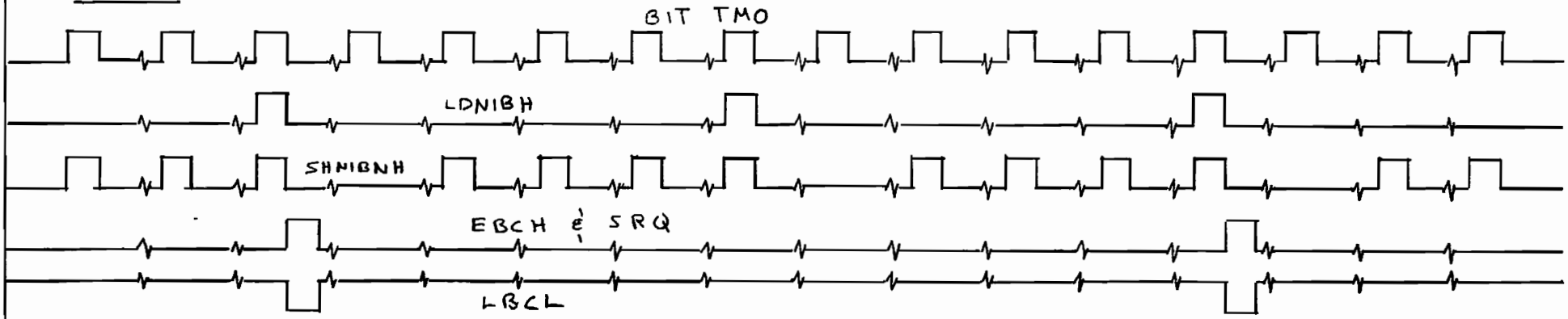
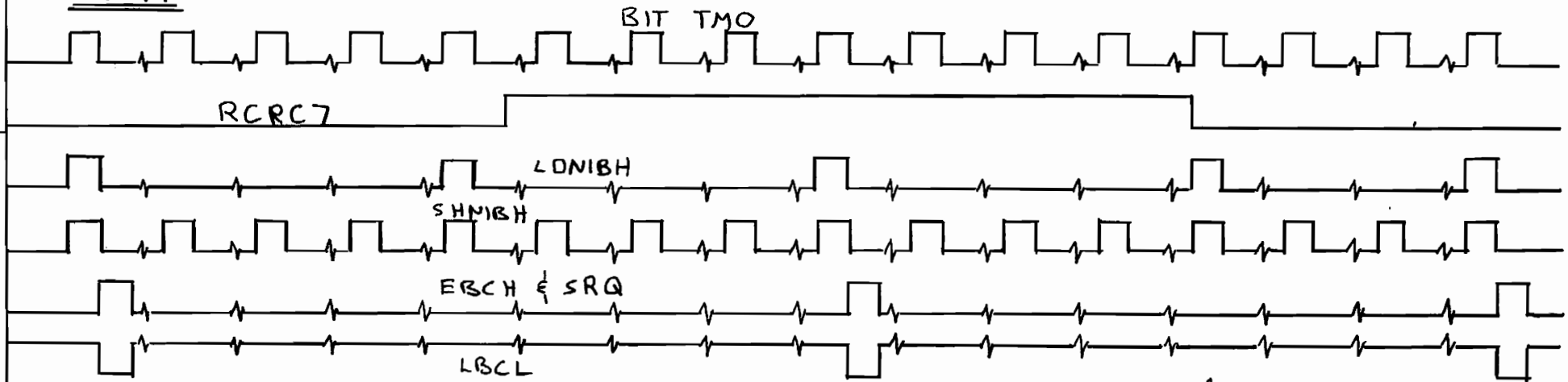
FPLA controlled Read Data Timing 8-81 Rev B3

GCR



Above is the timing generated by the FPLA to read a GCR record. This is not the timing for aquiring lock in a sync zone, for looking for the next sync zone, nor for looking for the framing character. There after, though, it is the timing. SLOKH is asserted always except when aquiring lock in a sync zone. CRSTL is asserted once each blockette simultaneous with LDNIBH. The width of the pulses above ≈ 160 ns. The spacing between the BIT TMO pulses ≈ 3 microsecond ± 20% That is approximately once every 20 instructions.

MFM



Above is the timing to read a 15 bit CRC. The Micro-code sets RCRC7 to indicate the 2nd. byte of the CRC is only 7 bits long. When reading data, the process loops on 8 bits. This is not the timing for Aquiring lock in a sync zone nor for looking for ID zone. SLOKH is always asserted except when aquiring lock. CRSTL is asserted twice each record simultaneously with LDNIBH. Each pulse ≈ 160 ns wide. The spacing between BIT TMO pulses is ≈ 5 microsecond ± 20% That is approximately once each 32 instructions

High Density

Byte	Bit#	Prom	
0	0-E		Aquire [CRST + LDNIB + SHNIB = DC]
0	F		EBC at Bt
1	0-E		
1	F		EBC at Bt
2	0-E		
2	F		EBC at Bt
3	0-0		
3	1		EBC at Bt
4	0-F		if Data = 1E then EBC at Bt
5	0	6	if Data = 1D then Aquire + EBC + LBC
5	1	6	if Data = 18 then
5	2	6	if Data = 13 then
5	3	8	if Data = 05 then LBC + EBC + SHNIB + LDNIB + CRST at Bt [framing → chip IE]
5	4	6	Aquire + EBC + LBC always
6	0-E	6	Aquire if Data ≠ 1F, then LBC + EBC at Bt
6	F	6	Aquire . EBC at Bt if Data ≠ 1F, then LBC + EBC at Bt
7	0-3	6	Aquire if Data ≠ 1F, then LBC + EBC at Bt
7	4	4	LBC + EBC always
X	X	6	if DUM then Aquire + LBC + EBC Always *

* Asserting "DUM" in high density causes an unconditional branch to #60 which looks for a re-sync zone. All Prom locations with bit "DUM" asserted must contain "6"

DC = Dont Care. Bt = Bit time (when signal BIT TMO is asserted high)
Any signal not mentioned must be inactive.

To enter a GCR record from an IRG - make this change in RDCM from: #00 to: #E0, Do this as soon as Read Data is detected

Monitor Read Data Detected + RDR. If it drops out, re-idle and start over again. If RDR, the Read Data in RDATA should be = #AB, indicating this is the byte count blockette. Time all this to find either sync zone late (after 25 feet) or ID late (if RDR is not 100 bits after the sync zone began). Aquiring sync is done by "Aquiring" for approx 50 sync bits and then looking for the blockette header.

To re-sync in successive blockettes, wether it be because the previous record was in error or is behind us:

Assert DUM for 1 instruction to begin a search which will Aquire until 25 "one" bits are found. Then it will look for the ID zone and if anything besides 1's or ID zone occurs, it will re-sync itself again. Again, monitor Read Data Detected + RDR. If a dropout occurs, idle till Data is detected and then attempt re-sync again. If RDR occurs, read the expected blockette header. Time all this so you know when it is late

FPLA code 8-31 Rev B3

High Density

FPLA code

Byte	Bit #	From	
8	0		
8	1		SHNIB at Bt
8	2		↓
8	3		
8	4	B	SHNIB + LONIB at Bt [blockette ID → chip IE]
8	5		
8	6		SHNIB at Bt
8	7		↓
8	8		
8	9	A	SHNIB + LONIB at Bt [1st nibble of blockette # → chip IE]
8	A	9	LBC + EBC + SRQ ALWAYS [RDR: RDATA = blockette header which is incorrect since it's supposed to be illegal GCR]
9	0		
9	1		SHNIB at Bt
9	2		↓
9	3		
9	4	B	SHNIB + LONIB at Bt [2nd nibble of blockette # → chip IE]
9	5		
9	6		SHNIB at Bt
9	7		↓
9	8		
9	9	A	SHNIB + LONIB at Bt [1st nibble of data → chip IE]
9	A	9	LBC + EBC + SRQ ALWAYS [RDR: RDATA = blockette number]

8-F	4	if Data = 00XXX then LBC at Bt
8-F	4	if Data = XXX00 then LBC at Bt
8-F	4	if Data = X000X then LBC at Bt
8-F	4	if Data = 11111 then LBC at Bt
8-F	9	if Data = 00XXX then LBC at Bt
8-F	9	if Data = XXX00 then LBC at Bt
8-F	9	if Data = X000X then LBC at Bt
8-F	9	if Data = 11111 then LBC at Bt

Above are the instructions to read in bytes of GCR. Also, above is the error detection. If, when you respond to RDR, RDSTS = 9, the data in RDATA was a legal GCR code. In the case of the blockette headers, this would be an error since they are illegal GCR codes. In general, heres the correspondence between RDSTS and the data in RDATA

RDSTS:	RDATA
#X9	this byte was legal GCR code
#XA	this byte is legal - The next one is illegal. RDL will be set in 163ns
#XB	this byte is legal. The next one is illegal
#XC	not used
#XD	this byte is illegal
#XE	this and the next byte are illegal. RDL will be set in 163ns
#XF	this and the next byte are illegal.

note: BIT 1 says the next byte is illegal.
BIT 2 says this byte is illegal

8-81 Rev B3

High Density

Byte	Bit#	Prom	
A	A	B	LBC + EBC + SRQ ALWAYS [RDR: RDATA = OK Byte]
B	0		
B	1		SHNIB at Bt
B	2		↓
B	3		
B	4	B	SHNIB + LONIB at Bt [2nd nibble of byte → chip 1E]
B	5		
B	6		SHNIB at Bt
B	7		↓
B	8		
B	9	E	SHNIB + LONIB at Bt [1st nibble of next byte → chip 1E]
B	A	D	LBC + EBC + SRQ ALWAYS [RDR = RDATA = Bad Byte]
D	0		
D	1		SHNIB at Bt
D	2		↓
D	3		
D	4	F	SHNIB + LONIB at Bt [2nd nibble of byte → chip 1E]
D	5		
D	6		SHNIB at Bt
D	7		↓
D	8		
D	9	A	SHNIB + LONIB at Bt [1st nibble of next byte → chip 1E]
D	A	9	LBC + EBC + SRQ ALWAYS [RDR: RDATA = OK BYTE]
E	A	F	LBC + EBC + SRQ ALWAYS [RDR: RDATA = Bad Byte]
F	0		
F	1		SHNIB at Bt
F	2		↓
F	3		
F	4	F	SHNIB + LONIB at Bt [2nd nibble of byte → chip 1E]
F	5		
F	6		SHNIB at Bt
F	7		↓
F	8		
F	9	E	SHNIB + LONIB at Bt [1st nibble of next byte → chip 1E]
F	A	D	LBC + EBC + SRQ ALWAYS [RDR: RDATA = Bad Byte]

Above are the routines which read bytes in after a byte has been an illegal GCR code.

FPLA code

8-81 Rev B2

Low Density

Byte	Bit*	Prom		
0	0-E		Aquire	[CRST+LONIB+SHNIB = DC]
0	F			EBC at Bt
1	0-E			
1	F			EBC at Bt
2	0-E			
2	F			EBC at Bt
3	0-9			
3	A			EBC at Bt
4	0-F	8	if Data = ØF then LBC+EBC+SHNIB+LONIB at Bt [framing → chip IE]	

The bit "DUM" is a don't care in low density mode.

To enter a MFM record, do exactly as you would with GCR.
note the differences:
"Aquiring" lasts approx 60 bits
there is no re-syncing

note also that in both MFM & GCR, the byte count in RDSTS gives an approximation of how long Aquiring sync has been going on

RDSTS = 0 0-15
1 16-31
2 32-47
3 47-50 or 60

= how many bits we have tried to aquire sync in ±20%

FPLA code

8-81 REV RN

Low Density

Byte	Bit#	Prom	
8	0		SHNIB at Bt [CRST=DC]
8	1		
8	2		
8	3		LDNIB at Bt [loads 2nd nibble of FF → chip 1E]
8	4		
8	5		
8	6		
8	7		LDNIB at Bt [loads 1st nibble of I.D. → chip 1E]
8	8	9	LBC+EBC+SRQ ALWAYS [RDR: RDATA = framing header FF]
9	0		
9	1		
9	2		
9	3		LDNIB at Bt [loads 2nd nibble of I.D. → chip 1E]
9	4		
9	5		
9	6		
9	7		LDNIB+CRST at Bt [loads 1st nibble of W.C. → chip 1E]
9	8	A	LBC+EBC+SRQ ALWAYS [RDR: RDATA = I.D. zone]
A	0		
A	1		
A	2		
A	3		LDNIB at Bt [loads 2nd nibble of W.C. → chip 1E]
A	4		
A	5		
A	6		
A	7		LDNIB at Bt [loads 3rd nibble of W.C. → chip 1E]
A	8	B	LBC+EBC+SRQ ALWAYS [RDR: RDATA = LSW C assert CHCRC (H) disassert CRCINIT (L)]
B	0		
B	1		
B	2		
B	3		LDNIB at Bt [loads 4th nibble of W.C. → chip 1E]
B	4		
B	5		
B	6		
B	7		LDNIB at Bt [loads 1st 4 bits of CRC → chip 1E]
B	8	C	LBC+EBC+SRQ ALWAYS [RDR: RDATA = MSW C assert RCRC7 (H)]

Above reads in MFM 16 bit words by byte + handles the 15 bit CRC

FPLA code

2-81 Rev B3

Low Density

Byte	Bit#	Prom	
C	0		SHMB at Bt
C	1		
C	2		
C	3		LDNIB at Bt [loads 2nd 4 bits of CRC → chip 1E]
C	4		
C	5		
C	6		
C	7		LDNIB at Bt [loads 3rd 4 bits of CRC → chip 1E]
C	8	D	LBC+EBC+SRQ ALWAYS [RDR: RDATA = Garbage disassert CHCRC (H) disassert RCRC7 (H)]
D	0		
D	1		
D	2		
D	3		LDNIB at Bt [loads 4th 3 bits of CRC → chip 1E]
D	4		
D	5		
D	6		
D	7		LDNIB+CRSTS at Bt [loads 1st nibble of data → chip 1E]
D	8	E	LBC+EBC+SRQ ALWAYS [RDR: RDATA = Garbage check CRCERR for W.C. CRC error now]
E	0		
E	1		
E	2		
E	3		LDNIB at Bt [loads 2nd nibble of byte → chip 1E]
E	4		
E	5		
E	6		
E	7		LDNIB at Bt [loads 1st nibble of next byte → chip 1E]
E	8	E	LBC+EBC+SRQ ALWAYS [RDR: RDATA = byte of data]

"E" is the loop that just reads in bytes till you tell it to stop by idleing it again.

