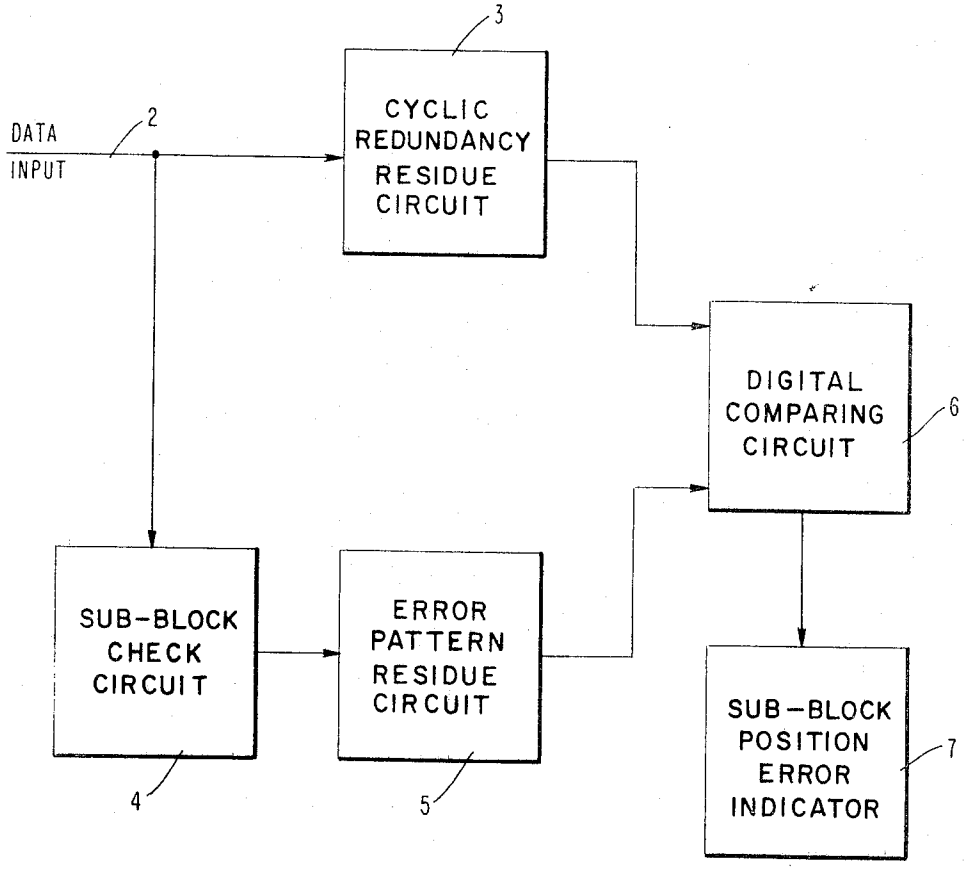
April 21, 1970 D. T. BROWN 3,508,194

ERROR DETECTION AND CORRECTION SYSTEM

Filed April 6, 1964 3 Sheets-Sheet 1

FIG. 1




INVENTOR
DAVID T. BROWN

BY

Bernard M. Goldman
ATTORNEY

FIG. 2

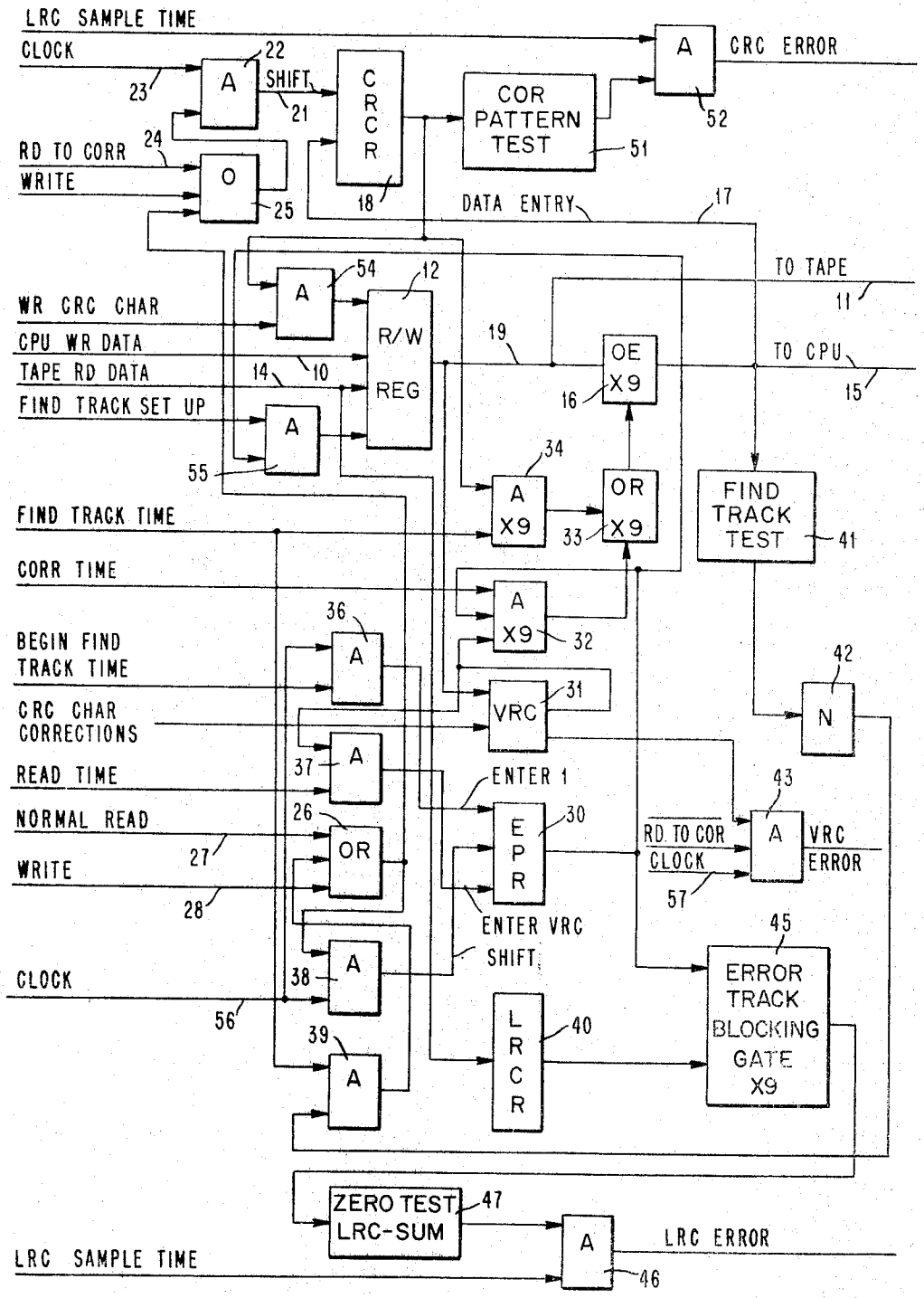
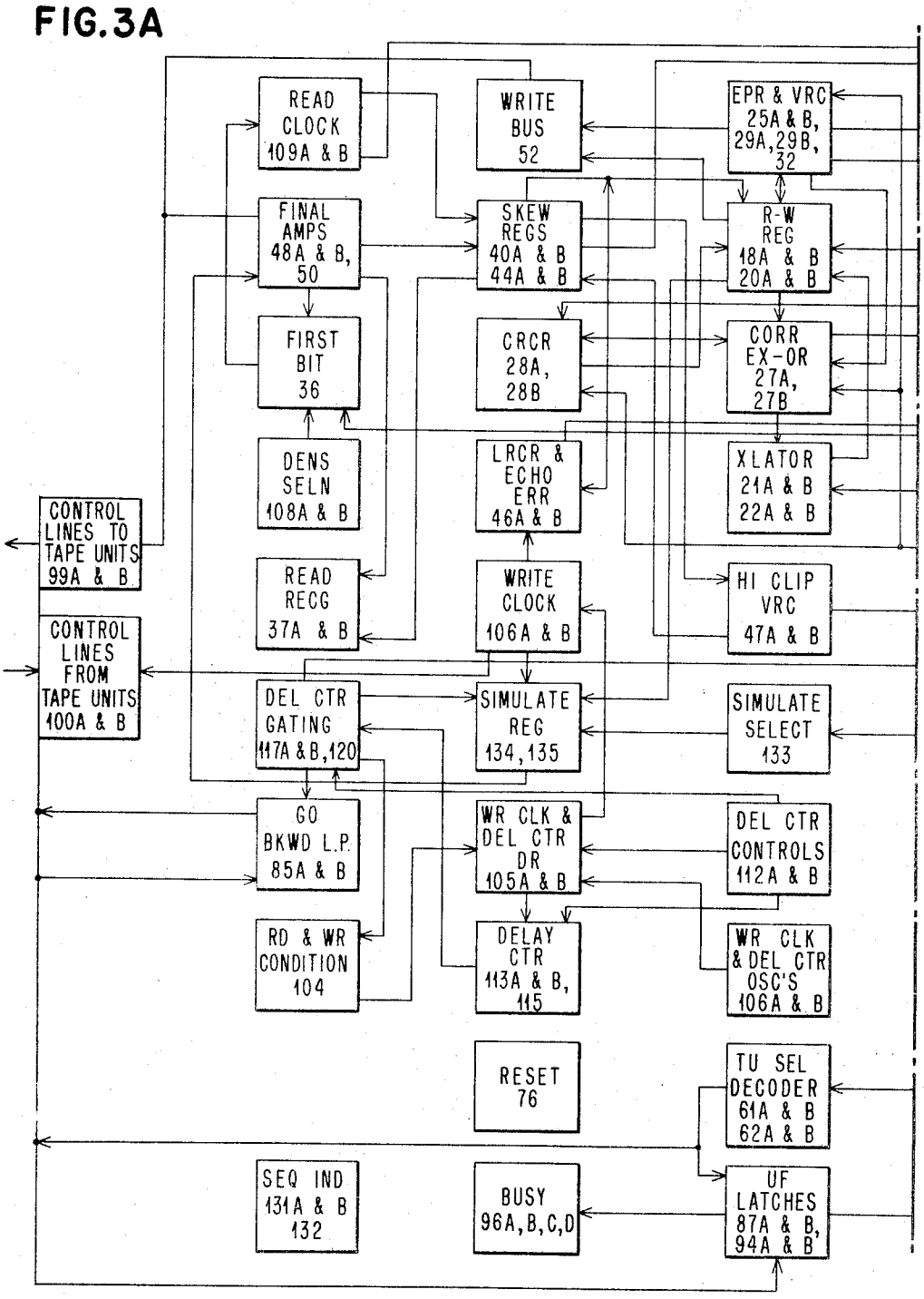

LRC SAMPLE TIME
CLOCK
RD TO CORR
WRITE
A
O
SHIFT
C R C R
COR PATTERN TEST
CRC ERROR
DATA ENTRY
TO TAPE
TO CPU
WR CRC CHAR
CPU WR DATA
TAPE RD DATA
FIND TRACK SET UP
R/W REG
OE X9
OR X9
A X9
FIND TRACK TIME
FIND TRACK TEST
CORR TIME
BEGIN FIND TRACK TIME
CRC CHAR CORRECTIONS
READ TIME
NORMAL READ
WRITE
CLOCK
VRC
N
ENTER 1
E P R
ENTER VRC
SHIFT
RD TO COR
CLOCK
VRC ERROR
ERROR TRACK BLOCKING GATE X9
L R C R
ZERO TEST LRC-SUM
LRC ERROR
LRC SAMPLE TIME

FIG. 3A


READ CLOCK 109A & B
WRITE BUS 52
EPR & VRC 25A & B, 29A, 29B, 32
FINAL AMPS 48A & B, 50
SKEW REGS 40A & B 44A & B
R-W REG 18A & B 20A & B
FIRST BIT 36
CRCR 28A, 28B
CORR EX-OR 27A, 27B
DENS SELN 108A & B
LRCR & ECHO ERR 46A & B
XLATOR 21A & B 22A & B
CONTROL LINES TO TAPE UNITS 99A & B
READ RECG 37A & B
WRITE CLOCK 106A & B
HI CLIP VRC 47A & B
CONTROL LINES FROM TAPE UNITS 100A & B
DEL CTR GATING 117A & B, 120
SIMULATE REG 134, 135
SIMULATE SELECT 133
GO BKWD L.P. 85A & B
WR CLK & DEL CTR DR 105A & B
DEL CTR CONTROLS 112A & B
RD & WR CONDITION 104
DELAY CTR 113A & B, 115
WR CLK & DEL CTR OSC'S 106A & B
RESET 76
TU SEL DECODER 61A & B 62A & B
SEQ IND 131A & B 132
BUSY 96A, B, C, D
UF LATCHES 87A & B, 94A & B

ERROR DETECTION AND CORRECTION SYSTEM

Filed April 6, 1964 316 Sheets-Sheet 4

FIG.3B

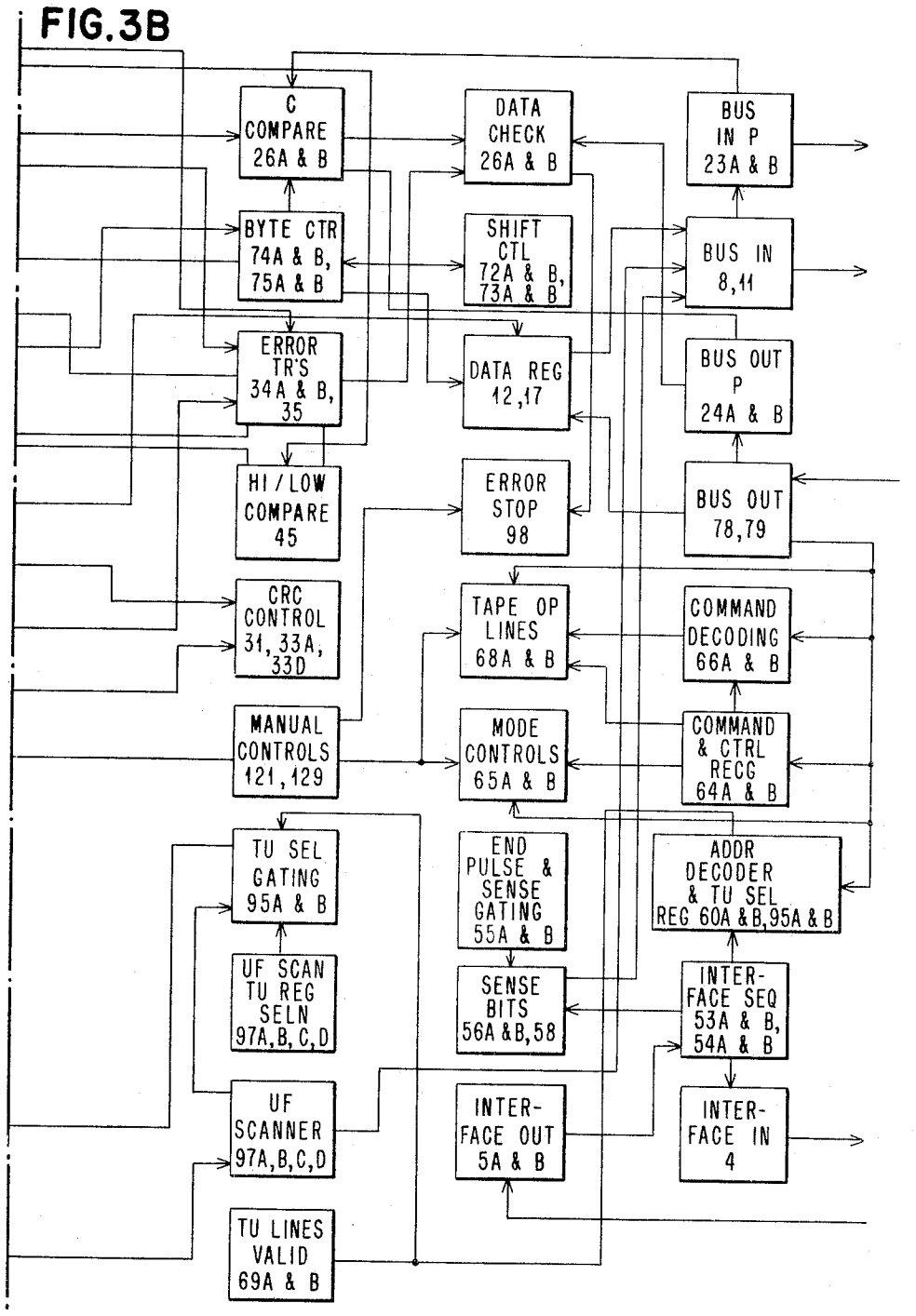


FIG. 4

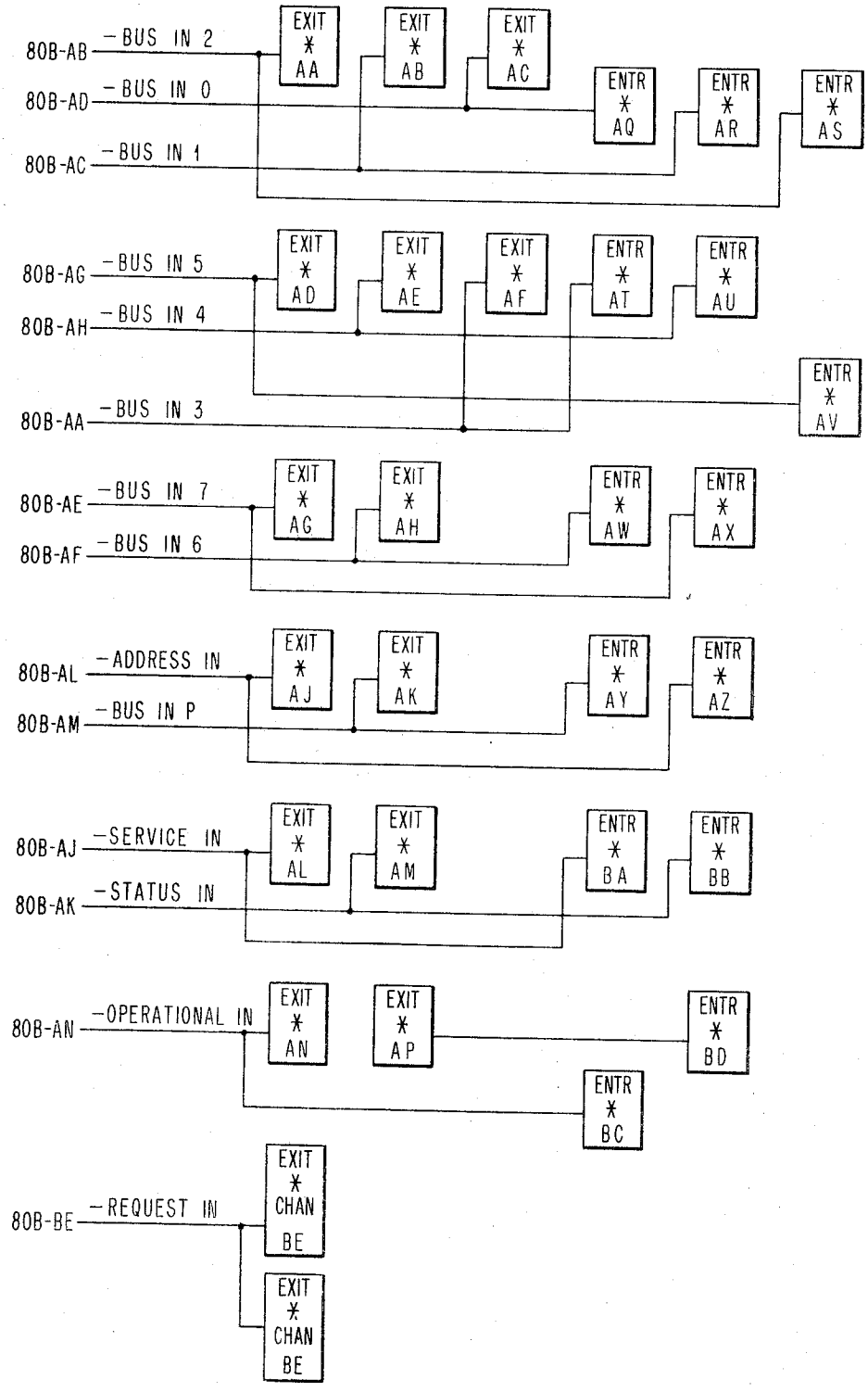

80B-AB
-BUS IN 2
80B-AD
-BUS IN 0
80B-AC
-BUS IN 1
EXIT * AA
EXIT * AB
EXIT * AC
ENTR * AQ
ENTR * AR
ENTR * AS
80B-AG
-BUS IN 5
80B-AH
-BUS IN 4
80B-AA
-BUS IN 3
EXIT * AD
EXIT * AE
EXIT * AF
ENTR * AT
ENTR * AU
ENTR * AV
80B-AE
-BUS IN 7
80B-AF
-BUS IN 6
EXIT * AG
EXIT * AH
ENTR * AW
ENTR * AX
80B-AL
-ADDRESS IN
80B-AM
-BUS IN P
EXIT * AJ
EXIT * AK
ENTR * AY
ENTR * AZ
80B-AJ
-SERVICE IN
80B-AK
-STATUS IN
EXIT * AL
EXIT * AM
ENTR * BA
ENTR * BB
80B-AN
-OPERATIONAL IN
EXIT * AN
EXIT * AP
ENTR * BD
ENTR * BC
80B-BE
-REQUEST IN
EXIT * CHAN BE
EXIT * CHAN BE

FIG.5

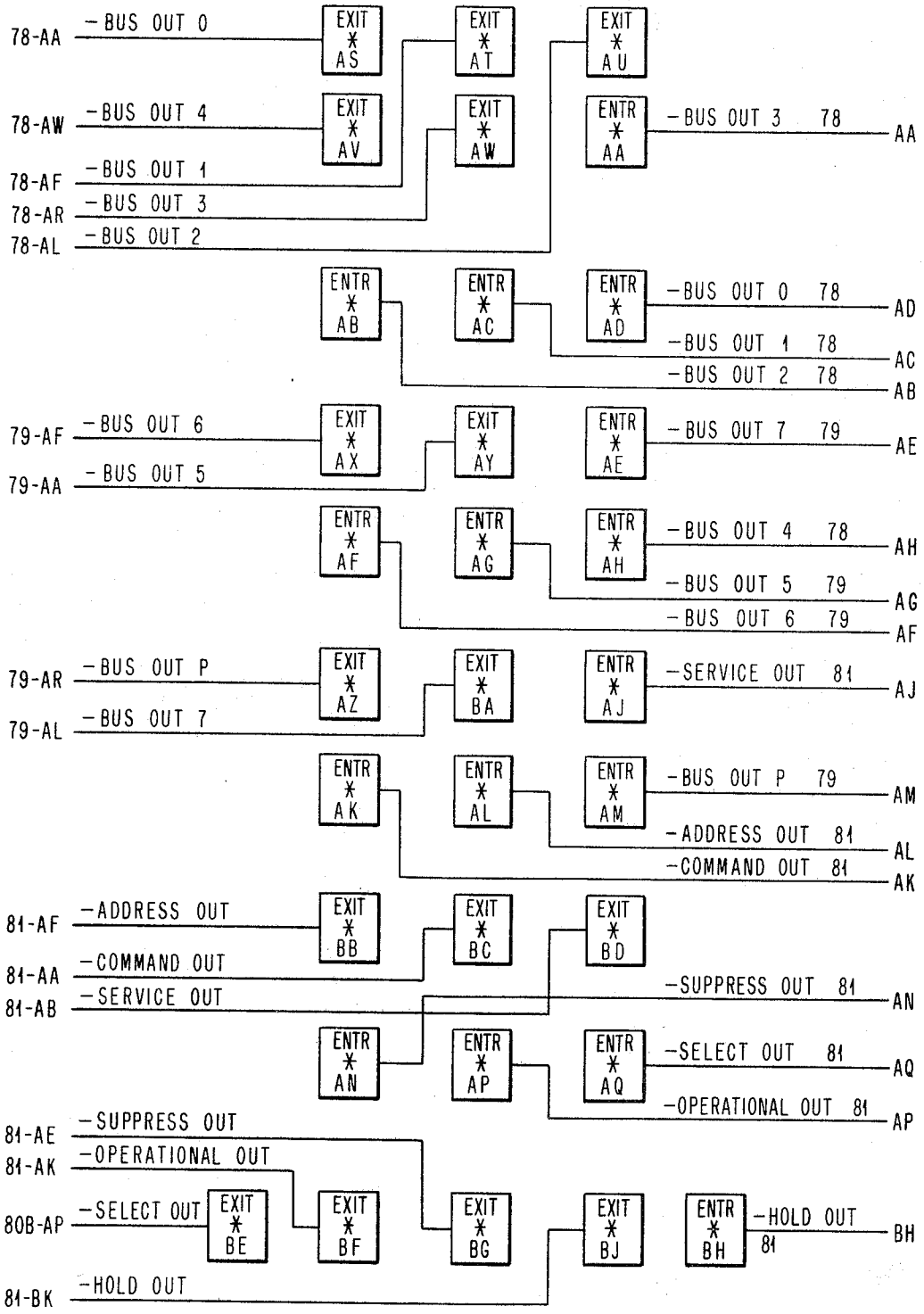

78-AA
-BUS OUT 0
EXIT * AS
EXIT * AT
EXIT * AU
78-AW
-BUS OUT 4
EXIT * AV
EXIT * AW
ENTR * AA
-BUS OUT 3 78
AA
78-AF
-BUS OUT 1
78-AR
-BUS OUT 3
78-AL
-BUS OUT 2
ENTR * AB
ENTR * AC
ENTR * AD
-BUS OUT 0 78
AD
-BUS OUT 1 78
AC
-BUS OUT 2 78
AB
79-AF
-BUS OUT 6
EXIT * AX
EXIT * AY
ENTR * AE
-BUS OUT 7 79
AE
79-AA
-BUS OUT 5
ENTR * AF
ENTR * AG
ENTR * AH
-BUS OUT 4 78
AH
-BUS OUT 5 79
AG
-BUS OUT 6 79
AF
79-AR
-BUS OUT P
EXIT * AZ
EXIT * BA
ENTR * AJ
-SERVICE OUT 81
AJ
79-AL
-BUS OUT 7
ENTR * AK
ENTR * AL
ENTR * AM
-BUS OUT P 79
AM
-ADDRESS OUT 81
AL
-COMMAND OUT 81
AK
81-AF
-ADDRESS OUT
EXIT * BB
EXIT * BC
EXIT * BD
81-AA
-COMMAND OUT
81-AB
-SERVICE OUT
-SUPPRESS OUT 81
AN
ENTR * AN
ENTR * AP
ENTR * AQ
-SELECT OUT 81
AQ
-OPERATIONAL OUT 81
AP
81-AE
-SUPPRESS OUT
81-AK
-OPERATIONAL OUT
80B-AP
-SELECT OUT
EXIT * BE
EXIT * BF
EXIT * BG
EXIT * BJ
ENTR * BH
-HOLD OUT 81
BH
81-BK
-HOLD OUT

FIG.6

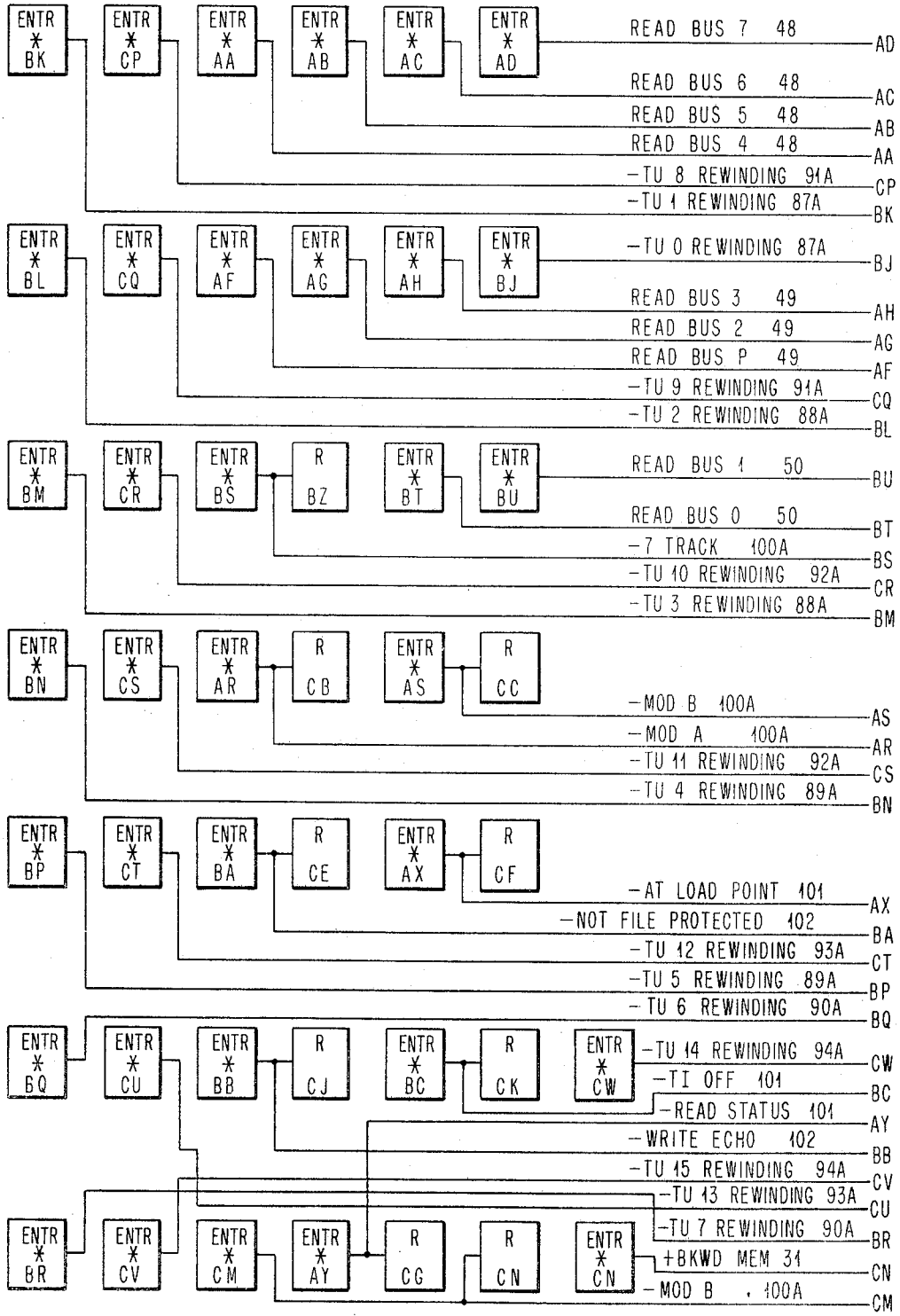

ENTR * BK
ENTR * CP
ENTR * AA
ENTR * AB
ENTR * AC
ENTR * AD
READ BUS 7 48
AD
READ BUS 6 48
AC
READ BUS 5 48
AB
READ BUS 4 48
AA
-TU 8 REWINDING 91A
CP
-TU 1 REWINDING 87A
BK
ENTR * BL
ENTR * CQ
ENTR * AF
ENTR * AG
ENTR * AH
ENTR * BJ
-TU 0 REWINDING 87A
BJ
READ BUS 3 49
AH
READ BUS 2 49
AG
READ BUS P 49
AF
-TU 9 REWINDING 91A
CQ
-TU 2 REWINDING 88A
BL
ENTR * BM
ENTR * CR
ENTR * BS
R BZ
ENTR * BT
ENTR * BU
READ BUS 1 50
BU
READ BUS 0 50
BT
-7 TRACK 100A
BS
-TU 10 REWINDING 92A
CR
-TU 3 REWINDING 88A
BM
ENTR * BN
ENTR * CS
ENTR * AR
R CB
ENTR * AS
R CC
-MOD B 100A
AS
-MOD A 100A
AR
-TU 11 REWINDING 92A
CS
-TU 4 REWINDING 89A
BN
ENTR * BP
ENTR * CT
ENTR * BA
R CE
ENTR * AX
R CF
-AT LOAD POINT 101
AX
-NOT FILE PROTECTED 102
BA
-TU 12 REWINDING 93A
CT
-TU 5 REWINDING 89A
BP
-TU 6 REWINDING 90A
BQ
ENTR * BQ
ENTR * CU
ENTR * BB
R CJ
ENTR * BC
R CK
ENTR * CW
-TU 14 REWINDING 94A
CW
-TI OFF 101
BC
-READ STATUS 101
AY
-WRITE ECHO 102
BB
-TU 15 REWINDING 94A
CV
-TU 13 REWINDING 93A
CU
-TU 7 REWINDING 90A
BR
+BKWD MEM 31
CN
-MOD B .100A
CM
ENTR * BR
ENTR * CV
ENTR * CM
ENTR * AY
R CG
R CN
ENTR * CN

FIG. 7

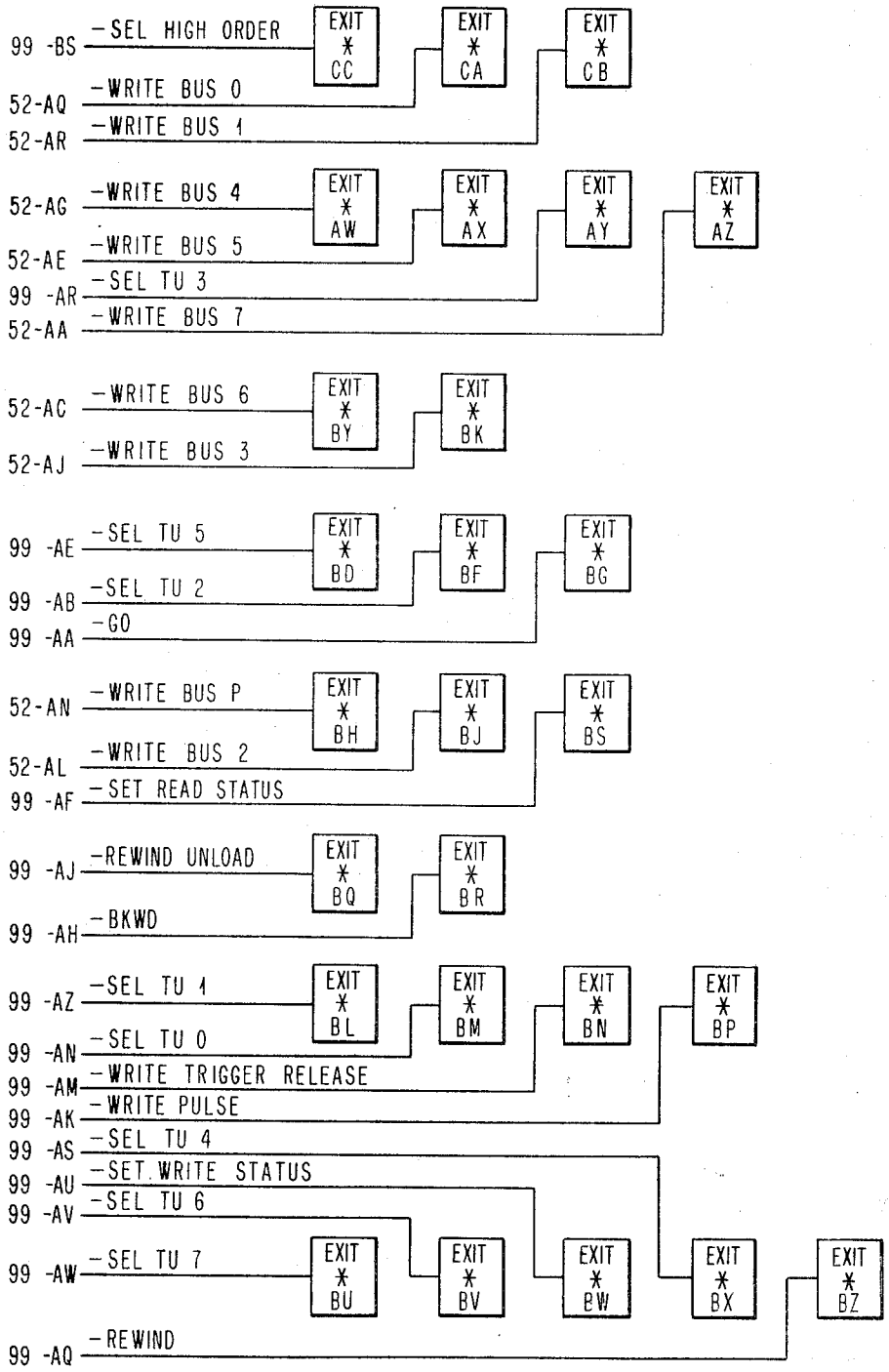

99 -BS
-SEL HIGH ORDER
52-AQ
-WRITE BUS 0
52-AR
-WRITE BUS 1
EXIT * CC
EXIT * CA
EXIT * CB
52-AG
-WRITE BUS 4
52-AE
-WRITE BUS 5
99 -AR
-SEL TU 3
52-AA
-WRITE BUS 7
EXIT * AW
EXIT * AX
EXIT * AY
EXIT * AZ
52-AC
-WRITE BUS 6
52-AJ
-WRITE BUS 3
EXIT * BY
EXIT * BK
99 -AE
-SEL TU 5
99 -AB
-SEL TU 2
99 -AA
-GO
EXIT * BD
EXIT * BF
EXIT * BG
52-AN
-WRITE BUS P
52-AL
-WRITE BUS 2
99 -AF
-SET READ STATUS
EXIT * BH
EXIT * BJ
EXIT * BS
99 -AJ
-REWIND UNLOAD
99 -AH
-BKWD
EXIT * BQ
EXIT * BR
99 -AZ
-SEL TU 1
99 -AN
-SEL TU 0
99 -AM
-WRITE TRIGGER RELEASE
99 -AK
-WRITE PULSE
99 -AS
-SEL TU 4
99 -AU
-SET WRITE STATUS
99 -AV
-SEL TU 6
99 -AW
-SEL TU 7
99 -AQ
-REWIND
EXIT * BL
EXIT * BM
EXIT * BN
EXIT * BP
EXIT * BU
EXIT * BV
EXIT * BW
EXIT * BX
EXIT * BZ