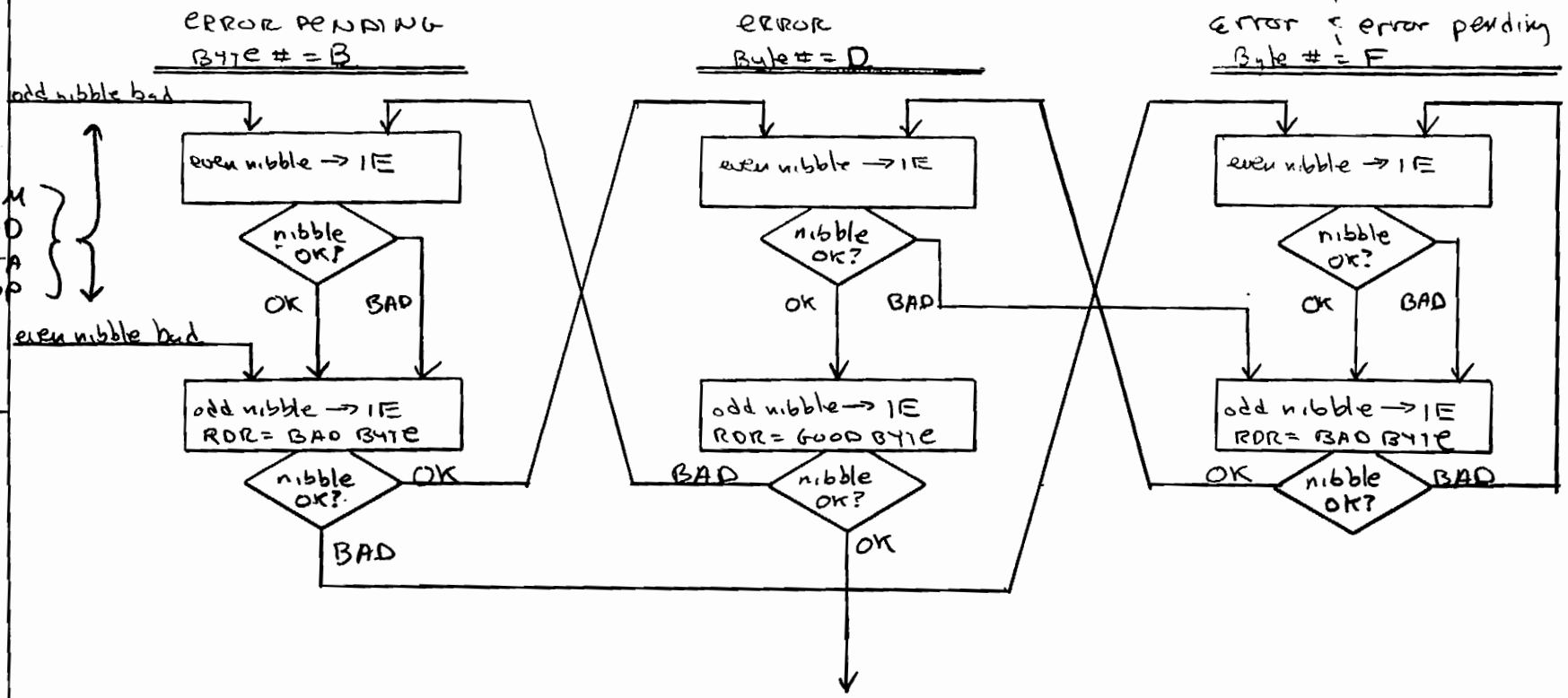
To read Data CRC, follow same timing procedure as reading WORD COUNT CRC. This is also true for GCR.

note that address "D Ø" is never actually executed since the EBC with RCRC7 set loads a 1 into the Bit counter instead of a zero

FPLA code

8, 21 RCE RW

How Read logic tells Processor that a GCR error has occurred



USING THIS ALGORYTHM, IF I look at the Byte number when RDR tells me to read a byte, I can interpret it like this

A, B → the next byte is in error

D → this byte is in error

E, F → this & the next byte are in error

82S100-I,N • 82S101-I,N

16X48X8 FPLA PROGRAM TABLE

THIS PORTION TO BE COMPLETED BY SIGNETICS

CF (XXXX) _____
CUSTOMER SYMBOLIZED PART # _____
DATE RECEIVED _____
COMMENTS _____

CUSTOMER NAME _____
PURCHASE ORDER # _____
SIGNETICS DEVICE # _____
TOTAL NUMBER OF PARTS _____
PROGRAM TABLE # _____
REV _____ DATE _____

PROGRAM TABLE ENTRIES

INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL	
Im	$\overline{Im}$	Don't Care	Prod. Term Present in FP	Prod. Term Not Present in FP	Active High	Active Low
H	L	— (dash)	A	• (period)	H	L

NOTE
Enter (—) for unused inputs of used P-terms

NOTES
1. Entries independent of output polarity.
2. Enter (A) for unused outputs of used P-terms.

NOTES
1. Polarity programmed once only.
2. Enter (H) for all unused outputs.

DEN BTO DUM PRODUCT TERM*

NO.	INPUT VARIABLE* 1 5	1 4	1 3	1 2	1 1	1 0	DATA 9	8	BYTE # 7	6	5	4	BIT # 3	2	1	0
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																

ACTIVE LEVEL* H L L H H L H L

OUTPUT FUNCTION* 7	6	5	4	3	2	1	0

LDNIB H | DSHNIB L | CRST L | SRQ H | AQUIKE L | EBC H | LBC L

* Input and Output fields of unused P-terms can be left blank. Unused inputs and outputs are FPLA terminals left floating.

16X48X8 FPLA PROGRAM TABLE

THIS PORTION TO BE COMPLETED BY SIGNETICS

CF (XXXX) _____

CUSTOMER SYMBOLIZED PART # _____

DATE RECEIVED _____

COMMENTS Read MFM

CUSTOMER NAME _____

PURCHASE ORDER # _____

SIGNETICS DEVICE # _____

TOTAL NUMBER OF PARTS _____

PROGRAM TABLE # _____

REV 83 DATE 8-81

Re-sync GCR

PROGRAM TABLE ENTRIES

INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL	
Im	$\overline{Im}$	Don't Care	Prod. Term Present in FP	Prod. Term Not Present in FP	Active High	Active Low
H	L	— (dash)	A	• (period)	H	L

NOTE: Enter (—) for unused inputs of used P-terms.

NOTES: 1. Entries independent of output polarity. 2. Enter (A) for unused outputs of used P-terms.

NOTES: 1. Polarity programmed once only. 2. Enter (H) for all unused outputs.

DEN BTO DUM PRODUCT TERM*

INPUT VARIABLE* (NO. 15 14 13 12 11 10 9 8 DATA) BYTE** (7 6 5 4) BIT** (3 2 1 0)

ACTIVE LEVEL* H L L H H L H L

OUTPUT FUNCTION* (7 6 5 4 3 2 1 0)

NO.	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
0	•	○	•															A							SHNIB ALL
2	•	•	•						○	○	•	○										A			
3	•	•	•						○	○	○	•										A			
4	•	—	•						○	○	•	•	—	—	—	—							A		Aquire All
6	○	—	•						○	○	—	—	—	○	—	○							A		
7	○	—	•	○	—	—	—	—	○	—	○	○					A						A	A	MFM frame
9	○	—	•						—	•	•	•	○	•	—	—	A								
10	○	•	•						—	•	•	•	—	○	○	○				A			A	A	Read MFM
11	○	—	•						—	•	○	—	○	—	—	—			A						
13	—	—	•						○	○	—	—	○	○	○	—							A		
14	—	—	•	—	—	—	—	○	○	—	○	○											A		
15	—	•	•	—	—	—	—	—	○	—	○	—	○	○	○	○					A	A	A	A	ECC
16	—	•	•	—	—	○	○	○	○	—	○	—	○	○	○	—					A	A	A	A	
17	—	•	•	—	○	○	—	—	○	—	○	—	○	○	—	○					A	A	A	A	
18	—	—	•	○	○	—	○	—	○	—	○	—	○	○	—	—	A		A			A	A	A	Find GCR frame
19	—	•	•						○	—	○	—	○	—	○	○					A	A	A	A	
21	—	•	•						—	•	•	•	○	○	○	○		A							SHNIB GCR
22	—	•	•						—	•	•	•	○	—	○	—		A							
24	—	—	•						—	•	•	•	○	—	○	○	A								
25	—	—	•						—	•	•	•	—	○	○	—	A								Read GCR
26	—	•	○						—	•	•	•	—	○	—	○				A			A	A	
28	—	—	•	○	○	•	•	•	—	•	•	•	○	—	○	○								A	
29	—	—	•	•	•	○	○		—	•	•	•	○	—	○	○								A	
30	—	—	•	•	○	○	○	•	—	•	•	•	○	—	○	○								A	
31	—	—	•	•	—	—	—	•	—	•	•	•	○	—	○	○								A	
32	—	—	•	○	○	•	•	•	—	•	•	•	—	○	○	—								A	
33	—	—	•	•	•	•	○	○	—	•	•	•	—	○	○	—								A	
34	—	—	•	•	○	○	○	•	—	•	•	•	—	○	○	—								A	
35	—	—	•	•	—	—	—	—	—	•	•	•	—	○	○	—								A	
37	—	•	—	○	•	•	•	•	○	—	—	•										A	A	A	Detect Illegal GCR code
38	—	—	•	○	○	•	•	•	○	—	—	•											A	A	
39	—	—	•	•	•	○	•	•	○	—	—	•											A	A	
40	—	—	•	•	•	•	○	•	○	—	—	•											A	A	
41	—	—	•	•	•	•	•	○	○	—	—	•											A	A	
42	—	—	•	•	•	•	•	•	○	—	—	•											A	A	
43	—	—	•						○	—	—	○	—	—	—	—							A		
44	—	•	•						○	—	—	○	○	—	○	○							A	A	
45	—	•	•						○	—	—	—										A			
46	—	•	•						○	—	—	—	○	○	•	•						A			
47																									

LBC L 0 · EBC H 1 · AQUIRE L 2 · SRQ H 3 · CRST L 4 · DSHNIB L 5 · LANIG H 6 · 7

* Input and Output fields of unused P-terms can be left blank. Unused inputs and outputs are FPLA terminals left floating.

82S100-I,N • 82S101-I,N

16X48X8 FPLA PROGRAM TABLE

THIS PORTION TO BE COMPLETED BY SIGNETICS

CF (XXXX) _____ CUSTOMER SYMBOLIZED PART # _____

DATE RECEIVED _____ COMMENTS _____

CUSTOMER NAME _____

PURCHASE ORDER # _____

SIGNETICS DEVICE # _____

TOTAL NUMBER OF PARTS _____

PROGRAM TABLE # _____ REV 03 DATE 8-8-81

PROGRAM TABLE ENTRIES

INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL	
Im	$\overline{Im}$	Don't Care	Prod. Term Present in FP	Prod. Term Not Present in FP	Active High	Active Low
H	L	— (dash)	A	• (period)	H	L

NOTE: Enter (—) for unused inputs of used P-terms.

NOTES: 1. Entries independent of output polarity. 2. Enter (A) for unused outputs of used P-terms.

NOTES: 1. Polarity programmed once only. 2. Enter (H) for all unused outputs

NO.	PRODUCT TERM* INPUT VARIABLE* 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ACTIVE LEVEL* OUTPUT FUNCTION* 7	6	5	4	3	2	1	0
0	–	L	–	–	–	–	–	–	L	L	–	L	–	–	–	–	•	A	•	•	•	•	•	•
1	–	–	–	–	–	–	–	–	L	L	–	–	–	–	–	–	•	•	•	•	•	A	•	•
2	–	–	–	–	–	–	–	–	L	L	L	–	–	–	–	–	•	•	•	•	•	A	•	•
3	–	H	–	–	–	–	–	–	L	L	–	–	H	H	H	H	•	•	•	•	•	•	A	•
4	L	H	–	–	–	–	–	–	L	L	H	H	H	L	H	L	•	•	•	•	•	•	A	–
5	L	H	–	L	H	H	H	H	L	H	L	L	–	–	–	–	A	•	•	•	•	•	A	A
6	L	H	–	–	–	–	–	–	H	–	–	–	L	–	H	H	A	•	•	•	•	•	•	•
7	L	–	–	–	–	–	–	–	H	–	–	–	H	L	L	L	•	•	•	A	•	•	A	A
8	L	H	–	–	–	–	–	–	H	–	L	H	L	H	H	H	•	•	A	•	•	•	•	•
9	H	H	–	–	–	–	–	–	L	L	H	H	L	L	L	H	•	•	•	•	•	•	A	•
10	H	H	–	H	H	H	H	L	L	H	L	L	–	–	–	–	•	•	•	•	•	•	A	•
11	H	–	–	H	H	H	L	H	L	H	L	H	L	L	L	L	•	•	•	•	•	A	A	A
12	H	–	–	H	H	L	L	L	L	H	L	H	L	L	L	H	•	•	•	•	•	A	A	A
13	H	–	–	H	L	L	H	H	L	H	L	H	L	L	H	L	•	•	•	•	•	A	A	A
14	H	H	–	L	L	H	L	H	L	H	L	H	L	L	H	H	A	•	A	•	•	•	A	A
15	H	–	–	–	–	–	–	–	L	H	L	H	L	H	L	L	•	•	•	•	•	A	A	A
16	H	–	–	–	–	–	–	–	H	–	–	–	L	L	L	L	•	A	•	•	•	•	•	•
17	H	–	–	–	–	–	–	–	H	–	–	–	L	H	L	H	•	A	•	•	•	•	•	•
18	H	H	–	–	–	–	–	–	H	–	–	–	L	H	L	L	A	•	•	•	•	•	•	•
19	H	H	–	–	–	–	–	–	H	–	–	–	H	L	L	H	A	•	•	•	•	•	•	•
20	H	–	L	–	–	–	–	–	H	–	–	–	H	L	H	L	•	•	•	A	•	•	A	A
21	H	H	–	L	L	–	–	–	H	–	–	–	L	H	L	L	•	•	•	•	•	•	•	A
22	H	H	–	–	–	–	L	L	H	–	–	–	L	H	L	L	•	•	•	•	•	•	•	A
23	H	H	–	–	L	L	L	–	H	–	–	–	L	H	L	L	•	•	•	•	•	•	•	A
24	H	H	–	H	H	H	H	H	H	–	–	–	L	H	L	L	•	•	•	•	•	•	•	A
25	H	H	–	L	L	–	–	–	H	–	–	–	H	L	L	H	•	•	•	•	•	•	•	A
26	H	H	–	–	–	–	L	L	H	–	–	–	H	L	L	H	•	•	•	•	•	•	•	A
27	H	H	–	–	L	L	L	–	H	–	–	–	H	L	L	H	•	•	•	•	•	•	•	A
28	H	H	–	H	H	H	H	H	H	–	–	–	H	L	L	H	•	•	•	•	•	•	•	A
29	H	–	H	–	–	–	–	–	–	–	–	–	–	–	–	–	•	•	•	•	•	A	A	A
30	H	H	–	L	–	–	–	–	L	H	H	–	–	–	–	–	•	•	•	•	•	•	A	A
31	H	H	–	–	L	–	–	–	L	H	H	–	–	–	–	–	•	•	•	•	•	•	A	A
32	H	H	–	–	–	L	–	–	L	H	H	–	–	–	–	–	•	•	•	•	•	•	A	A
33	H	H	–	–	–	–	L	–	L	H	H	–	–	–	–	–	•	•	•	•	•	•	A	A
34	H	H	–	–	–	–	–	L	L	H	H	–	–	–	–	–	•	•	•	•	•	•	A	A
35	H	H	–	–	–	–	–	–	L	H	H	L	H	H	H	H	•	•	•	•	•	•	A	•
36	H	–	–	–	–	–	–	–	L	H	H	H	L	H	L	L	•	•	•	•	•	•	A	A
37	H	–	–	–	–	–	–	–	L	H	H	L	–	–	–	–	•	•	•	•	•	A	•	•
38	H	–	–	–	–	–	–	–	L	H	H	H	L	L	–	–	•	•	•	•	•	A	•	•
39																								
40																								
41																								
42																								
43																								
44																								
45																								
46																								
47																								

* Input and Output fields of unused P-terms can be left blank. Unused inputs and outputs are FPLA terminals left floating.

82S100-I,N • 82S101-I,N

16X48X8 FPLA PROGRAM TABLE

THIS PORTION TO BE COMPLETED BY SIGNETICS

CF (XXXX) _____

CUSTOMER SYMBOLIZED PART # _____

DATE RECEIVED _____

COMMENTS _____

CUSTOMER NAME Raw Part = 82S100 #0104-0123

PURCHASE ORDER # Burned Part = #0121-0113

SIGNETICS DEVICE # Prom set = #0125-0050

TOTAL NUMBER OF PARTS _____

PROGRAM TABLE # _____

REV 03 DATE 8-8-81

PROGRAM TABLE ENTRIES

INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL	
Im	$\overline{Im}$	Don't Care	Prod. Term Present in FP	Prod. Term Not Present in FP	Active High	Active Low
H	L	— (dash)	A	• (period)	H	L

NOTE: Enter (—) for unused inputs of used P-terms.

NOTES: 1. Entries independent of output polarity. 2. Enter (A) for unused outputs of used P-terms.

NOTES: 1. Polarity programmed once only. 2. Enter (H) for all unused outputs

NO.	PRODUCT TERM* INPUT VARIABLE* 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ACTIVE LEVEL* OUTPUT FUNCTION* 7	6	5	4	3	2	1	0
0	—	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—	•	A	•	•	•	•	•	•
1	—	—	—	—	—	—	—	—	L	L	—	L	—	—	—	—	•	•	•	•	•	A	•	•
2	—	—	—	—	—	—	—	—	L	L	L	—	—	—	—	—	•	•	•	•	•	A	•	•
3	—	H	—	—	—	—	—	—	L	L	—	—	H	H	H	H	•	•	•	•	•	•	A	•
4	L	H	—	—	—	—	—	—	L	L	H	H	H	L	H	L	•	•	•	•	•	•	A	-
5	L	H	—	L	H	H	H	H	L	H	L	L	—	—	—	—	A	•	•	•	•	•	A	A
6	L	H	—	—	—	—	—	—	H	—	—	—	L	—	H	H	A	•	•	•	•	•	•	•
7	L	—	—	—	—	—	—	—	H	—	—	—	H	L	L	L	•	•	•	A	•	•	A	A
8	L	H	—	—	—	—	—	—	H	—	L	H	L	H	H	H	•	•	A	•	•	•	•	•
9	H	H	—	—	—	—	—	—	L	L	H	H	L	L	L	H	•	•	•	•	•	•	A	•
10	H	H	—	H	H	H	H	L	L	H	L	L	—	—	—	—	•	•	•	•	•	•	A	•
11	H	—	—	H	H	H	L	H	L	H	L	H	L	L	L	L	•	•	•	•	•	A	A	A
12	H	—	—	H	H	L	L	L	L	H	L	H	L	L	L	H	•	•	•	•	•	A	A	A
13	H	—	—	H	L	L	H	H	L	H	L	H	L	L	H	L	•	•	•	•	•	A	A	A
14	H	H	—	L	L	H	L	H	L	H	L	H	L	L	H	H	A	•	A	•	•	•	A	A
15	H	—	—	—	—	—	—	—	L	H	L	H	L	H	L	L	•	•	•	•	•	A	A	A
16	H	—	—	—	—	—	—	—	H	—	—	—	L	L	L	L	•	A	•	•	•	•	•	•
17	H	—	—	—	—	—	—	—	H	—	—	—	L	H	L	H	•	A	•	•	•	•	•	•
18	H	H	—	—	—	—	—	—	H	—	—	—	L	H	L	L	A	•	•	•	•	•	•	•
19	H	H	—	—	—	—	—	—	H	—	—	—	H	L	L	H	A	•	•	•	•	•	•	•
20	H	—	L	—	—	—	—	—	H	—	—	—	H	L	H	L	•	•	•	A	•	•	A	A
21	H	H	—	L	L	—	—	—	A	—	—	—	L	H	L	L	•	•	•	•	•	•	•	A
22	H	H	—	—	—	—	L	L	H	—	—	—	L	H	L	L	•	•	•	•	•	•	•	A
23	H	H	—	—	L	L	L	—	H	—	—	—	L	H	L	L	•	•	•	•	•	•	•	A
24	H	H	—	H	H	H	H	H	H	—	—	—	L	H	L	L	•	•	•	•	•	•	•	A
25	H	H	—	L	L	—	—	—	H	—	—	—	H	L	L	H	•	•	•	•	•	•	•	A
26	H	H	—	—	—	—	L	L	H	—	—	—	H	L	L	H	•	•	•	•	•	•	•	A
27	H	H	—	—	L	L	L	—	H	—	—	—	H	L	L	H	•	•	•	•	•	•	•	A
28	H	H	—	H	H	H	H	H	H	—	—	—	H	L	L	H	•	•	•	•	•	•	•	A
29	H	—	H	—	—	—	—	—	—	—	—	—	—	—	—	—	•	•	•	•	•	A	A	A
30	H	H	—	L	—	—	—	—	L	H	H	—	—	—	—	—	•	•	•	•	•	•	A	A
31	H	H	—	—	L	—	—	—	L	H	H	—	—	—	—	—	•	•	•	•	•	•	A	A
32	H	H	—	—	—	L	—	—	L	H	H	—	—	—	—	—	•	•	•	•	•	•	A	A
33	H	H	—	—	—	—	L	—	L	H	H	—	—	—	—	—	•	•	•	•	•	•	A	A
34	H	H	—	—	—	—	—	L	L	H	H	—	—	—	—	—	•	•	•	•	•	•	A	A
35	H	H	—	—	—	—	—	—	L	H	H	L	H	H	H	H	•	•	•	•	•	•	A	•
36	H	—	—	—	—	—	—	—	L	H	H	H	L	H	L	L	•	•	•	•	•	•	A	A
37	H	—	—	—	—	—	—	—	L	H	H	L	—	—	—	—	•	•	•	•	•	A	•	•
38	H	—	—	—	—	—	—	—	L	H	H	H	L	L	—	—	•	•	•	•	•	A	•	•
39																								
40																								
41																								
42																								
43																								
44																								
45																								
46																								
47																								

* Input and Output fields of unused P-terms can be left blank. Unused inputs and outputs are FPLA terminals left floating

TEXT = FPLA3.TEXT ACCOUNT = UCODE.CTC
RAW PART = N82S100 #0104-0124
BURNED PART = #0121-0113 REV. LEVEL = B3
LOCATION = 1F

		PRODUCT TERM INPUT VARIABLE																	ACTIVE LEVEL OUTPUT FUNCTION							
N		1	1	1	1	1	1												H	L	L	H	H	L	H	L
0.		5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
**	**	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	****	*	*	*	*	*	*	*	*
0		-	L	-	-	-	-	-	-	-	-	-	-	-	-	-	-		.	A	.	.	.	.	.	.
1		-	-	-	-	-	-	-	-	L	L	-	L	-	-	-	-		.	.	.	.	.	A	.	.
2		-	-	-	-	-	-	-	-	L	L	L	-	-	-	-	-		.	.	.	.	.	A	.	.
3		-	H	-	-	-	-	-	-	L	L	-	-	H	H	H	H		.	.	.	.	.	.	A	.
4		L	H	-	-	-	-	-	-	L	L	H	H	H	L	H	L		.	.	.	.	.	.	A	.
5		L	H	-	L	H	H	H	H	L	H	L	L	-	-	-	-		A	.	.	.	.	.	A	A
6		L	H	-	-	-	-	-	-	H	-	-	-	L	-	H	H		A	.	.	.	.	.	.	.
7		L	-	-	-	-	-	-	-	H	-	-	-	H	L	L	L		.	.	.	A	.	.	A	A
8		L	H	-	-	-	-	-	-	H	-	L	H	L	H	H	H		.	.	A	.	.	.	.	.
9		H	H	-	-	-	-	-	-	L	L	H	H	L	L	L	H		.	.	.	.	.	.	A	.
10		H	H	-	-	-	-	-	L	L	H	L	L	-	-	-	-		.	.	.	.	.	.	A	.
11		H	-	-	H	H	H	L	H	L	H	L	H	L	L	L	L		.	.	.	.	.	A	A	A
12		H	-	-	H	H	L	L	L	L	H	L	H	L	L	L	H		.	.	.	.	.	A	A	A
13		H	-	-	H	L	L	H	H	L	H	L	H	L	L	H	L		.	.	.	.	.	A	A	A
14		H	H	-	L	L	H	L	H	L	H	L	H	L	L	H	H		A	.	A	.	.	.	A	A
15		H	-	-	-	-	-	-	-	L	H	L	H	L	H	L	L		.	.	.	.	.	A	A	A
16		H	-	-	-	-	-	-	-	H	-	-	-	L	L	L	L		.	A	.	.	.	.	.	.
17		H	-	-	-	-	-	-	-	H	-	-	-	L	H	L	H		.	A	.	.	.	.	.	.
18		H	H	-	-	-	-	-	-	H	-	-	-	L	H	L	L		A	.	.	.	.	.	.	.
19		H	H	-	-	-	-	-	-	H	-	-	-	H	L	L	H		A	.	.	.	.	.	.	.
20		H	-	L	-	-	-	-	-	H	-	-	-	H	L	H	L		.	.	.	A	.	.	A	A
21		H	H	-	L	L	-	-	-	H	-	-	-	L	H	L	L		.	.	.	.	.	.	.	A
22		H	H	-	-	-	-	L	L	H	-	-	-	L	H	L	L		.	.	.	.	.	.	.	A
23		H	H	-	-	L	L	L	-	H	-	-	-	L	H	L	L		.	.	.	.	.	.	.	A
24		H	H	-	H	H	H	H	H	H	-	-	-	L	H	L	L		.	.	.	.	.	.	.	A
25		H	H	-	L	L	-	-	-	H	-	-	-	H	L	L	H		.	.	.	.	.	.	.	A
26		H	H	-	-	-	-	L	L	H	-	-	-	H	L	L	H		.	.	.	.	.	.	.	A
27		H	H	-	-	L	L	L	-	H	-	-	-	H	L	L	H		.	.	.	.	.	.	.	A
28		H	H	-	H	H	H	H	H	H	-	-	-	H	L	L	H		.	.	.	.	.	.	.	A
29		H	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-		.	.	.	.	.	A	A	A
30		H	H	-	L	-	-	-	-	L	H	H	-	-	-	-	-		.	.	.	.	.	.	A	A
31		H	H	-	-	L	-	-	-	L	H	H	-	-	-	-	-		.	.	.	.	.	.	A	A
32		H	H	-	-	-	L	-	-	L	H	H	-	-	-	-	-		.	.	.	.	.	.	A	A
33		H	H	-	-	-	-	L	-	L	H	H	-	-	-	-	-		.	.	.	.	.	.	A	A
34		H	H	-	-	-	-	-	L	L	H	H	-	-	-	-	-		.	.	.	.	.	.	A	A
35		H	H	-	-	-	-	-	-	L	H	H	L	H	H	H	H		.	.	.	.	.	.	A	.
36		H	-	-	-	-	-	-	-	L	H	H	H	L	H	L	L		.	.	.	.	.	.	A	A
37		H	-	-	-	-	-	-	-	L	H	H	L	-	-	-	-		.	.	.	.	.	A	.	.
38		H	-	-	-	-	-	-	-	L	H	H	H	L	L	-	-		.	.	.	.	.	A	.	.

THIS CHART CONFORMS TO SIGNETICS PROAGRAMMING FORMATS FOR FPLA'S

Jump Table PROM in Read Logic

HDEN	DUM	Address: Byte #				Address: Bit #				Contents
9	8	7	6	5	4	3	2	1	0	Hex
0	X	0	1	0	0	X	X	X	X	8
0	X	1	0	0	0	1	0	0	0	9
0	X	1	0	0	1	1	0	0	0	A
0	X	1	0	1	0	1	0	0	0	B
0	X	1	0	1	1	1	0	0	0	C
0	X	1	1	0	0	1	0	0	0	D
0	X	1	1	0	1	1	0	0	0	E
0	X	1	1	1	0	1	0	0	0	E
1	0	0	1	0	1	0	0	0	0	6
1	0	0	1	0	1	0	0	0	1	6
1	0	0	1	0	1	0	0	1	0	6
1	0	0	1	0	1	0	0	1	1	8
1	0	0	1	0	1	0	1	0	0	6
1	0	0	1	1	0	X	X	X	X	6
1	0	0	1	1	1	0	0	X	X	6
1	0	0	1	1	1	0	1	0	0	4
1	1	X	X	X	X	X	X	X	X	6
1	0	1	0	0	0	0	1	0	0	B
1	0	1	0	0	0	1	0	0	1	A
1	0	1	0	0	0	1	0	1	0	9
1	0	1	0	0	1	0	1	0	0	B
1	0	1	0	0	1	1	0	0	1	A
1	0	1	0	0	1	1	0	1	0	9
1	0	1	0	1	0	1	0	1	0	B
1	0	1	0	1	1	0	1	0	0	B
1	0	1	0	1	1	1	0	0	1	E
1	0	1	0	1	1	1	0	1	0	D
1	0	1	1	0	1	0	1	0	0	F
1	0	1	1	0	1	1	0	0	1	A
1	0	1	1	0	1	1	0	1	0	9
1	0	1	1	1	0	1	0	1	0	F
1	0	1	1	1	1	0	1	0	0	F
1	0	1	1	1	1	1	0	0	1	E
1	0	1	1	1	1	1	0	1	0	D

All unspecified locations = ∅

Jump Table PROM in Read Logic 8-81 Rev B3

TEXT=JMPTBL3.TEXT BINARY=JMPTBL3.BIN ACCOUNT=UCODE.CTC
RAW PART = 82S137 OR 93453 #0106-0014 OR #0106-0027
BURNED PART = #0121-0111 REV. LEVEL = B3
LOCATION = 2D

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
010:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
030:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040:	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
050:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
060:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
070:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
080:	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
090:	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0
0A0:	0	0	0	0	0	0	0	0	B	0	0	0	0	0	0	0
0B0:	0	0	0	0	0	0	0	0	C	0	0	0	0	0	0	0
0C0:	0	0	0	0	0	0	0	0	D	0	0	0	0	0	0	0
0D0:	0	0	0	0	0	0	0	0	E	0	0	0	0	0	0	0
0E0:	0	0	0	0	0	0	0	0	E	0	0	0	0	0	0	0
0F0:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
110:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
130:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
140:	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
150:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
160:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
170:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
180:	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
190:	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0
1A0:	0	0	0	0	0	0	0	0	B	0	0	0	0	0	0	0
1B0:	0	0	0	0	0	0	0	0	C	0	0	0	0	0	0	0
1C0:	0	0	0	0	0	0	0	0	D	0	0	0	0	0	0	0
1D0:	0	0	0	0	0	0	0	0	E	0	0	0	0	0	0	0
1E0:	0	0	0	0	0	0	0	0	E	0	0	0	0	0	0	0
1F0:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
200:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
210:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
220:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
230:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
240:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
250:	6	6	6	8	6	0	0	0	0	0	0	0	0	0	0	0
260:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
270:	6	6	6	6	4	0	0	0	0	0	0	0	0	0	0	0
280:	0	0	0	0	B	0	0	0	0	A	9	0	0	0	0	0
290:	0	0	0	0	B	0	0	0	0	A	9	0	0	0	0	0
2A0:	0	0	0	0	0	0	0	0	0	0	B	0	0	0	0	0
2B0:	0	0	0	0	B	0	0	0	0	E	D	0	0	0	0	0
2C0:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2D0:	0	0	0	0	F	0	0	0	0	A	9	0	0	0	0	0