FIG. 8

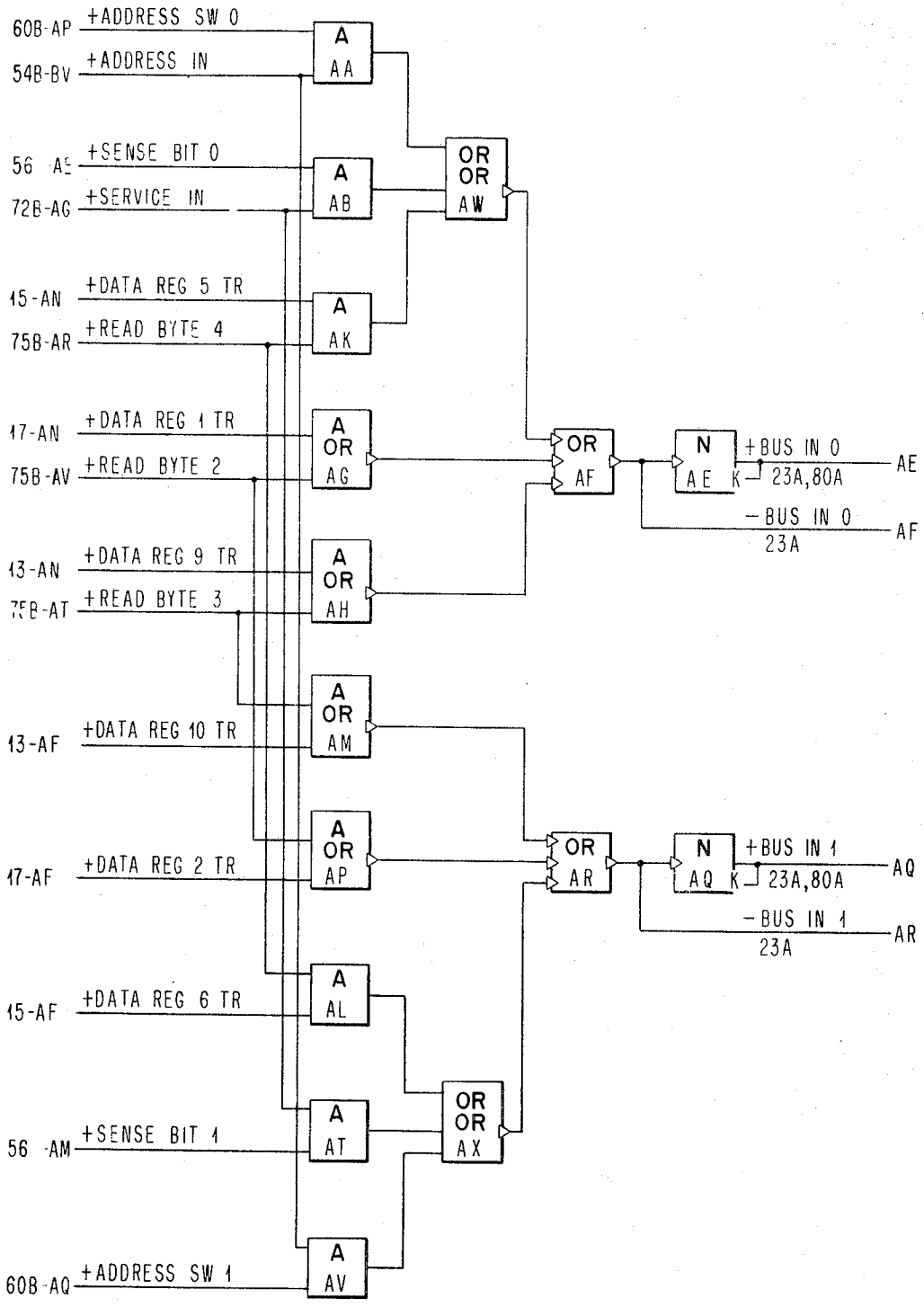

60B-AP +ADDRESS SW 0
54B-BV +ADDRESS IN
A AA
56 AE +SENSE BIT 0
72B-AG +SERVICE IN
A AB
OR OR AW
15-AN +DATA REG 5 TR
75B-AR +READ BYTE 4
A AK
17-AN +DATA REG 1 TR
75B-AV +READ BYTE 2
A OR AG
OR AF
N AE K
+BUS IN 0
23A,80A
AE
-BUS IN 0
23A
AF
13-AN +DATA REG 9 TR
75B-AT +READ BYTE 3
A OR AH
13-AF +DATA REG 10 TR
A OR AM
17-AF +DATA REG 2 TR
A OR AP
OR AR
N AQ K
+BUS IN 1
23A,80A
AQ
-BUS IN 1
23A
AR
15-AF +DATA REG 6 TR
A AL
56 AM +SENSE BIT 1
A AT
OR OR AX
60B-AQ +ADDRESS SW 1
A AV

FIG.9

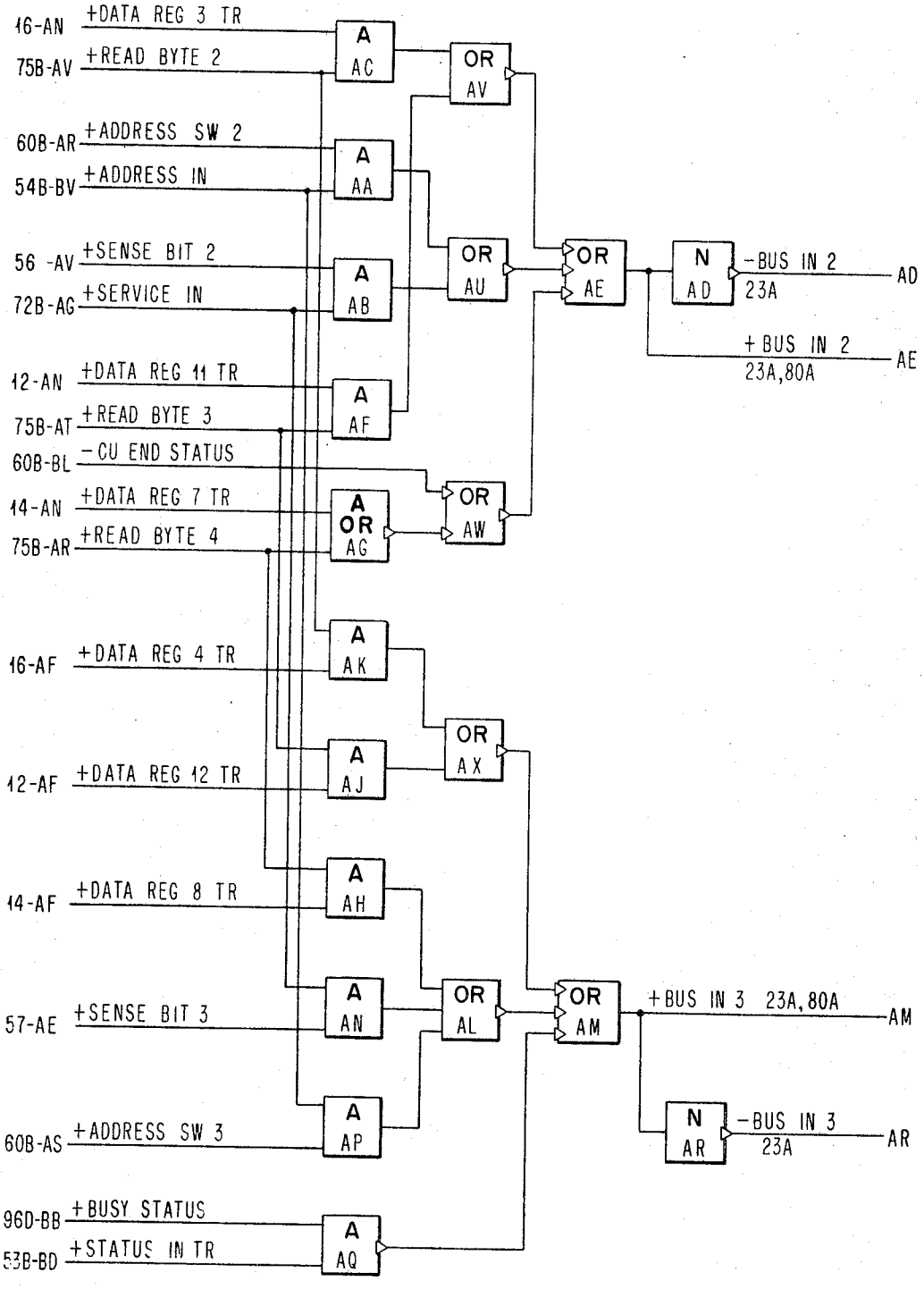

16-AN +DATA REG 3 TR
75B-AV +READ BYTE 2
A AC
OR AV
60B-AR +ADDRESS SW 2
54B-BV +ADDRESS IN
A AA
56 -AV +SENSE BIT 2
72B-AG +SERVICE IN
A AB
OR AU
OR AE
N AD
-BUS IN 2 23A
AD
+BUS IN 2 23A,80A
AE
12-AN +DATA REG 11 TR
75B-AT +READ BYTE 3
A AF
60B-BL -CU END STATUS
14-AN +DATA REG 7 TR
75B-AR +READ BYTE 4
A OR AG
OR AW
16-AF +DATA REG 4 TR
A AK
OR AX
12-AF +DATA REG 12 TR
A AJ
14-AF +DATA REG 8 TR
A AH
57-AE +SENSE BIT 3
A AN
OR AL
OR AM
+BUS IN 3 23A,80A
AM
60B-AS +ADDRESS SW 3
A AP
N AR
-BUS IN 3 23A
AR
96D-BB +BUSY STATUS
53B-BD +STATUS IN TR
A AQ

FIG.10

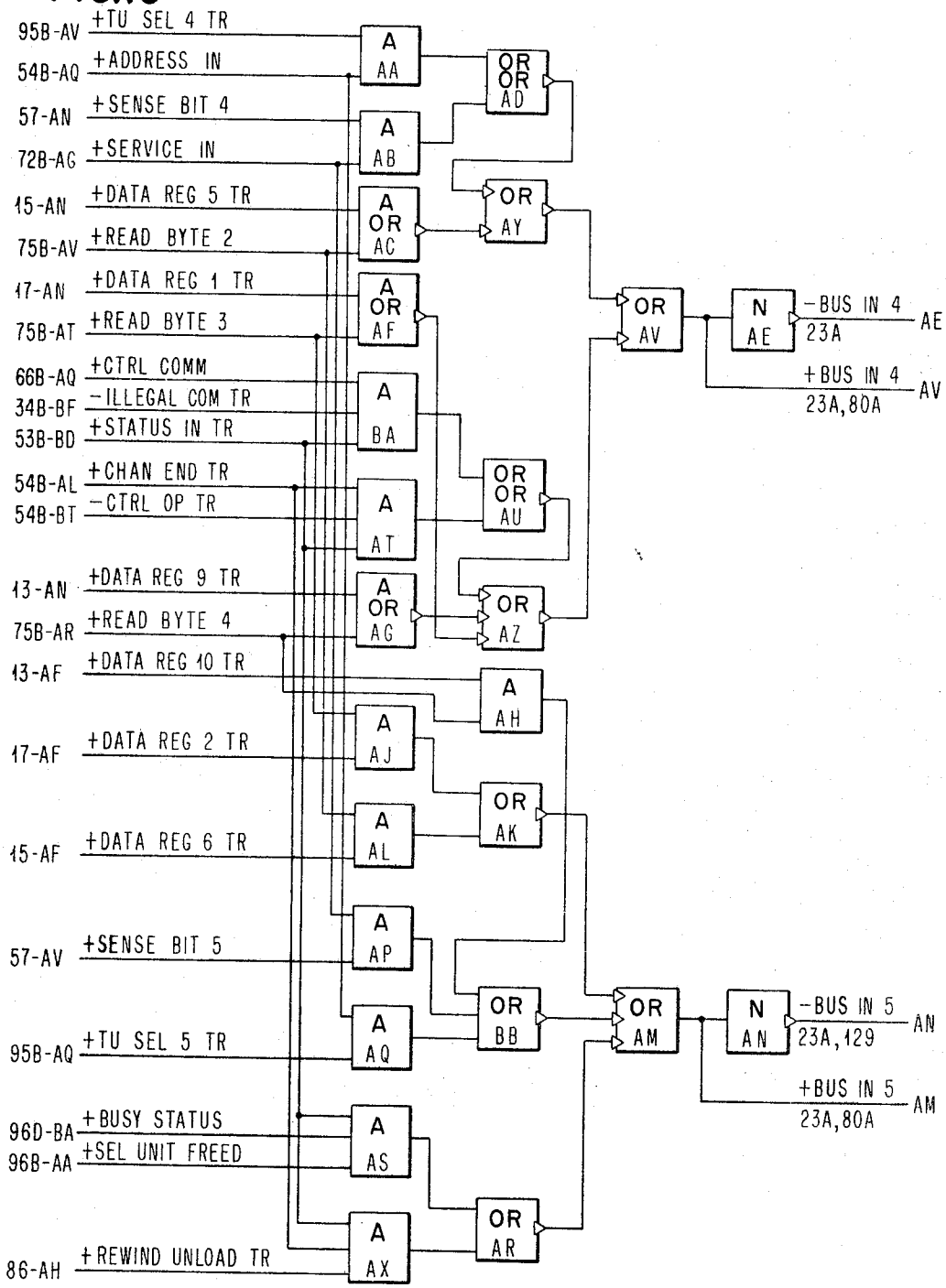

95B-AV +TU SEL 4 TR
54B-AQ +ADDRESS IN
57-AN +SENSE BIT 4
72B-AG +SERVICE IN
15-AN +DATA REG 5 TR
75B-AV +READ BYTE 2
17-AN +DATA REG 1 TR
75B-AT +READ BYTE 3
66B-AQ +CTRL COMM
34B-BF -ILLEGAL COM TR
53B-BD +STATUS IN TR
54B-AL +CHAN END TR
54B-BT -CTRL OP TR
13-AN +DATA REG 9 TR
75B-AR +READ BYTE 4
13-AF +DATA REG 10 TR
17-AF +DATA REG 2 TR
15-AF +DATA REG 6 TR
57-AV +SENSE BIT 5
95B-AQ +TU SEL 5 TR
96D-BA +BUSY STATUS
96B-AA +SEL UNIT FREED
86-AH +REWIND UNLOAD TR
A AA
A AB
A OR AC
A OR AF
A BA
A AT
A OR AG
A AH
A AJ
A AL
A AP
A AQ
A AS
A AX
OR OR AD
OR AY
OR AV
OR OR AU
OR AZ
OR AK
OR BB
OR AR
OR AM
N AE
-BUS IN 4
23A
AE
+BUS IN 4
23A,80A
AV
N AN
-BUS IN 5
23A,129
AN
+BUS IN 5
23A,80A
AM

FIG.11

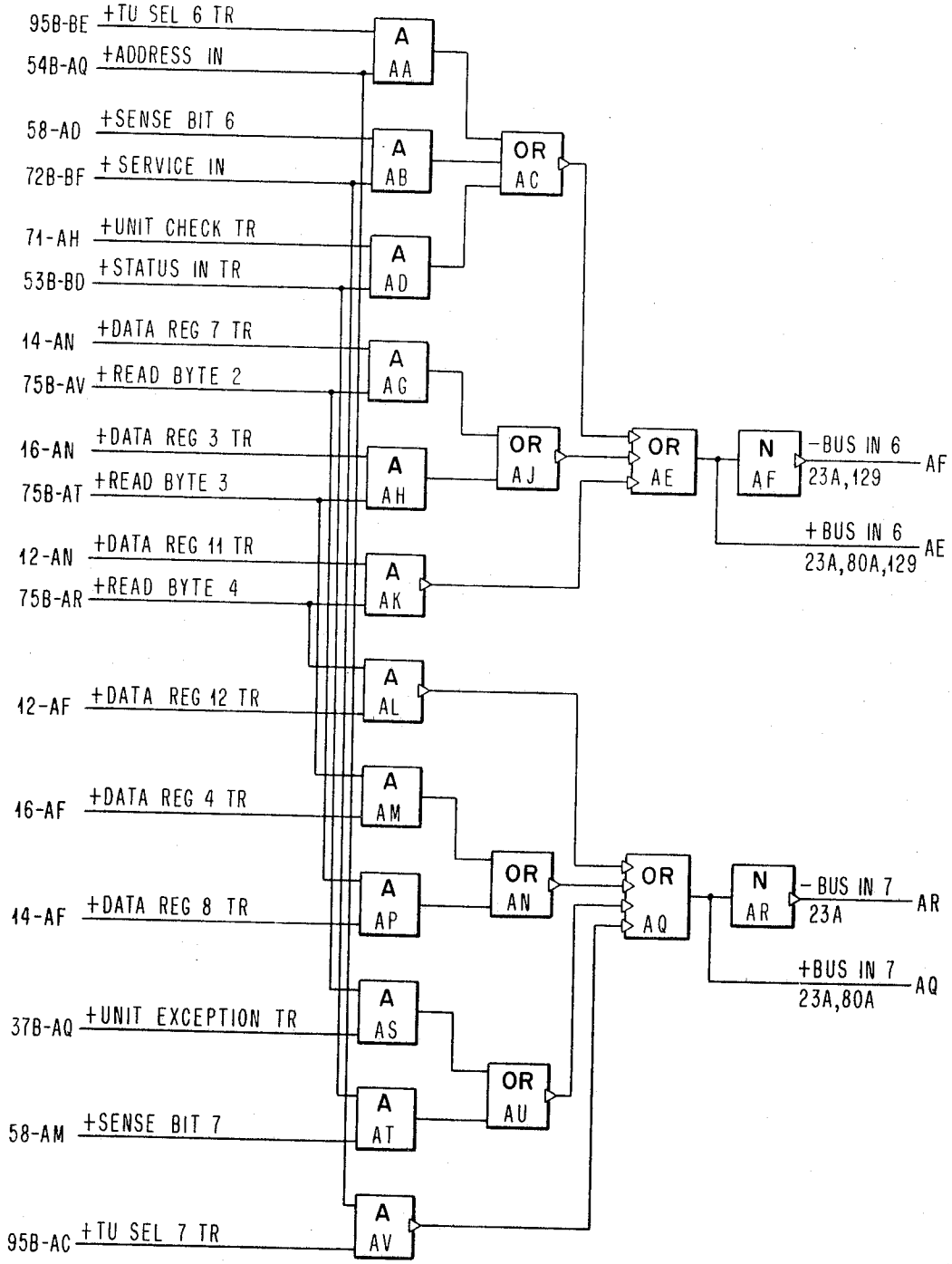

95B-BE +TU SEL 6 TR
54B-AQ +ADDRESS IN
A AA
58-AD +SENSE BIT 6
72B-BF +SERVICE IN
A AB
OR AC
71-AH +UNIT CHECK TR
53B-BD +STATUS IN TR
A AD
14-AN +DATA REG 7 TR
75B-AV +READ BYTE 2
A AG
16-AN +DATA REG 3 TR
75B-AT +READ BYTE 3
A AH
OR AJ
OR AE
N AF
-BUS IN 6
23A,129
AF
+BUS IN 6
23A,80A,129
AE
12-AN +DATA REG 11 TR
75B-AR +READ BYTE 4
A AK
12-AF +DATA REG 12 TR
A AL
16-AF +DATA REG 4 TR
A AM
14-AF +DATA REG 8 TR
A AP
OR AN
OR AQ
N AR
-BUS IN 7
23A
AR
+BUS IN 7
23A,80A
AQ
37B-AQ +UNIT EXCEPTION TR
A AS
58-AM +SENSE BIT 7
A AT
OR AU
95B-AC +TU SEL 7 TR
A AV

FIG. 12

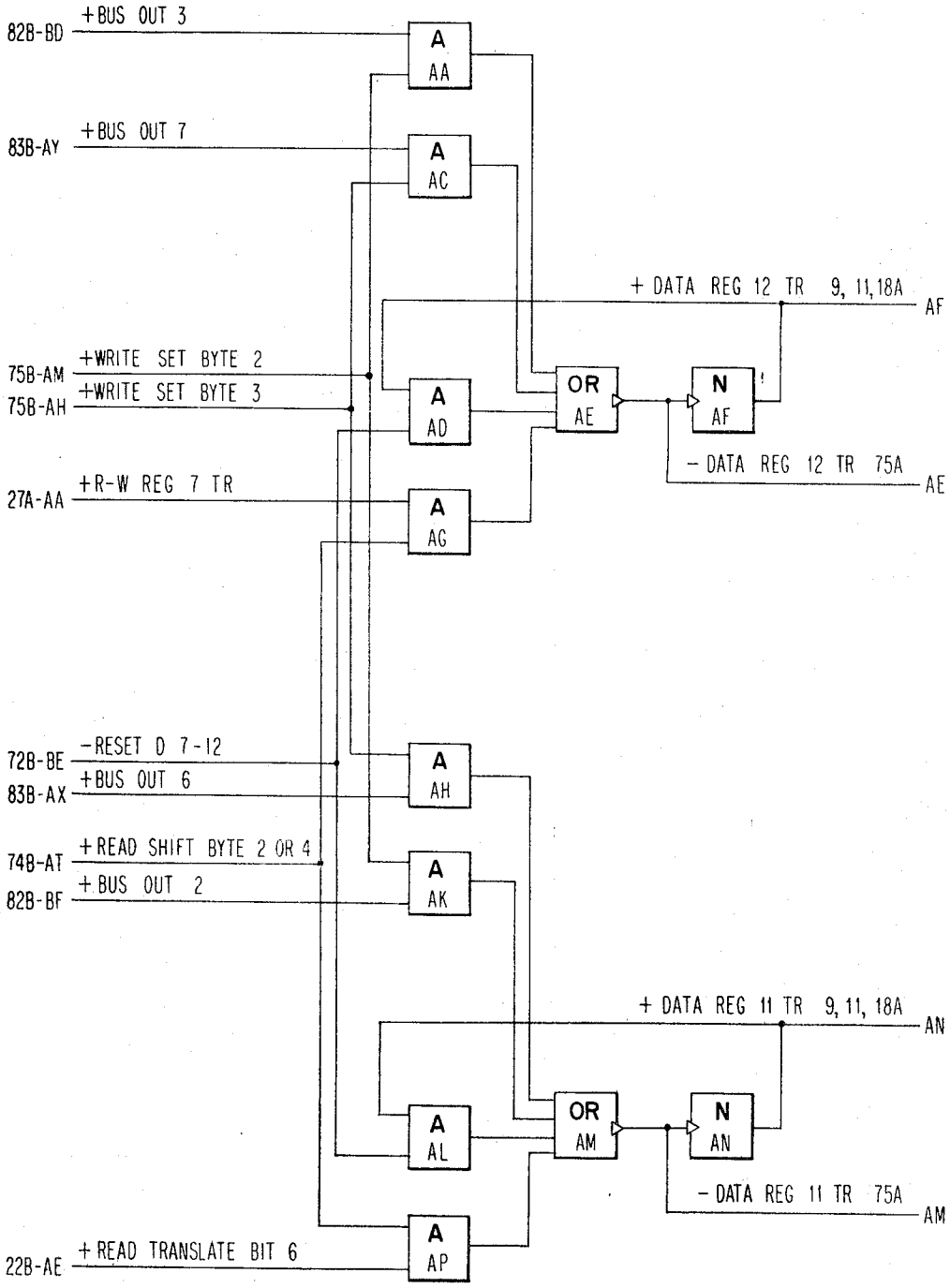

82B-BD
+BUS OUT 3
A AA
83B-AY
+BUS OUT 7
A AC
+ DATA REG 12 TR 9, 11, 18A
AF
75B-AM
+WRITE SET BYTE 2
75B-AH
+WRITE SET BYTE 3
A AD
OR AE
N AF
- DATA REG 12 TR 75A
AE
27A-AA
+R-W REG 7 TR
A AG
72B-BE
-RESET D 7-12
83B-AX
+BUS OUT 6
A AH
74B-AT
+READ SHIFT BYTE 2 OR 4
82B-BF
+BUS OUT 2
A AK
+ DATA REG 11 TR 9, 11, 18A
AN
A AL
OR AM
N AN
- DATA REG 11 TR 75A
AM
22B-AE
+READ TRANSLATE BIT 6
A AP

FIG. 13

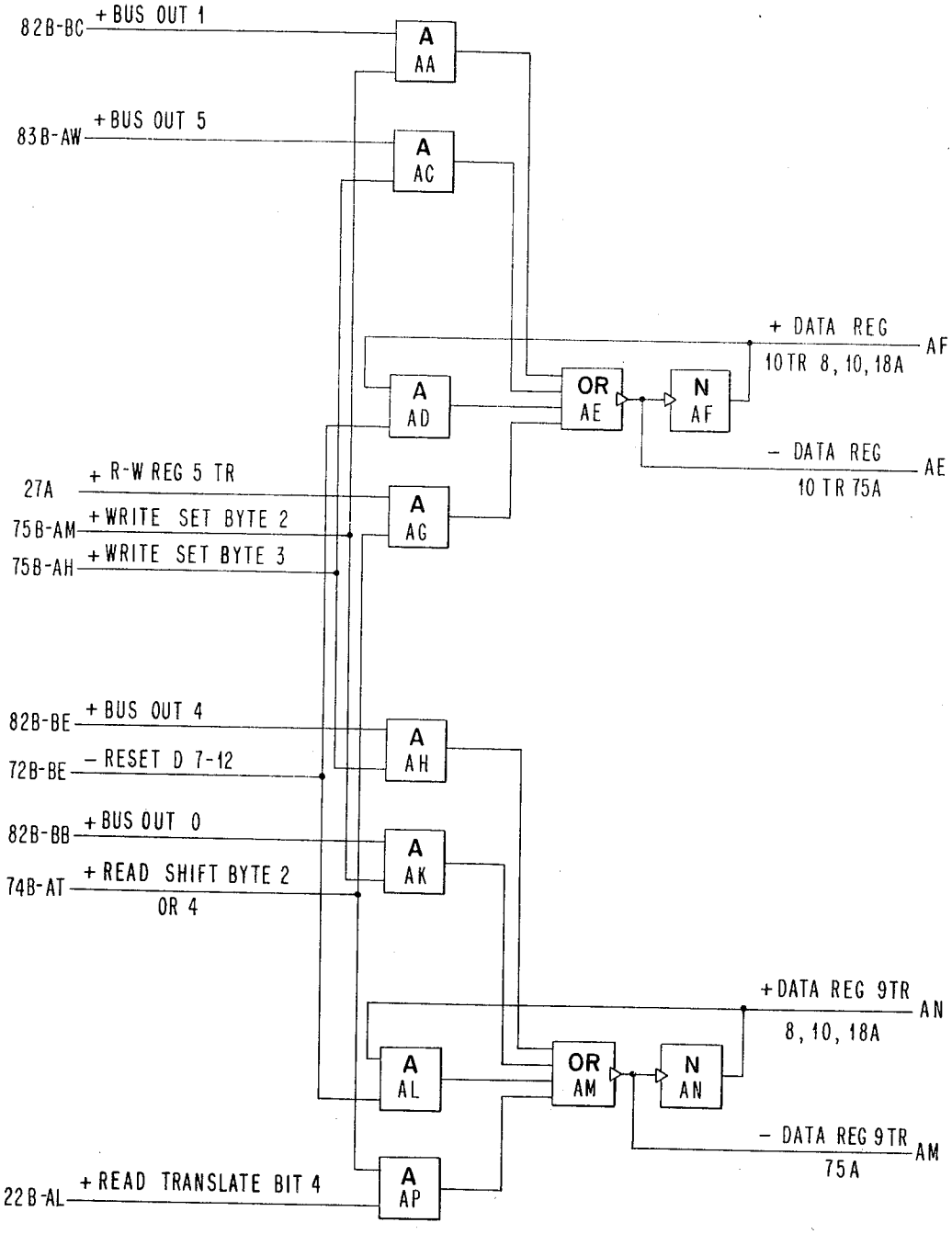

82B-BC
+BUS OUT 1
A AA
83B-AW
+BUS OUT 5
A AC
+ DATA REG 10TR 8, 10, 18A
AF
A AD
OR AE
N AF
- DATA REG 10 TR 75A
AE
27A
+ R-W REG 5 TR
75B-AM
+WRITE SET BYTE 2
75B-AH
+WRITE SET BYTE 3
A AG
82B-BE
+BUS OUT 4
72B-BE
- RESET D 7-12
A AH
82B-BB
+BUS OUT 0
74B-AT
+READ SHIFT BYTE 2 OR 4
A AK
+DATA REG 9TR 8, 10, 18A
AN
A AL
OR AM
N AN
- DATA REG 9TR 75A
AM
22B-AL
+READ TRANSLATE BIT 4
A AP

FIG. 14

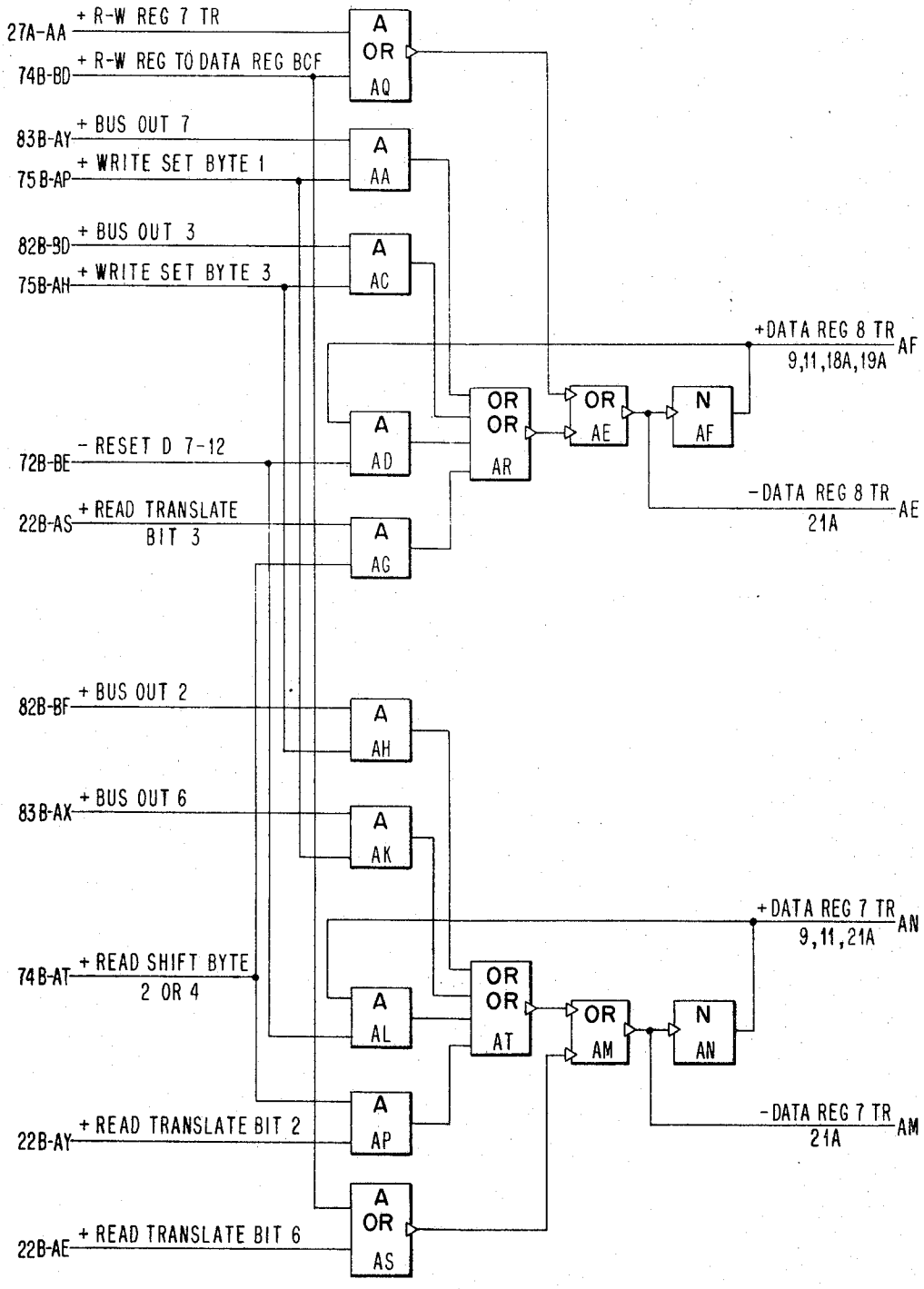

27A-AA
+ R-W REG 7 TR
74B-BD
+ R-W REG TO DATA REG BCF
A OR AQ
83B-AY
+ BUS OUT 7
75B-AP
+ WRITE SET BYTE 1
A AA
82B-BD
+ BUS OUT 3
75B-AH
+ WRITE SET BYTE 3
A AC
+DATA REG 8 TR
9,11,18A,19A
AF
OR OR AR
OR AE
N AF
72B-BE
- RESET D 7-12
A AD
-DATA REG 8 TR
21A
AE
22B-AS
+ READ TRANSLATE BIT 3
A AG
82B-BF
+ BUS OUT 2
A AH
83B-AX
+ BUS OUT 6
A AK
+DATA REG 7 TR
9,11,21A
AN
74B-AT
+ READ SHIFT BYTE 2 OR 4
A AL
OR OR AT
OR AM
N AN
-DATA REG 7 TR
21A
AM
22B-AY
+ READ TRANSLATE BIT 2
A AP
22B-AE
+ READ TRANSLATE BIT 6
A OR AS

FIG. 15

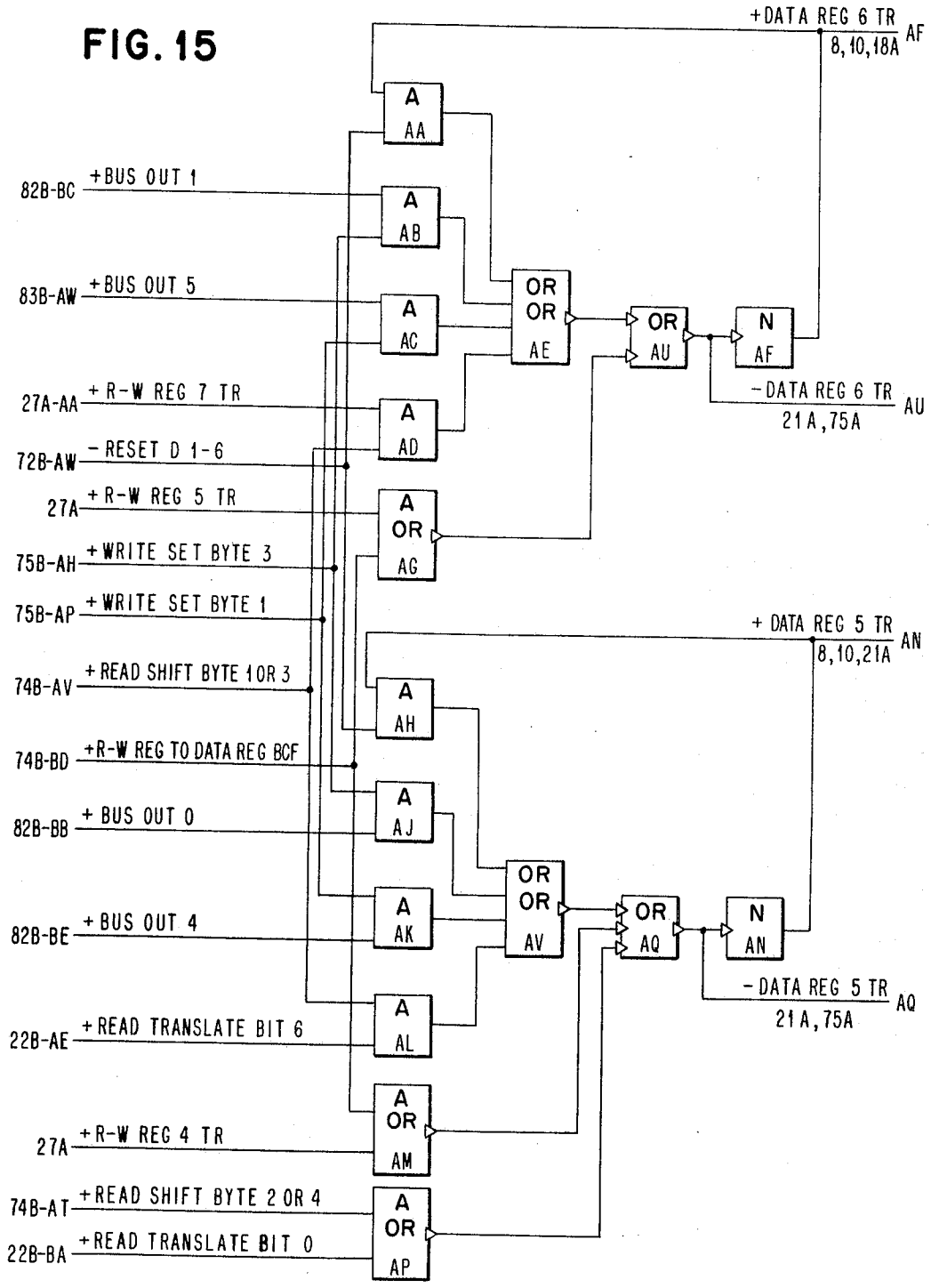

+DATA REG 6 TR
8,10,18A
AF
A AA
82B-BC +BUS OUT 1
A AB
83B-AW +BUS OUT 5
A AC
OR OR AE
OR AU
N AF
27A-AA + R-W REG 7 TR
A AD
-DATA REG 6 TR
21A,75A
AU
72B-AW - RESET D 1-6
27A +R-W REG 5 TR
A OR AG
75B-AH +WRITE SET BYTE 3
75B-AP +WRITE SET BYTE 1
+ DATA REG 5 TR
8,10,21A
AN
74B-AV +READ SHIFT BYTE 1 OR 3
A AH
74B-BD +R-W REG TO DATA REG BCF
82B-BB + BUS OUT 0
A AJ
OR OR AV
OR AQ
N AN
82B-BE + BUS OUT 4
A AK
-DATA REG 5 TR
21A,75A
AQ
22B-AE +READ TRANSLATE BIT 6
A AL
A OR AM
27A +R-W REG 4 TR
74B-AT +READ SHIFT BYTE 2 OR 4
A OR AP
22B-BA +READ TRANSLATE BIT 0

FIG. 16

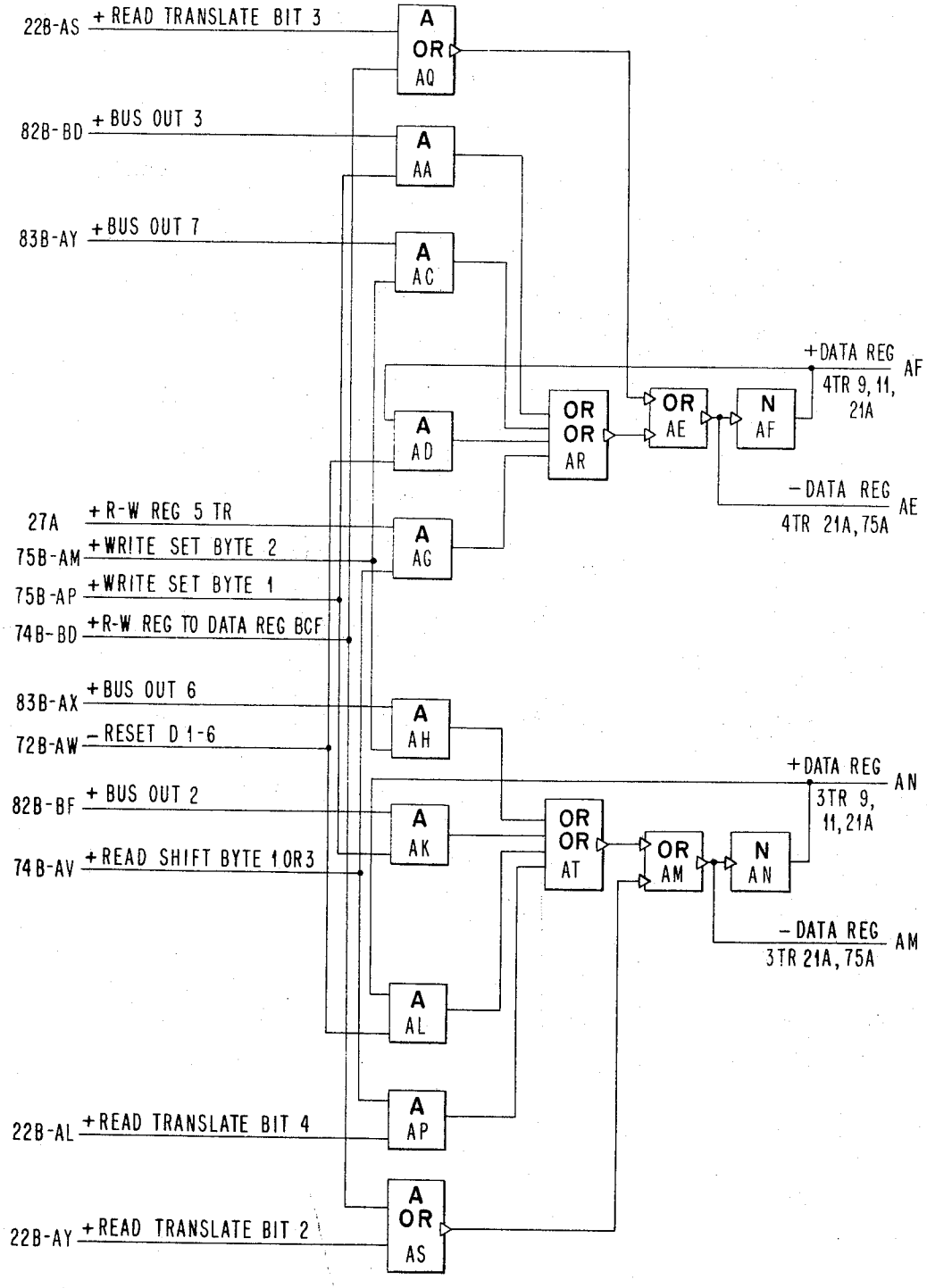

22B-AS +READ TRANSLATE BIT 3
A OR AQ
82B-BD +BUS OUT 3
A AA
83B-AY +BUS OUT 7
A AC
+DATA REG 4TR 9, 11, 21A AF
A AD
OR OR AR
OR AE
N AF
-DATA REG 4TR 21A, 75A AE
27A +R-W REG 5 TR
75B-AM +WRITE SET BYTE 2
75B-AP +WRITE SET BYTE 1
74B-BD +R-W REG TO DATA REG BCF
A AG
83B-AX +BUS OUT 6
72B-AW -RESET D 1-6
A AH
+DATA REG 3TR 9, 11, 21A AN
82B-BF +BUS OUT 2
74B-AV +READ SHIFT BYTE 1 OR 3
A AK
OR OR AT
OR AM
N AN
-DATA REG 3TR 21A, 75A AM
A AL
22B-AL +READ TRANSLATE BIT 4
A AP
22B-AY +READ TRANSLATE BIT 2
A OR AS

FIG.17

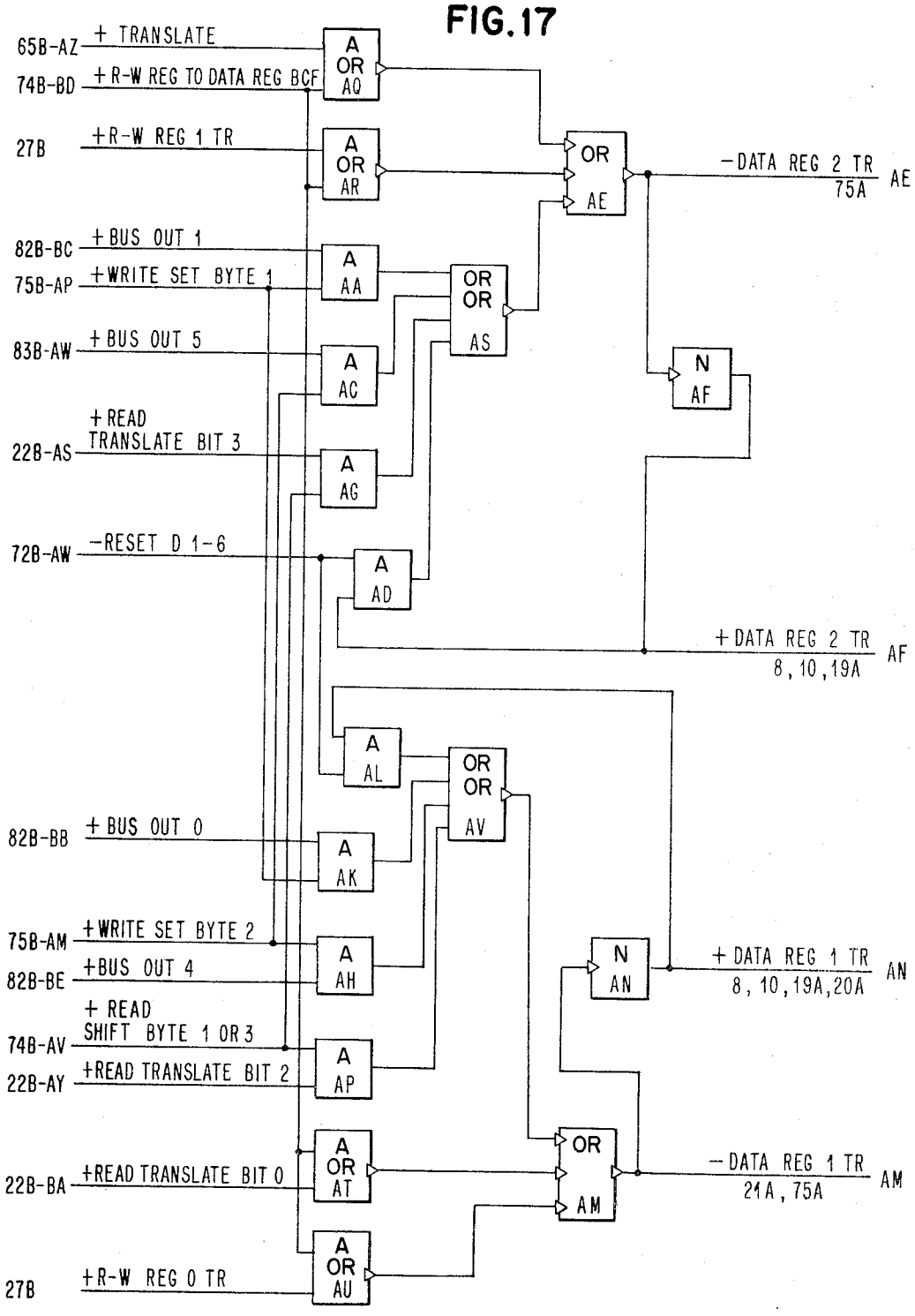

65B-AZ
+ TRANSLATE
74B-BD
+R-W REG TO DATA REG BCF
A OR AQ
27B
+R-W REG 1 TR
A OR AR
OR AE
-DATA REG 2 TR 75A
AE
82B-BC
+BUS OUT 1
75B-AP
+WRITE SET BYTE 1
A AA
OR OR AS
83B-AW
+BUS OUT 5
A AC
N AF
22B-AS
+READ TRANSLATE BIT 3
A AG
72B-AW
-RESET D 1-6
A AD
+DATA REG 2 TR 8,10,19A
AF
A AL
OR OR AV
82B-BB
+BUS OUT 0
A AK
75B-AM
+WRITE SET BYTE 2
82B-BE
+BUS OUT 4
A AH
N AN
+DATA REG 1 TR 8,10,19A,20A
AN
74B-AV
+ READ SHIFT BYTE 1 OR 3
22B-AY
+READ TRANSLATE BIT 2
A AP
22B-BA
+READ TRANSLATE BIT 0
A OR AT
OR AM
-DATA REG 1 TR 21A,75A
AM
27B
+R-W REG 0 TR
A OR AU

FIG.18A

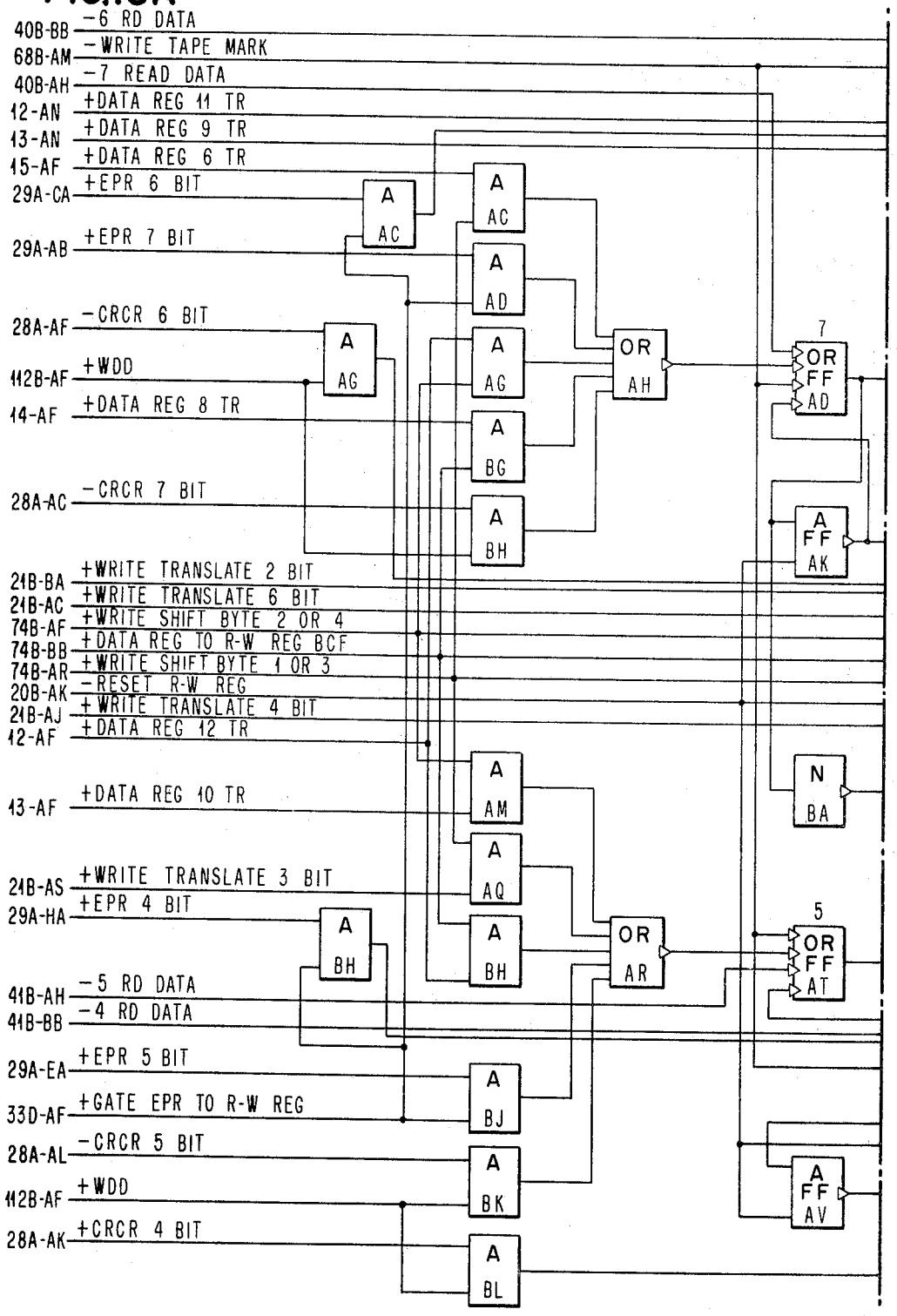

40B-BB −6 RD DATA
68B-AM −WRITE TAPE MARK
40B-AH −7 READ DATA
12-AN +DATA REG 11 TR
13-AN +DATA REG 9 TR
15-AF +DATA REG 6 TR
29A-CA +EPR 6 BIT
29A-AB +EPR 7 BIT
28A-AF −CRCR 6 BIT
112B-AF +WDD
14-AF +DATA REG 8 TR
28A-AC −CRCR 7 BIT
21B-BA +WRITE TRANSLATE 2 BIT
21B-AC +WRITE TRANSLATE 6 BIT
74B-AF +WRITE SHIFT BYTE 2 OR 4
74B-BB +DATA REG TO R-W REG BCF
74B-AR +WRITE SHIFT BYTE 1 OR 3
20B-AK −RESET R-W REG
21B-AJ +WRITE TRANSLATE 4 BIT
12-AF +DATA REG 12 TR
13-AF +DATA REG 10 TR
21B-AS +WRITE TRANSLATE 3 BIT
29A-HA +EPR 4 BIT
41B-AH −5 RD DATA
41B-BB −4 RD DATA
29A-EA +EPR 5 BIT
33D-AF +GATE EPR TO R-W REG
28A-AL −CRCR 5 BIT
112B-AF +WDD
28A-AK +CRCR 4 BIT
A AC
A AC
A AD
A AG
A AG
A BG
A BH
A BH
A AM
A AQ
A BH
A BJ
A BK
A BL
OR AH
OR AR
7
OR FF AD
A FF AK
N BA
5
OR FF AT
A FF AV

FIG.18B

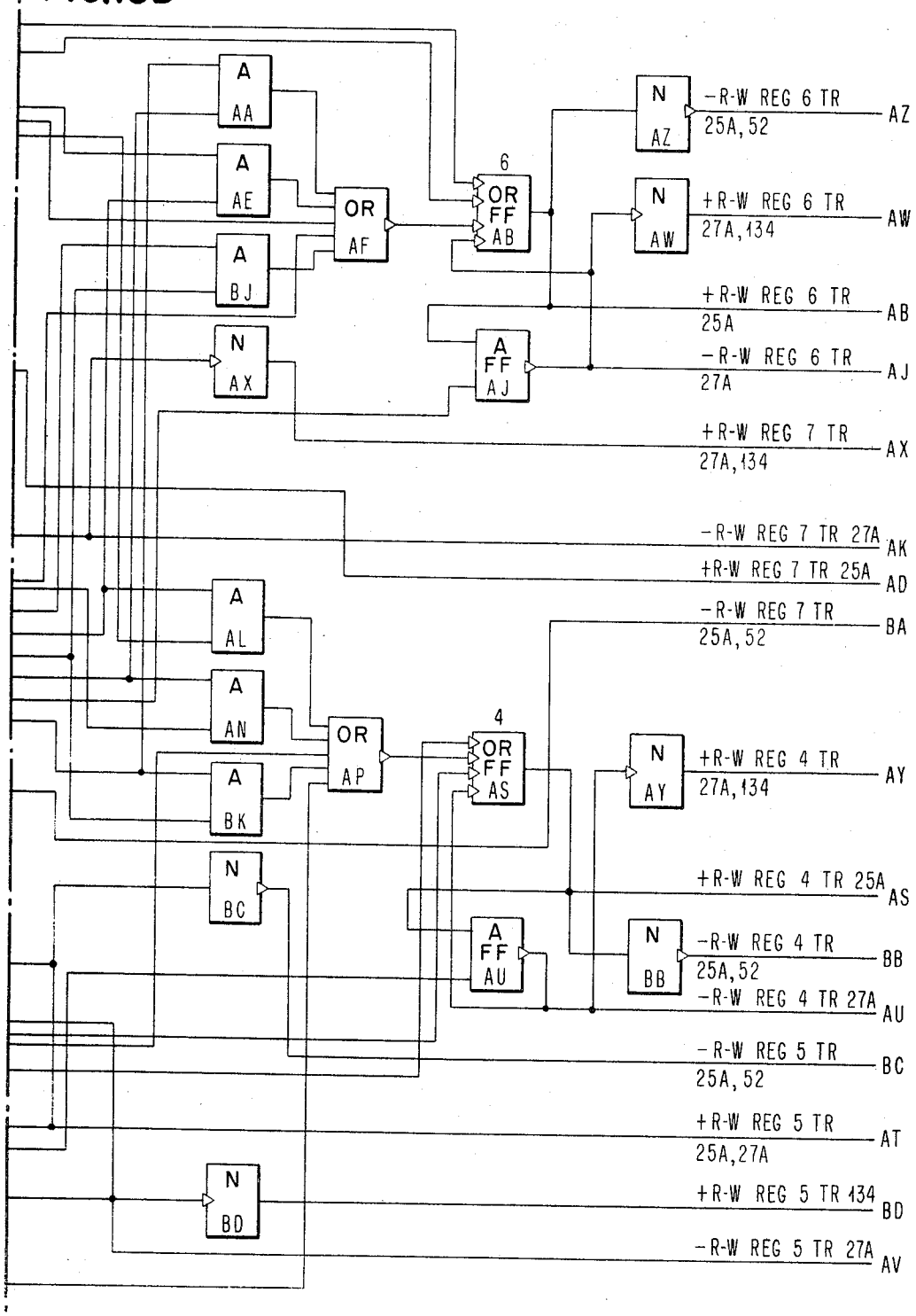

A AA
A AE
A BJ
N AX
OR AF
6
OR FF AB
A FF AJ
N AZ
N AW
-R-W REG 6 TR 25A,52 AZ
+R-W REG 6 TR 27A,134 AW
+R-W REG 6 TR 25A AB
-R-W REG 6 TR 27A AJ
+R-W REG 7 TR 27A,134 AX
-R-W REG 7 TR 27A AK
+R-W REG 7 TR 25A AD
A AL
A AN
A BK
N BC
OR AP
4
OR FF AS
A FF AU
N AY
N BB
-R-W REG 7 TR 25A,52 BA
+R-W REG 4 TR 27A,134 AY
+R-W REG 4 TR 25A AS
-R-W REG 4 TR 25A,52 BB
-R-W REG 4 TR 27A AU
-R-W REG 5 TR 25A,52 BC
+R-W REG 5 TR 25A,27A AT
N BD
+R-W REG 5 TR 134 BD
-R-W REG 5 TR 27A AV

FIG.19A

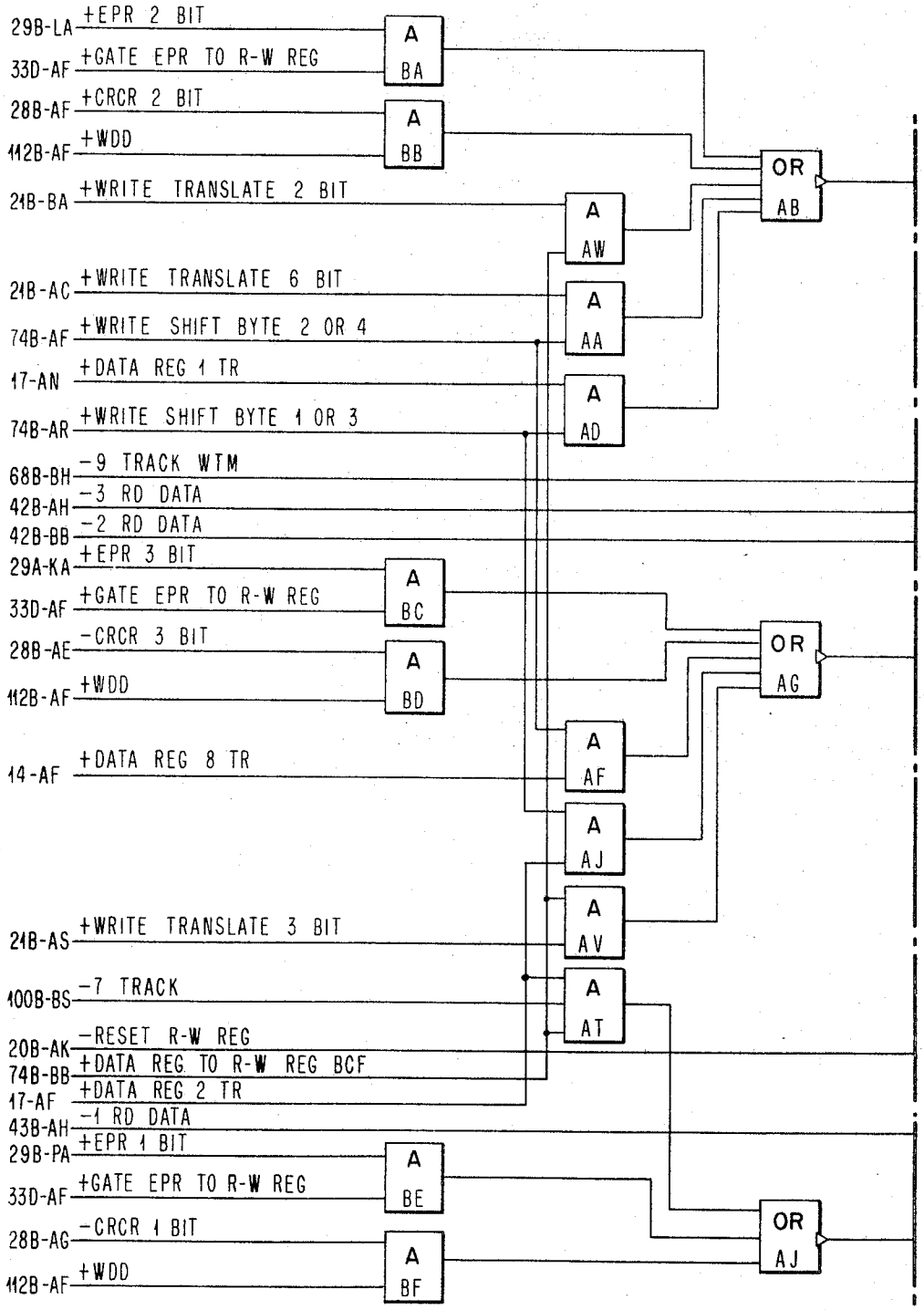

29B-LA +EPR 2 BIT
33D-AF +GATE EPR TO R-W REG
A BA
28B-AF +CRCR 2 BIT
112B-AF +WDD
A BB
OR AB
21B-BA +WRITE TRANSLATE 2 BIT
A AW
21B-AC +WRITE TRANSLATE 6 BIT
74B-AF +WRITE SHIFT BYTE 2 OR 4
A AA
17-AN +DATA REG 1 TR
74B-AR +WRITE SHIFT BYTE 1 OR 3
A AD
68B-BH -9 TRACK WTM
42B-AH -3 RD DATA
42B-BB -2 RD DATA
29A-KA +EPR 3 BIT
33D-AF +GATE EPR TO R-W REG
A BC
28B-AE -CRCR 3 BIT
112B-AF +WDD
A BD
OR AG
14-AF +DATA REG 8 TR
A AF
A AJ
21B-AS +WRITE TRANSLATE 3 BIT
A AV
100B-BS -7 TRACK
A AT
20B-AK -RESET R-W REG
74B-BB +DATA REG TO R-W REG BCF
17-AF +DATA REG 2 TR
43B-AH -1 RD DATA
29B-PA +EPR 1 BIT
33D-AF +GATE EPR TO R-W REG
A BE
28B-AG -CRCR 1 BIT
112B-AF +WDD
A BF
OR AJ

FIG.19B

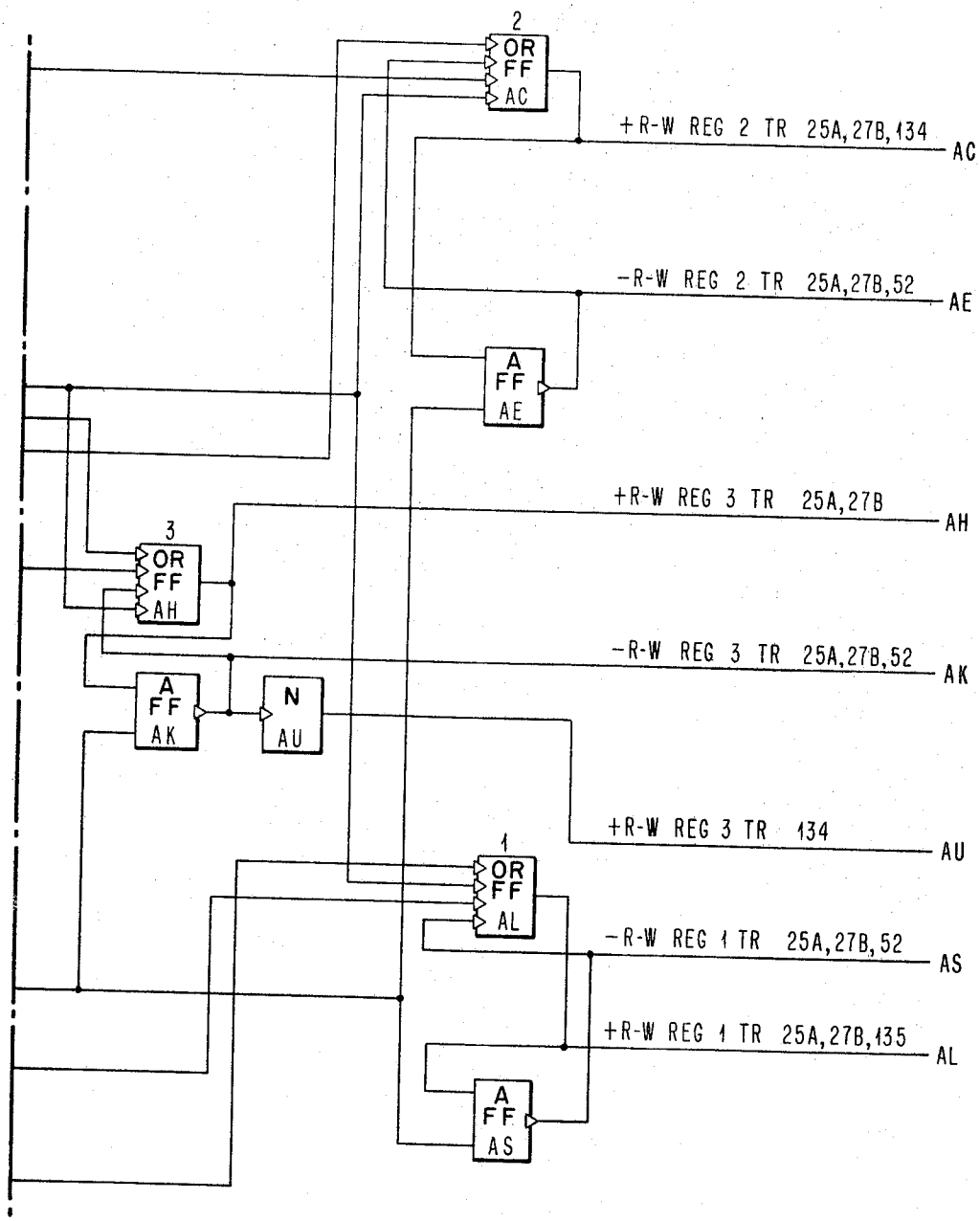

2
OR
FF
AC
+R-W REG 2 TR 25A,27B,134
AC
-R-W REG 2 TR 25A,27B,52
AE
A
FF
AE
+R-W REG 3 TR 25A,27B
AH
3
OR
FF
AH
-R-W REG 3 TR 25A,27B,52
AK
A
FF
AK
N
AU
+R-W REG 3 TR 134
AU
1
OR
FF
AL
-R-W REG 1 TR 25A,27B,52
AS
+R-W REG 1 TR 25A,27B,135
AL
A
FF
AS

FIG.20A

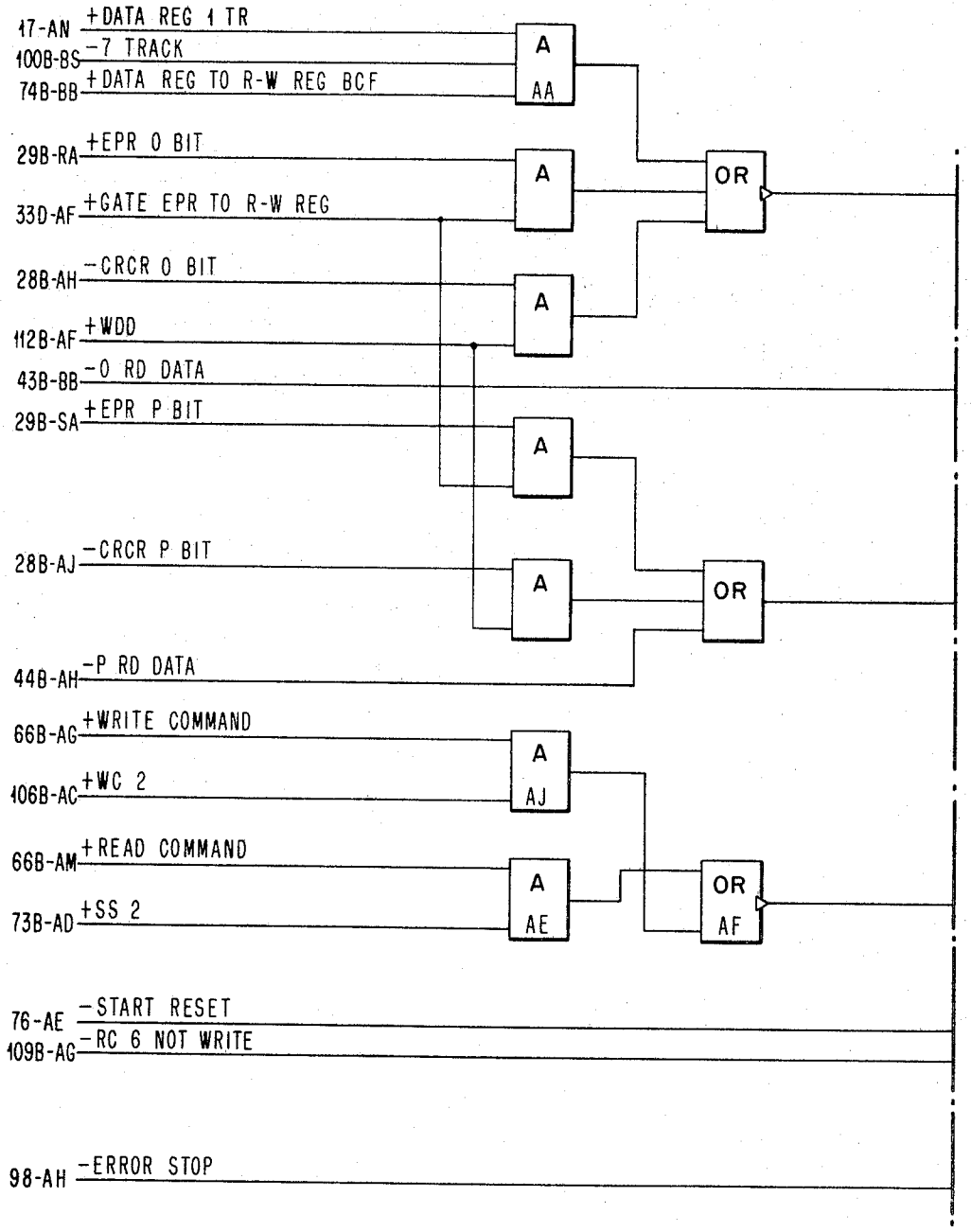

17-AN +DATA REG 1 TR
100B-BS -7 TRACK
74B-BB +DATA REG TO R-W REG BCF
A
AA
29B-RA +EPR 0 BIT
33D-AF +GATE EPR TO R-W REG
A
28B-AH -CRCR 0 BIT
112B-AF +WDD
43B-BB -0 RD DATA
A
OR
29B-SA +EPR P BIT
A
28B-AJ -CRCR P BIT
44B-AH -P RD DATA
A
OR
66B-AG +WRITE COMMAND
106B-AC +WC 2
A
AJ
66B-AM +READ COMMAND
73B-AD +SS 2
A
AE
OR
AF
76-AE -START RESET
109B-AG -RC 6 NOT WRITE
98-AH -ERROR STOP

FIG.20B

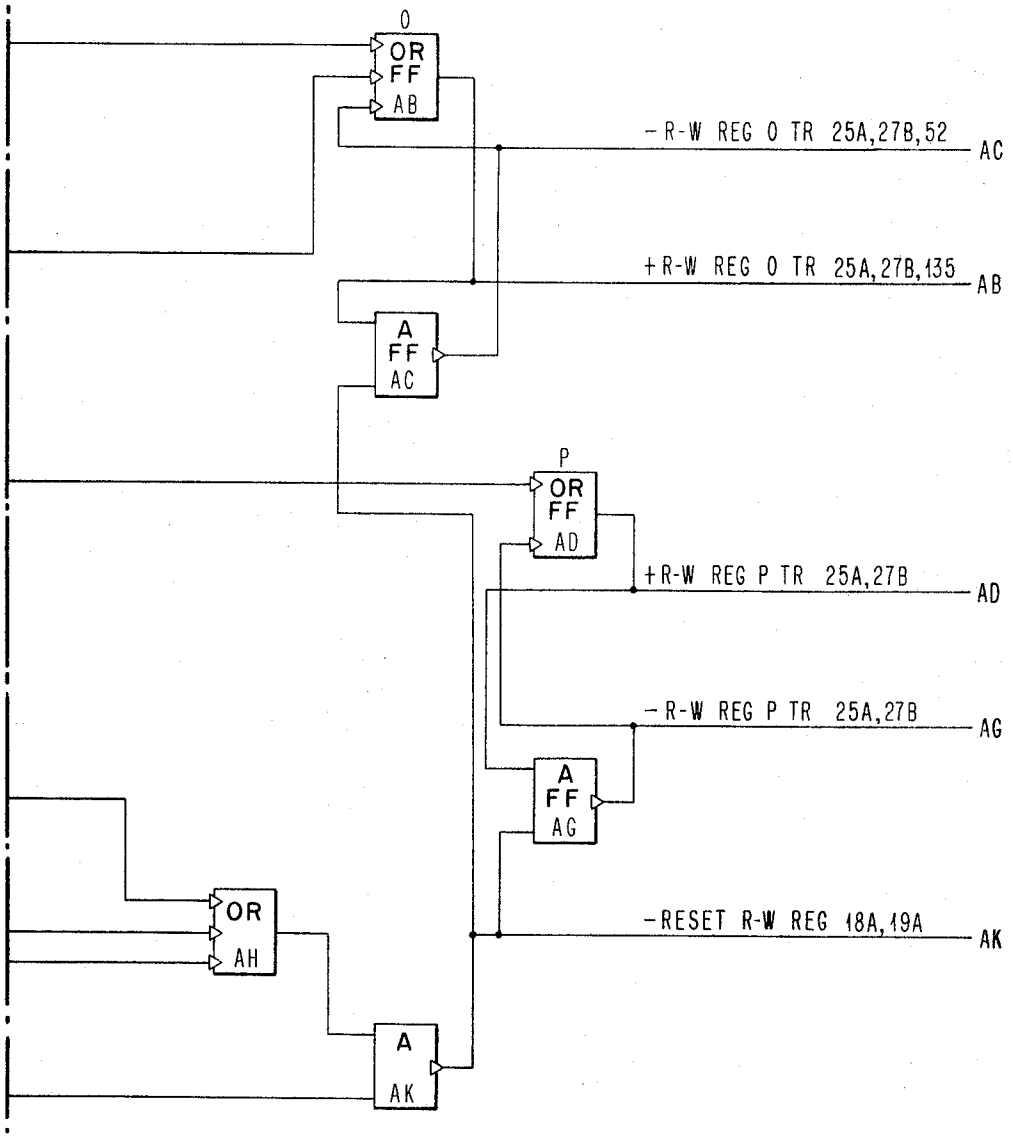

0
OR
FF
AB
-R-W REG 0 TR 25A,27B,52
AC
+R-W REG 0 TR 25A,27B,135
AB
A
FF
AC
P
OR
FF
AD
+R-W REG P TR 25A,27B
AD
-R-W REG P TR 25A,27B
AG
A
FF
AG
OR
AH
-RESET R-W REG 18A,19A
AK
A
AK