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2E0:	0	0	0	0	0	0	0	0	0	0	F	0	0	0	0	0
2F0:	0	0	0	0	F	0	0	0	0	E	D	0	0	0	0	0
300:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
310:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
320:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
330:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
340:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
350:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
360:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
370:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
380:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
390:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3A0:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3B0:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3C0:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3D0:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3E0:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3F0:	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6

GCR	Hex (legal)	Hex (illegal)	COMMENTS
0 0 0 0 0		0	
0 0 0 0 1		1	
0 0 0 1 0		2	
0 0 0 1 1		7	
0 0 1 0 0		6	
0 0 1 0 1		B	framing character / Blockette header B
0 0 1 1 0		C	Blockette header C
0 0 1 1 1		A	Blockette header A
0 1 0 0 0		4	
0 1 0 0 1	1		
0 1 0 1 0	0		
0 1 0 1 1	2		
0 1 1 0 0		3	
0 1 1 0 1	3		
0 1 1 1 0	6		
0 1 1 1 1	5		
1 0 0 0 0		8	
1 0 0 0 1		9	
1 0 0 1 0	8		
1 0 0 1 1	7		
1 0 1 0 0		D	Framing character, reversed
1 0 1 0 1	9		
1 0 1 1 0	C		
1 0 1 1 1	B		
1 1 0 0 0		E	
1 1 0 0 1	E		
1 1 0 1 0	4		
1 1 0 1 1	F		
1 1 1 0 0		5	Blockette header A reversed
1 1 1 0 1	D		
1 1 1 1 0	A		
1 1 1 1 1		F	SYNC CHARACTER / Blockette header F

↑ written last
written first

GCR code table

All these codes are defined such that if you read the GCR backwards and translate that, you'll get the backwards hex number

ie

A = 1010 rev 0101 = 5 (Hex → binary → Hex)

A = 11110 rev 01111 = 5 (Hex → GCR → Hex)

10/15/80 Rev B2

GCR Decoder PROM in Read Logic

Address									output	HEX DIGIT
Hi DEN	REV	DUM	4	3	2	1	0			
1	X	X	0	0	0	0	0		0	0
1	X	X	0	0	0	0	1		8	1
1	X	X	0	0	0	1	0		4	2
1	X	X	0	0	0	1	1		E	7
1	X	X	0	0	1	0	0		6	6
1	X	X	0	0	1	0	1		D	B
1	X	X	0	0	1	1	0		3	C
1	X	X	0	0	1	1	1		5	A
1	X	X	0	1	0	0	0		2	4
1	X	X	0	1	0	0	1		8	1
1	X	X	0	1	0	1	0		0	0
1	X	X	0	1	0	1	1		4	2
1	X	X	0	1	1	0	0		C	3
1	X	X	0	1	1	0	1		C	3
1	X	X	0	1	1	1	0		6	6
1	X	X	0	1	1	1	1		A	5
1	X	X	1	0	0	0	0		1	8
1	X	X	1	0	0	0	1		9	9
1	X	X	1	0	0	1	0		1	8
1	X	X	1	0	0	1	1		E	7
1	X	X	1	0	1	0	0		B	D
1	X	X	1	0	1	0	1		9	9
1	X	X	1	0	1	1	0		3	C
1	X	X	1	0	1	1	1		D	B
1	X	X	1	1	0	0	0		7	E
1	X	X	1	1	0	0	1		7	E
1	X	X	1	1	0	1	0		2	4
1	X	X	1	1	0	1	1		F	F
1	X	X	1	1	1	0	0		A	5
1	X	X	1	1	1	0	1		B	D
1	X	X	1	1	1	1	0		5	A
1	X	X	1	1	1	1	1		F	F

last bit read
first bit read

Address									OUTPUT	HEX DIGIT
Hi DEN	REV	DUM	4	3	2	1	0			
0	X	X	X	0	0	0	0		0	0
0	X	X	X	0	0	0	1		1	8
0	X	X	X	0	0	1	0		2	4
0	X	X	X	0	0	1	1		3	C
0	X	X	X	0	1	0	0		4	2
0	X	X	X	0	1	0	1		5	A
0	X	X	X	0	1	1	0		6	6
0	X	X	X	0	1	1	1		7	E
0	X	X	X	1	0	0	0		8	1
0	X	X	X	1	0	0	1		9	9
0	X	X	X	1	0	1	0		A	5
0	X	X	X	1	0	1	1		B	D
0	X	X	X	1	1	0	0		C	3
0	X	X	X	1	1	0	1		D	B
0	X	X	X	1	1	1	0		E	7
0	X	X	X	1	1	1	1		F	F

last bit read
first bit read

"Hex digit" refers to the hex digit which will eventually be sent to the PPU which results from this bit pattern.

X's ought to be programmed for both the low and the high state

GCR Decoder Prom

8-81 Rev B2

```
*      CTC GCR DECODER / READ LOGIC                                  7-15-81
*
*          TEXT=GCREAD2.TEXT     BINARY=GCREAD2.BIN     ACCOUNT=UCODE,CTC
*          RAW PART = 82S131 [512*4]      #0106-0010
*          BURNED PART = #0121-0109     REV. LEVEL = B2
*          LOCATION = 2E
*
*                0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
*             .....................................................
*               :
*           00: 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
            10: 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
            20: 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
            30: 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
            40: 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
            50: 0  1  2  3  4  5  6  7  8  9  A  B  C  D  F  F
            60: 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
            70: 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
            80: 0  8  4  F  6  D  3  5  2  8  0  4  C  C  6  A
            90: 1  9  1  E  B  9  3  D  7  7  2  F  A  B  5  F
            A0: 0  8  4  E  6  D  3  5  2  8  0  4  C  C  6  A
            B0: 1  9  1  E  B  9  3  D  7  7  2  F  A  B  5  F
            C0: 0  8  4  E  6  D  3  5  2  8  0  4  C  C  6  A
            D0: 1  9  1  E  B  9  3  D  7  7  2  F  A  B  5  F
            E0: 0  8  4  E  6  D  3  5  2  8  0  4  C  C  6  A
            F0: 1  9  1  E  B  9  3  D  7  7  2  F  A  B  5  F
*             :.....................................................
```

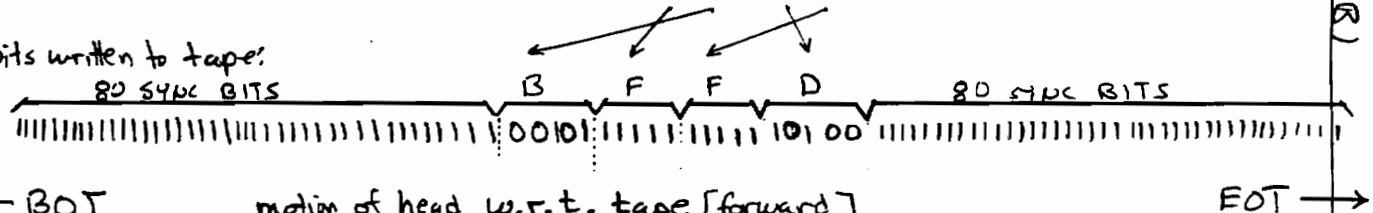
10/20/80

GCR Encoder PROM in Write Logic

Address	Contents	legal hex digit	illegal hex digit	comments
00	0A	0		
01	12	8		
02	1A	4		
03	16	C		
04	0B	2		
05	1E	A		
06	0E	6		
07	19	E		
08	09	1		
09	15	9		
0A	0F	5		
0B	1D	D		
0C	0D	3		
0D	17	B		
0E	13	7		
0F	1B	F		
10	00		0	
11	10		8	
12	08		4	
13	06		C	Blockette header C
14	02		2	
15	07		A	Blockette header A
16	04		6	
17	18		E	
18	01		1	
19	11		9	
1A	1C		5	Blockette header A reversed
1B	14		D	Framing character reversed
1C	0C		3	
1D	05		B	Framing character / Blockette header B
1E	03		7	
1F	1F		F	sync zone / Blockette header F

Here's what a File Mark looks like:

sent to write logic: FF, FF, FF, FF, FB, DF, FF, FF, FF, FF
(FF, FF, FF, FF: sync zone; FB: FM.; DF: FM. Rev.; FF, FF, FF, FF: sync zone)



GCR Encoder PROM
10-20-80 Rev B)

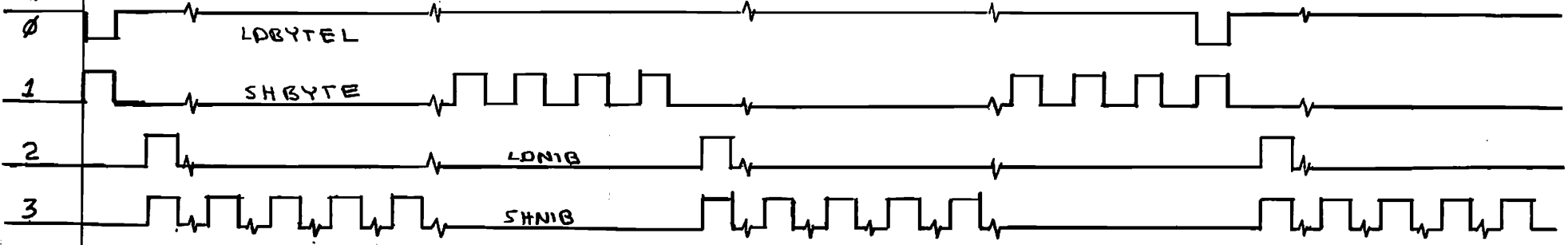
```
*      CTC GCR ENCODER PROM / WRITE LOGIC                          11-18-80
*
*          TEXT=GCRWRT1.TEXT    BINARY=GCRWRT1.BIN     ACCOUNT=UCODE.CTC
*          RAW PART = 82S123 [32*8]   #0106-0009
*          BURNED PART = #0121-0108     REV. LEVEL = B1
*          LOCATION = 13E
*
*                0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
*             ..................................................
*             :
*           00: 0A 12 1A 16 0B 1E 0E 19 09 15 0F 1D 0D 17 13 1B
*           10: 00 10 08 06 02 07 04 18 01 11 1C 14 0C 05 03 1F
*             :.................................................
```

WRITE CONTROL PROM timing

10-15-80 Rev B1

PROM bit #

GCR

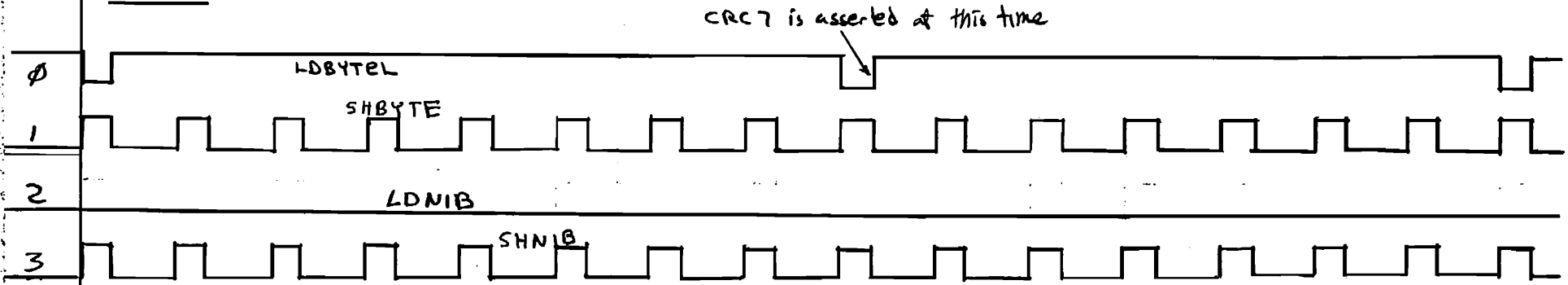


199 ∅ 1 21 41 61 81 93 95 97 99 101 121 141 161 181 193 195 197 199 ∅ 1 21 41 61 81

PROM Address (in base 10) displacement

The above table should appear in the PROM twice. Once beginning at address 512, once at 768
Odd Addresses 713-767 and 969-1023 should assert both LDBYTEL & SHNIB (#8)
Above, 1 flux reversal time = 20 instructions. 1 byte time = 200 instructions
unspecified locations should assert nothing. Thats a #1.

MFM



255 31 63 95 127 159 191 223 255 287 319 351 383 415 447 479

PROM Address (in base 10)

The above table should appear in PROM at the given addresses. Odd Addresses 481-511 should assert both LDBYTEL & SHNIB (#8). 1 flux reversal = 1 bit time = 32 instructions. 1 byte time = 256 instructions.
Unspecified locations should assert nothing: Thats a #1.

10/15/80

WRITE CONTROL PROM

Address dec.	Address hex	contents hex	
31	1F	B	MFM
63	3F	B	
95	5F	B	
127	7F	B	
159	9F	B	
191	BF	B	
223	DF	B	
255	FF	A	
287	11F	B	MFM's last 7 bits of CRC — CRC7 Asserted
319	13F	B	
351	15F	B	
383	17F	B	
415	19F	B	
447	1BF	B	
479	1DF	A	
513	201	D	GCR
533	215	9	
553	229	9	
573	23D	9	
593	251	9	
605	25D	3	
607	25F	3	
609	261	3	
611	263	3	
613	265	D	
633	279	9	
653	28D	9	
673	2A1	9	
693	2B5	9	
705	2C1	3	
707	2C3	3	
709	2C5	3	
711	2C7	2	

Address dec.	Address hex	contents hex	
769	301	D	GCR with CRC7 Asserted
789	315	9	
809	329	9	
829	33D	9	
849	351	9	
861	35D	3	
863	35F	3	
865	361	3	
867	363	3	
869	365	D	
889	379	9	
909	38D	9	
929	3A1	9	
949	3B5	9	
961	3C1	3	
963	3C3	3	
965	3C5	3	
967	3C7	2	

other

1) Odd locations which should never be accessed should contain a "reset" which is the value "#8".

dec	hex	contents
481-511	1E1-1FF	8
713-767	2C9-2FF	8
969-1023	3C9-3FF	8

(odd addresses only)

2) all even addresses and all unspecified addresses = #1

Write Control Prom

10/15/80 REV B1

CTC WRITE CONTROL / WRITE LOGIC 11-18-80

TEXT=WRTCTL1.TEXT BINARY=WRTCTL1.BIN ACCOUNT=UCODE.CTC
RAW PART = 82S137 OR 93453 #0106-0014 OR #0106-0027
BURNED PART = #0121-0112 REV. LEVEL = B1
LOCATION = 11F