FIG. 21A

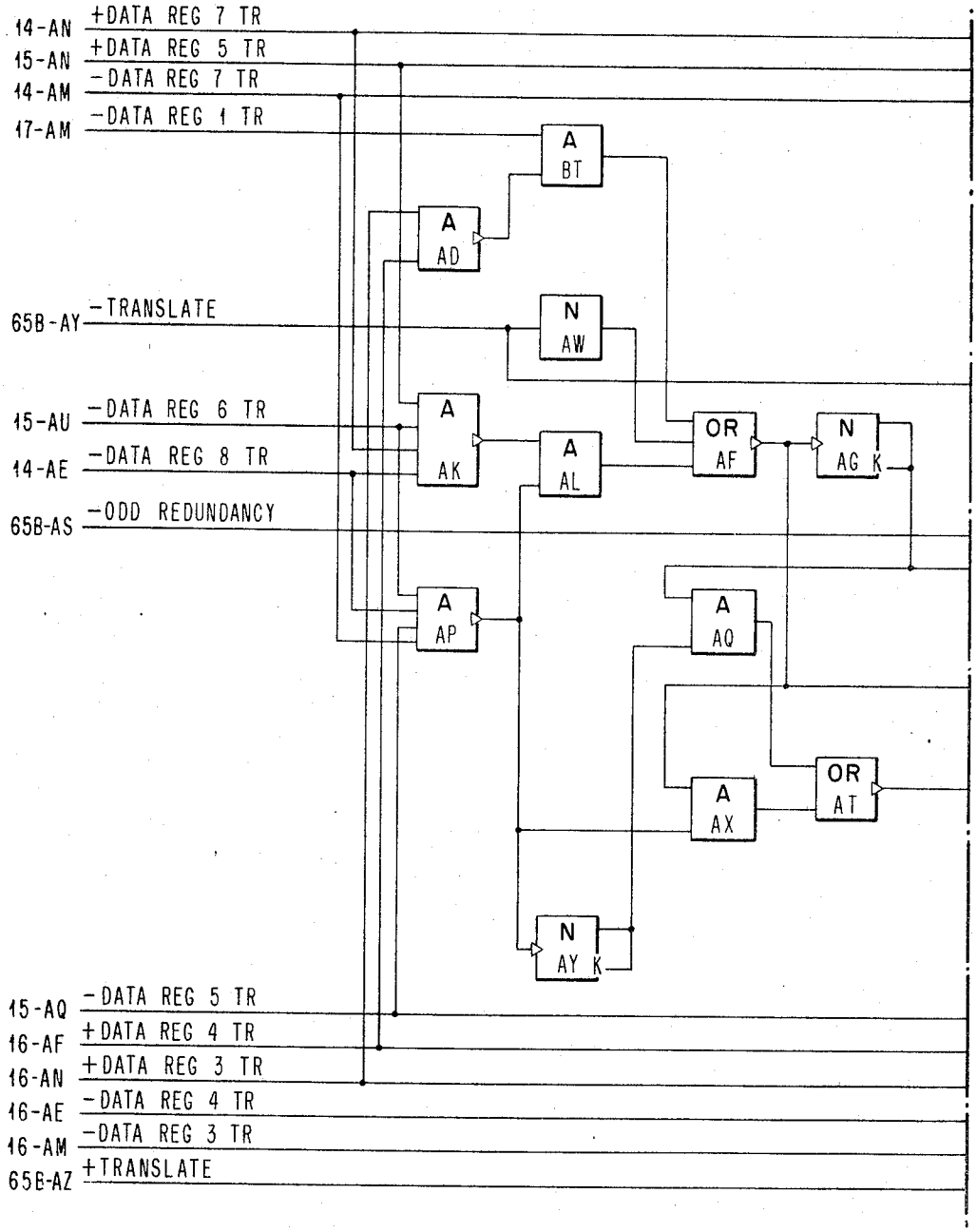

14-AN +DATA REG 7 TR
15-AN +DATA REG 5 TR
14-AM -DATA REG 7 TR
17-AM -DATA REG 1 TR
A BT
A AD
65B-AY -TRANSLATE
N AW
15-AU -DATA REG 6 TR
14-AE -DATA REG 8 TR
65B-AS -ODD REDUNDANCY
A AK
A AL
OR AF
N AG K
A AP
A AQ
OR AT
A AX
N AY K
15-AQ -DATA REG 5 TR
16-AF +DATA REG 4 TR
16-AN +DATA REG 3 TR
16-AE -DATA REG 4 TR
16-AM -DATA REG 3 TR
65B-AZ +TRANSLATE

FIG. 21B

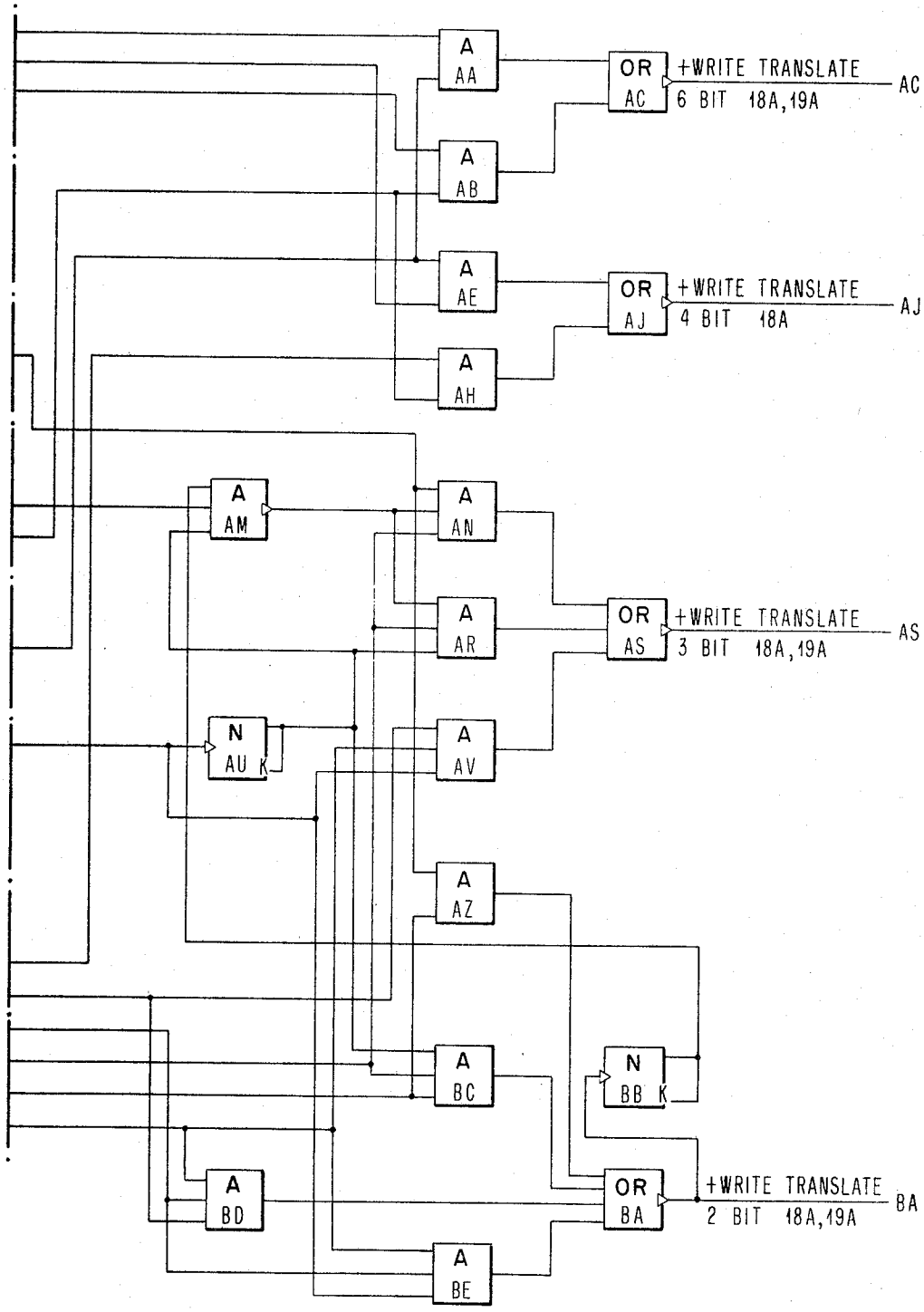

A AA
A AB
A AE
A AH
OR AC
+WRITE TRANSLATE
6 BIT 18A,19A
AC
OR AJ
+WRITE TRANSLATE
4 BIT 18A
AJ
A AM
A AN
A AR
A AV
A AZ
N AU K
OR AS
+WRITE TRANSLATE
3 BIT 18A,19A
AS
A BC
N BB K
A BD
A BE
OR BA
+WRITE TRANSLATE
2 BIT 18A,19A
BA

FIG. 22A

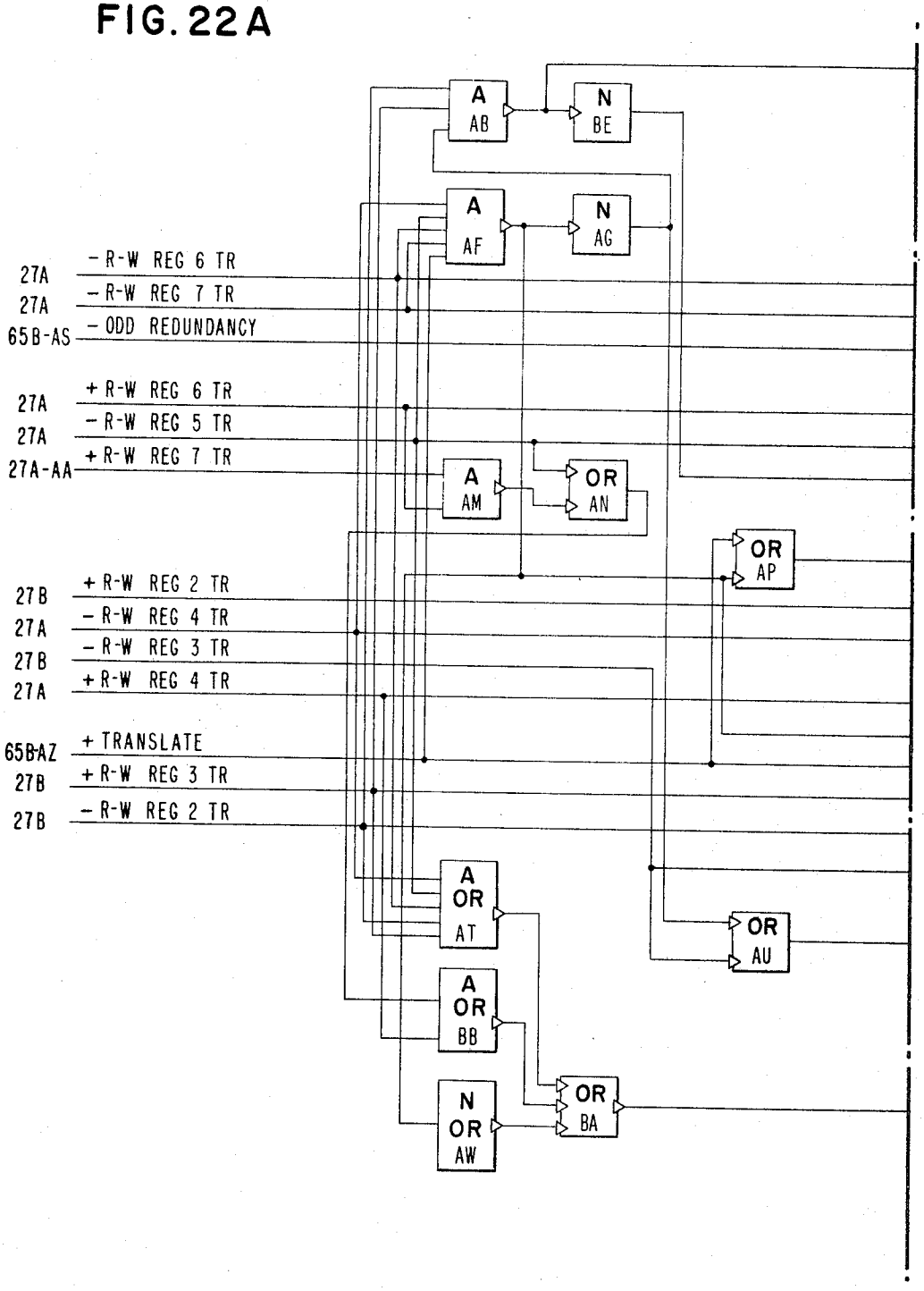

A AB
N BE
A AF
N AG
27A - R-W REG 6 TR
27A - R-W REG 7 TR
65B-AS - ODD REDUNDANCY
27A + R-W REG 6 TR
27A - R-W REG 5 TR
27A-AA + R-W REG 7 TR
A AM
OR AN
OR AP
27B + R-W REG 2 TR
27A - R-W REG 4 TR
27B - R-W REG 3 TR
27A + R-W REG 4 TR
65B-AZ + TRANSLATE
27B + R-W REG 3 TR
27B - R-W REG 2 TR
A OR AT
OR AU
A OR BB
N OR AW
OR BA

FIG. 22B

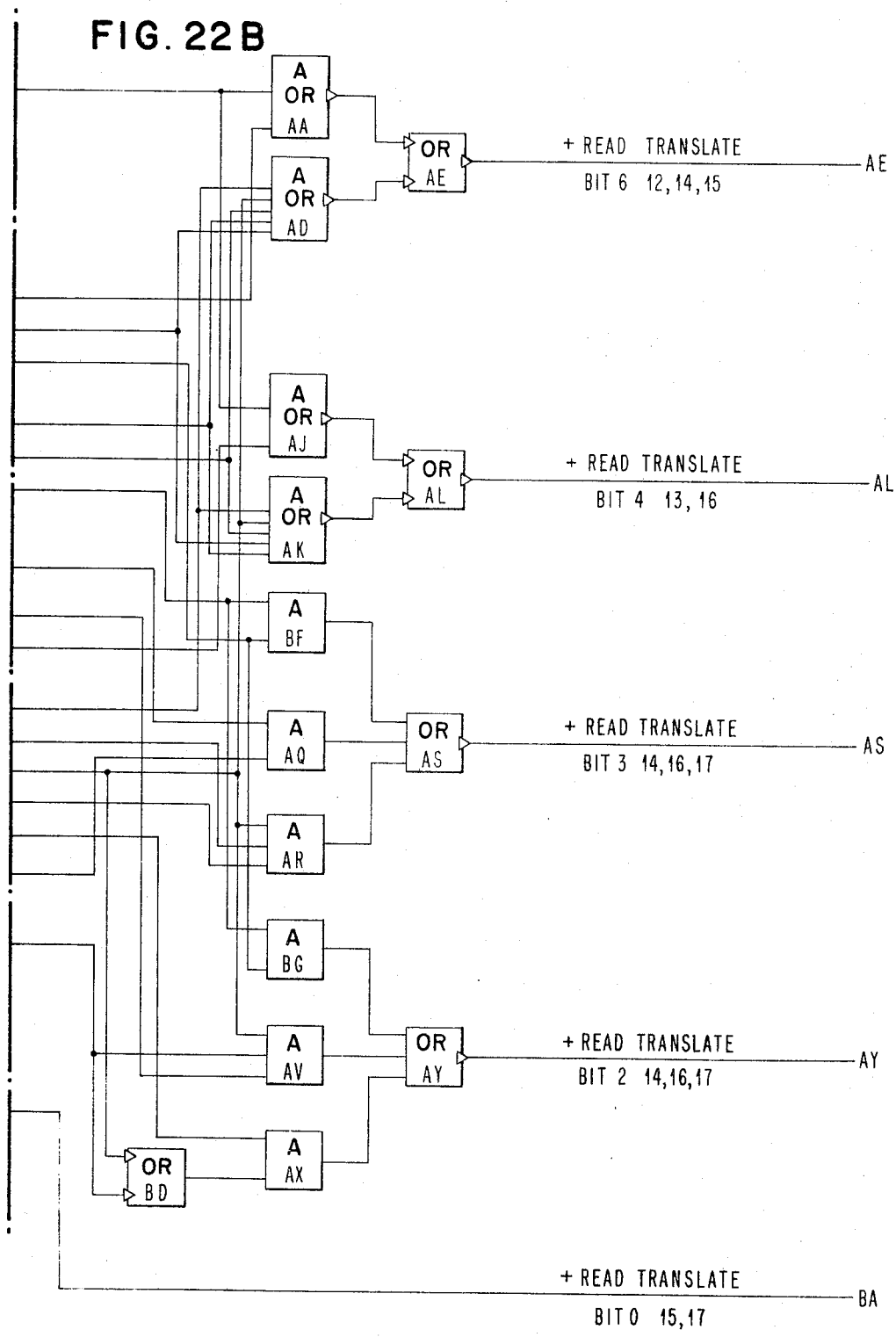

A OR AA
A OR AD
OR AE
+ READ TRANSLATE
BIT 6 12,14,15
AE
A OR AJ
A OR AK
OR AL
+ READ TRANSLATE
BIT 4 13, 16
AL
A BF
A AQ
OR AS
+ READ TRANSLATE
BIT 3 14,16,17
AS
A AR
A BG
A AV
OR AY
+ READ TRANSLATE
BIT 2 14,16,17
AY
OR BD
A AX
+ READ TRANSLATE
BIT 0 15,17
BA

FIG. 23A

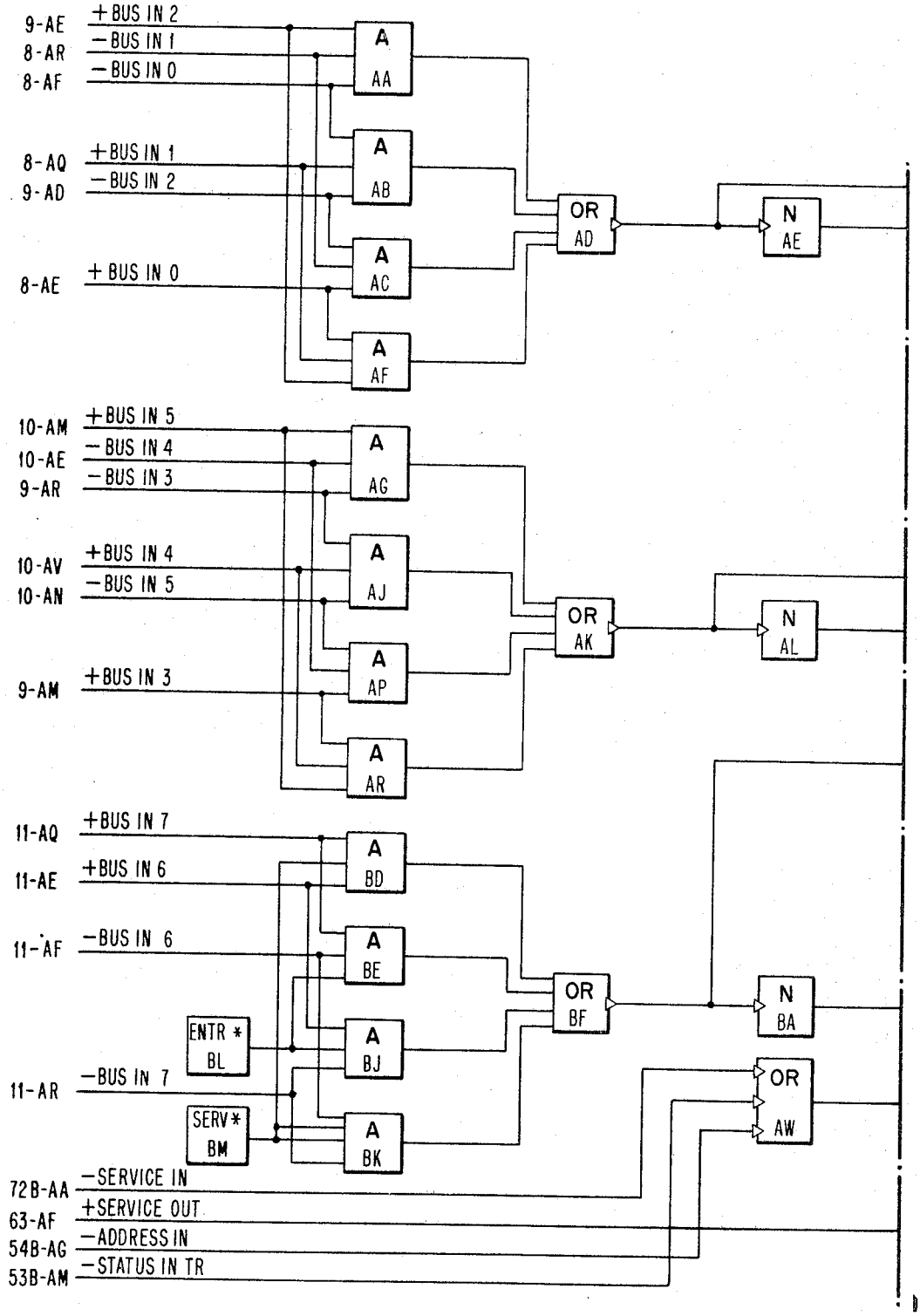

9-AE
8-AR
8-AF
+BUS IN 2
−BUS IN 1
−BUS IN 0
A AA
A AB
A AC
A AF
8-AQ
9-AD
8-AE
+BUS IN 1
−BUS IN 2
+BUS IN 0
OR AD
N AE
10-AM
10-AE
9-AR
+BUS IN 5
−BUS IN 4
−BUS IN 3
A AG
A AJ
A AP
A AR
10-AV
10-AN
9-AM
+BUS IN 4
−BUS IN 5
+BUS IN 3
OR AK
N AL
11-AQ
11-AE
11-AF
+BUS IN 7
+BUS IN 6
−BUS IN 6
A BD
A BE
A BJ
A BK
ENTR * BL
SERV * BM
11-AR
−BUS IN 7
OR BF
N BA
OR AW
72B-AA
63-AF
54B-AG
53B-AM
−SERVICE IN
+SERVICE OUT
−ADDRESS IN
−STATUS IN TR

FIG. 23B

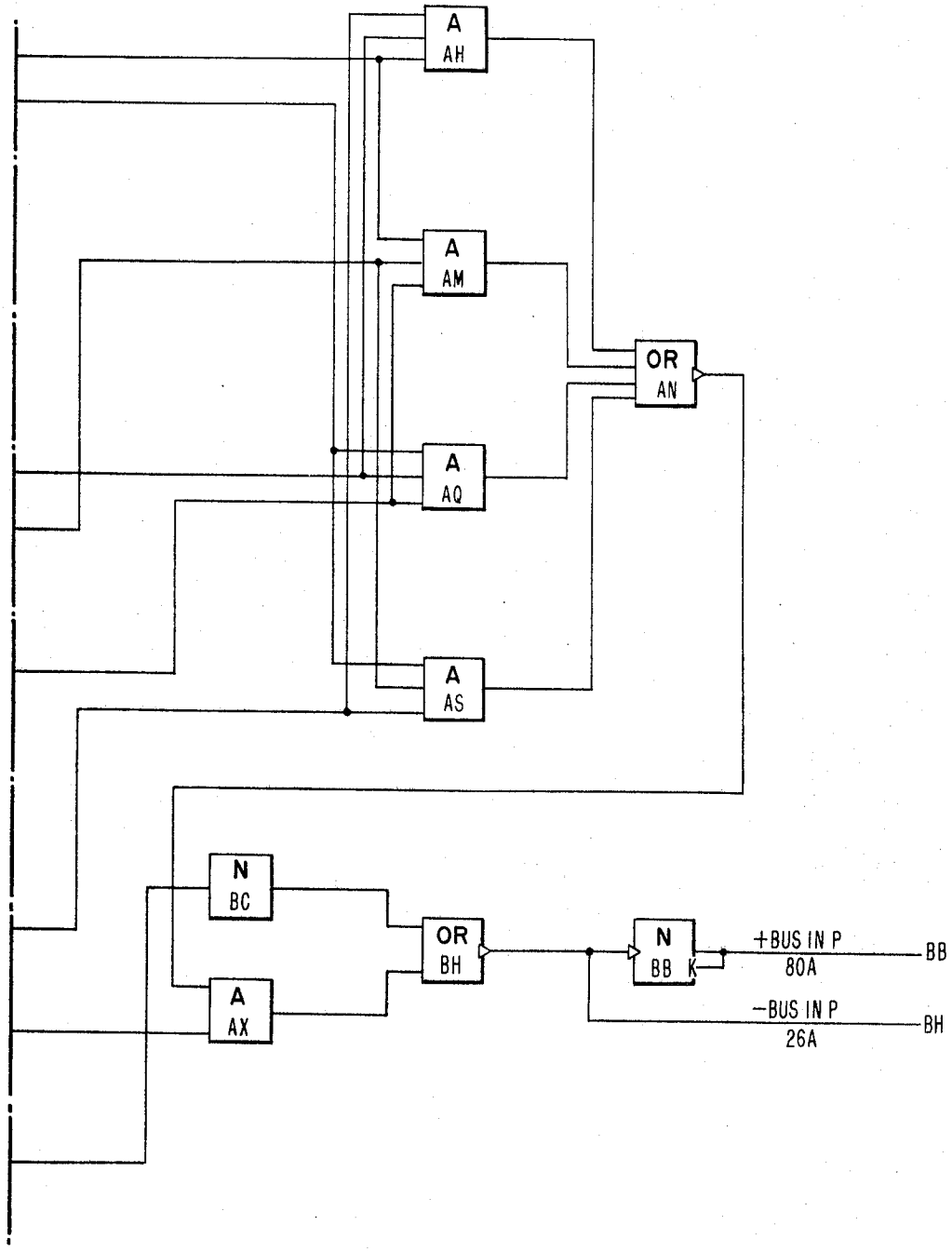

A
AH
A
AM
OR
AN
A
AQ
A
AS
N
BC
A
AX
OR
BH
N
BB K
+BUS IN P
80A
BB
−BUS IN P
26A
BH

FIG. 24A

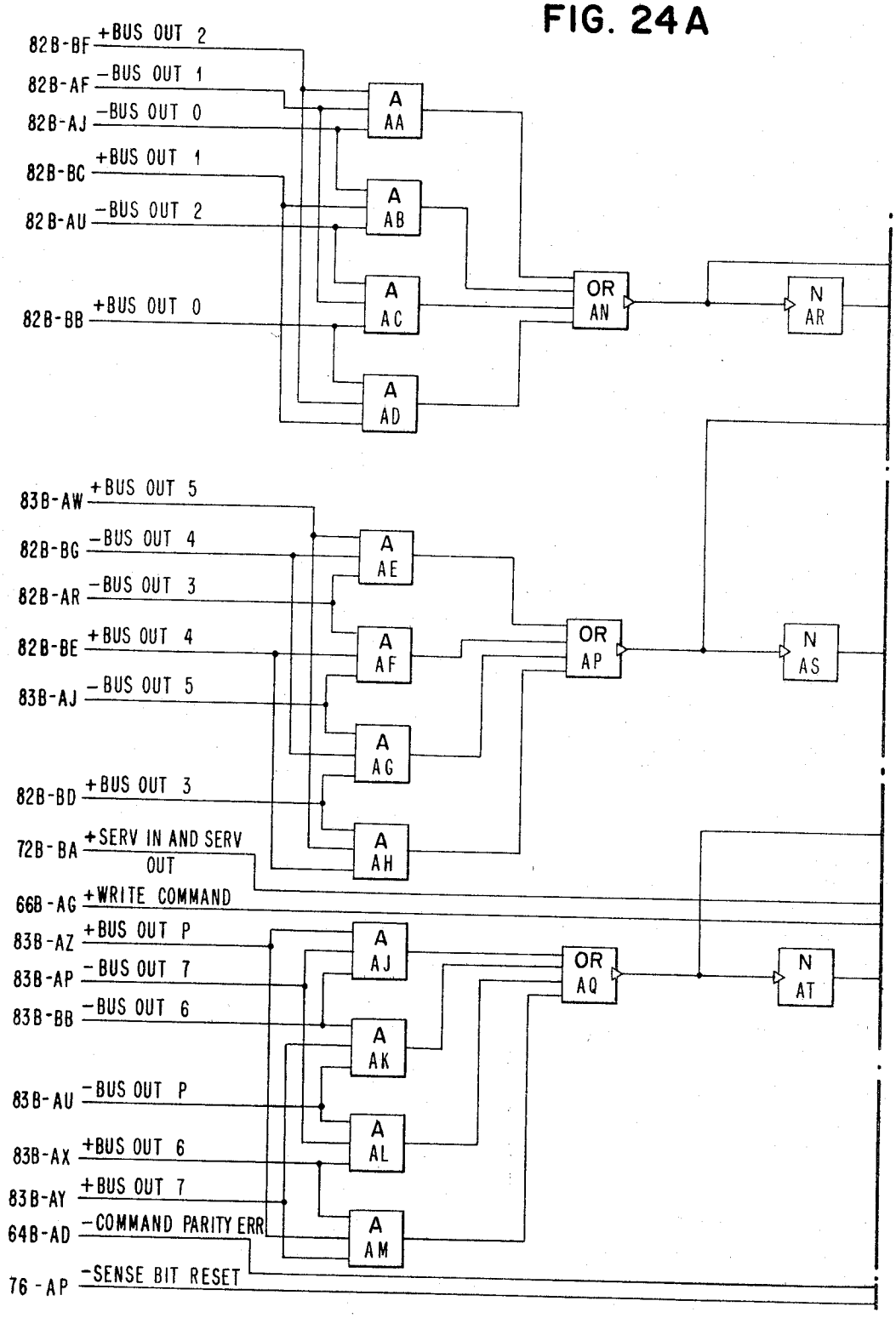

82B-BF +BUS OUT 2
82B-AF -BUS OUT 1
82B-AJ -BUS OUT 0
82B-BC +BUS OUT 1
82B-AU -BUS OUT 2
82B-BB +BUS OUT 0
A AA
A AB
A AC
A AD
OR AN
N AR
83B-AW +BUS OUT 5
82B-BG -BUS OUT 4
82B-AR -BUS OUT 3
82B-BE +BUS OUT 4
83B-AJ -BUS OUT 5
82B-BD +BUS OUT 3
72B-BA +SERV IN AND SERV OUT
A AE
A AF
A AG
A AH
OR AP
N AS
66B-AG +WRITE COMMAND
83B-AZ +BUS OUT P
83B-AP -BUS OUT 7
83B-BB -BUS OUT 6
83B-AU -BUS OUT P
83B-AX +BUS OUT 6
83B-AY +BUS OUT 7
64B-AD -COMMAND PARITY ERR
76-AP -SENSE BIT RESET
A AJ
A AK
A AL
A AM
OR AQ
N AT

FIG. 24 B

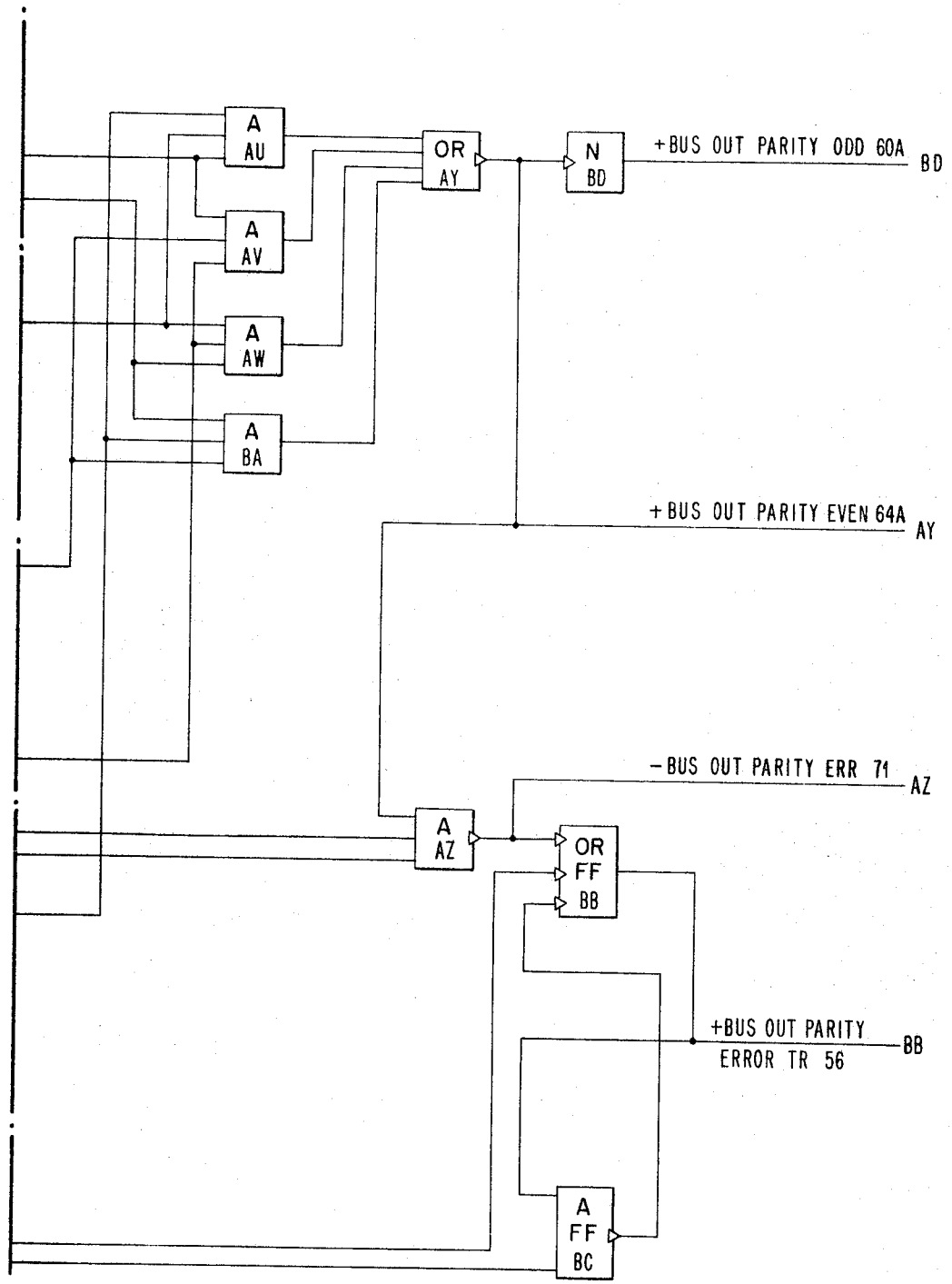

A AU
A AV
A AW
A BA
OR AY
N BD
+BUS OUT PARITY ODD 60A
BD
+BUS OUT PARITY EVEN 64A
AY
-BUS OUT PARITY ERR 71
AZ
A AZ
OR FF BB
+BUS OUT PARITY ERROR TR 56
BB
A FF BC

FIG.25A

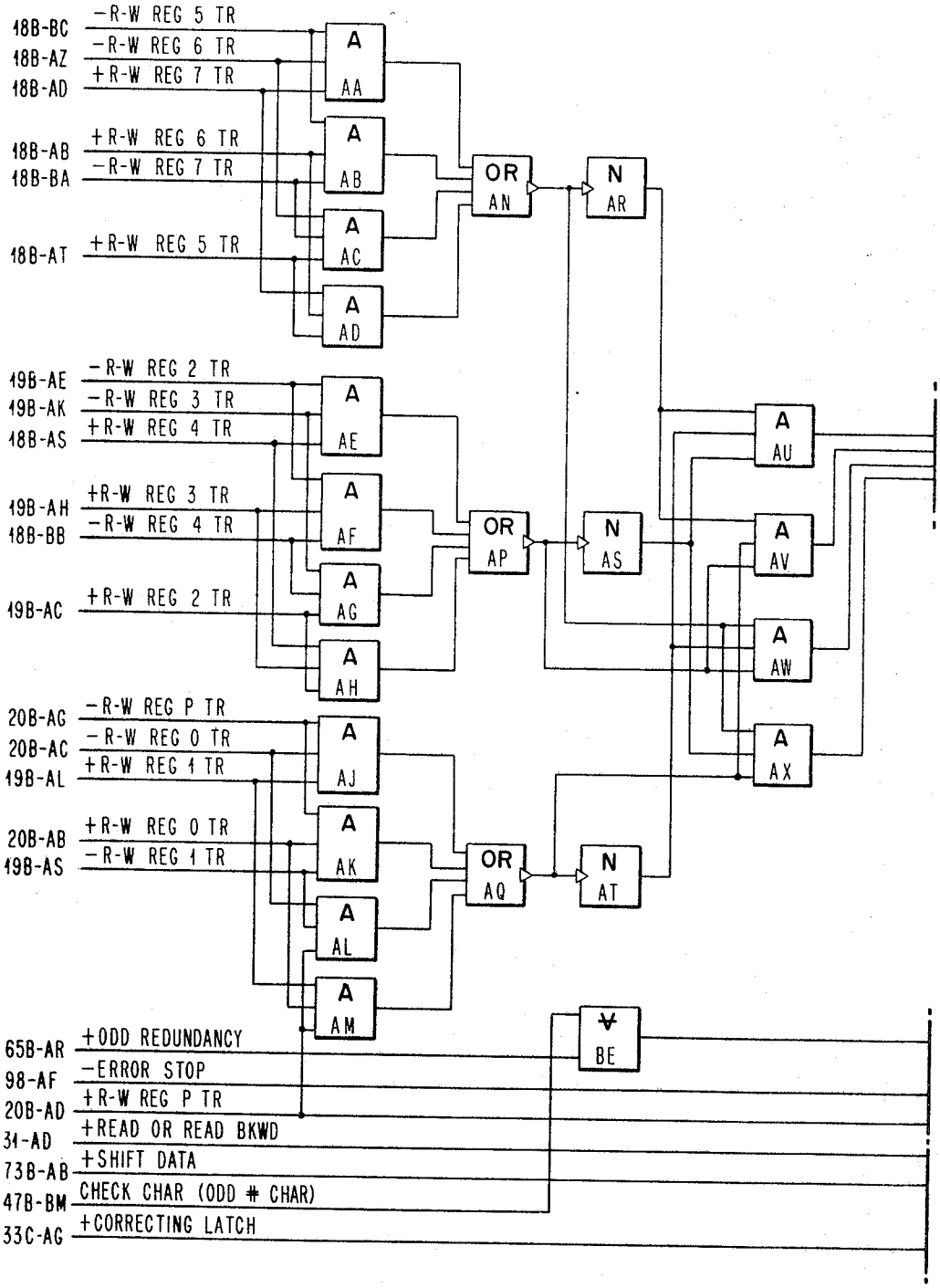

18B-BC
18B-AZ
18B-AD
−R-W REG 5 TR
−R-W REG 6 TR
+R-W REG 7 TR
A AA
18B-AB
18B-BA
+R-W REG 6 TR
−R-W REG 7 TR
A AB
18B-AT
+R-W REG 5 TR
A AC
A AD
OR AN
N AR
19B-AE
19B-AK
18B-AS
−R-W REG 2 TR
−R-W REG 3 TR
+R-W REG 4 TR
A AE
19B-AH
18B-BB
+R-W REG 3 TR
−R-W REG 4 TR
A AF
19B-AC
+R-W REG 2 TR
A AG
A AH
OR AP
N AS
A AU
A AV
A AW
A AX
20B-AG
20B-AC
19B-AL
−R-W REG P TR
−R-W REG 0 TR
+R-W REG 1 TR
A AJ
20B-AB
19B-AS
+R-W REG 0 TR
−R-W REG 1 TR
A AK
A AL
A AM
OR AQ
N AT
65B-AR
98-AF
20B-AD
31-AD
73B-AB
47B-BM
33C-AG
+ODD REDUNDANCY
−ERROR STOP
+R-W REG P TR
+READ OR READ BKWD
+SHIFT DATA
CHECK CHAR (ODD # CHAR)
+CORRECTING LATCH
V BE

FIG.25B

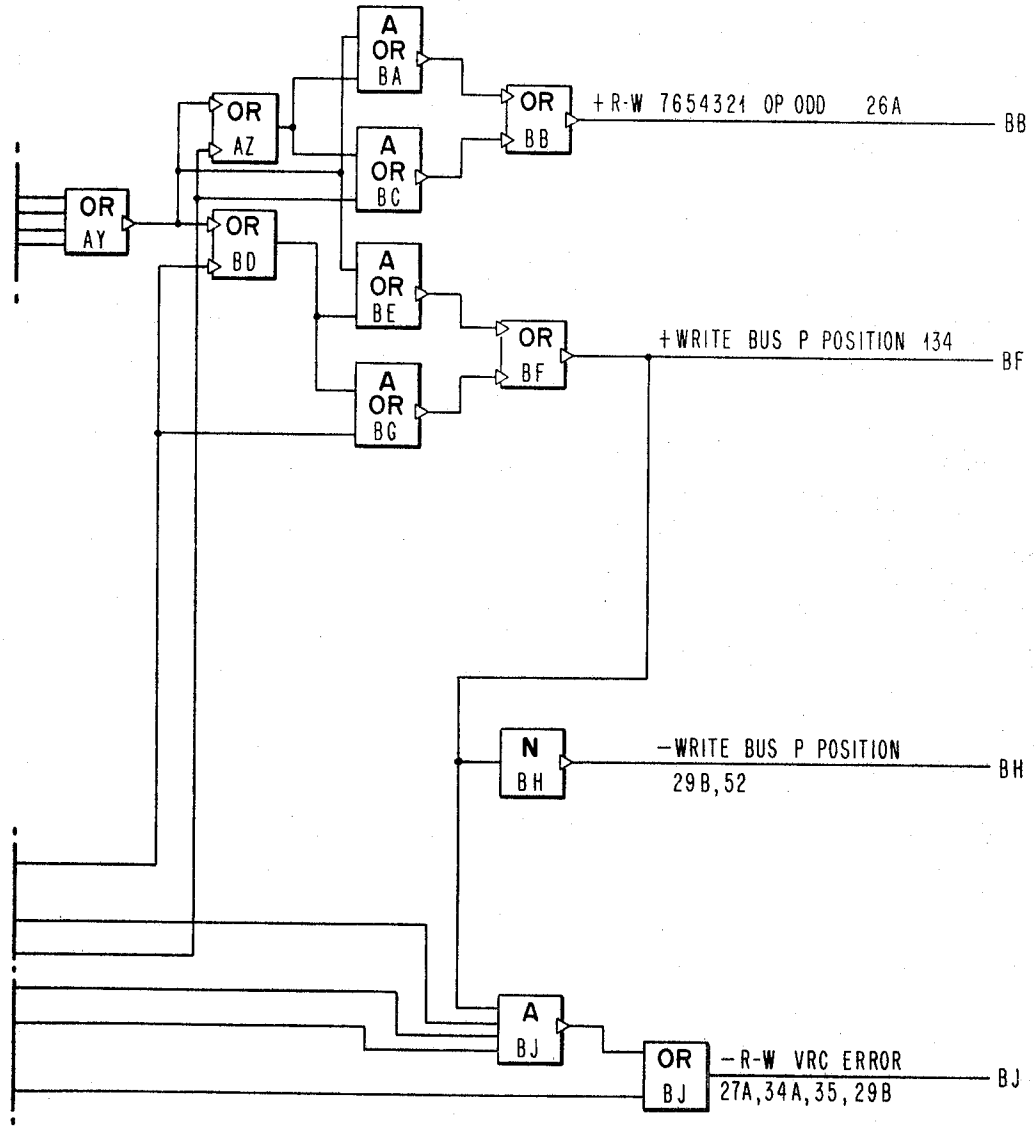

A OR BA
OR AZ
OR BB
+R-W 7654321 OP ODD 26A
BB
A OR BC
OR AY
OR BD
A OR BE
OR BF
+WRITE BUS P POSITION 134
BF
A OR BG
N BH
-WRITE BUS P POSITION
29B,52
BH
A BJ
OR BJ
-R-W VRC ERROR
27A,34A,35,29B
BJ

FIG. 26A

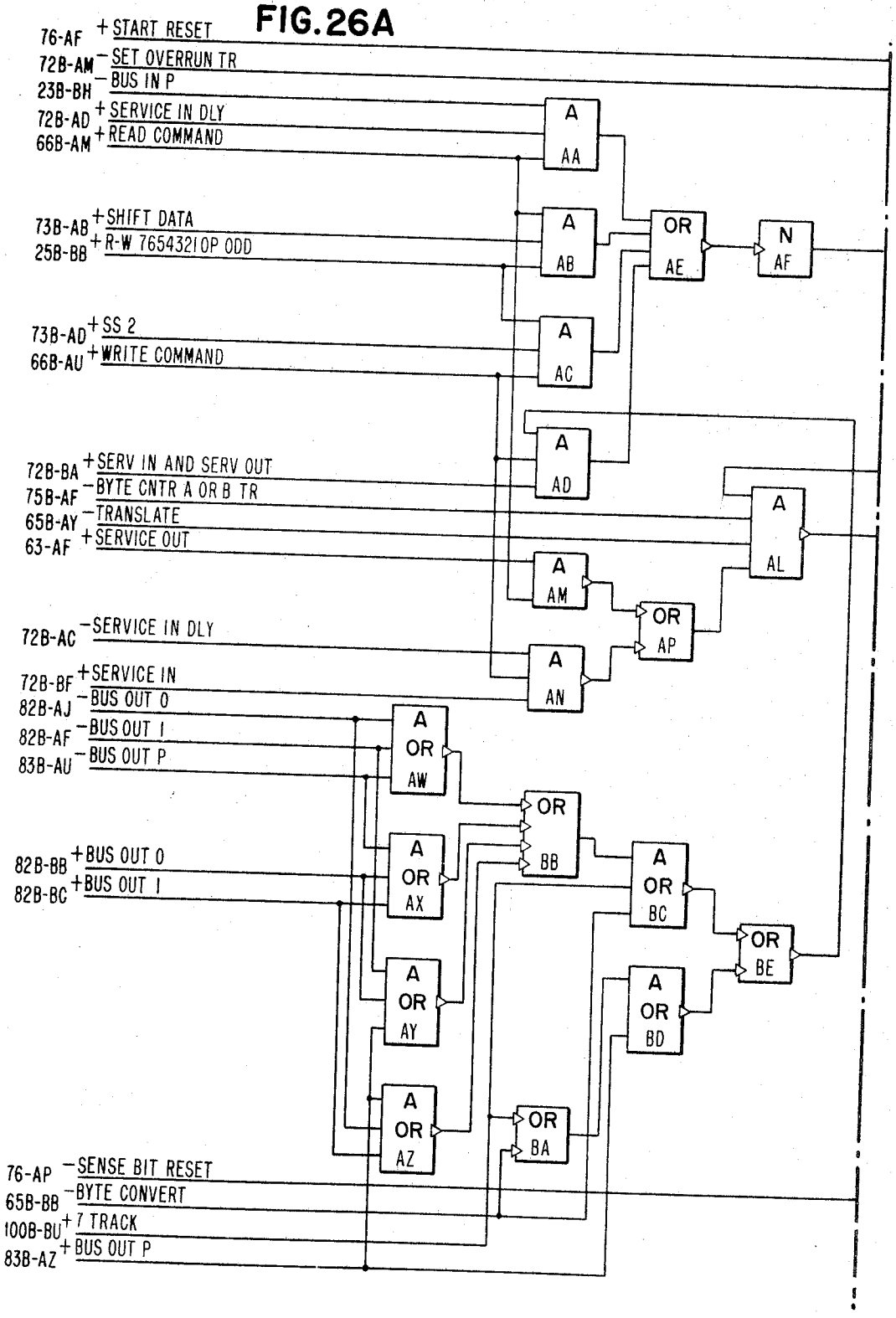

76-AF +START RESET
72B-AM −SET OVERRUN TR
23B-BH −BUS IN P
72B-AD +SERVICE IN DLY
66B-AM +READ COMMAND
A AA
73B-AB +SHIFT DATA
25B-BB +R-W 7654321OP ODD
A AB
OR AE
N AF
73B-AD +SS 2
66B-AU +WRITE COMMAND
A AC
A AD
72B-BA +SERV IN AND SERV OUT
75B-AF −BYTE CNTR A OR B TR
65B-AY −TRANSLATE
63-AF +SERVICE OUT
A AL
A AM
OR AP
72B-AC −SERVICE IN DLY
72B-BF +SERVICE IN
82B-AJ −BUS OUT 0
82B-AF −BUS OUT 1
83B-AU −BUS OUT P
A OR AW
OR BB
82B-BB +BUS OUT 0
82B-BC +BUS OUT 1
A OR AX
A OR BC
OR BE
A OR AY
A OR BD
A OR AZ
OR BA
76-AP −SENSE BIT RESET
65B-BB −BYTE CONVERT
100B-BU +7 TRACK
83B-AZ +BUS OUT P

FIG. 26B

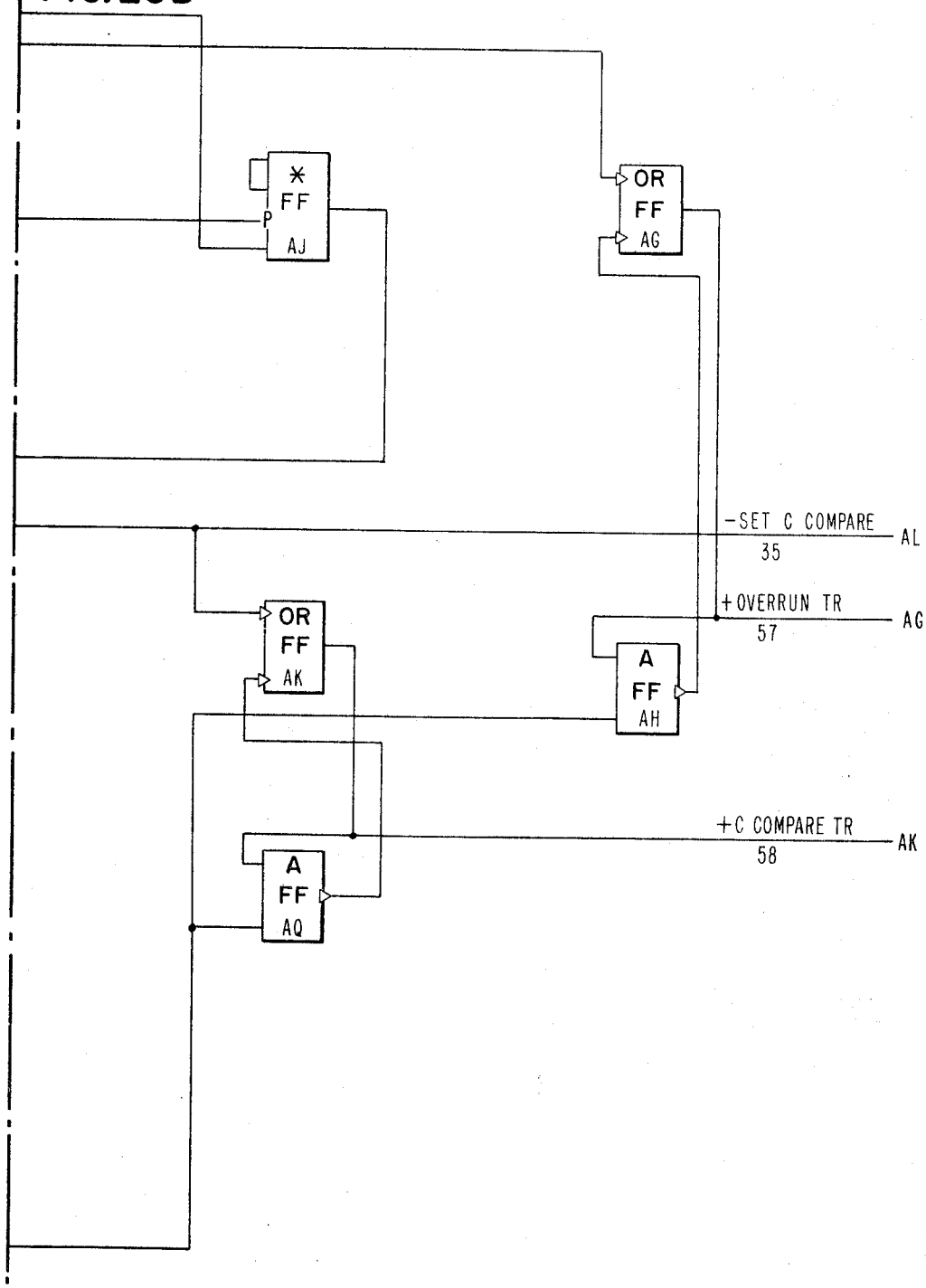
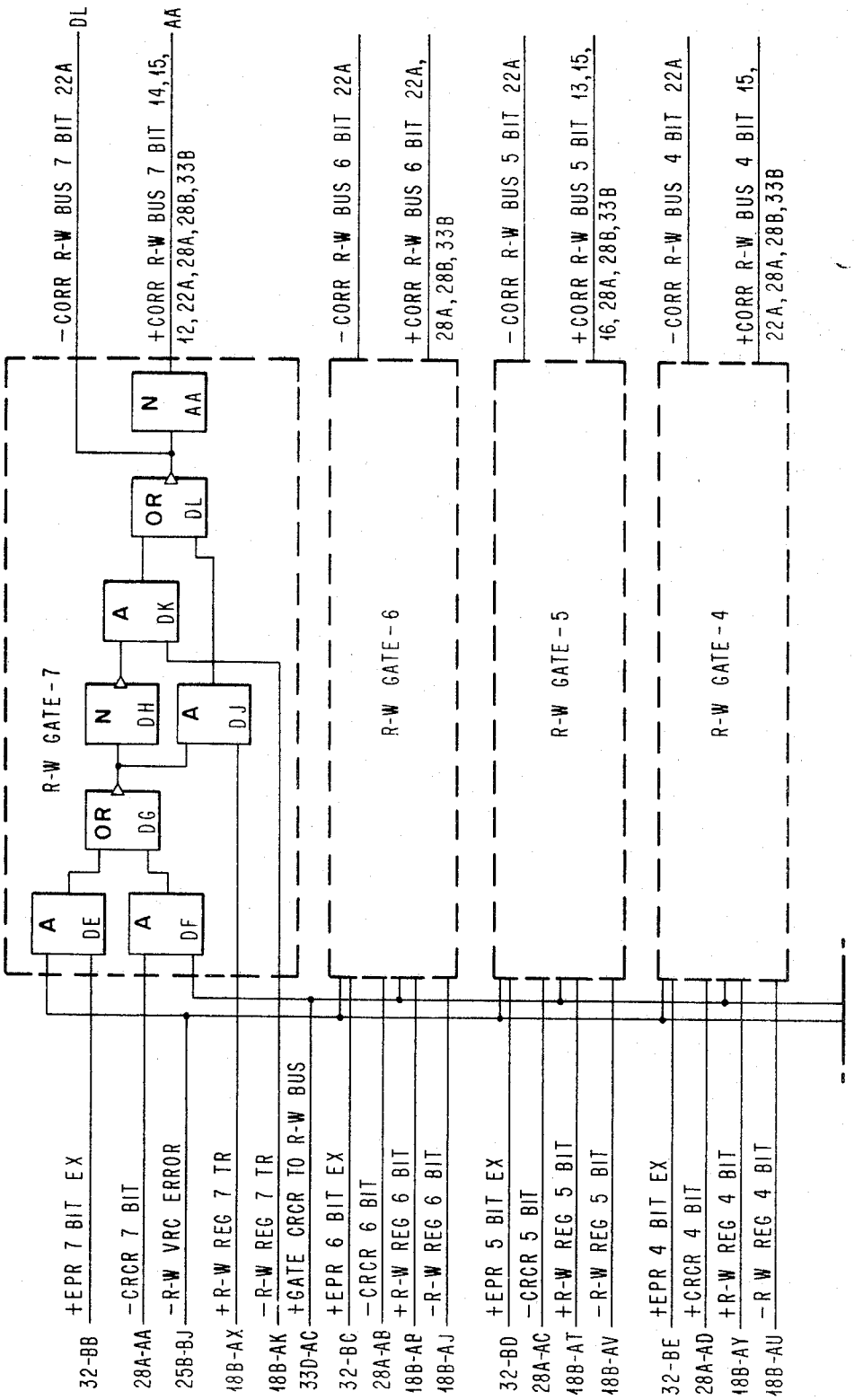

*
FF
P
AJ
OR
FF
AG
-SET C COMPARE
35
AL
+OVERRUN TR
57
AG
OR
FF
AK
A
FF
AH
+C COMPARE TR
58
AK
A
FF
AQ

FIG.27A


R-W GATE-7
R-W GATE-6
R-W GATE-5
R-W GATE-4
A DE
A DF
OR DG
N DH
A DJ
A DK
OR DL
N AA
-CORR R-W BUS 7 BIT 22A DL
+CORR R-W BUS 7 BIT 14,15, 12, 22A, 28A, 28B, 33B AA
-CORR R-W BUS 6 BIT 22A
+CORR R-W BUS 6 BIT 22A, 28A, 28B, 33B
-CORR R-W BUS 5 BIT 22A
+CORR R-W BUS 5 BIT 13,15, 16, 28A, 28B, 33B
-CORR R-W BUS 4 BIT 22A
+CORR R-W BUS 4 BIT 15, 22A, 28A, 28B, 33B
32-BB +EPR 7 BIT EX
28A-AA -CRCR 7 BIT
25B-BJ -R-W VRC ERROR
18B-AX +R-W REG 7 TR
18B-AK -R-W REG 7 TR
33D-AC +GATE CRCR TO R-W BUS
32-BC +EPR 6 BIT EX
28A-AB -CRCR 6 BIT
18B-AR +R-W REG 6 BIT
18B-AJ -R-W REG 6 BIT
32-BD +EPR 5 BIT EX
28A-AC -CRCR 5 BIT
18B-AT +R-W REG 5 BIT
18B-AV -R-W REG 5 BIT
32-BE +EPR 4 BIT EX
28A-AD +CRCR 4 BIT
18B-AY +R-W REG 4 BIT
18B-AU -R W REG 4 BIT

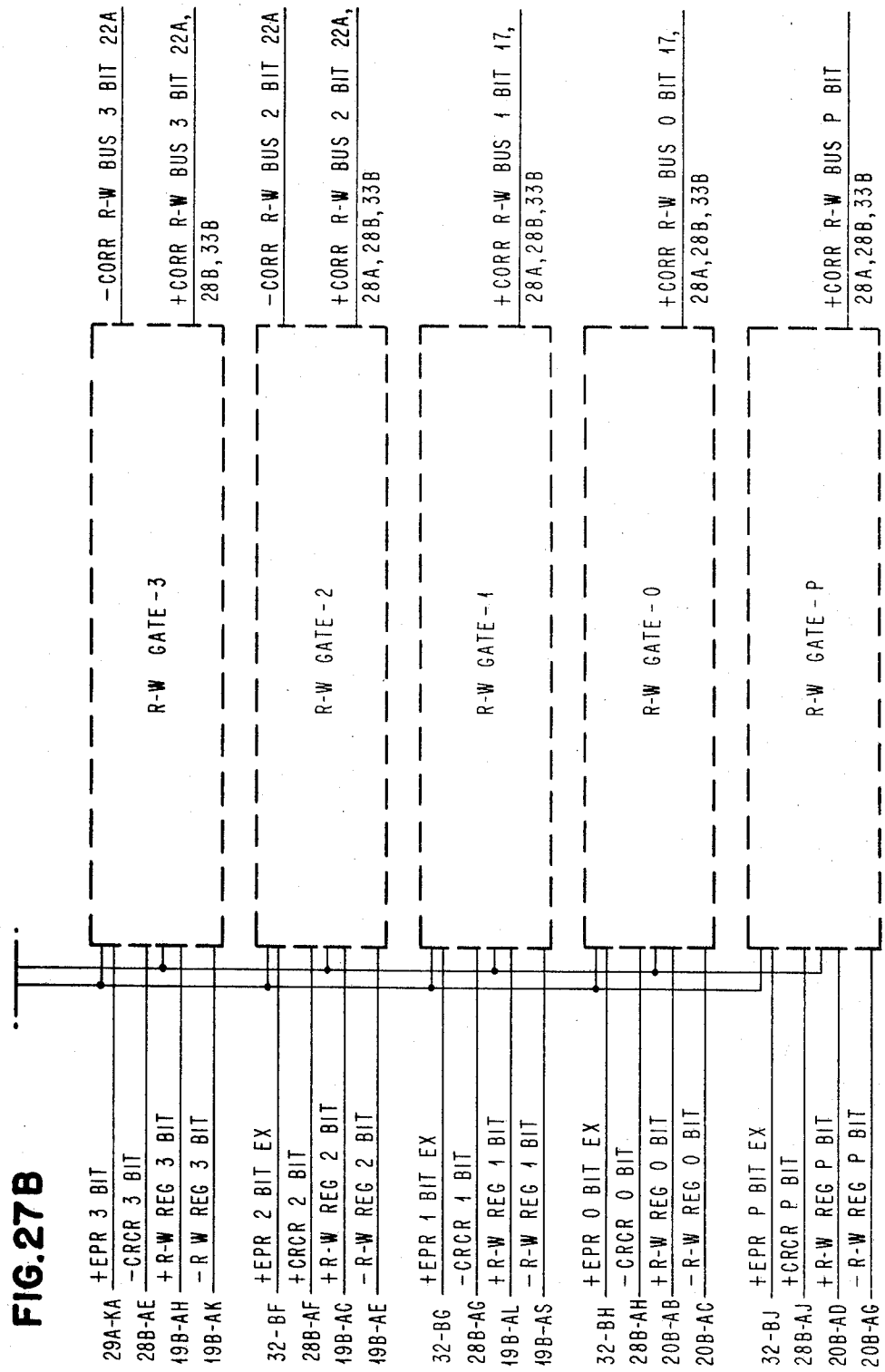


FIG.27B
29A-KA +EPR 3 BIT
28B-AE -CRCR 3 BIT
19B-AH + R-W REG 3 BIT
19B-AK - R W REG 3 BIT
32-BF +EPR 2 BIT EX
28B-AF +CRCR 2 BIT
19B-AC + R-W REG 2 BIT
19B-AE - R-W REG 2 BIT
32-BG +EPR 1 BIT EX
28B-AG -CRCR 1 BIT
19B-AL + R-W REG 1 BIT
19B-AS - R-W REG 1 BIT
32-BH +EPR 0 BIT EX
28B-AH - CRCR 0 BIT
20B-AB + R-W REG 0 BIT
20B-AC - R-W REG 0 BIT
32-BJ +EPR P BIT EX
28B-AJ +CRCR P BIT
20B-AD + R-W REG P BIT
20B-AG - R-W REG P BIT
R-W GATE-3
R-W GATE-2
R-W GATE-1
R-W GATE-0
R-W GATE-P
-CORR R-W BUS 3 BIT 22A
+CORR R-W BUS 3 BIT 22A, 28B, 33B
-CORR R-W BUS 2 BIT 22A
+CORR R-W BUS 2 BIT 22A, 28A, 28B, 33B
+CORR R-W BUS 1 BIT 17, 28A, 28B, 33B
+CORR R-W BUS 0 BIT 17, 28A, 28B, 33B
+CORR R-W BUS P BIT 28A, 28B, 33B

FIG.28A

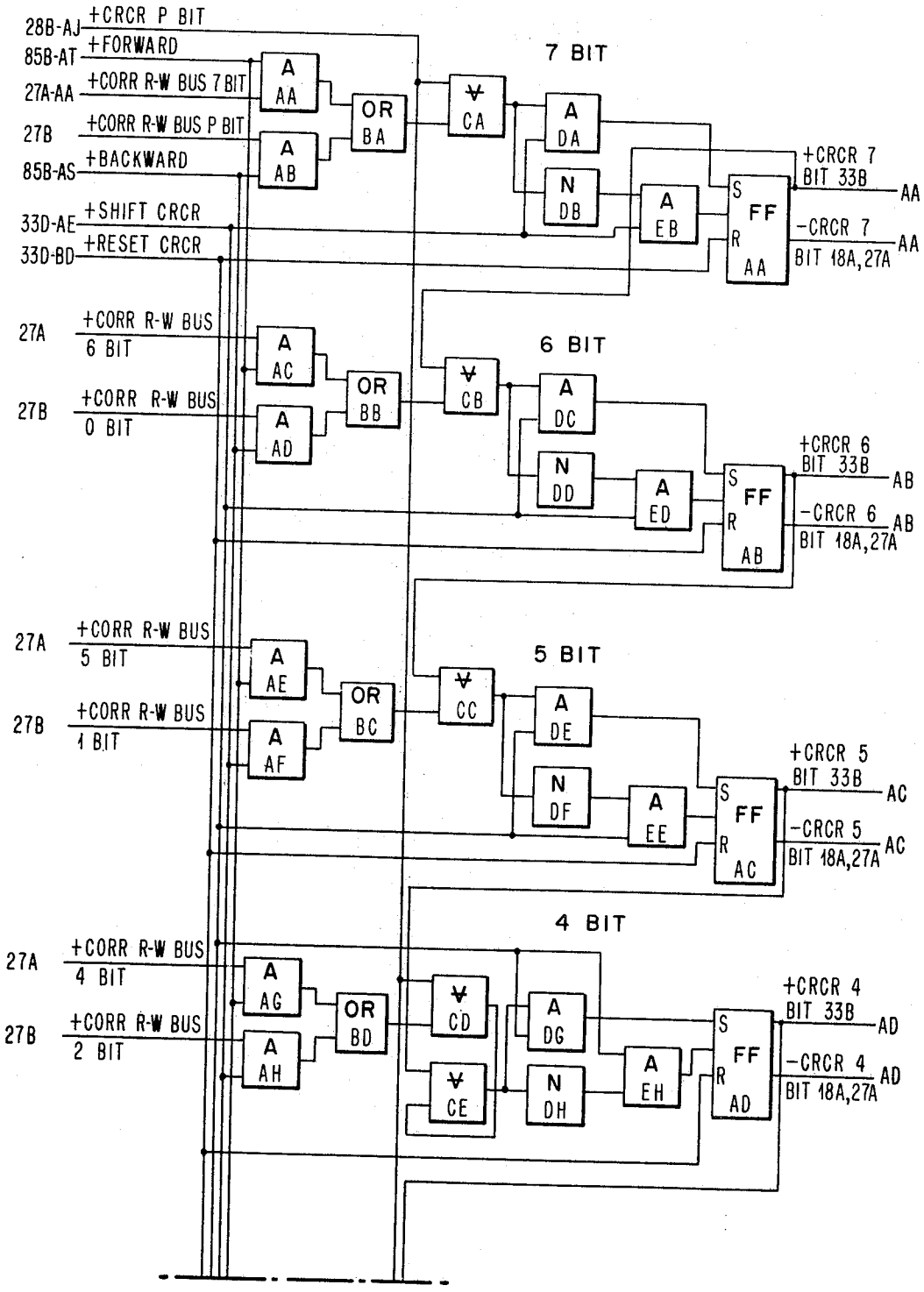

28B-AJ +CRCR P BIT
85B-AT +FORWARD
27A-AA +CORR R-W BUS 7 BIT
27B +CORR R-W BUS P BIT
85B-AS +BACKWARD
33D-AE +SHIFT CRCR
33D-BD +RESET CRCR
7 BIT
A AA
OR BA
A AB
CA
A DA
N DB
A EB
S FF R AA
+CRCR 7 BIT 33B AA
-CRCR 7 BIT 18A,27A AA
27A +CORR R-W BUS 6 BIT
27B +CORR R-W BUS 0 BIT
6 BIT
A AC
OR BB
A AD
CB
A DC
N DD
A ED
S FF R AB
+CRCR 6 BIT 33B AB
-CRCR 6 BIT 18A,27A AB
27A +CORR R-W BUS 5 BIT
27B +CORR R-W BUS 1 BIT
5 BIT
A AE
OR BC
A AF
CC
A DE
N DF
A EE
S FF R AC
+CRCR 5 BIT 33B AC
-CRCR 5 BIT 18A,27A AC
27A +CORR R-W BUS 4 BIT
27B +CORR R-W BUS 2 BIT
4 BIT
A AG
OR BD
A AH
CD
CE
A DG
N DH
A EH
S FF R AD
+CRCR 4 BIT 33B AD
-CRCR 4 BIT 18A,27A AD

FIG.28B

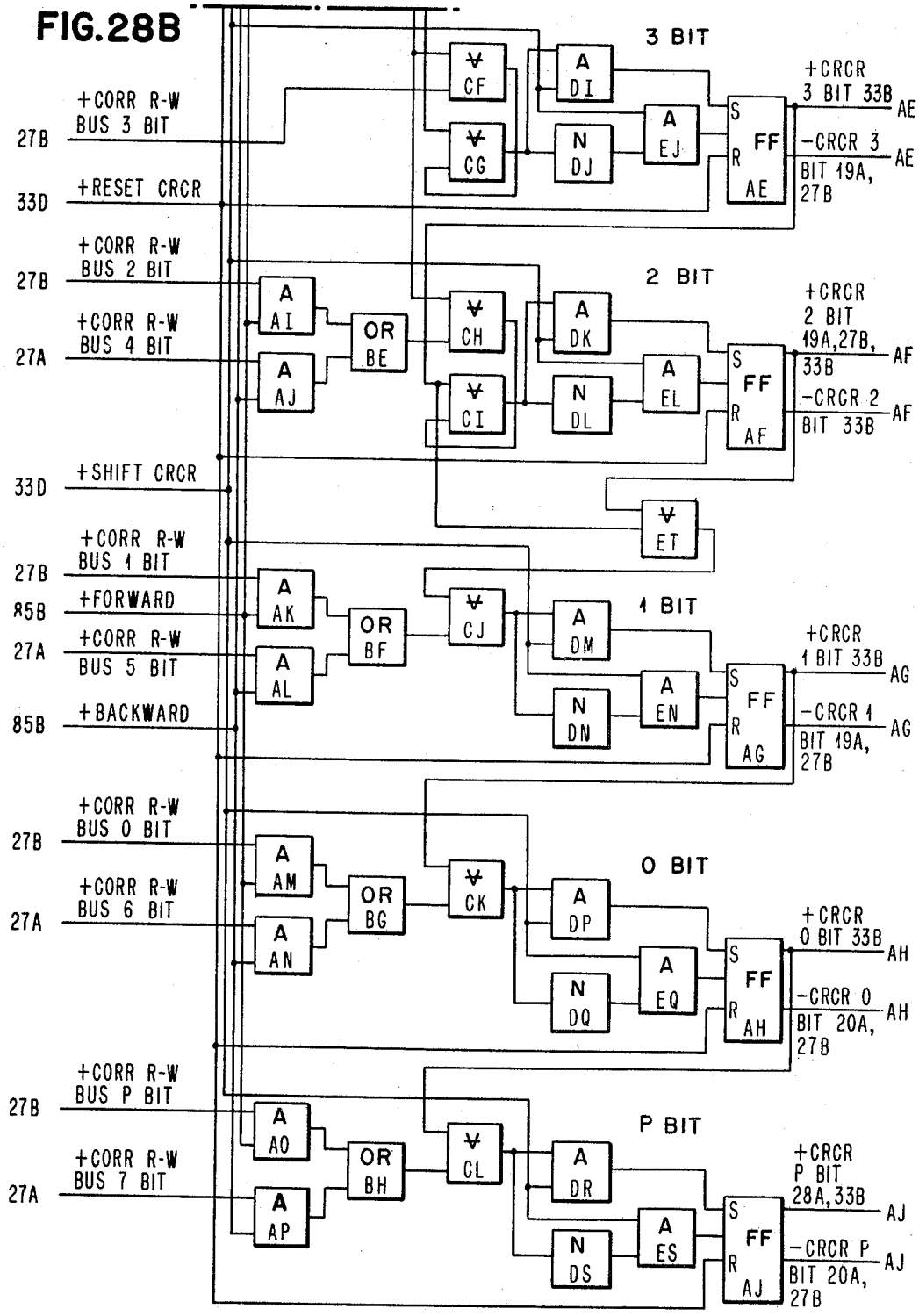

3 BIT
+CORR R-W BUS 3 BIT
27B
+RESET CRCR
33D
CF
CG
A DI
N DJ
A EJ
S FF R AE
+CRCR 3 BIT 33B AE
-CRCR 3 BIT 19A, 27B AE
2 BIT
+CORR R-W BUS 2 BIT
27B
+CORR R-W BUS 4 BIT
27A
A AI
A AJ
OR BE
CH
CI
A DK
N DL
A EL
S FF R AF
+CRCR 2 BIT 19A,27B, 33B AF
-CRCR 2 BIT 33B AF
+SHIFT CRCR
33D
ET
1 BIT
+CORR R-W BUS 1 BIT
27B
+FORWARD
85B
+CORR R-W BUS 5 BIT
27A
+BACKWARD
85B
A AK
A AL
OR BF
CJ
A DM
N DN
A EN
S FF R AG
+CRCR 1 BIT 33B AG
-CRCR 1 BIT 19A, 27B AG
0 BIT
+CORR R-W BUS 0 BIT
27B
+CORR R-W BUS 6 BIT
27A
A AM
A AN
OR BG
CK
A DP
N DQ
A EQ
S FF R AH
+CRCR 0 BIT 33B AH
-CRCR 0 BIT 20A, 27B AH
P BIT
+CORR R-W BUS P BIT
27B
+CORR R-W BUS 7 BIT
27A
A AO
A AP
OR BH
CL
A DR
N DS
A ES
S FF R AJ
+CRCR P BIT 28A,33B AJ
-CRCR P BIT 20A, 27B AJ