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
010:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
020:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
030:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
040:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
050:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
060:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
070:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
080:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
090:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
0A0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0B0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
0C0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0D0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
0E0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0F0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	A
100:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
110:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
120:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
130:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
140:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
150:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
160:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
170:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
180:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
190:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
1A0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1B0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
1C0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1D0:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	A
1E0:	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8
1F0:	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8
200:	1	D	1	1	1	1	1	1	1	1	1	1	1	1	1	1
210:	1	1	1	1	1	9	1	1	1	1	1	1	1	1	1	1
220:	1	1	1	1	1	1	1	1	1	9	1	1	1	1	1	1
230:	1	1	1	1	1	1	1	1	1	1	1	1	1	9	1	1
240:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
250:	1	9	1	1	1	1	1	1	1	1	1	1	1	3	1	3
260:	1	3	1	3	1	D	1	1	1	1	1	1	1	1	1	1
270:	1	1	1	1	1	1	1	1	1	9	1	1	1	1	1	1
280:	1	1	1	1	1	1	1	1	1	1	1	1	1	9	1	1
290:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2A0:	1	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2B0:	1	1	1	1	1	9	1	1	1	1	1	1	1	1	1	1
2C0:	1	3	1	3	1	3	1	2	1	8	1	8	1	8	1	8
2D0:	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2E0:	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8
2F0:	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8
300:	1	D	1	1	1	1	1	1	1	1	1	1	1	1	1	1
310:	1	1	1	1	1	9	1	1	1	1	1	1	1	1	1	1
320:	1	1	1	1	1	1	1	1	1	9	1	1	1	1	1	1
330:	1	1	1	1	1	1	1	1	1	1	1	1	1	9	1	1
340:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
350:	1	9	1	1	1	1	1	1	1	1	1	1	1	3	1	3
360:	1	3	1	3	1	D	1	1	1	1	1	1	1	1	1	1
370:	1	1	1	1	1	1	1	1	1	9	1	1	1	1	1	1
380:	1	1	1	1	1	1	1	1	1	1	1	1	1	9	1	1
390:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3A0:	1	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3B0:	1	1	1	1	1	9	1	1	1	1	1	1	1	1	1	1
3C0:	1	3	1	3	1	3	1	3	1	8	1	8	1	8	1	8
3D0:	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8
3E0:	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8
3F0:	1	8	1	8	1	8	1	8	1	8	1	8	1	8	1	8

REVISIONS				
LET-TER	DESCRIPTION	DATE	CHNG	APPV'D
A		11/17/76	Lg	WWV
B		2/1/77	XP	WWV
C		4/25/77	Dg	WWV

Received May 2, 1977

DEI DATA ELECTRONICS, INC.			
DRAWN: Dg CHECKED: APPROVED: WWV 11/15/76	TITLE: Data Format and Code Specification		
	SIZE: A	DOCUMENT NO: 301,177	SCALE SHT 1 OF 5
APPLICATION			

1.0 SCOPE

This specification covers the encoding, decoding, and format specifications for the 6400 bpi drive and cartridge. It shall be used in conjunction with the Drive Specification to define all aspects of the 6400 bpi formats.

2.0 ENCODING

2.1 Write Data Code:

The code used shall be Modified Frequency Modulation (MFM) which follows these definitions:

- A 'one' is defined by a flux change in the center of the bit cell (a bit cell period is 5.208×10^{-6} sec.).
- A 'zero' shall not require a flux change unless there are two or more sequential zeros whereupon a flux change shall occur between the zeros bit cells.
- The nominal input data frequency is 192,000 bits/sec.

2.2 Pre-Stressing:

The write data delivered to the drive shall be capable of being shifted in time to partially compensate for the recovered data phase error. The 'pre-stress' (time shift) is $\pm .488\mu$sec, $\pm .04\mu$ sec. (3/32 bit cell).

When the adjacent periods to any flux transition are nominally ~~constant~~ equal, no stressing is required for the given transition. Nominally constant is defined without stressing considerations.

When changing from a 2T period to a longer period the intervening transition shall be shifted so as to occur $.488\mu$sec (3/32 of a bit cell) early. When changing from a long period (3T or 4T) to a 2T period, the intervening transition shall be shifted so as to occur $.488\mu$sec (3/32 bit cell) late.

2.3 Equalization:

After the input data is encoded and pre-stressed, the following modulation pattern is to be added by the encoder. There are three possible periods between the sequential flux transition: 2T, 3T, and 4T. At 192,000 bits/sec. (6400 bpi), 1T is 2.604μsec. The rule for equalization is:

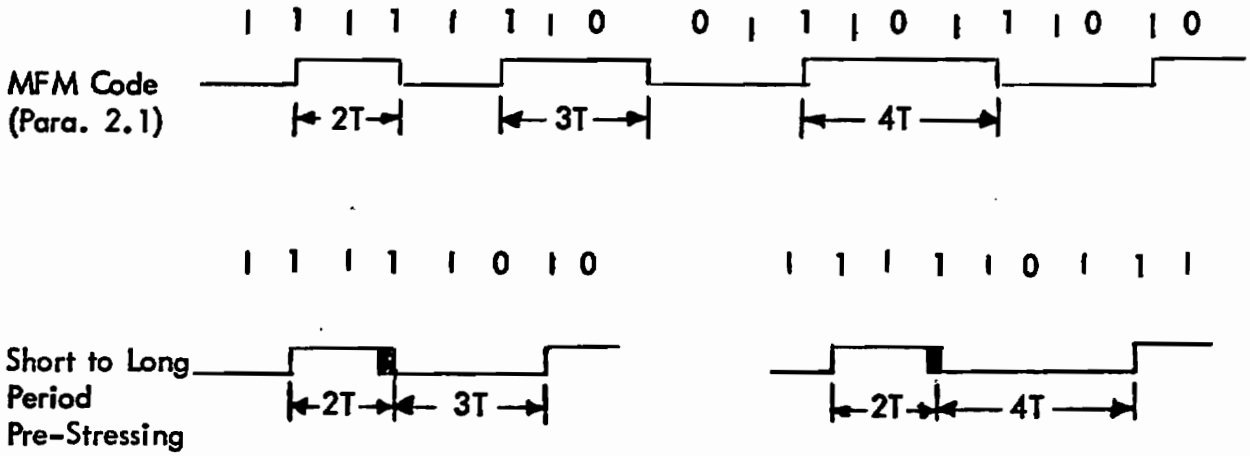
- The 2T period shall be unchanged.

- The 3T period shall have two (2) additional flux transitions appended to it as follows: within the 3T period two transitions are added starting at 4.88μsec (15/16 of a cell) as measured from the transition which begins the 3T period. The period between these two additional flux transitions is .814μsec. (5/32 of a cell).

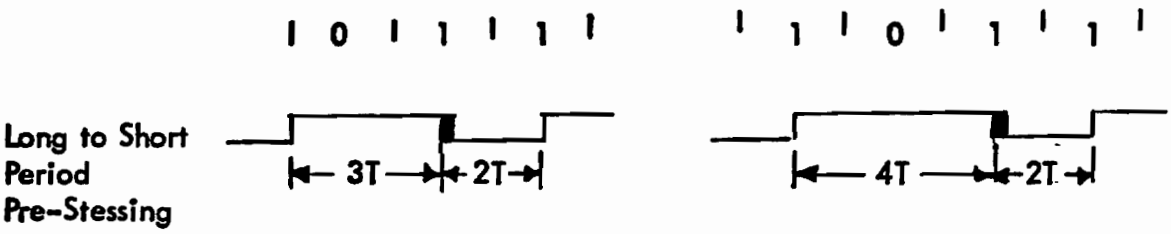
- The 4T period shall have four (4) additional flux transitions appended to it as follows: within the 4T period the leading edge of the first transition pair shall occur at 4.88μsec. (15/16 of a cell) as measured from the transition which begins the 4T period. The leading edge of the second transition pair shall occur at 2.604μsec. (1/2 of a cell) as measured from the leading edge of the first transition pair. The period between the transitions in a pair is .814μsec. (5/32 of a cell).

- In the absence of any Write Data a symmetrical ~~1.30~~μsec. (~~1/4~~ of a cell) period frequency shall be placed on the WDA+/WDA- signal lines. The encoder shall provide this 1.30μsec. period signal 4.069μsec. (25/32 of a cell) after the last zero bit of the postamble is sent to the drive.

2.4 Example:



(Note: Shaded areas represent early pre-stressing period.)



(Note: Shaded areas represent late pre-stressing period.)

3.0 DECODING

3.1 Read Data:

The data returned from the drive shall be a replica of the encoded data except:

1) The equalization signals which were added shall not be returned.

2) The pre-stress shall have been essentially eliminated and in its place shall be up to the maximum drive portion of phase error: ±33% of the recovery window, ± .434μsec.

3) The data will additionally have total speed or time base error of up to maximum of 40% peak to peak (±14% short term; 6% long term). The bandwidth of these data rate changes is DC to 200 Hz (half power point).

4) The DAD signal will go true ≤16μsec. after the first transition being read from tape. The DAD signal may stay true up to 62μsec. after the last transition is read from the tape. During this period the Read Data Signal may contain apparent spurious transitions. The Read Data Signal will consist of spurious oscillations when not in data.

   A digital filter must be provided to prevent these and possibly other non-data transitions which can occur in the data pattern from passing to the data separator.

4.0 FORMAT

4.1 Pre and Postambles:

An appended non-data bearing pattern of at least thirty-nine (39) 'ones' followed by one (1) 'zero' shall be added to the beginning of each data block. This pattern is called preamble. The reverse pattern (i.e., one (1) 'zero' followed by at least thirty-nine (39) 'ones' shall be added to the end of each data block. This pattern is called a postamble.

Hence a data block is:

≥ 39 'ones'; 1 'zero', Data, 1 'zero', and ≥ 39 'ones'.

4.2* Check Characters:

Any check characters shall be agreed to by the interchanging parties.

4.3* Minimum Data Block Size:

A data block shall contain at least 128 data bits.

4.4* Tape Mark:

A tape mark shall consist of a preamble, 8 data bits and a postamble.

4.5 The Interblock Gap:

The minimum interblock gap (IBG) shall be 1.2 inches. The interblock gaps shall be erased using an A.C. type erase technique.

4.6 Number of Tracks:

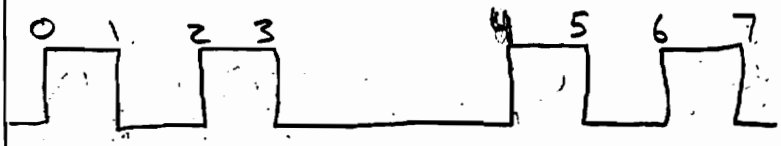
There shall be 4 recordable tracks.

*These items are not a part of the data reliability standard practice requirements and hence other practices are acceptable.

GCR PRESTRESS

(no equalization / no prestress)

1111001111 = TEST PATTERN(5A)



1) +1	+3	+3	+3	+3
2) -3	-3	-2	-1	-1
3) +11	+13	+15	+14	+13
4) -14	-11	-10	-7	-8
5) +3	+4	+4	+4	+5
6) -5	-3	-3	-1	-2
7) +2	+3	+4	+4	+4
-5	+6	+11	+16	+14

← jitter in the tape!

Assuming 0 is in the right place, each successive flux transition is displaced by x/32 of a bit cell, as shown in the table at left, in 5 successive samples.

It is clear that flux transition #3 needs to be moved backwards in time by approx 12/32's of a bit cell. or 7/20's of a bit cell.

[I measured only the distance between adjacent flux transitions]

WRITING A FLUX TRANSITION PUSHES THE LAST ONE BACK, HOW FAR DEPENDS ON HOW CLOSE IT IS!

#3 is right where it should be now

1) +4	+3	+5
2) -7	-8	-11
3) +6	+4	+6
4) +4	-1	-2
5) +5	+3	+4
6) -2	-4	-4
7) +4	+2	+2
+14	-1	0

and 1 behind

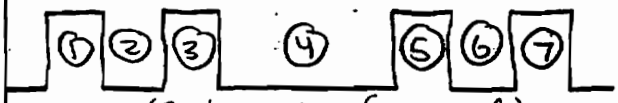
Pushing #3 back in time crowded #2 so it also moved back.

It seems that what counts in pre-stress is the flux transitions that come after, rather than before. If you think about the head on the tape, this makes some sense that the next bit written will affect the one just behind while the one just behind will not affect the one being written. This suggests having the prestress prom look at the 3 bits ahead rather than 2 ahead + 2 behind. First, lets check out the equalization though.

GCR Prestress

forward + reverse, no equalization

1111001111 = Test Pattern (#5A)



(Read moving forward)

+1	-3	+11	-14	+3	-5	+2
+3	-3	+13	-11	+4	-3	+3
+3	-2	+15	-10	+4	-3	+4
+3	-1	+14	-7	+4	-1	+4
+3	-1	+13	-8	+5	-2	+4

(Read moving reverse)

+3	-2	+7	-9	+10	-3	+3
+3	-2	+7	-7	+10	-2	+3
+3	-2	+7	-7	+10	-2	+3
+3	-2	+8	-8	+10	-3	+3
+3	-3	+6	-9	+10	-4	+3

11011101011 = Test Pattern = #FF

1	2	3	4	5	6	7
+12	-6	+4	-3	+12	-6	+4
+12	-9	+3	-4	+11	-8	+3
+11	-8	+2	-3	+10	-8	+3
+12	-8	+3	-3	+11	-7	+3
+12	-6	+4	-3	+13	-6	+4
+10	-11	+2	-5	+11	-11	+2

7	6	5	4	3	2	1
+6	-5	+9	-3	+7	-5	+9
+6	-5	+7	-3	+6	-6	+8
+6	-6	+8	-3	+5	-6	+9
+6	-5	+8	-3	+6	-6	+9
+7	-6	+9	-4	+6	-6	+9
+8	-4	+10	-2	+7	-4	+10

LOOKING AT THE ANALOG WAVEFORM, FORWARD + REVERSE, THE FORWARD WAVE FORM IS DISTORTED, THE REVERSE WAVEFORM IS IDEAL (not the location of the zero crossings, but the aesthetic value of the waveform). One might consider that there is write distortion and forward read distortion that is additive. The reverse read distortion is subtractive.

11010010011 = Test Pattern = #24

1	2	3	4	5
+9	+5	+1	-4	+1
+10	+4	+2	-4	+1
+10	+5	+2	-2	+2
+7	+1	-5	-8	-1
+9	+3	0	-4	0

5	4	3	2	1
+3	-1	+1	+3	+6
+3	-1	+1	+3	+6
+4	-1	+2	+3	+6
+4	-1	+3	+4	+6
+4	-1	+2	+3	+7

Data is the error in the distance between 2 flux transitions given in 10ths of a μsecond.

ALSO TRY #00 #18 #08 #11

GCR Prestress

(with equalization)

111100|111 = Test Pattern (5A)

equalization centered when flux transition should be and 4/20's of a bit cell wide

1) +2	+2	+2
2) -1	-1	-1
3) +9	+10	+10
4) -6	-8	-6
5) +4	+4	+4
6) -2	-2	-2
7) +2	+3	+3
+8	+8	+10

This equalization ~~prestress~~ has an expected small effect on the prestress that is needed.

From this small sample though, it would appear the data recovery is better behaved (less jitter) } probably co-incidence!

I guess equalization will have to be done by inspecting the linear waveform and attempting to get the prettiest one.