FIG.29A

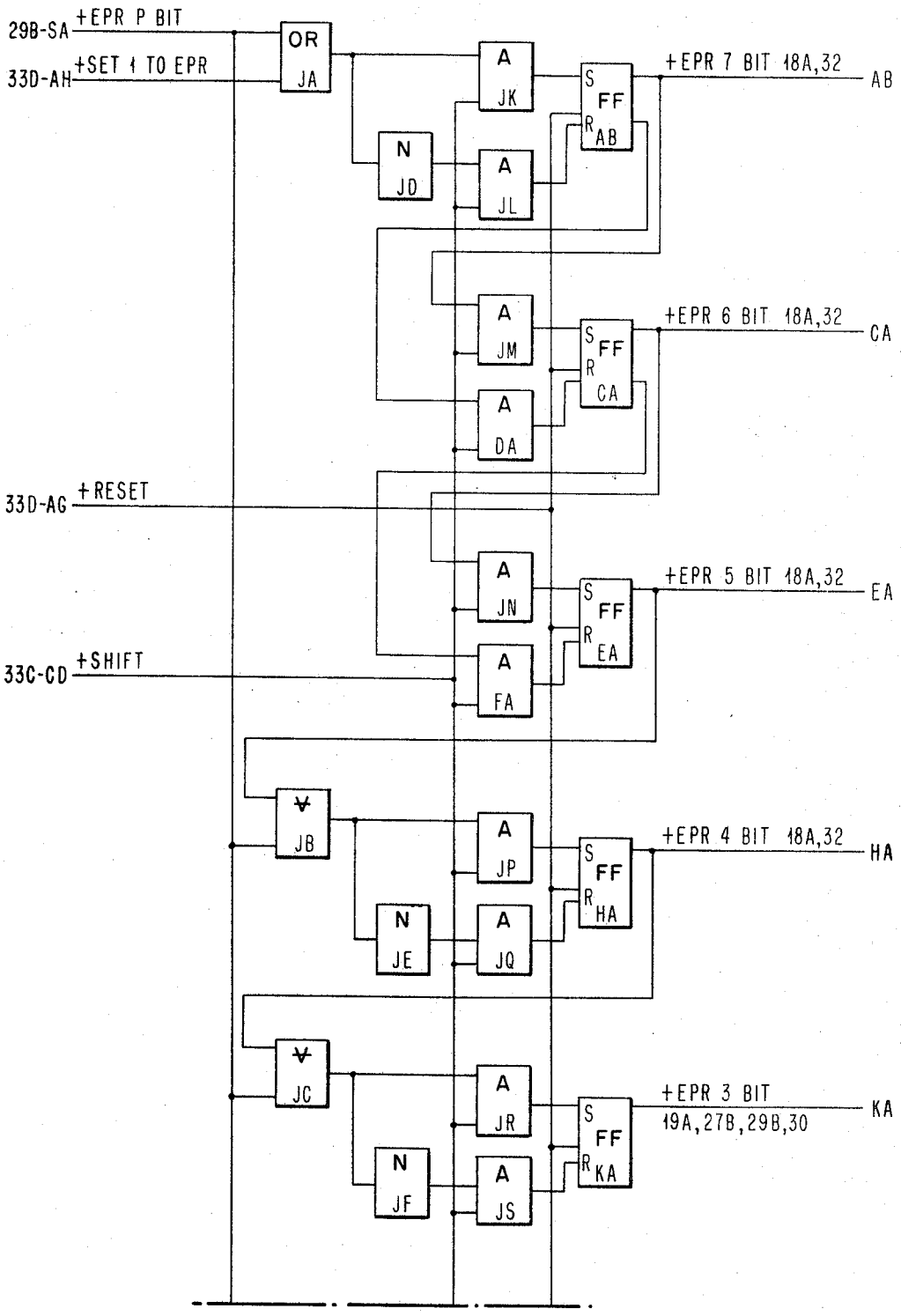

29B-SA
+EPR P BIT
33D-AH
+SET 1 TO EPR
OR
JA
N
JD
A
JK
A
JL
S
FF
R
AB
+EPR 7 BIT 18A,32
AB
A
JM
A
DA
S
FF
R
CA
+EPR 6 BIT 18A,32
CA
33D-AG
+RESET
A
JN
A
FA
S
FF
R
EA
+EPR 5 BIT 18A,32
EA
33C-CD
+SHIFT
JB
N
JE
A
JP
A
JQ
S
FF
R
HA
+EPR 4 BIT 18A,32
HA
JC
N
JF
A
JR
A
JS
S
FF
R
KA
+EPR 3 BIT
19A,27B,29B,30
KA

FIG.29B

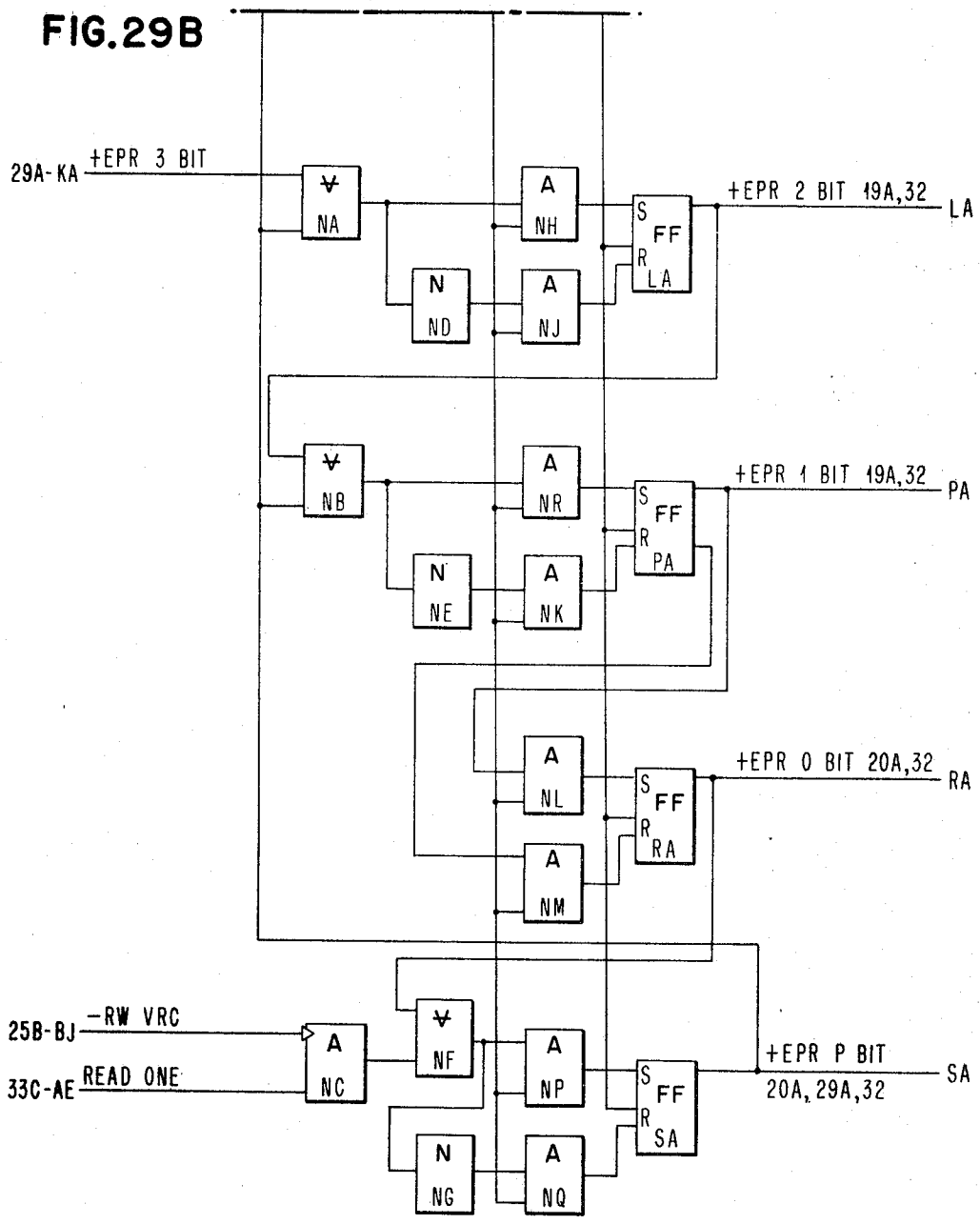

29A-KA +EPR 3 BIT
NA
NB
NC
ND
NE
NF
NG
NH
NJ
NK
NL
NM
NP
NQ
NR
FF LA
FF PA
FF RA
FF SA
+EPR 2 BIT 19A,32 LA
+EPR 1 BIT 19A,32 PA
+EPR 0 BIT 20A,32 RA
+EPR P BIT 20A, 29A,32 SA
25B-BJ -RW VRC
33C-AE READ ONE

FIG.30

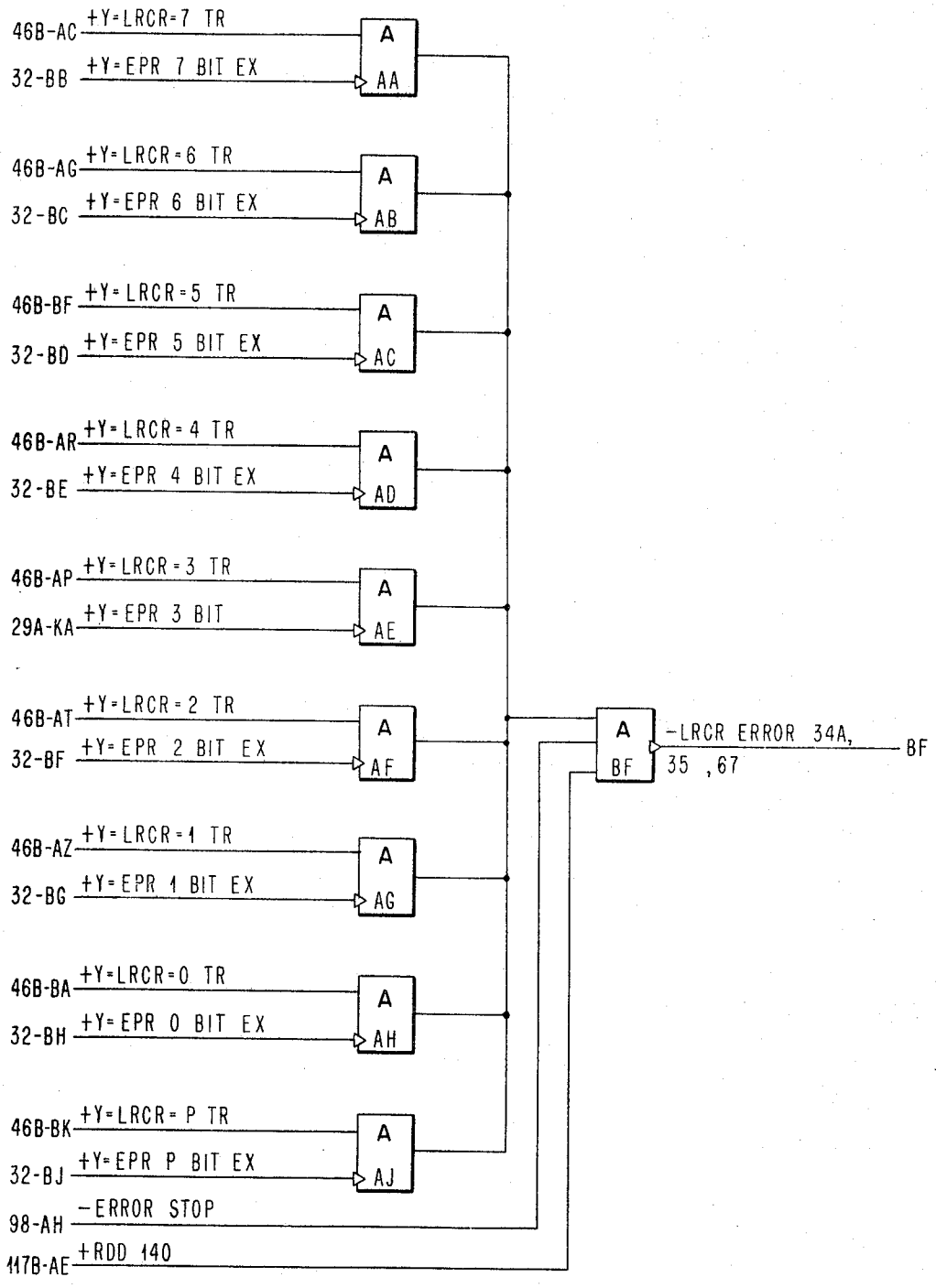

46B-AC
+Y=LRCR=7 TR
32-BB
+Y=EPR 7 BIT EX
A
AA
46B-AG
+Y=LRCR=6 TR
32-BC
+Y=EPR 6 BIT EX
A
AB
46B-BF
+Y=LRCR=5 TR
32-BD
+Y=EPR 5 BIT EX
A
AC
46B-AR
+Y=LRCR=4 TR
32-BE
+Y=EPR 4 BIT EX
A
AD
46B-AP
+Y=LRCR=3 TR
29A-KA
+Y=EPR 3 BIT
A
AE
46B-AT
+Y=LRCR=2 TR
32-BF
+Y=EPR 2 BIT EX
A
AF
46B-AZ
+Y=LRCR=1 TR
32-BG
+Y=EPR 1 BIT EX
A
AG
46B-BA
+Y=LRCR=0 TR
32-BH
+Y=EPR 0 BIT EX
A
AH
46B-BK
+Y=LRCR=P TR
32-BJ
+Y=EPR P BIT EX
A
AJ
98-AH
-ERROR STOP
117B-AE
+RDD 140
A
BF
-LRCR ERROR 34A, 35 ,67
BF

FIG.31

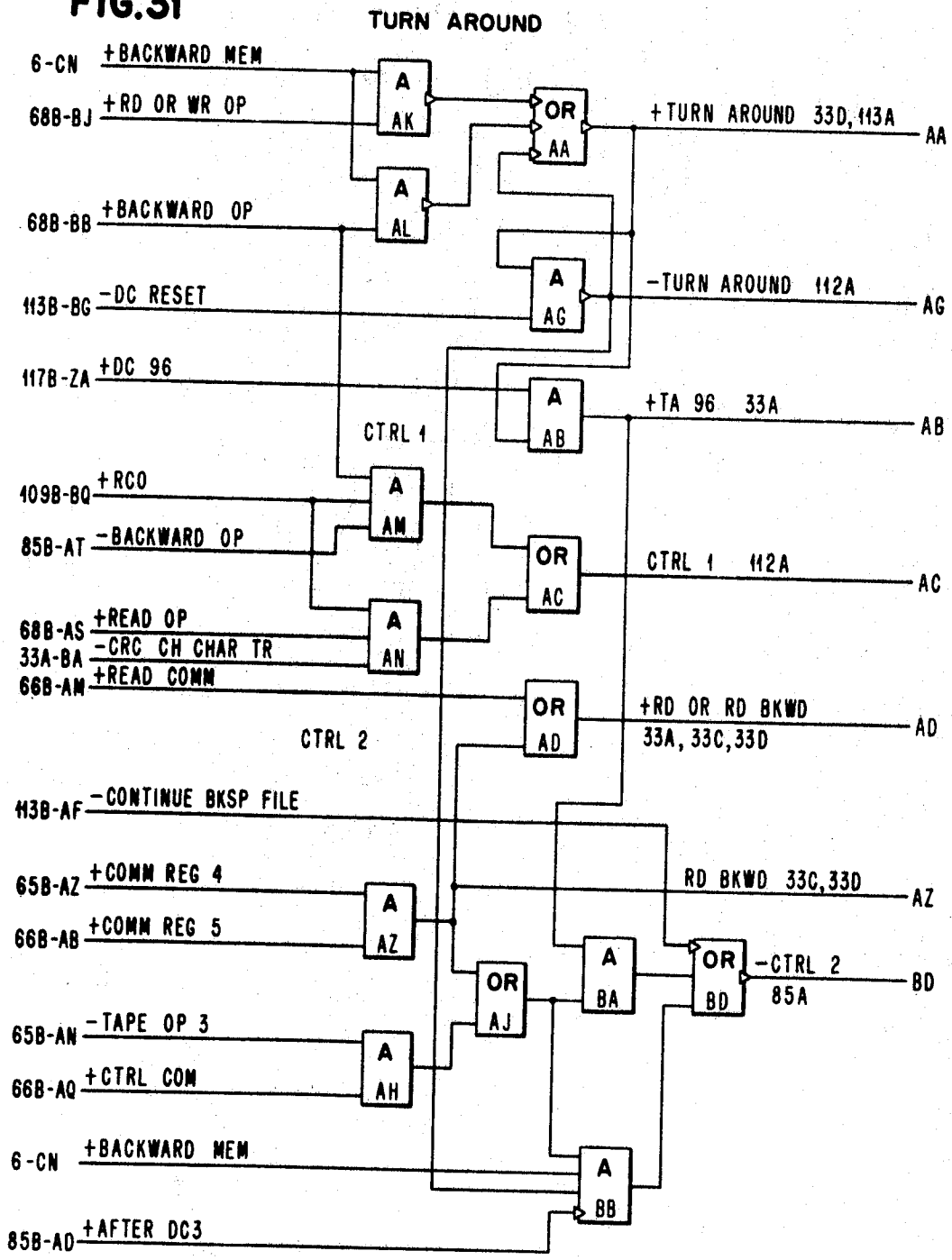

TURN AROUND
6-CN +BACKWARD MEM
68B-BJ +RD OR WR OP
A AK
OR AA
+TURN AROUND 33D, 113A
AA
A AL
68B-BB +BACKWARD OP
A AG
-TURN AROUND 112A
AG
113B-BG -DC RESET
117B-ZA +DC 96
A AB
+TA 96 33A
AB
CTRL 1
109B-BQ +RCO
A AM
85B-AT -BACKWARD OP
OR AC
CTRL 1 112A
AC
68B-AS +READ OP
33A-BA -CRC CH CHAR TR
A AN
66B-AM +READ COMM
OR AD
+RD OR RD BKWD 33A, 33C, 33D
AD
CTRL 2
113B-AF -CONTINUE BKSP FILE
65B-AZ +COMM REG 4
66B-AB +COMM REG 5
A AZ
RD BKWD 33C, 33D
AZ
OR AJ
A BA
OR BD
-CTRL 2 85A
BD
65B-AN -TAPE OP 3
66B-AQ +CTRL COM
A AH
6-CN +BACKWARD MEM
A BB
85B-AD +AFTER DC3

FIG.32

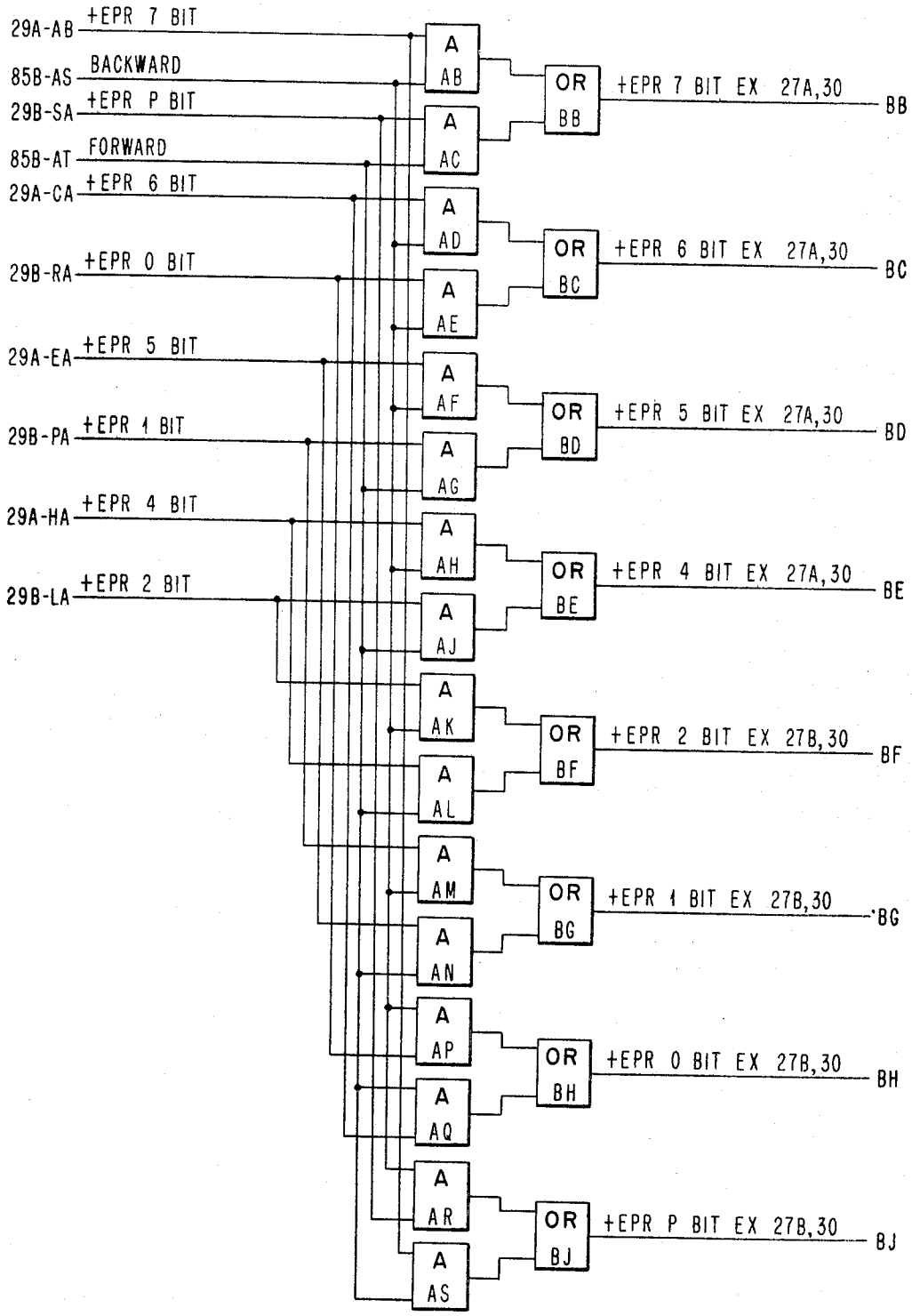

29A-AB +EPR 7 BIT
85B-AS BACKWARD
29B-SA +EPR P BIT
85B-AT FORWARD
29A-CA +EPR 6 BIT
29B-RA +EPR 0 BIT
29A-EA +EPR 5 BIT
29B-PA +EPR 1 BIT
29A-HA +EPR 4 BIT
29B-LA +EPR 2 BIT
A AB
A AC
A AD
A AE
A AF
A AG
A AH
A AJ
A AK
A AL
A AM
A AN
A AP
A AQ
A AR
A AS
OR BB
OR BC
OR BD
OR BE
OR BF
OR BG
OR BH
OR BJ
+EPR 7 BIT EX 27A,30 BB
+EPR 6 BIT EX 27A,30 BC
+EPR 5 BIT EX 27A,30 BD
+EPR 4 BIT EX 27A,30 BE
+EPR 2 BIT EX 27B,30 BF
+EPR 1 BIT EX 27B,30 BG
+EPR 0 BIT EX 27B,30 BH
+EPR P BIT EX 27B,30 BJ

FIG. 33A

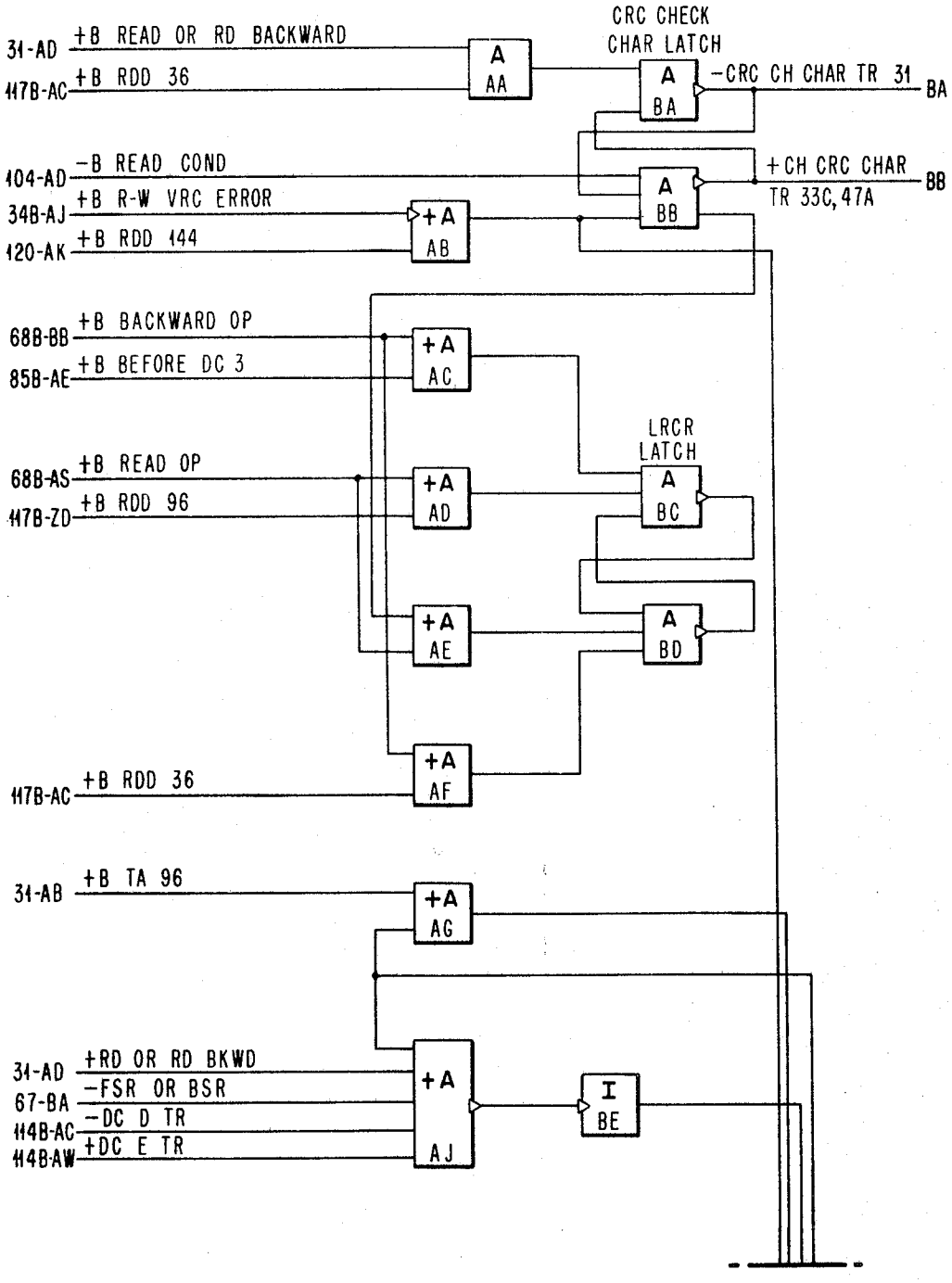

CRC CHECK CHAR LATCH
31-AD +B READ OR RD BACKWARD
117B-AC +B RDD 36
A AA
A BA
-CRC CH CHAR TR 31
BA
104-AD -B READ COND
34B-AJ +B R-W VRC ERROR
120-AK +B RDD 144
+A AB
A BB
+CH CRC CHAR TR 33C, 47A
BB
68B-BB +B BACKWARD OP
85B-AE +B BEFORE DC 3
+A AC
LRCR LATCH
68B-AS +B READ OP
117B-ZD +B RDD 96
+A AD
A BC
+A AE
A BD
117B-AC +B RDD 36
+A AF
31-AB +B TA 96
+A AG
31-AD +RD OR RD BKWD
67-BA -FSR OR BSR
114B-AC -DC D TR
114B-AW +DC E TR
+A AJ
I BE

FIG.33B

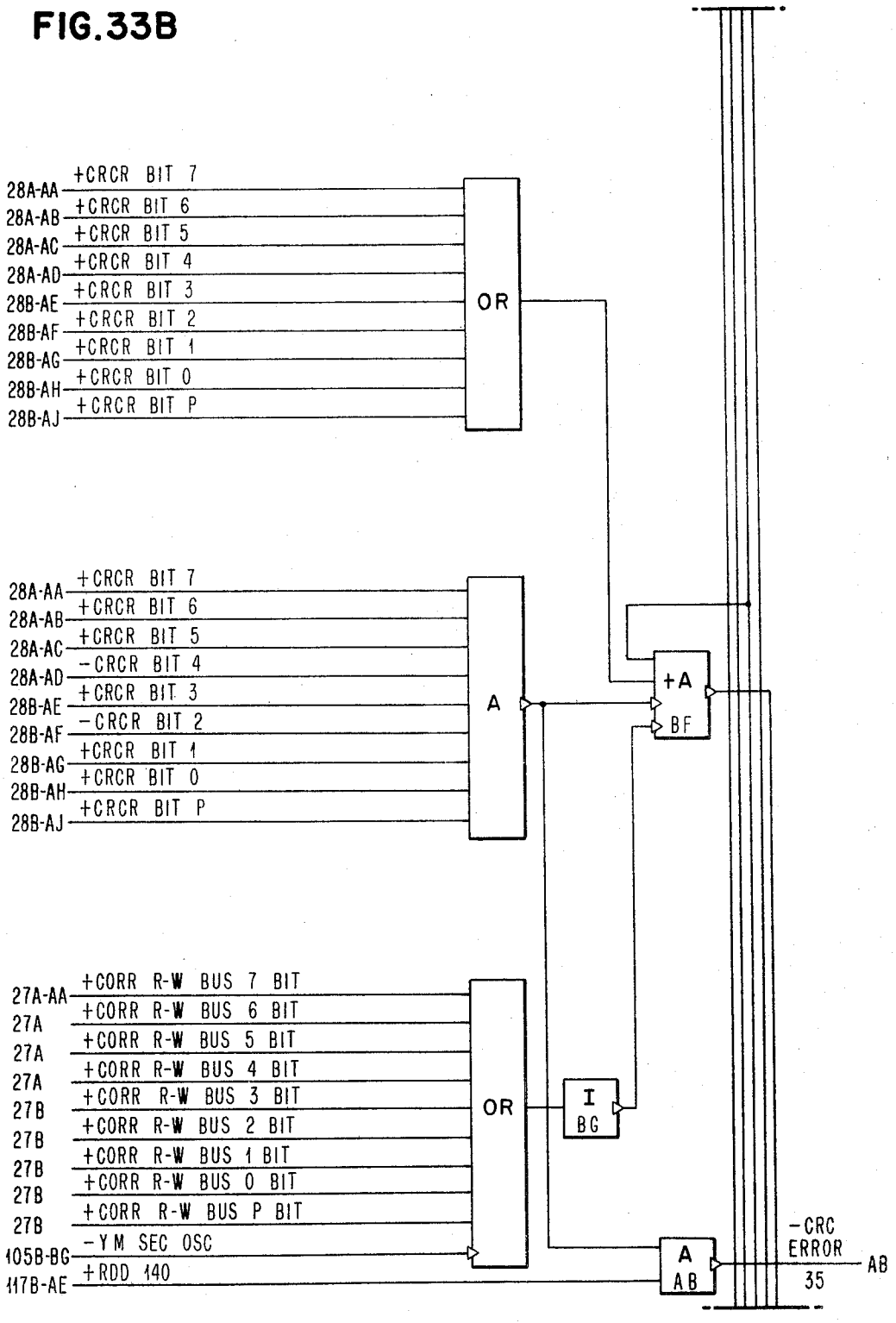

28A-AA +CRCR BIT 7
28A-AB +CRCR BIT 6
28A-AC +CRCR BIT 5
28A-AD +CRCR BIT 4
28B-AE +CRCR BIT 3
28B-AF +CRCR BIT 2
28B-AG +CRCR BIT 1
28B-AH +CRCR BIT 0
28B-AJ +CRCR BIT P
OR
28A-AA +CRCR BIT 7
28A-AB +CRCR BIT 6
28A-AC +CRCR BIT 5
28A-AD -CRCR BIT 4
28B-AE +CRCR BIT 3
28B-AF -CRCR BIT 2
28B-AG +CRCR BIT 1
28B-AH +CRCR BIT 0
28B-AJ +CRCR BIT P
A
+A
BF
27A-AA +CORR R-W BUS 7 BIT
27A +CORR R-W BUS 6 BIT
27A +CORR R-W BUS 5 BIT
27A +CORR R-W BUS 4 BIT
27B +CORR R-W BUS 3 BIT
27B +CORR R-W BUS 2 BIT
27B +CORR R-W BUS 1 BIT
27B +CORR R-W BUS 0 BIT
27B +CORR R-W BUS P BIT
105B-BG -YM SEC OSC
117B-AE +RDD 140
OR
I
BG
A
AB
-CRC
ERROR
35
AB

FIG.33C

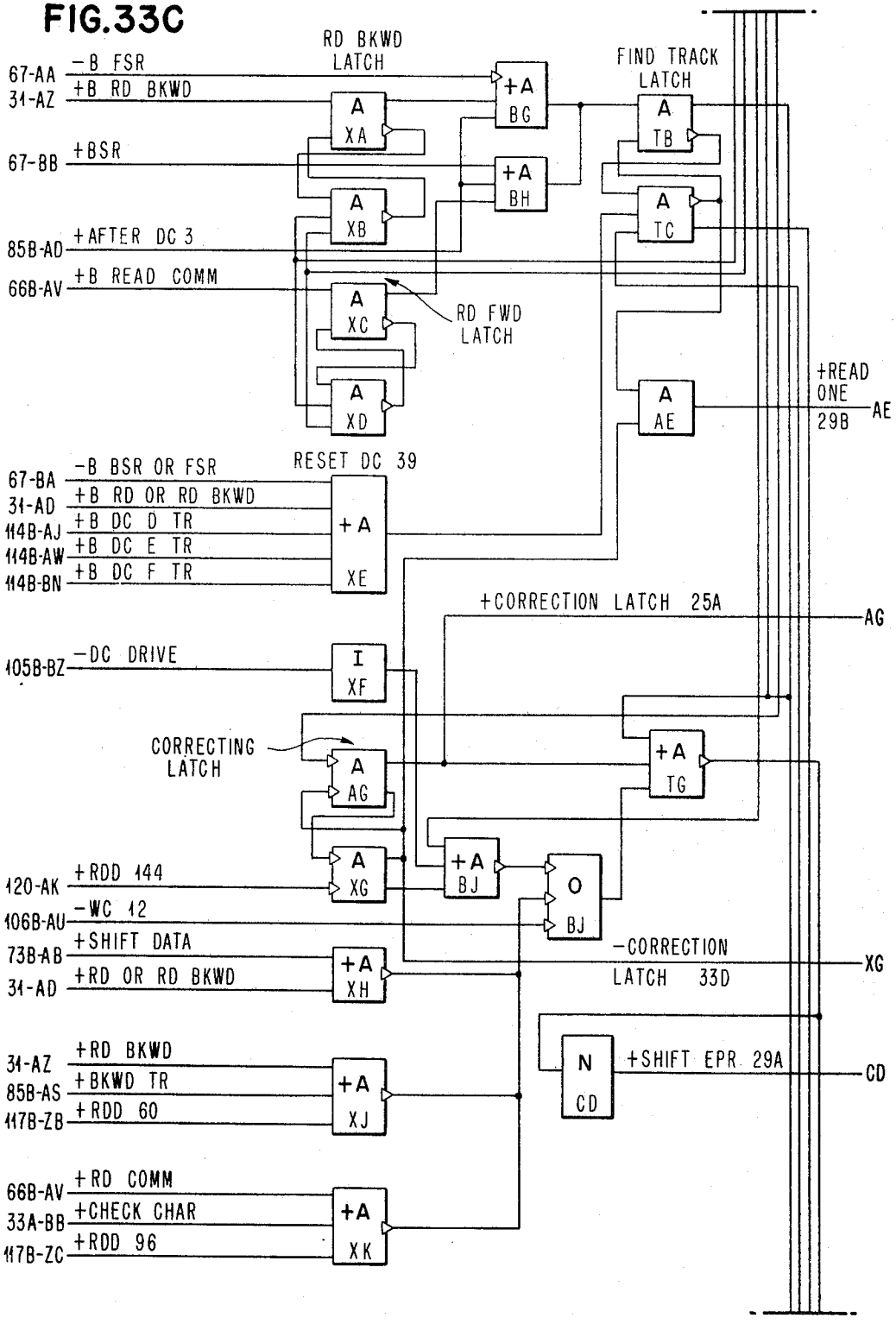

RD BKWD LATCH
FIND TRACK LATCH
67-AA −B FSR
31-AZ +B RD BKWD
67-BB +BSR
85B-AD +AFTER DC 3
66B-AV +B READ COMM
RD FWD LATCH
+READ ONE 29B
AE
RESET DC 39
67-BA −B BSR OR FSR
31-AD +B RD OR RD BKWD
114B-AJ +B DC D TR
114B-AW +B DC E TR
114B-BN +B DC F TR
+CORRECTION LATCH 25A
AG
105B-BZ −DC DRIVE
CORRECTING LATCH
120-AK +RDD 144
106B-AU −WC 12
73B-AB +SHIFT DATA
31-AD +RD OR RD BKWD
−CORRECTION LATCH 33D
XG
31-AZ +RD BKWD
85B-AS +BKWD TR
117B-ZB +RDD 60
+SHIFT EPR 29A
CD
66B-AV +RD COMM
33A-BB +CHECK CHAR
117B-ZC +RDD 96