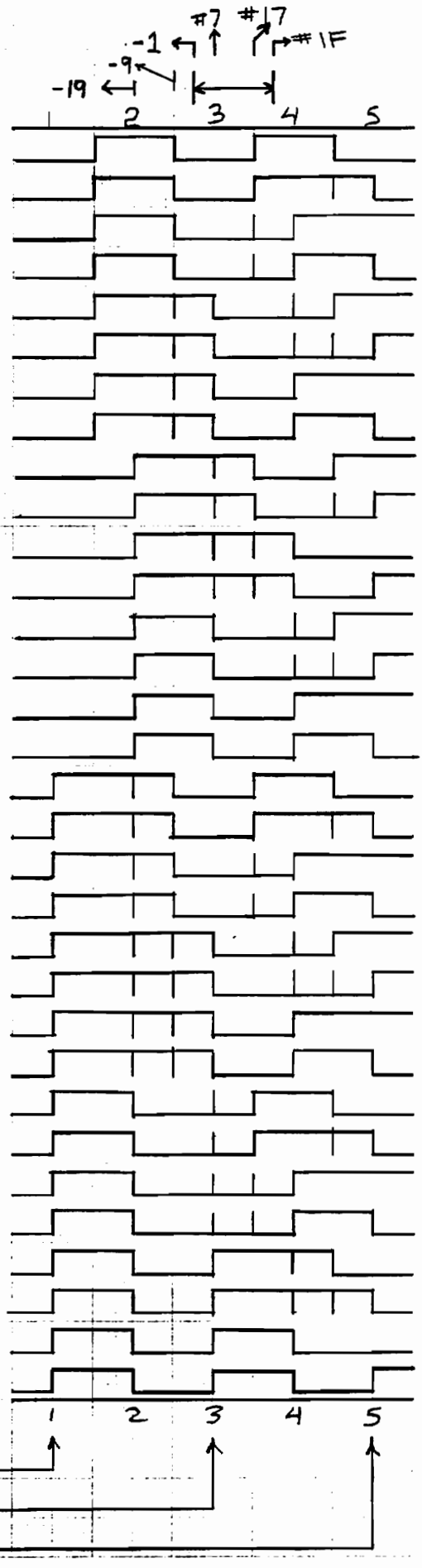
Inspection of the linear waveform prior to digitization during equalization experimenting yeilded this:

1) during a 2 zero sequence, a 3/20's bit cell with equalization pulse at about 2/3 of the way across the cell cleaned up the waveform to almost ideal
2) reverse reading will require completely different equalization

Decision: forget about equalization + work at a prestress table that is a compromise between what forward + reverse need, as if appear their needs are different. In any case, the ~~drop~~ droop between flux transitions in GCR is much less than that observed for MFM which works well + has for years.

MFM

Bits 1	2	3	4	5	where flux transition nominally should occur	Prestress	Result	Location of equalization flux transitions
0	0	0	0	0	17	0	17	
0	0	0	0	1	17	-3	14	
0	0	0	1	0				12,17
0	0	0	1	1				12,17
0	0	1	0	0	07	0	07	
0	0	1	0	1	07	0	07	
0	0	1	1	0	07	+3	0A	
0	0	1	1	1	07	+3	0A	
0	1	0	0	0	17	+3	1A	05,0A
0	1	0	0	1	17	0	17	05,0A
0	1	0	1	0				05,0A 15,1A
0	1	0	1	1				05,0A 15,1A
0	1	1	0	0	07	-3	04	
0	1	1	0	1	07	-3	04	
0	1	1	1	0	07	0	07	
0	1	1	1	1	07	0	07	
1	0	0	0	0	17	0	17	
1	0	0	0	1	17	-3	14	
1	0	0	1	0				15,1A
1	0	0	1	1				15,1A
1	0	1	0	0	07	0	07	
1	0	1	0	1	07	0	07	
1	0	1	1	0	07	+3	0A	
1	0	1	1	1	07	+3	0A	
1	1	0	0	0	17	+3	1A	02,07
1	1	0	0	1	17	0	17	02,07
1	1	0	1	0				02,07 12,17
1	1	0	1	1				02,07 12,17
1	1	1	0	0	07	-3	04	
1	1	1	0	1	07	-3	04	
1	1	1	1	0	07	0	07	
1	1	1	1	1	07	0	07	



Prestress From 3-3-81 Rev B@5

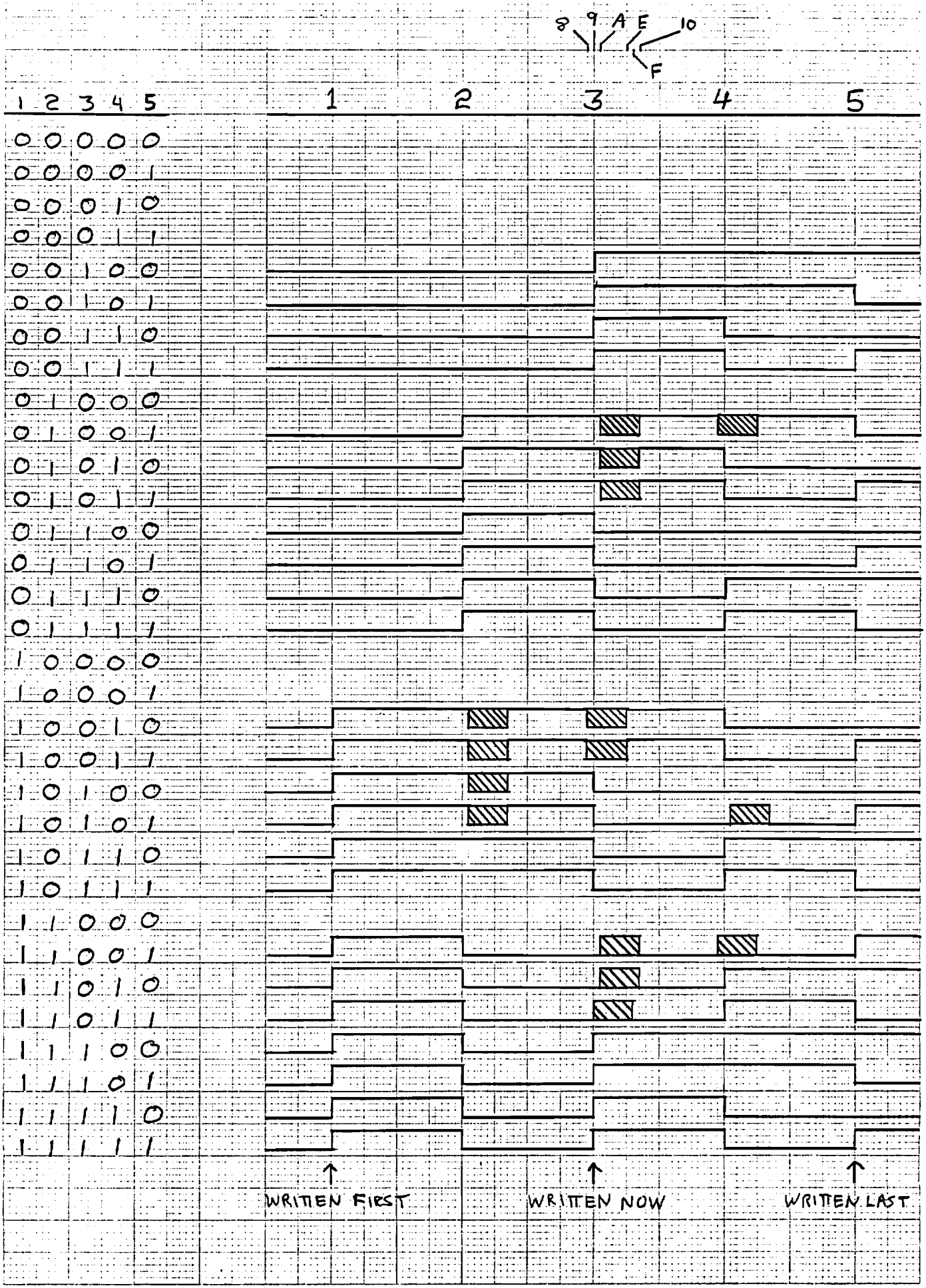
MFM

First bit N-1	now bit N	last bit N+1	XT4	XT3	XT2	XT1	XT0	Real PROM Address	Contents
0	0	0	1	0	1	1	1	05	8
0	0	0	1	0	1	0	0	0D	1
0	0	0	1	0	0	1	0	14	4
0	0	0	1	0	0	1	1	15	8
0	0	0	1	1	0	1	0	1C	4
0	0	0	1	1	0	1	1	1D	8
0	0	1	0	0	0	1	1	21	8
0	0	1	0	0	1	1	1	29	8
0	0	1	1	0	0	1	0	32	4
0	0	1	1	0	0	1	1	3A	4
0	1	0	0	0	1	1	0	46	2
0	1	0	0	0	0	1	1	41	4
0	1	0	0	0	0	1	0	42	8
0	1	0	0	0	1	1	1	4D	2
0	1	0	0	1	0	0	1	49	4
0	1	0	0	1	0	1	0	4A	2
0	1	0	1	0	0	0	1	51	4
0	1	0	1	0	0	1	0	52	2
0	1	0	1	0	1	0	1	55	4
0	1	0	1	0	1	1	0	56	2
0	1	0	1	1	0	0	1	59	4
0	1	0	1	1	0	1	0	5A	2
0	1	0	1	1	1	0	1	5D	4
0	1	0	1	1	1	1	0	5E	1
0	1	1	0	0	0	0	1	61	2
0	1	1	0	0	0	0	0	69	8
0	1	1	0	0	0	1	1	71	8
0	1	1	1	0	0	1	1	79	8
1	0	0	0	0	1	1	1	85	8
1	0	0	0	1	1	0	1	8D	1
1	0	0	1	0	1	0	1	95	2
1	0	0	1	0	1	1	0	96	4
1	0	0	1	1	1	0	1	9D	2
1	0	0	1	1	1	1	0	9E	4
1	0	1	0	0	0	0	1	A1	8
1	0	1	0	0	0	1	1	A9	8
1	0	1	1	0	0	1	0	B2	4
1	0	1	1	0	1	1	0	BA	4
1	1	0	0	0	1	1	0	C6	4
1	1	0	0	0	0	1	0	C0	8
1	1	0	0	0	1	1	1	C1	8
1	1	0	0	1	1	0	1	CD	4
1	1	0	0	1	0	0	0	C8	8
1	1	0	0	1	0	0	1	C9	8

First Bit N-1	now BIT N	LAST Bit N+1	XT4	XT3	XT2	XT1	XT0	Real PROM Address	Contents
1	0	1	0	0	0	1	0	D0	4
1	0	1	0	0	1	1	0	D1	8
1	0	1	0	1	0	1	0	D4	4
1	0	1	0	1	1	1	0	D5	8
1	0	1	1	0	0	1	0	D8	4
1	0	1	1	0	1	1	0	D9	8
1	0	1	1	1	0	1	0	DC	4
1	0	1	1	1	1	1	0	DD	8
1	1	0	0	0	1	1	0	E1	8
1	1	0	0	1	1	1	0	E9	8
1	1	0	1	0	1	1	0	F1	8
1	1	0	1	1	1	1	0	F9	8

Prestress Prom 3-3-81 Rev B5

GCR Pre-stress & Equalization Rev 5, 9-81



*

First bit N-1	Now bit N	Last bit N+1	XT4	XT3	XT2	XT1	XT0	Real Prom Address	Contents	
0	0	1	0	0	0	1	0	122	2	
0	0	1	0	1	0	1	0	12A	2	
0	0	1	1	0	0	1	0	132	2	
0	0	1	1	1	0	1	0	13A	2	
0	1	0	0	1	0	1	0	14A	4	*
0	1	0	0	1	1	0	0	14C	1	*
0	1	0	1	0	0	1	0	152	4	*
0	1	0	1	0	1	0	0	154	1	*
0	1	0	1	1	0	1	0	15A	4	*
0	1	0	1	1	1	0	0	15C	1	*
0	1	1	0	0	0	1	0	162	2	
0	1	1	0	1	0	1	0	16A	2	
0	1	1	1	0	0	1	0	172	2	
0	1	1	1	1	0	1	0	17A	2	
1	0	0	1	0	0	1	0	192	1	*
1	0	0	1	0	0	1	1	193	4	*
1	0	0	1	1	0	1	0	19A	1	*
1	0	0	1	1	0	1	1	19B	4	*
1	0	1	0	0	0	1	0	1A2	2	
1	0	1	0	1	0	1	0	1AA	2	
1	0	1	1	0	0	1	0	1B2	2	
1	0	1	1	1	0	1	0	1BA	2	
1	1	0	0	1	0	1	0	1CA	4	*
1	1	0	0	1	1	0	0	1CC	1	*
1	1	0	1	0	0	1	0	1D2	4	*
1	1	0	1	0	1	0	0	1D4	1	*
1	1	0	1	1	0	1	0	1DA	2	*
1	1	0	1	1	0	1	1	1DB	8	*
1	1	1	0	0	0	1	0	1E2	2	
1	1	1	0	1	0	1	0	1EA	2	
1	1	1	1	0	0	1	0	1F2	2	
1	1	1	1	1	0	1	0	1FA	2	

The * on this page indicate terms which are equalization pulses shown by ▨ on the previous page. Note no prestress is used.

The "1" of "122" in the real Prom Address is the high density (GCR) bit

The * on the previous page is the one equalization pulse which is not located exactly where all the others are.

Note not all equalization pulses on previous page are shown.

The flux transitions on the previous page are shown exactly where they occur, with the nominal location at the time when:

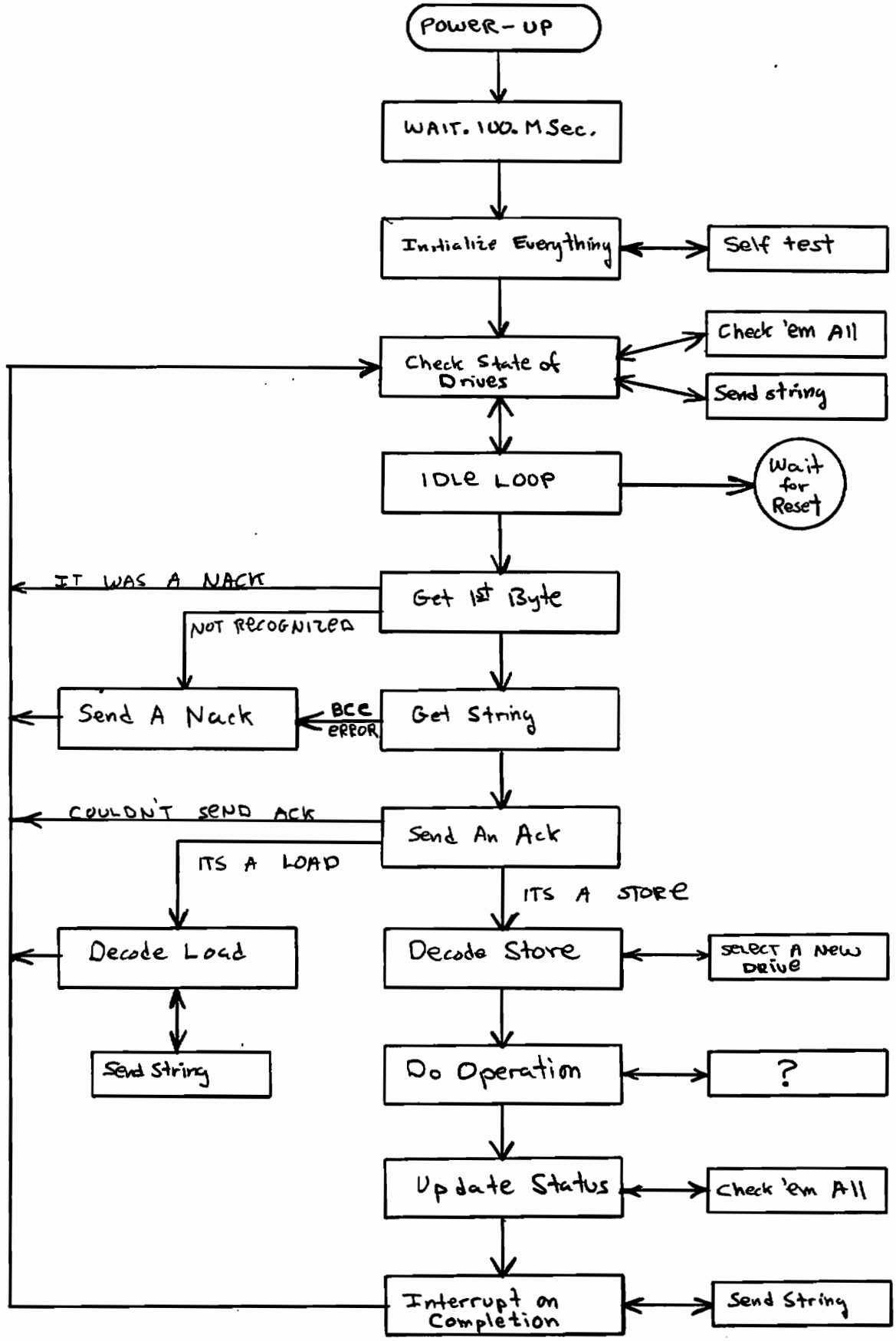
XT4, XT3, XT2, XT1, XT0 = 01001 = 9

CTC PRESTRESS / WRITE LOGIC 9-17-81

TEXT=PRSTRS5.TEXT BINARY=PRSTRS5.BIN ACCOUNT=UCODE.CTC
RAW PART = 82S131 [512*4] #0106-0010
BURNED PART = #0121-0110 REV. LEVEL = B5
LOCATION = 12F