FIG.33D

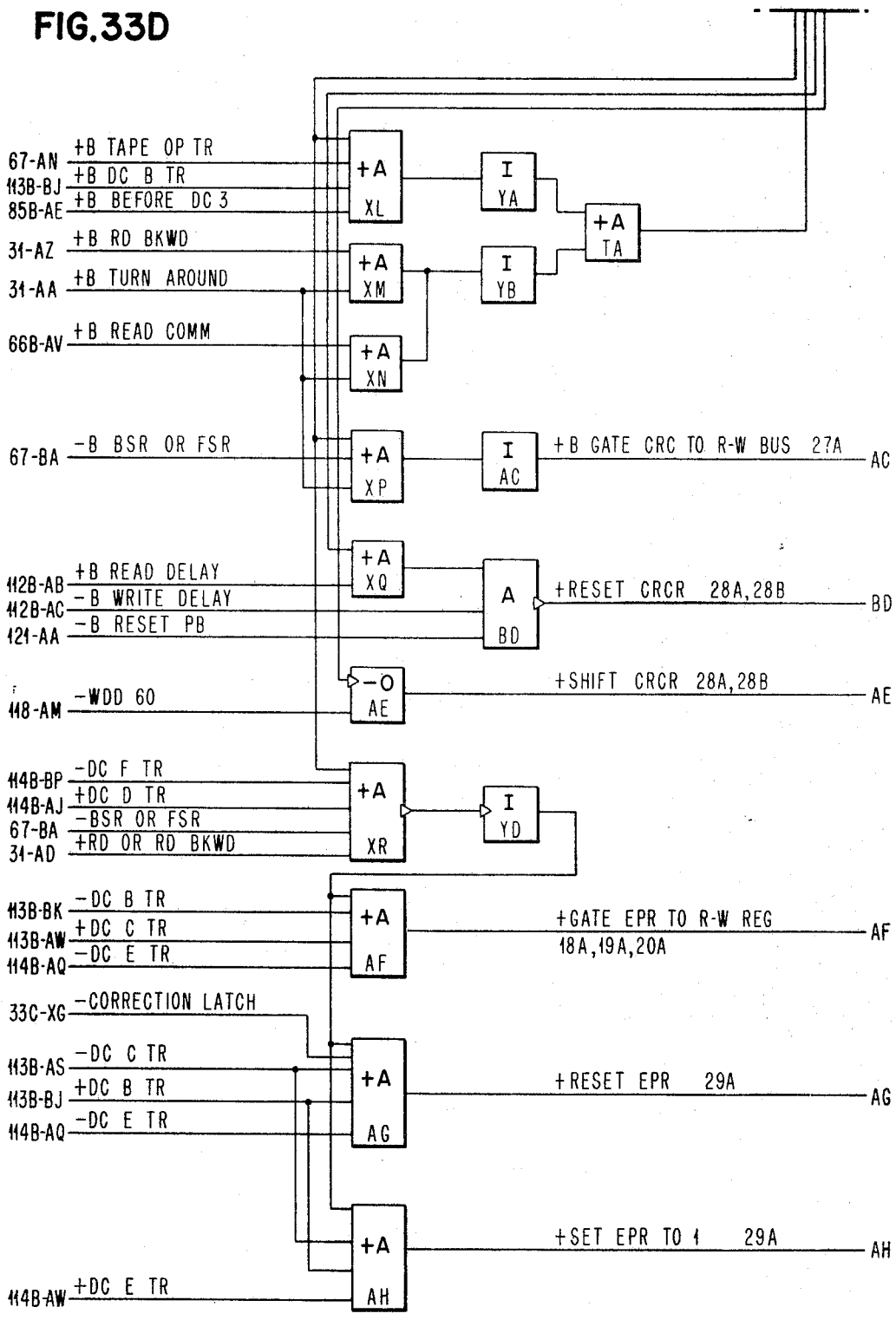

67-AN +B TAPE OP TR
113B-BJ +B DC B TR
85B-AE +B BEFORE DC 3
31-AZ +B RD BKWD
31-AA +B TURN AROUND
66B-AV +B READ COMM
+A XL
+A XM
+A XN
I YA
I YB
+A TA
67-BA -B BSR OR FSR
+A XP
I AC
+B GATE CRC TO R-W BUS 27A
AC
112B-AB +B READ DELAY
112B-AC -B WRITE DELAY
121-AA -B RESET PB
+A XQ
A BD
+RESET CRCR 28A,28B
BD
118-AM -WDD 60
-O AE
+SHIFT CRCR 28A,28B
AE
114B-BP -DC F TR
114B-AJ +DC D TR
67-BA -BSR OR FSR
31-AD +RD OR RD BKWD
+A XR
I YD
113B-BK -DC B TR
113B-AW +DC C TR
114B-AQ -DC E TR
+A AF
+GATE EPR TO R-W REG 18A,19A,20A
AF
33C-XG -CORRECTION LATCH
113B-AS -DC C TR
113B-BJ +DC B TR
114B-AQ -DC E TR
+A AG
+RESET EPR 29A
AG
+A AH
+SET EPR TO 1 29A
AH
114B-AW +DC E TR

FIG.34A

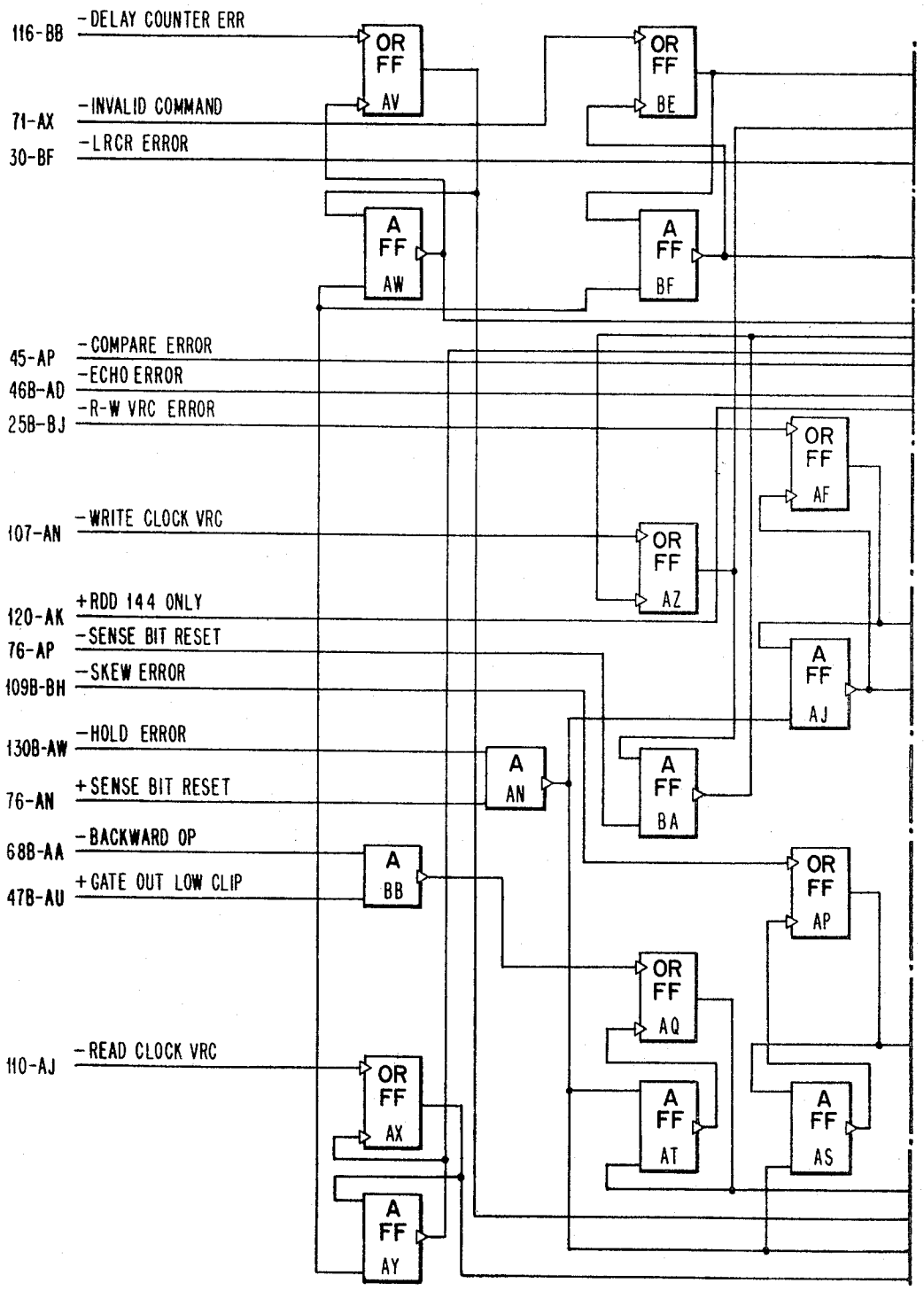

116-BB -DELAY COUNTER ERR
71-AX -INVALID COMMAND
30-BF -LRCR ERROR
45-AP -COMPARE ERROR
46B-AD -ECHO ERROR
25B-BJ -R-W VRC ERROR
107-AN -WRITE CLOCK VRC
120-AK +RDD 144 ONLY
76-AP -SENSE BIT RESET
109B-BH -SKEW ERROR
130B-AW -HOLD ERROR
76-AN +SENSE BIT RESET
68B-AA -BACKWARD OP
47B-AU +GATE OUT LOW CLIP
110-AJ -READ CLOCK VRC
OR FF AV
OR FF BE
A FF AW
A FF BF
OR FF AF
OR FF AZ
A FF AJ
A AN
A FF BA
A BB
OR FF AP
OR FF AQ
A FF AT
A FF AS
OR FF AX
A FF AY

FIG. 34B

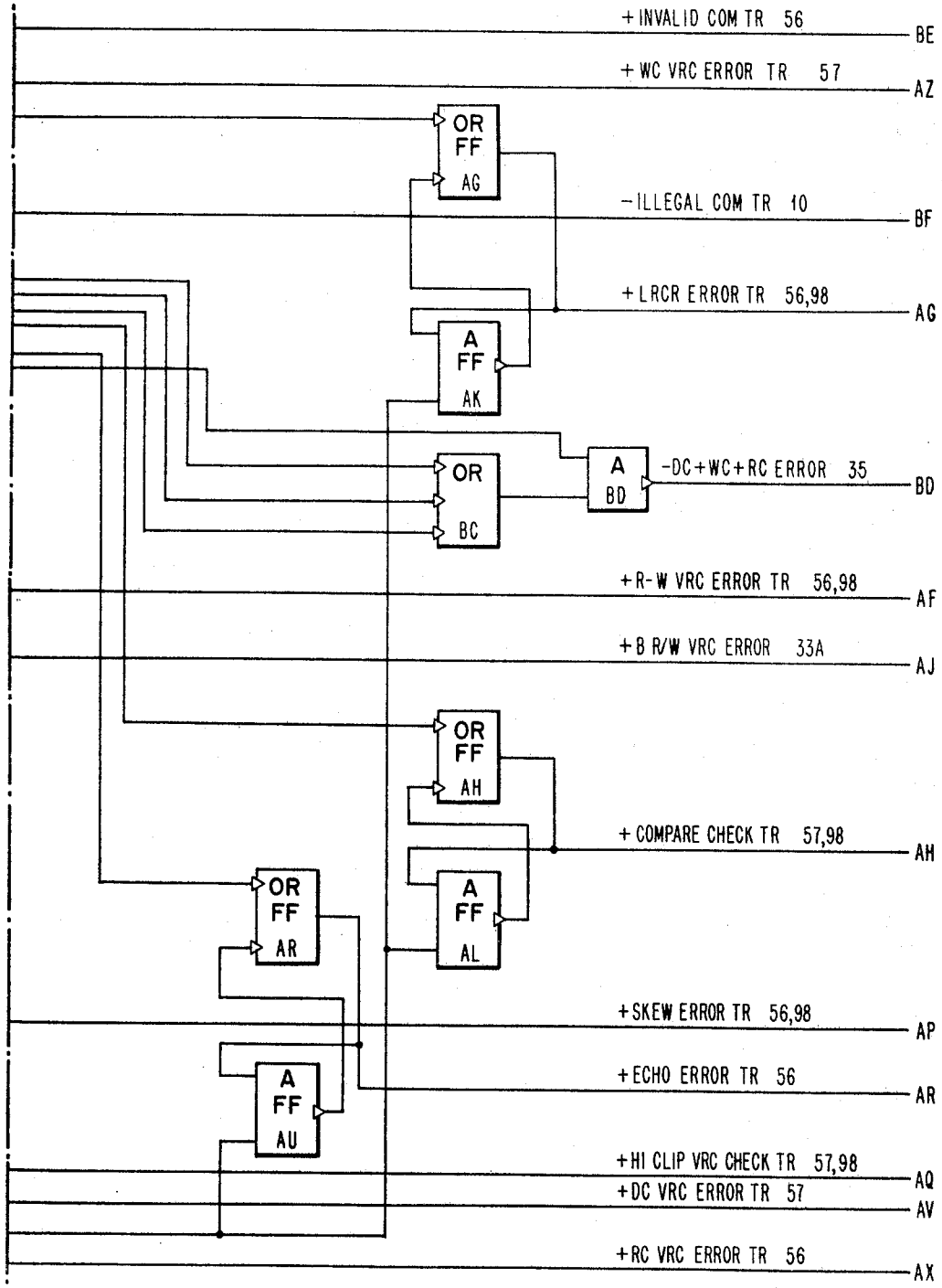

+INVALID COM TR 56
BE
+WC VRC ERROR TR 57
AZ
OR FF AG
-ILLEGAL COM TR 10
BF
+LRCR ERROR TR 56,98
AG
A FF AK
OR BC
A BD
-DC+WC+RC ERROR 35
BD
+R-W VRC ERROR TR 56,98
AF
+B R/W VRC ERROR 33A
AJ
OR FF AH
+COMPARE CHECK TR 57,98
AH
OR FF AR
A FF AL
+SKEW ERROR TR 56,98
AP
A FF AU
+ECHO ERROR TR 56
AR
+HI CLIP VRC CHECK TR 57,98
AQ
+DC VRC ERROR TR 57
AV
+RC VRC ERROR TR 56
AX

FIG. 35

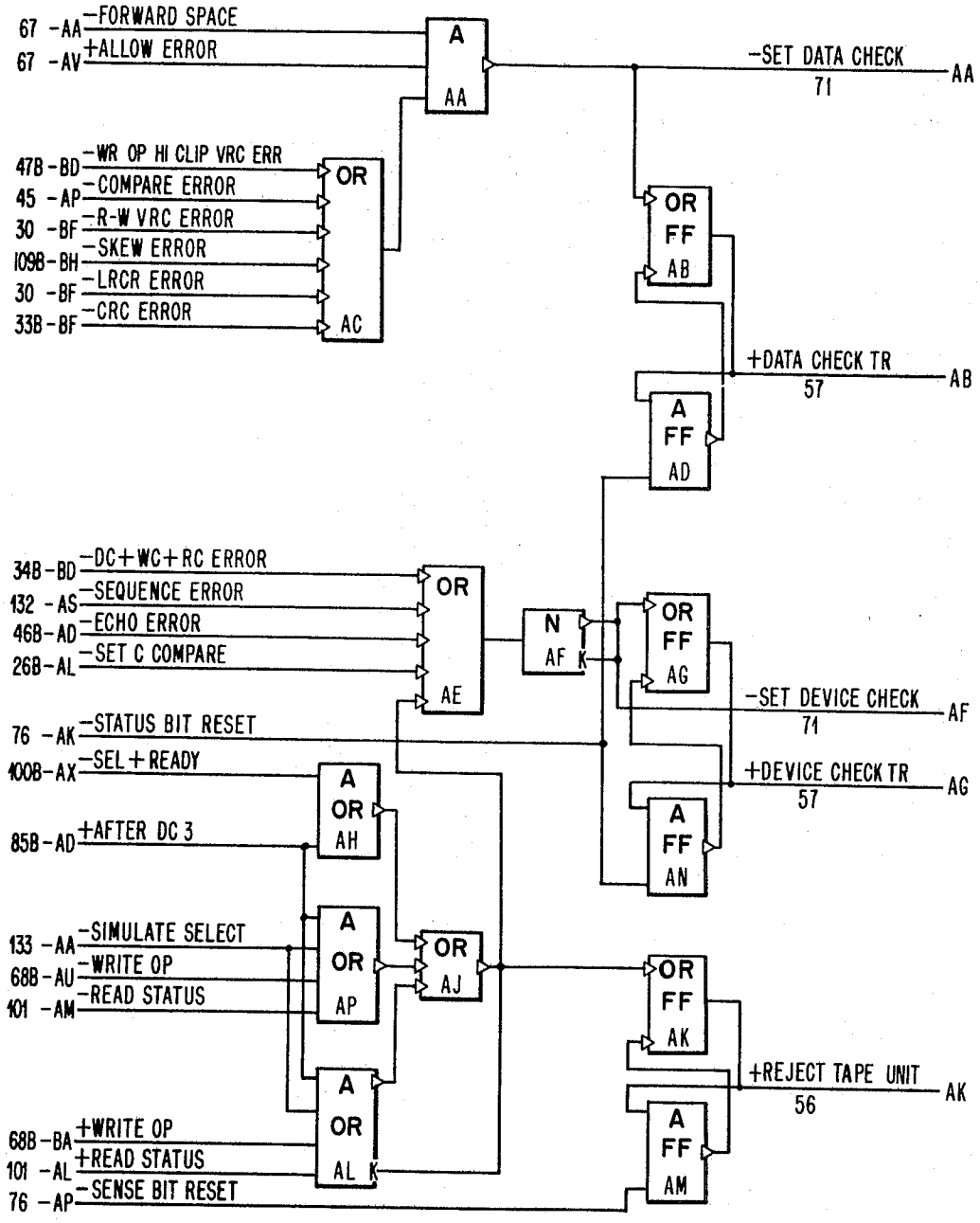

67 -AA
-FORWARD SPACE
67 -AV
+ALLOW ERROR
A
AA
-SET DATA CHECK
71
AA
47B-BD
-WR OP HI CLIP VRC ERR
45 -AP
-COMPARE ERROR
30 -BF
-R-W VRC ERROR
109B-BH
-SKEW ERROR
30 -BF
-LRCR ERROR
33B-BF
-CRC ERROR
OR
AC
OR
FF
AB
+DATA CHECK TR
57
AB
A
FF
AD
34B-BD
-DC+WC+RC ERROR
132 -AS
-SEQUENCE ERROR
46B-AD
-ECHO ERROR
26B-AL
-SET C COMPARE
OR
AE
N
AF K
OR
FF
AG
-SET DEVICE CHECK
71
AF
76 -AK
-STATUS BIT RESET
100B-AX
-SEL+READY
+DEVICE CHECK TR
57
AG
A
FF
AN
A
OR
AH
85B-AD
+AFTER DC 3
133 -AA
-SIMULATE SELECT
68B-AU
-WRITE OP
101 -AM
-READ STATUS
A
OR
AP
OR
AJ
OR
FF
AK
+REJECT TAPE UNIT
56
AK
A
OR
AL K
68B-BA
+WRITE OP
101 -AL
+READ STATUS
76 -AP
-SENSE BIT RESET
A
FF
AM

FIG. 36

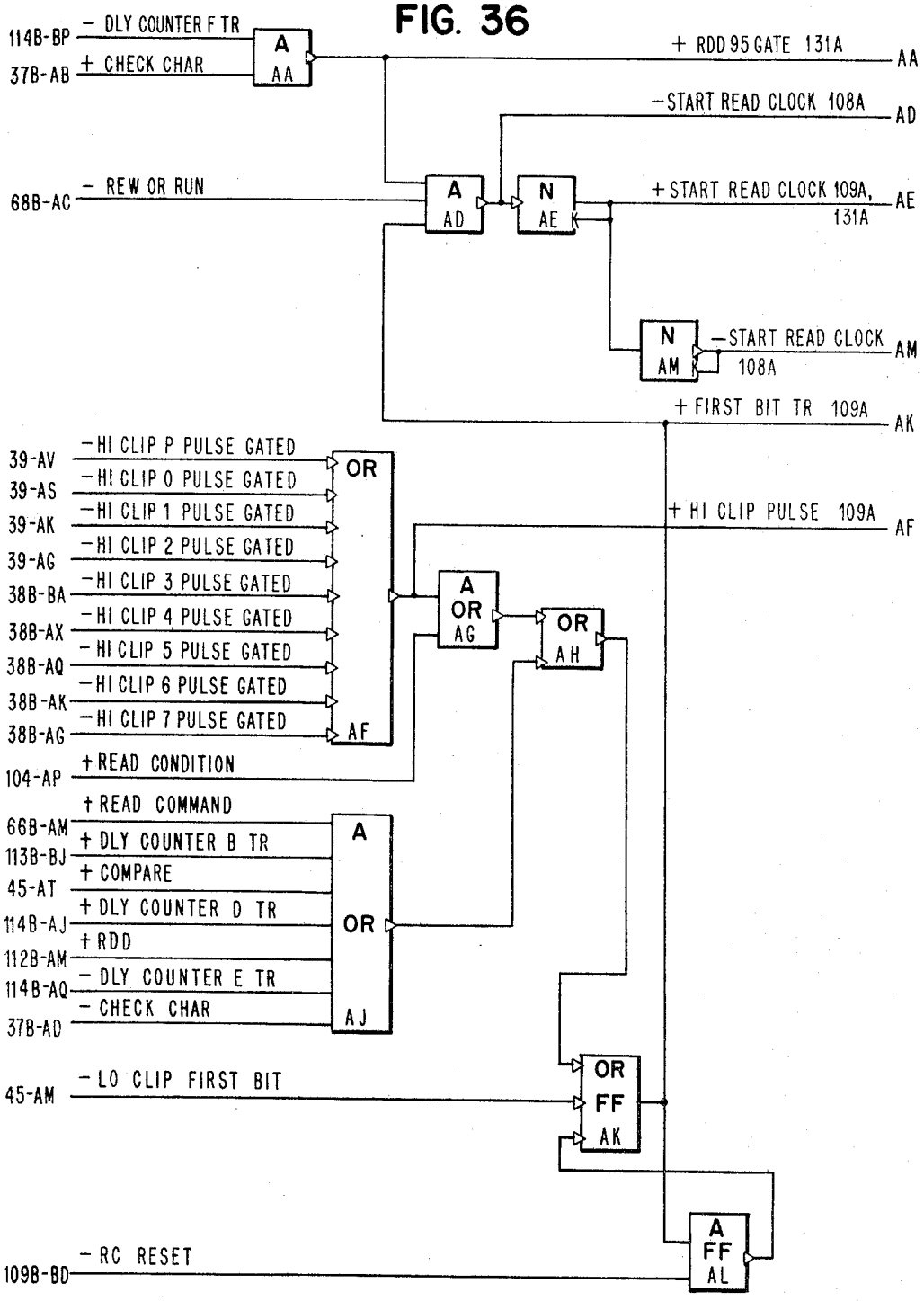

114B-BP
37B-AB
- DLY COUNTER F TR
+ CHECK CHAR
A AA
+ RDD 95 GATE 131A
AA
-START READ CLOCK 108A
AD
68B-AC
- REW OR RUN
A AD
N AE
+START READ CLOCK 109A, 131A
AE
N AM
-START READ CLOCK 108A
AM
+ FIRST BIT TR 109A
AK
39-AV
39-AS
39-AK
39-AG
38B-BA
38B-AX
38B-AQ
38B-AK
38B-AG
-HI CLIP P PULSE GATED
-HI CLIP O PULSE GATED
-HI CLIP 1 PULSE GATED
-HI CLIP 2 PULSE GATED
-HI CLIP 3 PULSE GATED
-HI CLIP 4 PULSE GATED
-HI CLIP 5 PULSE GATED
-HI CLIP 6 PULSE GATED
-HI CLIP 7 PULSE GATED
OR
AF
+ HI CLIP PULSE 109A
AF
104-AP
+READ CONDITION
A OR AG
OR AH
66B-AM
113B-BJ
45-AT
114B-AJ
112B-AM
114B-AQ
37B-AD
+READ COMMAND
+ DLY COUNTER B TR
+ COMPARE
+ DLY COUNTER D TR
+RDD
- DLY COUNTER E TR
- CHECK CHAR
A OR AJ
45-AM
- LO CLIP FIRST BIT
OR FF AK
109B-BD
- RC RESET
A FF AL

FIG. 37A

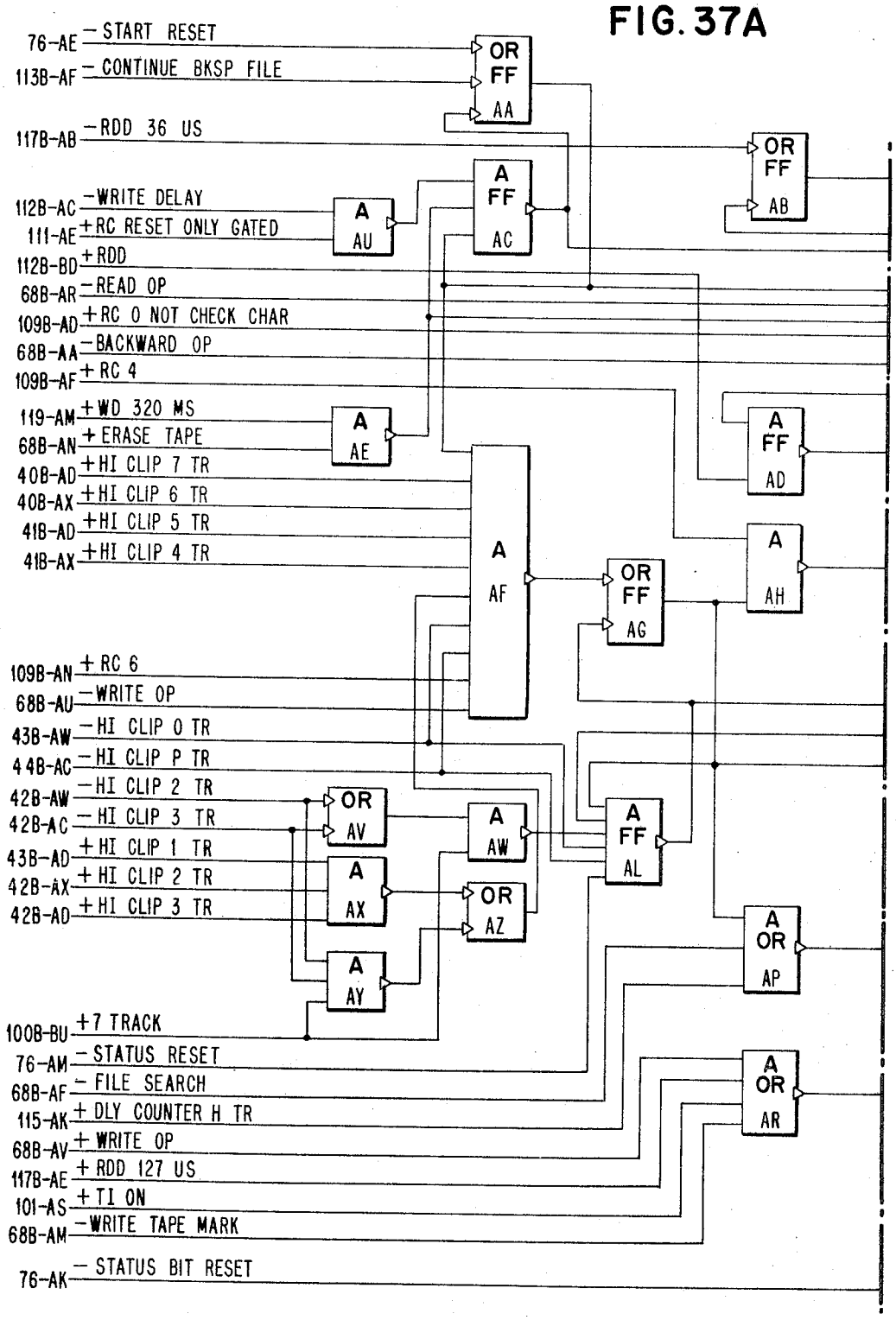

76-AE -START RESET
113B-AF -CONTINUE BKSP FILE
OR FF AA
117B-AB -RDD 36 US
OR FF AB
112B-AC -WRITE DELAY
111-AE +RC RESET ONLY GATED
A AU
A FF AC
112B-BD +RDD
68B-AR -READ OP
109B-AD +RC 0 NOT CHECK CHAR
68B-AA -BACKWARD OP
109B-AF +RC 4
119-AM +WD 320 MS
68B-AN +ERASE TAPE
A AE
A FF AD
40B-AD +HI CLIP 7 TR
40B-AX +HI CLIP 6 TR
41B-AD +HI CLIP 5 TR
41B-AX +HI CLIP 4 TR
A AF
OR FF AG
A AH
109B-AN +RC 6
68B-AU -WRITE OP
43B-AW -HI CLIP 0 TR
44B-AC -HI CLIP P TR
42B-AW -HI CLIP 2 TR
42B-AC -HI CLIP 3 TR
43B-AD +HI CLIP 1 TR
42B-AX +HI CLIP 2 TR
42B-AD +HI CLIP 3 TR
OR AV
A AX
A AY
A AW
OR AZ
A FF AL
A OR AP
100B-BU +7 TRACK
76-AM -STATUS RESET
68B-AF -FILE SEARCH
115-AK +DLY COUNTER H TR
68B-AV +WRITE OP
117B-AE +RDD 127 US
101-AS +TI ON
68B-AM -WRITE TAPE MARK
A OR AR
76-AK -STATUS BIT RESET

FIG. 37B

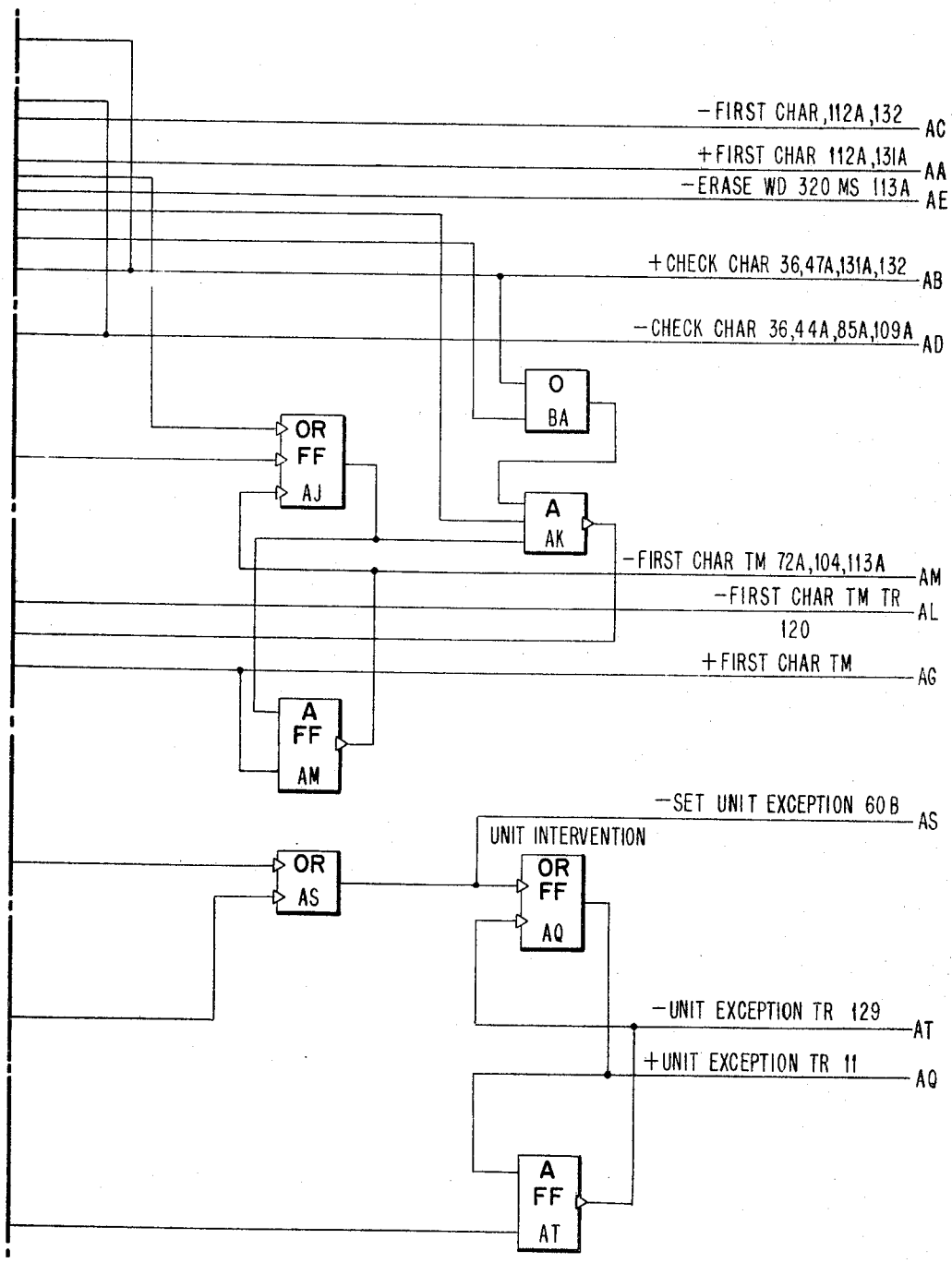

-FIRST CHAR,112A,132
AC
+FIRST CHAR 112A,131A
AA
-ERASE WD 320 MS 113A
AE
+CHECK CHAR 36,47A,131A,132
AB
-CHECK CHAR 36,44A,85A,109A
AD
O
BA
OR
FF
AJ
A
AK
-FIRST CHAR TM 72A,104,113A
AM
-FIRST CHAR TM TR
120
AL
+FIRST CHAR TM
AG
A
FF
AM
-SET UNIT EXCEPTION 60B
AS
UNIT INTERVENTION
OR
AS
OR
FF
AQ
-UNIT EXCEPTION TR 129
AT
+UNIT EXCEPTION TR 11
AQ
A
FF
AT

FIG.38A

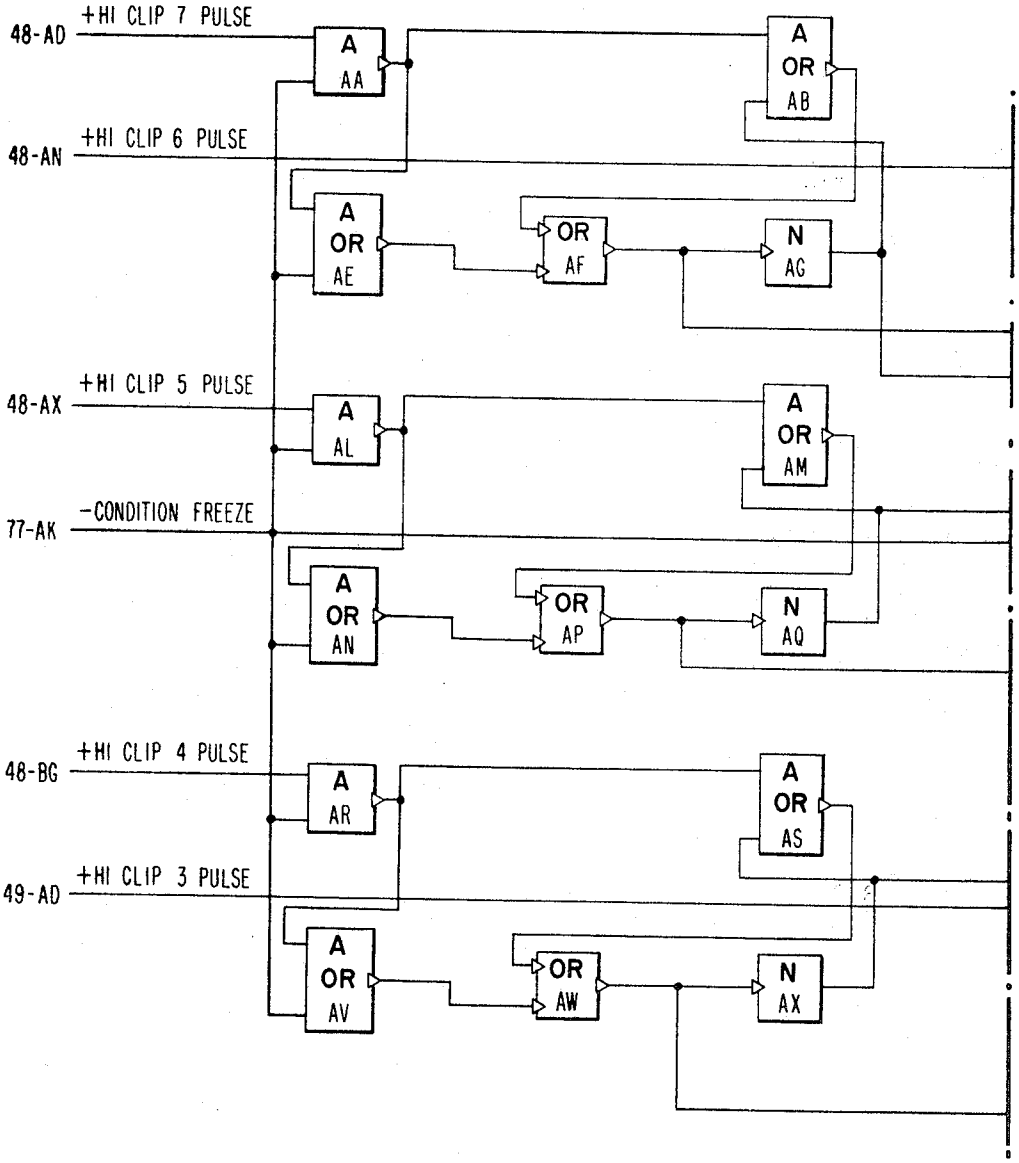

48-AD
+HI CLIP 7 PULSE
48-AN
+HI CLIP 6 PULSE
48-AX
+HI CLIP 5 PULSE
77-AK
-CONDITION FREEZE
48-BG
+HI CLIP 4 PULSE
49-AD
+HI CLIP 3 PULSE
A AA
A OR AB
A OR AE
OR AF
N AG
A AL
A OR AM
A OR AN
OR AP
N AQ
A AR
A OR AS
A OR AV
OR AW
N AX