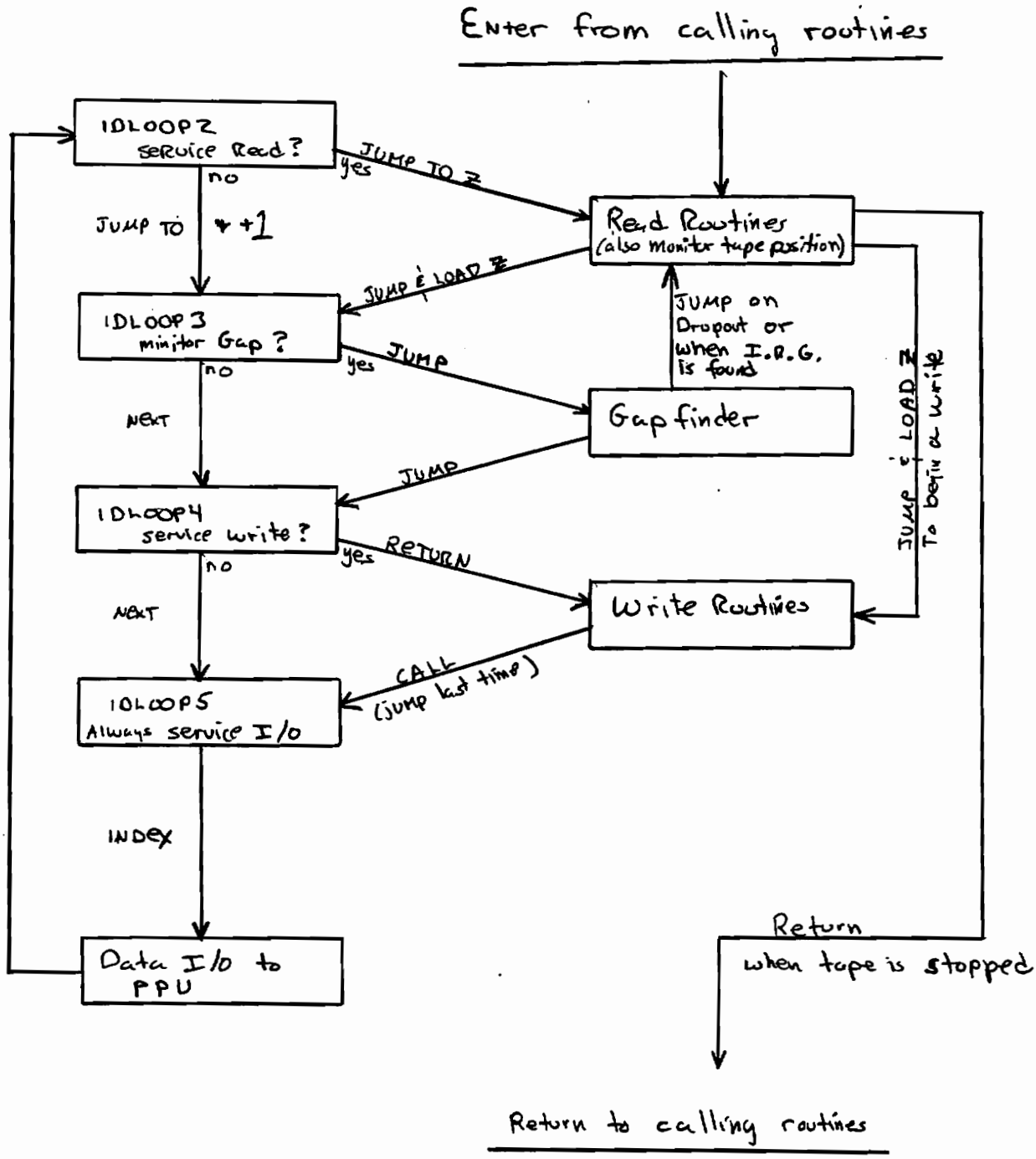
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000:	0	0	0	0	0	8	0	0	0	0	0	0	0	1	0	0
010:	0	0	0	0	4	8	0	0	0	0	0	0	4	8	0	0
020:	0	8	0	0	0	0	0	0	0	8	0	0	0	0	0	0
030:	0	0	4	0	0	0	0	0	0	0	4	0	0	0	0	0
040:	0	2	4	0	0	0	4	0	0	2	4	0	0	8	0	0
050:	0	2	4	0	0	2	4	0	0	2	4	0	0	2	4	0
060:	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
070:	0	8	0	0	0	0	0	0	0	8	0	0	0	0	0	0
080:	0	0	0	0	0	8	0	0	0	0	0	0	0	1	0	0
090:	0	0	0	0	0	2	4	0	0	0	0	0	0	2	4	0
0A0:	0	8	0	0	0	0	0	0	0	8	0	0	0	0	0	0
0B0:	0	0	4	0	0	0	0	0	0	0	4	0	0	0	0	0
0C0:	4	8	0	0	0	0	4	0	4	8	0	0	0	8	0	0
0D0:	4	8	0	0	4	8	0	0	4	8	0	0	4	8	0	0
0E0:	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0F0:	0	8	0	0	0	0	0	0	0	8	0	0	0	0	0	0
100:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
110:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
120:	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0
130:	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0
140:	0	0	0	0	0	0	0	0	0	0	4	0	1	0	0	0
150:	0	0	4	0	1	0	0	0	0	0	4	0	1	0	0	0
160:	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0
170:	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0
180:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
190:	0	0	1	4	0	0	0	0	0	0	1	4	0	0	0	0
1A0:	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0
1B0:	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0
1C0:	0	0	0	0	0	0	0	0	0	0	4	0	1	0	0	0
1D0:	0	0	4	0	1	0	0	0	0	0	2	8	0	0	0	0
1E0:	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0
1F0:	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0

Top level CTC Microcode Structure

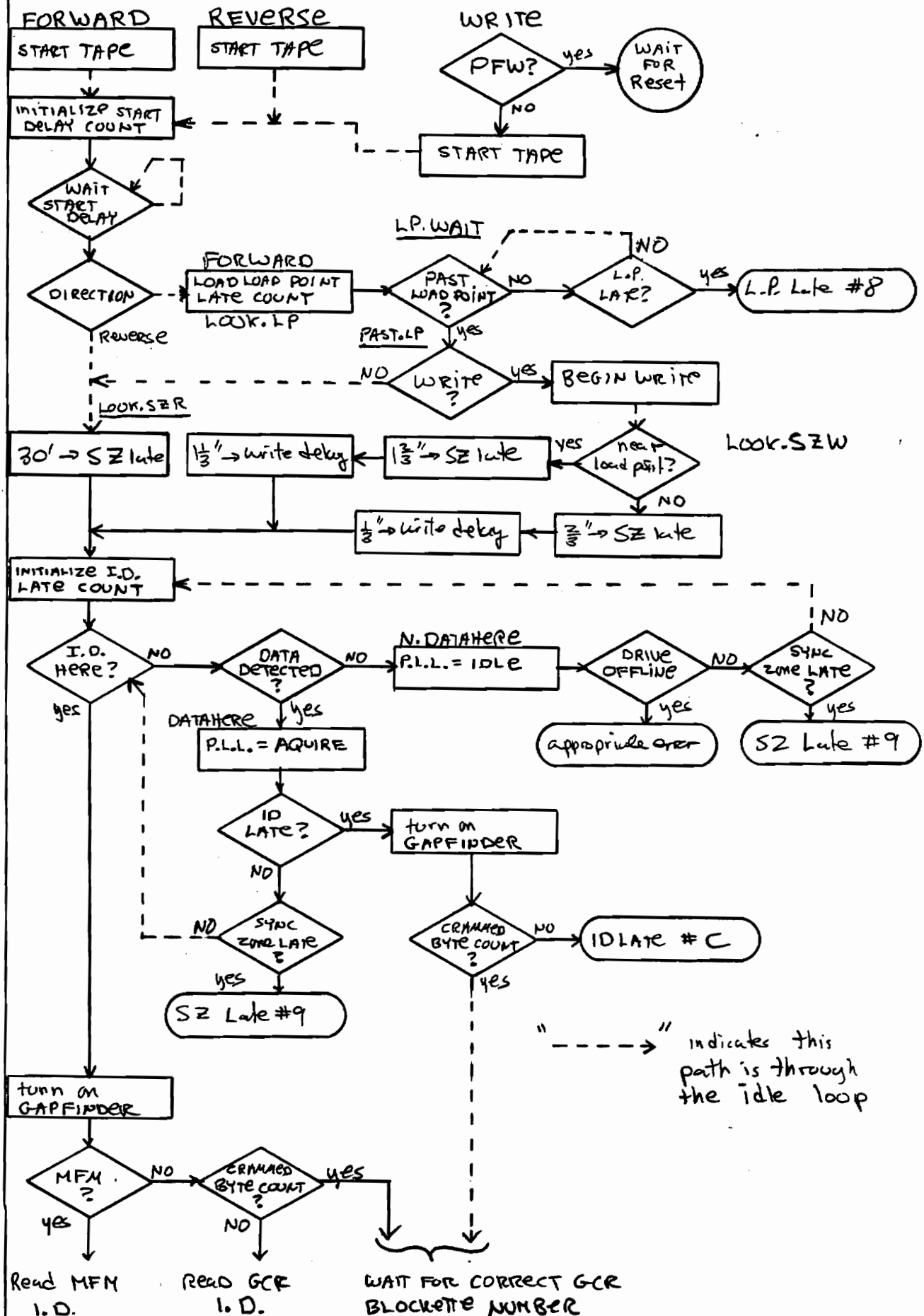


CTCC1

Read, Write, Skip, Search File Mark operations use the following structure to pseudo-simultaneously monitor tape position, write data, read data, monitor dropouts, transfer data to/from the PPU.



Read Routines : | START TILL Decode ID.



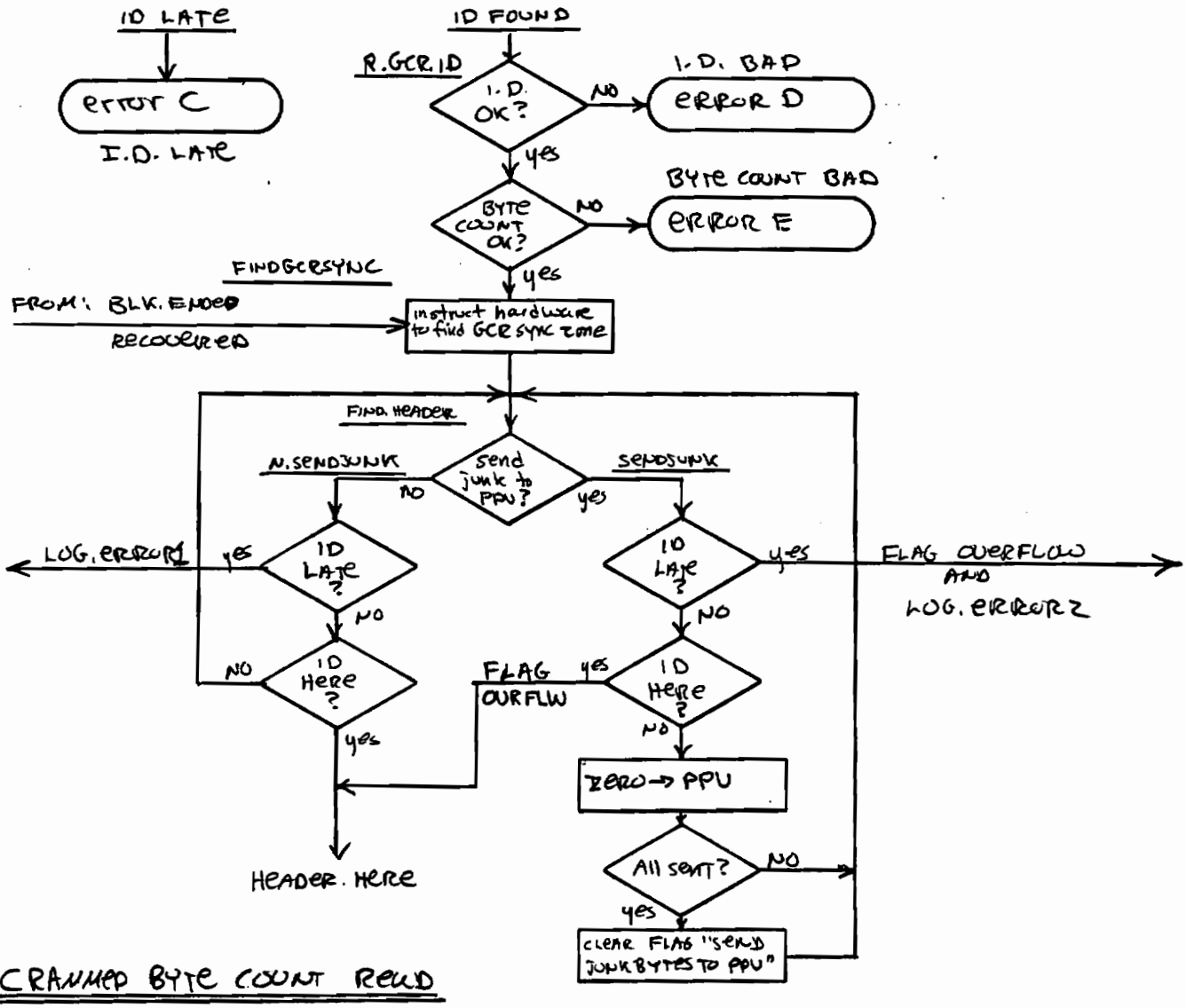
"- - - - →" indicates this path is through the idle loop

42-381 50 SHEETS 5 SQUARE
42-382 100 SHEETS 5 SQUARE
42-389 200 SHEETS 5 SQUARE
NATIONAL

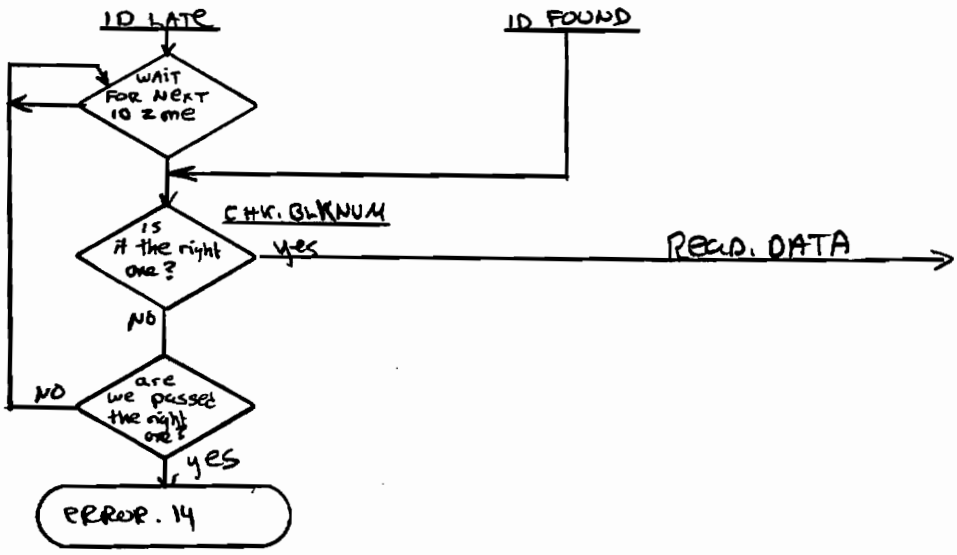
CTCC1

9-10-81

NORMAL GCR READ

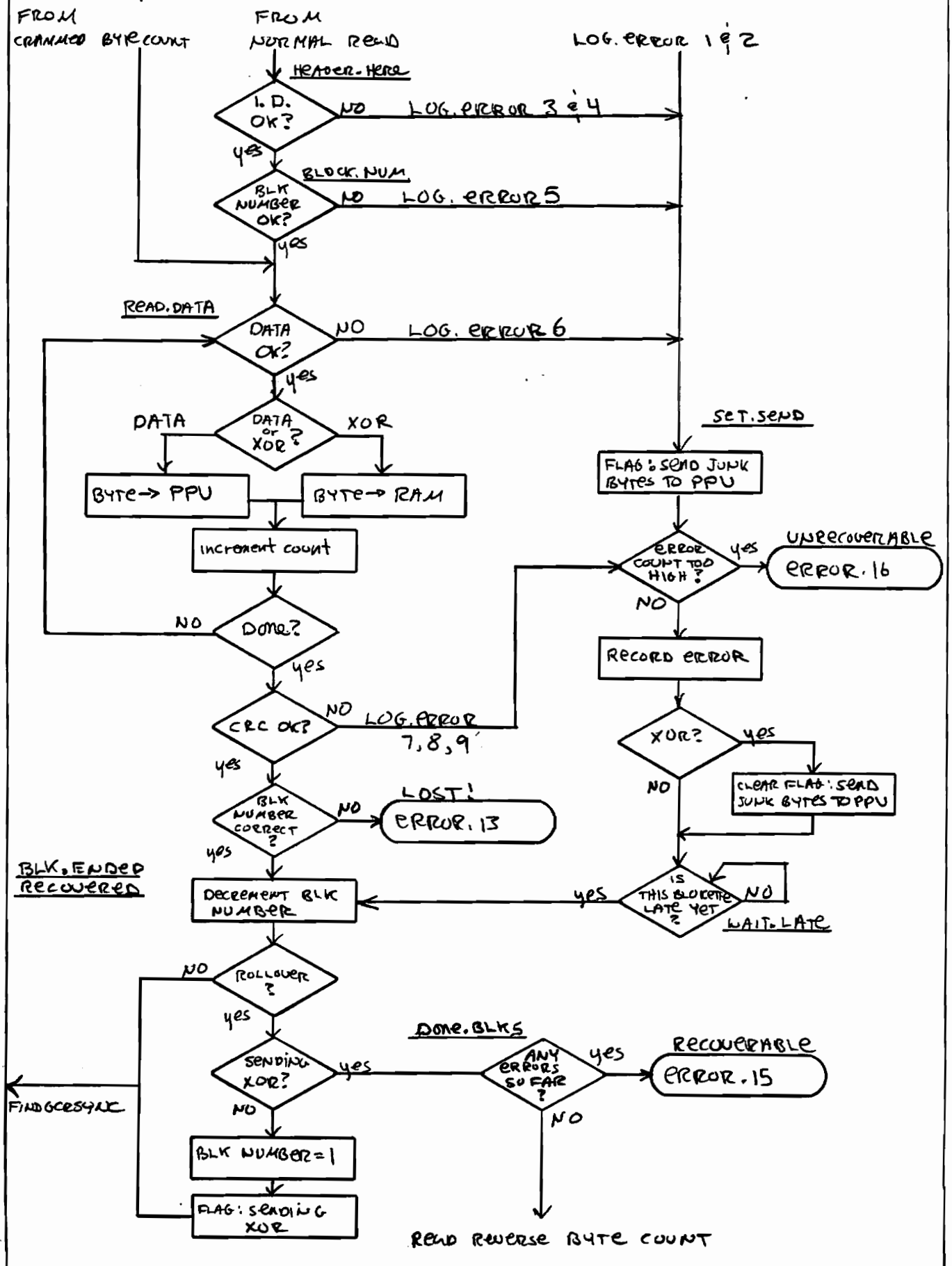


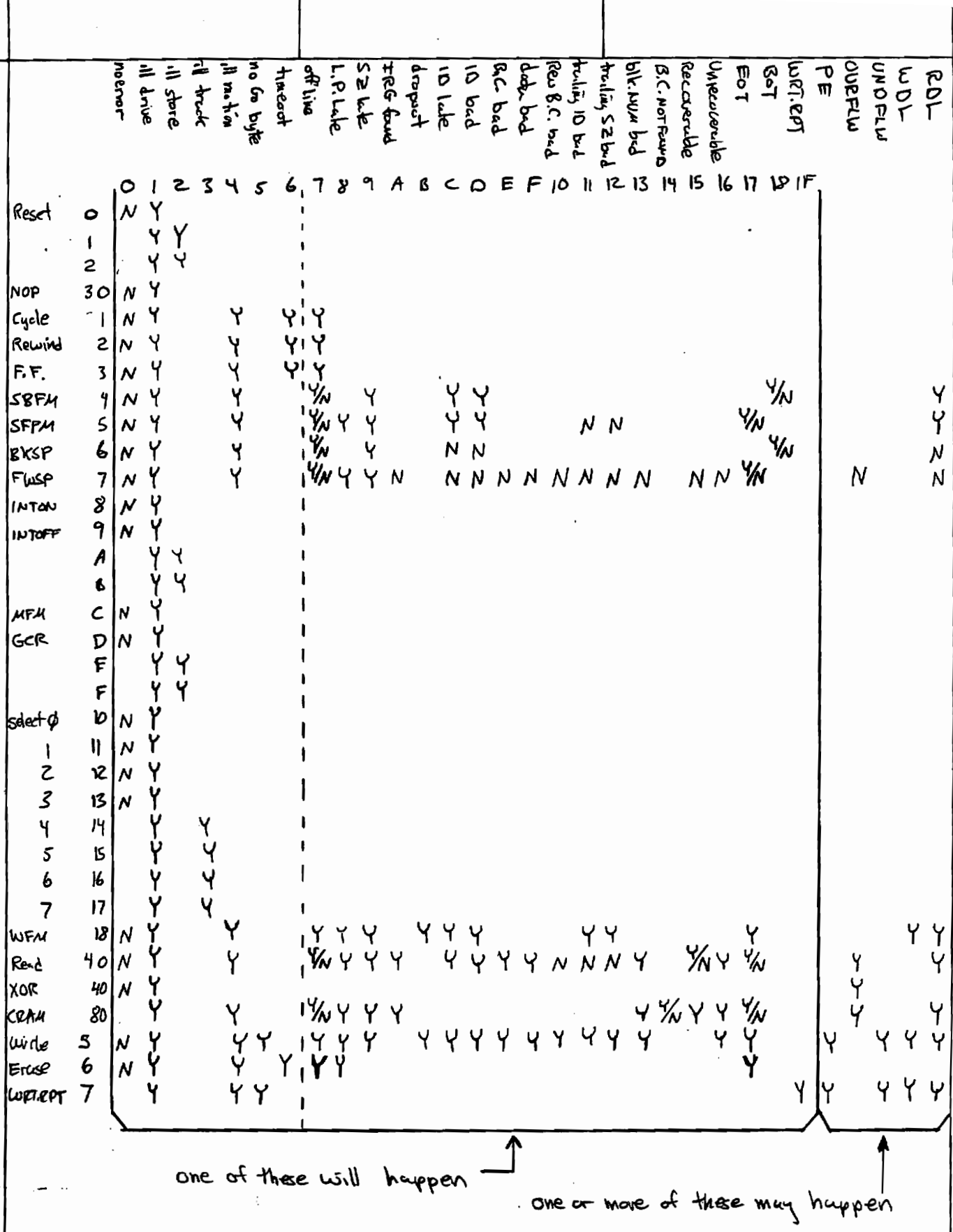
CRAMMED BYTE COUNT READ



GCR Reads

CTCC1 9/20/81





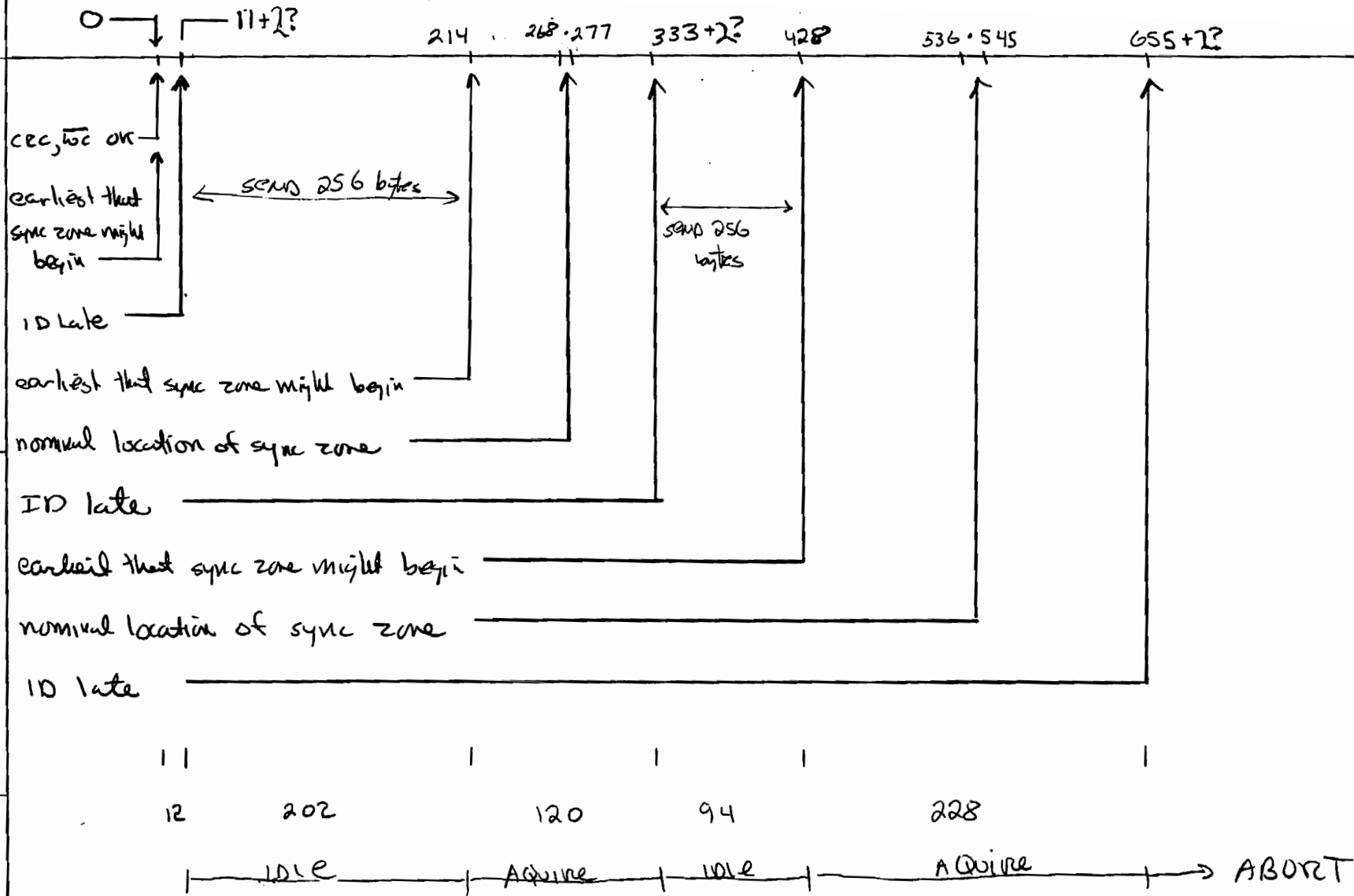
CTCC1

CTCC1

What errors may happen with each store and whether or not (Y/N) it will set the error bit.

ERROR 15 will not set the error bit if it is only an XOR error
error 14 will not set the error bit if the command byte count = 0000.
error 7, 17, 18 set the error bit if they replace errors which set the error bit, or if its a write operation

9-12-81



when CRC, $\overline{WC}$ is OKAY, LOAD 13 → ID LATE. (#142)
each time a blockette is found to be bad or is not findable, add 822 to the ID late counter, and do the cleanup and search for the new blockette. Timing of earliest sync zone may be found will not be done.

TIMERS FOR RE-SYNCING GCR

Lets suppose exactly 2 DATA Blockettes Drop out. The CRC of a blockette is OKAY. THE SYNC ZONE OF The 3rd blockette following is OKAY. All in between is garbage.

In theory, the 1st Byte of that 3rd blockette's sync zone is the 537th byte following the OKAY CRC byte. The Blockette header is the 545th byte. If tape speed was always 100% accurate, we should try to re-sync after 536 bytes, note the header is late after 546 bytes.

We can time these bytes using RTC and noting it may differ from actual tape speed by ±20%.

Thus, $536 - 20\% = 428$ bytes till sync zone can 1st happen
$546 + 20\% = 655 + 1 = 656$ bytes till header is late.

That "+1" is because the counter that counts RTC's may be off by +1, -0.

Now lets look for the 2nd Blockettes sync zone.
nominally, its 269th byte and the header is 277th byte.

$268 - 20\% = 214$
$277 + 20\% = 333 + 1 = 334$

Now lets look for the 1st Blockettes sync zone
nominally, in the 1st byte. The header is 9th byte

$0 - 20\% = 0$ till sync zone can happen
$9 + 20\% = 11 + 1 = 12$ till header late

Add one more to the ID-late's ① for margin
② cuz I check ID late before I check ID here!