FIG.38B

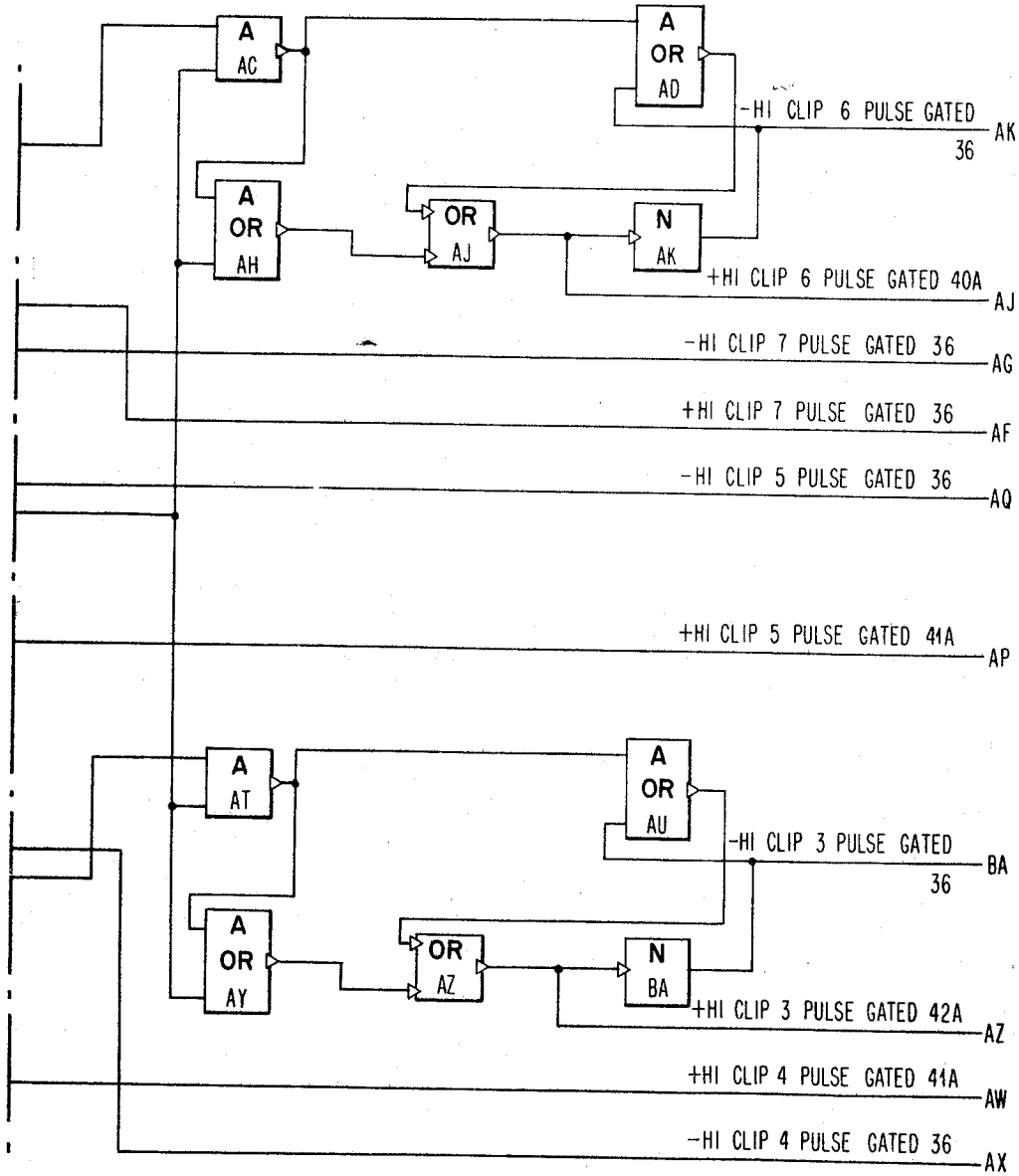

A
AC
A
OR
AD
-HI CLIP 6 PULSE GATED
36
AK
A
OR
AH
OR
AJ
N
AK
+HI CLIP 6 PULSE GATED 40A
AJ
-HI CLIP 7 PULSE GATED 36
AG
+HI CLIP 7 PULSE GATED 36
AF
-HI CLIP 5 PULSE GATED 36
AQ
+HI CLIP 5 PULSE GATED 41A
AP
A
AT
A
OR
AU
-HI CLIP 3 PULSE GATED
36
BA
A
OR
AY
OR
AZ
N
BA
+HI CLIP 3 PULSE GATED 42A
AZ
+HI CLIP 4 PULSE GATED 41A
AW
-HI CLIP 4 PULSE GATED 36
AX

FIG. 39

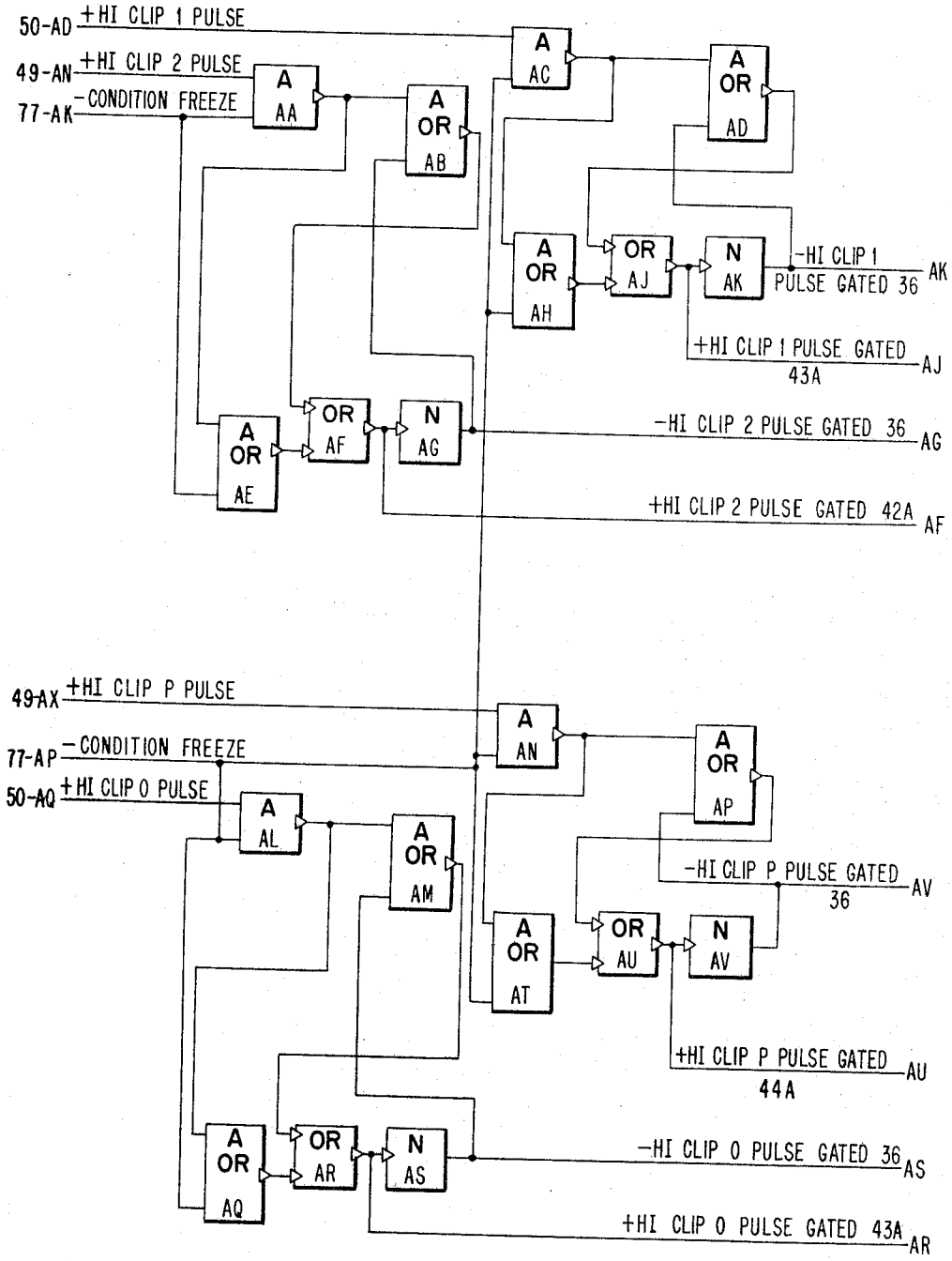

50-AD +HI CLIP 1 PULSE
49-AN +HI CLIP 2 PULSE
77-AK -CONDITION FREEZE
A AA
A OR AB
A AC
A OR AD
A OR AH
OR AJ
N AK
-HI CLIP 1 PULSE GATED 36 AK
+HI CLIP 1 PULSE GATED 43A AJ
A OR AE
OR AF
N AG
-HI CLIP 2 PULSE GATED 36 AG
+HI CLIP 2 PULSE GATED 42A AF
49-AX +HI CLIP P PULSE
77-AP -CONDITION FREEZE
50-AQ +HI CLIP 0 PULSE
A AL
A OR AM
A AN
A OR AP
A OR AT
OR AU
N AV
-HI CLIP P PULSE GATED 36 AV
+HI CLIP P PULSE GATED 44A AU
A OR AQ
OR AR
N AS
-HI CLIP 0 PULSE GATED 36 AS
+HI CLIP 0 PULSE GATED 43A AR

FIG. 40A

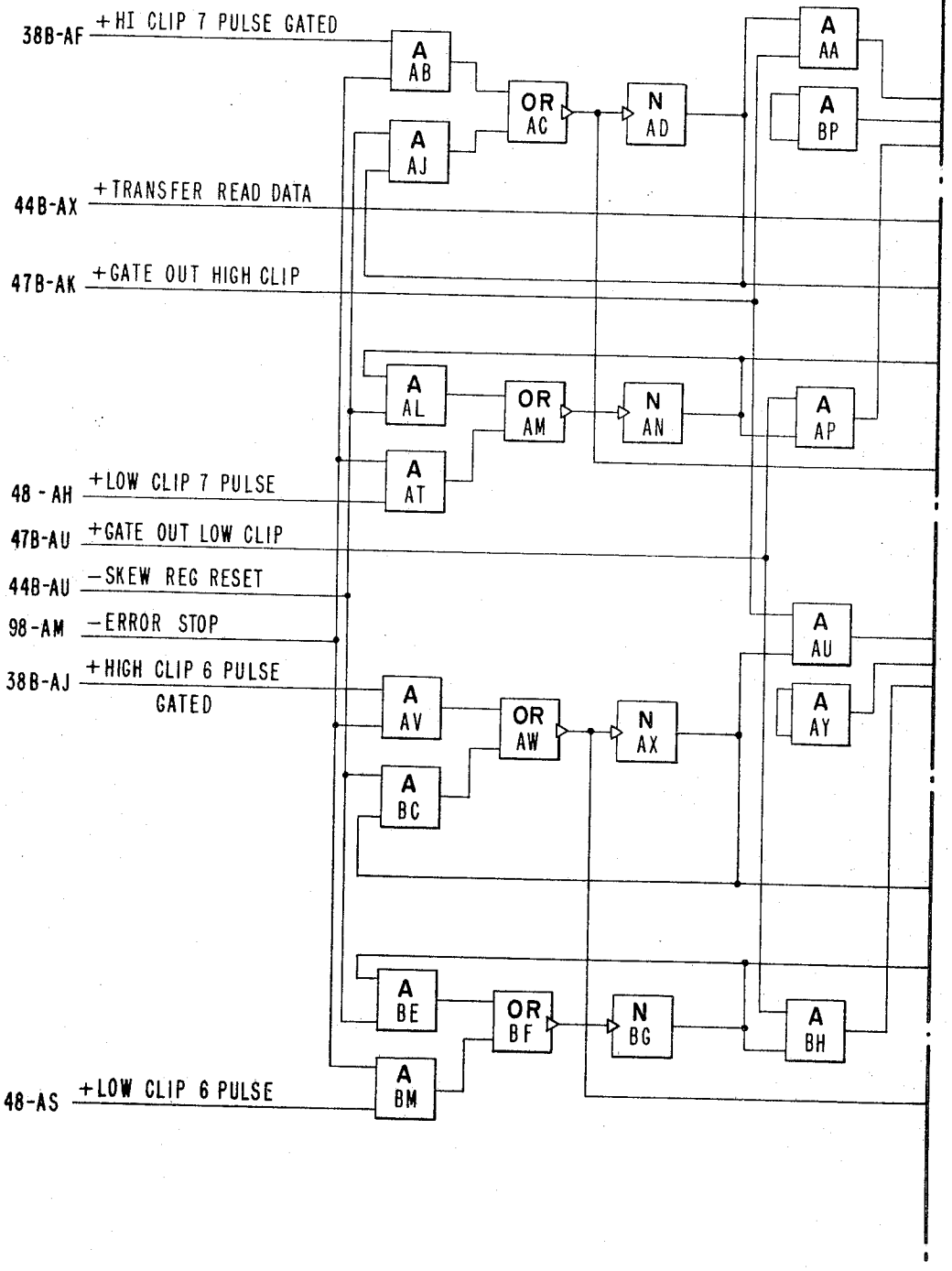

38B-AF +HI CLIP 7 PULSE GATED
44B-AX +TRANSFER READ DATA
47B-AK +GATE OUT HIGH CLIP
48 - AH +LOW CLIP 7 PULSE
47B-AU +GATE OUT LOW CLIP
44B-AU -SKEW REG RESET
98-AM -ERROR STOP
38B-AJ +HIGH CLIP 6 PULSE GATED
48-AS +LOW CLIP 6 PULSE
A AB
A AJ
OR AC
N AD
A AA
A BP
A AL
A AT
OR AM
N AN
A AP
A AV
A BC
OR AW
N AX
A AU
A AY
A BE
A BM
OR BF
N BG
A BH

FIG. 40B

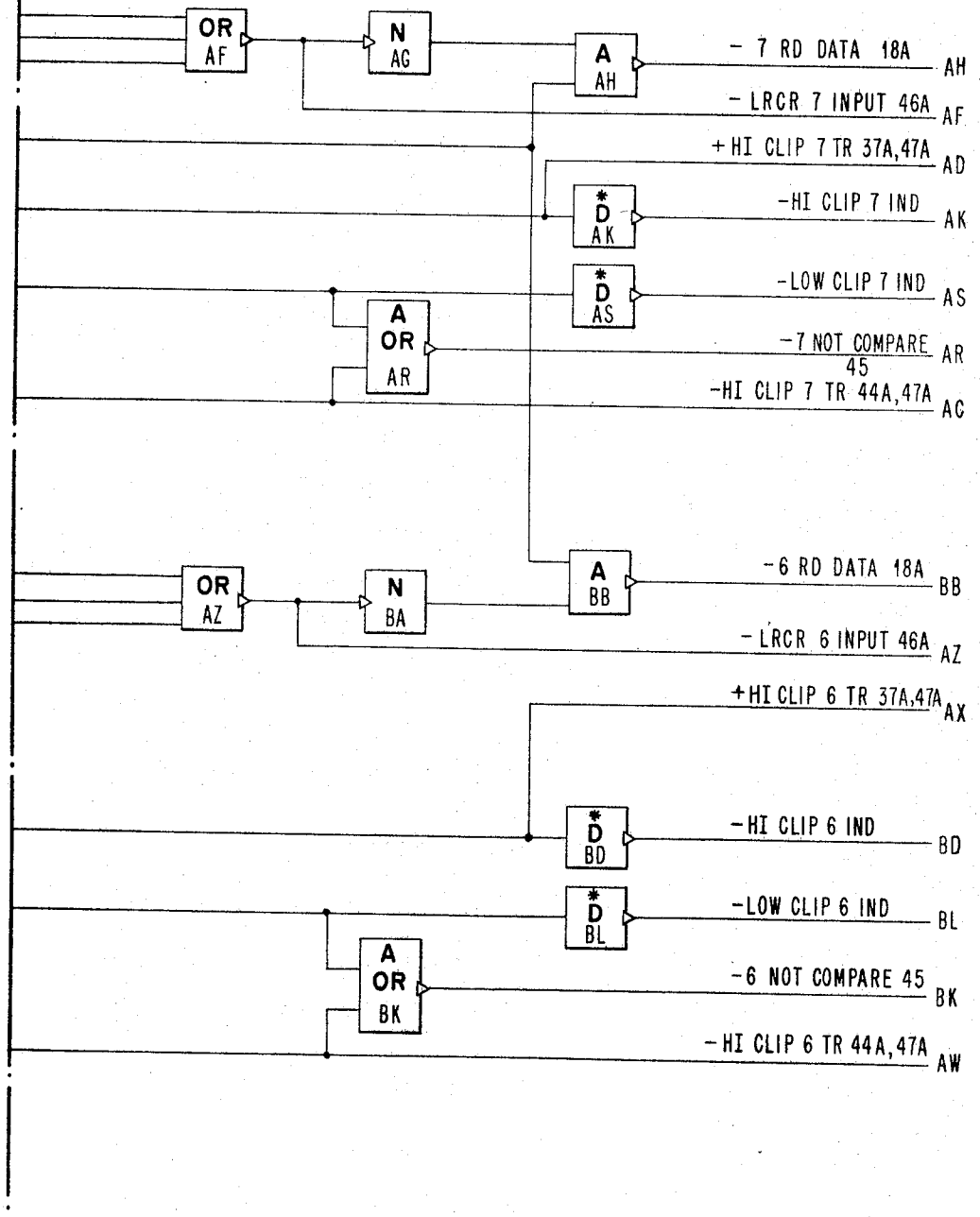

OR AF
N AG
A AH
- 7 RD DATA 18A
AH
- LRCR 7 INPUT 46A
AF
+HI CLIP 7 TR 37A,47A
AD
*D AK
-HI CLIP 7 IND
AK
*D AS
-LOW CLIP 7 IND
AS
A OR AR
-7 NOT COMPARE 45
AR
-HI CLIP 7 TR 44A,47A
AC
OR AZ
N BA
A BB
-6 RD DATA 18A
BB
- LRCR 6 INPUT 46A
AZ
+HI CLIP 6 TR 37A,47A
AX
*D BD
-HI CLIP 6 IND
BD
*D BL
-LOW CLIP 6 IND
BL
A OR BK
-6 NOT COMPARE 45
BK
-HI CLIP 6 TR 44A,47A
AW

FIG. 41A

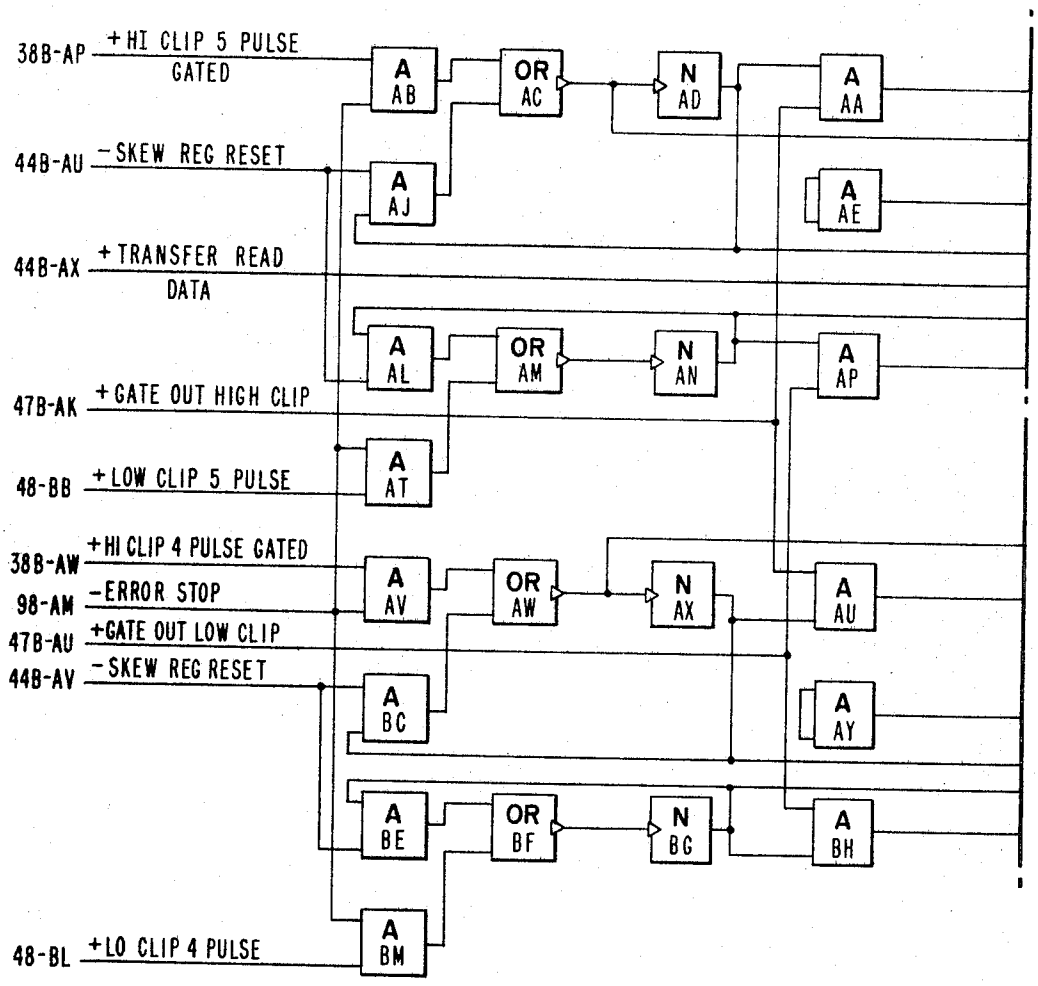

38B-AP
+HI CLIP 5 PULSE GATED
44B-AU
-SKEW REG RESET
44B-AX
+TRANSFER READ DATA
47B-AK
+GATE OUT HIGH CLIP
48-BB
+LOW CLIP 5 PULSE
38B-AW
+HI CLIP 4 PULSE GATED
98-AM
-ERROR STOP
47B-AU
+GATE OUT LOW CLIP
44B-AV
-SKEW REG RESET
48-BL
+LO CLIP 4 PULSE
A AB
OR AC
N AD
A AA
A AJ
A AE
A AL
OR AM
N AN
A AP
A AT
A AV
OR AW
N AX
A AU
A BC
A AY
A BE
OR BF
N BG
A BH
A BM

FIG. 41B

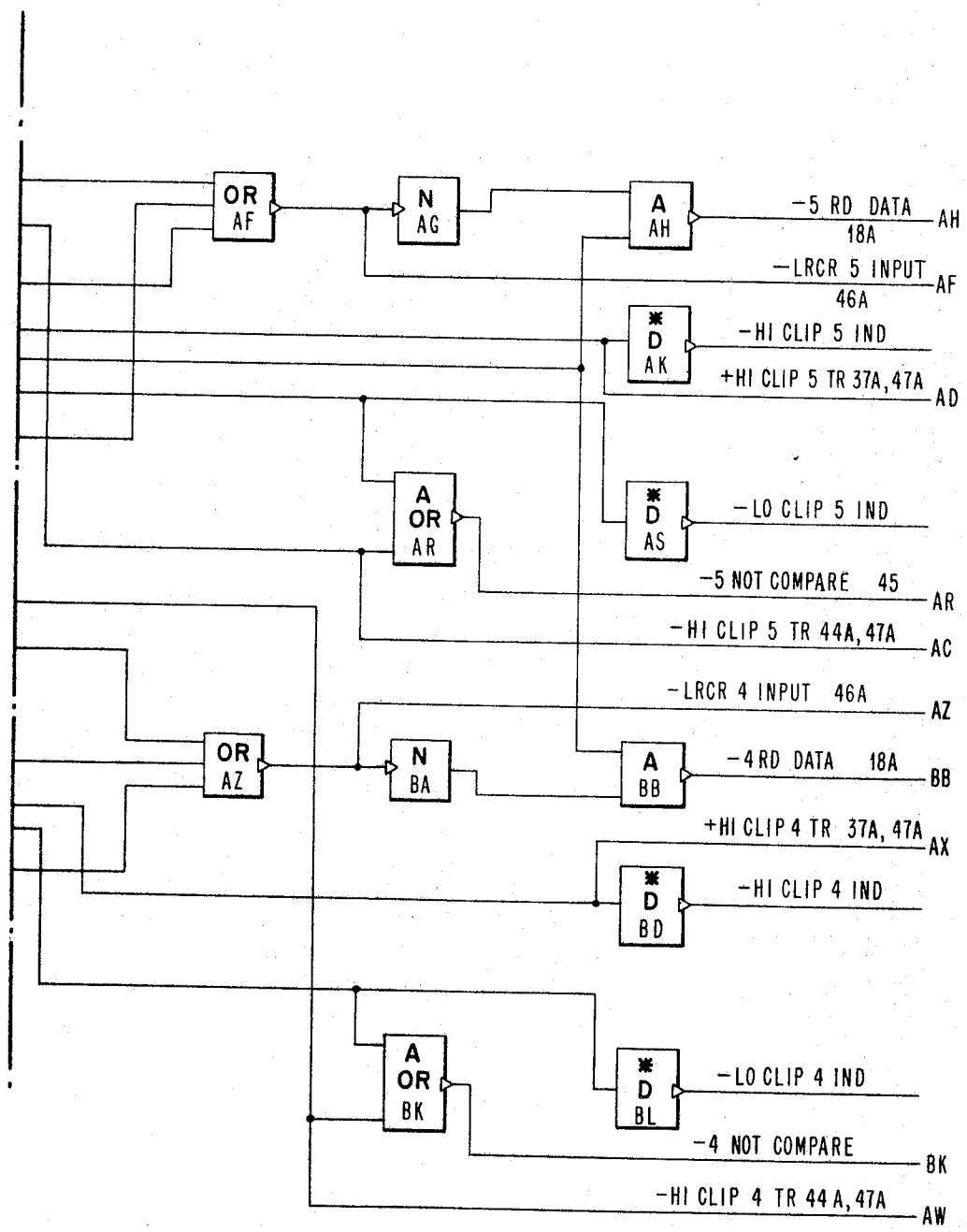

OR AF
N AG
A AH
-5 RD DATA 18A
AH
-LRCR 5 INPUT 46A
AF
✱D AK
-HI CLIP 5 IND
+HI CLIP 5 TR 37A,47A
AD
A OR AR
✱D AS
-LO CLIP 5 IND
-5 NOT COMPARE 45
AR
-HI CLIP 5 TR 44A,47A
AC
-LRCR 4 INPUT 46A
AZ
OR AZ
N BA
A BB
-4 RD DATA 18A
BB
+HI CLIP 4 TR 37A, 47A
AX
✱D BD
-HI CLIP 4 IND
A OR BK
✱D BL
-LO CLIP 4 IND
-4 NOT COMPARE
BK
-HI CLIP 4 TR 44A,47A
AW

FIG. 42A

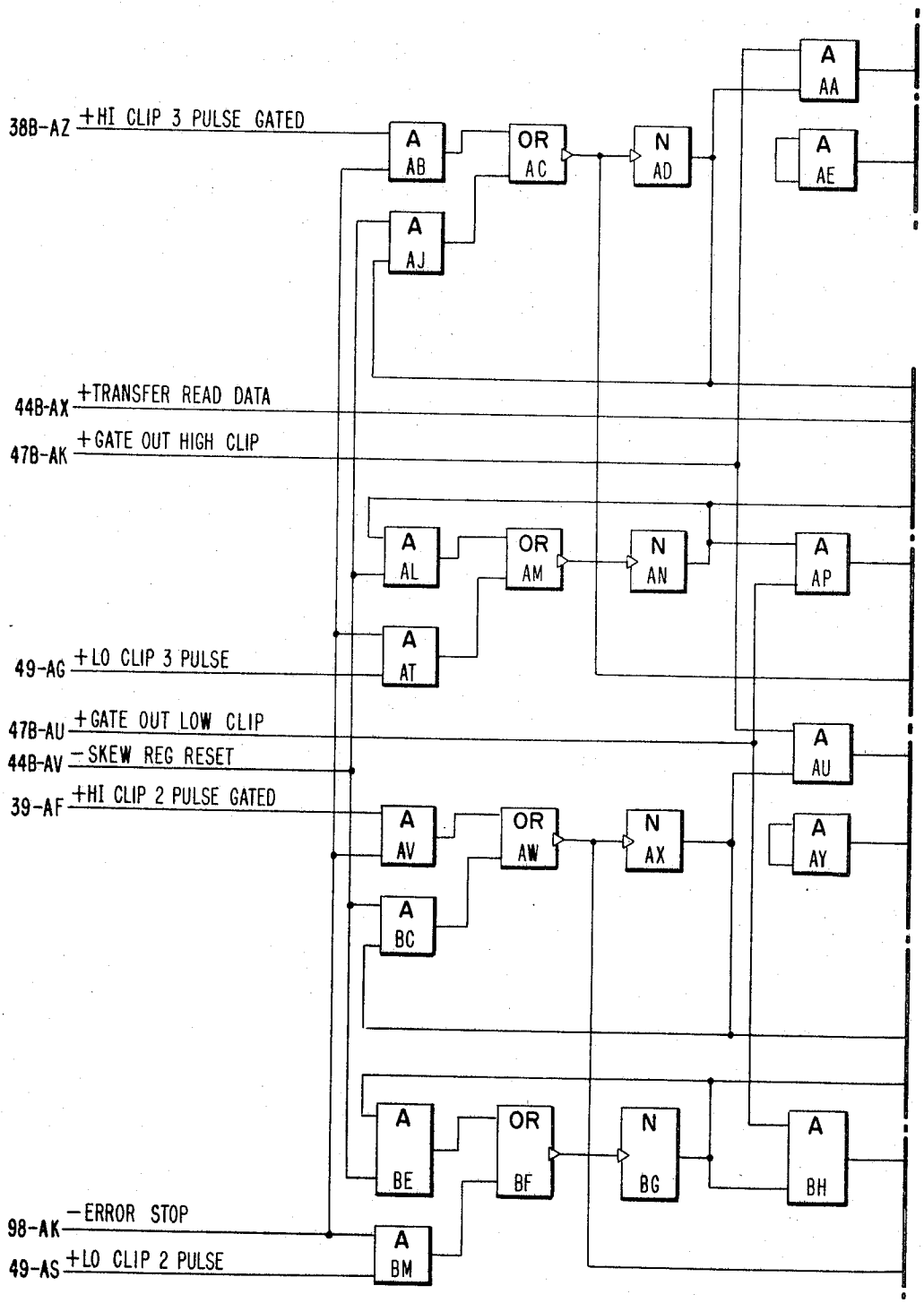

38B-AZ +HI CLIP 3 PULSE GATED
44B-AX +TRANSFER READ DATA
47B-AK +GATE OUT HIGH CLIP
49-AG +LO CLIP 3 PULSE
47B-AU +GATE OUT LOW CLIP
44B-AV -SKEW REG RESET
39-AF +HI CLIP 2 PULSE GATED
98-AK -ERROR STOP
49-AS +LO CLIP 2 PULSE
A AA
A AB
OR AC
N AD
A AE
A AJ
A AL
OR AM
N AN
A AP
A AT
A AU
A AV
OR AW
N AX
A AY
A BC
A BE
OR BF
N BG
A BH
A BM

FIG. 42B

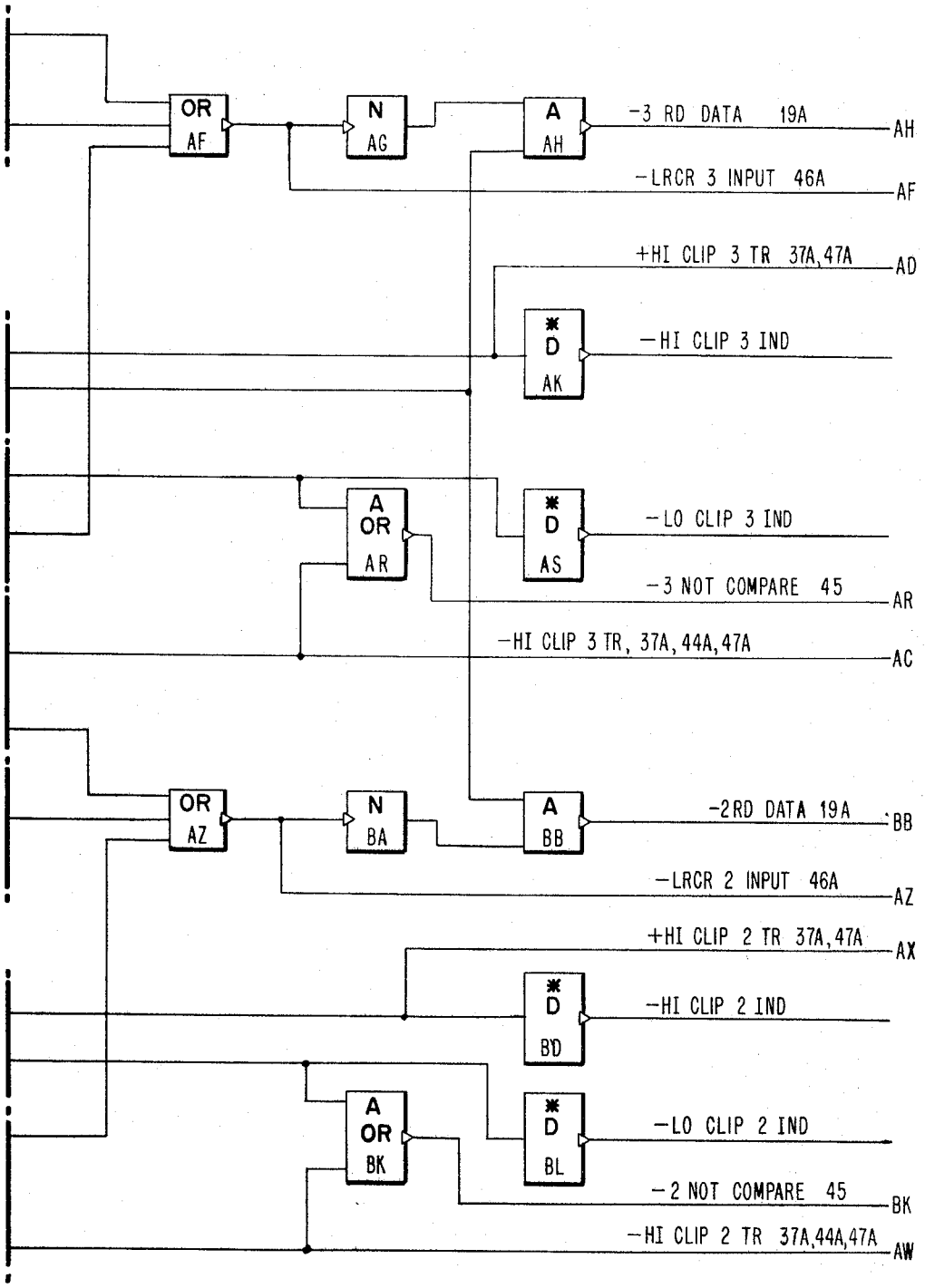

OR AF
N AG
A AH
-3 RD DATA 19A
AH
-LRCR 3 INPUT 46A
AF
+HI CLIP 3 TR 37A,47A
AD
✱D AK
-HI CLIP 3 IND
A OR AR
✱D AS
-LO CLIP 3 IND
-3 NOT COMPARE 45
AR
-HI CLIP 3 TR, 37A, 44A, 47A
AC
OR AZ
N BA
A BB
-2RD DATA 19A
BB
-LRCR 2 INPUT 46A
AZ
+HI CLIP 2 TR 37A,47A
AX
✱D BD
-HI CLIP 2 IND
A OR BK
✱D BL
-LO CLIP 2 IND
-2 NOT COMPARE 45
BK
-HI CLIP 2 TR 37A,44A,47A
AW

FIG. 43A

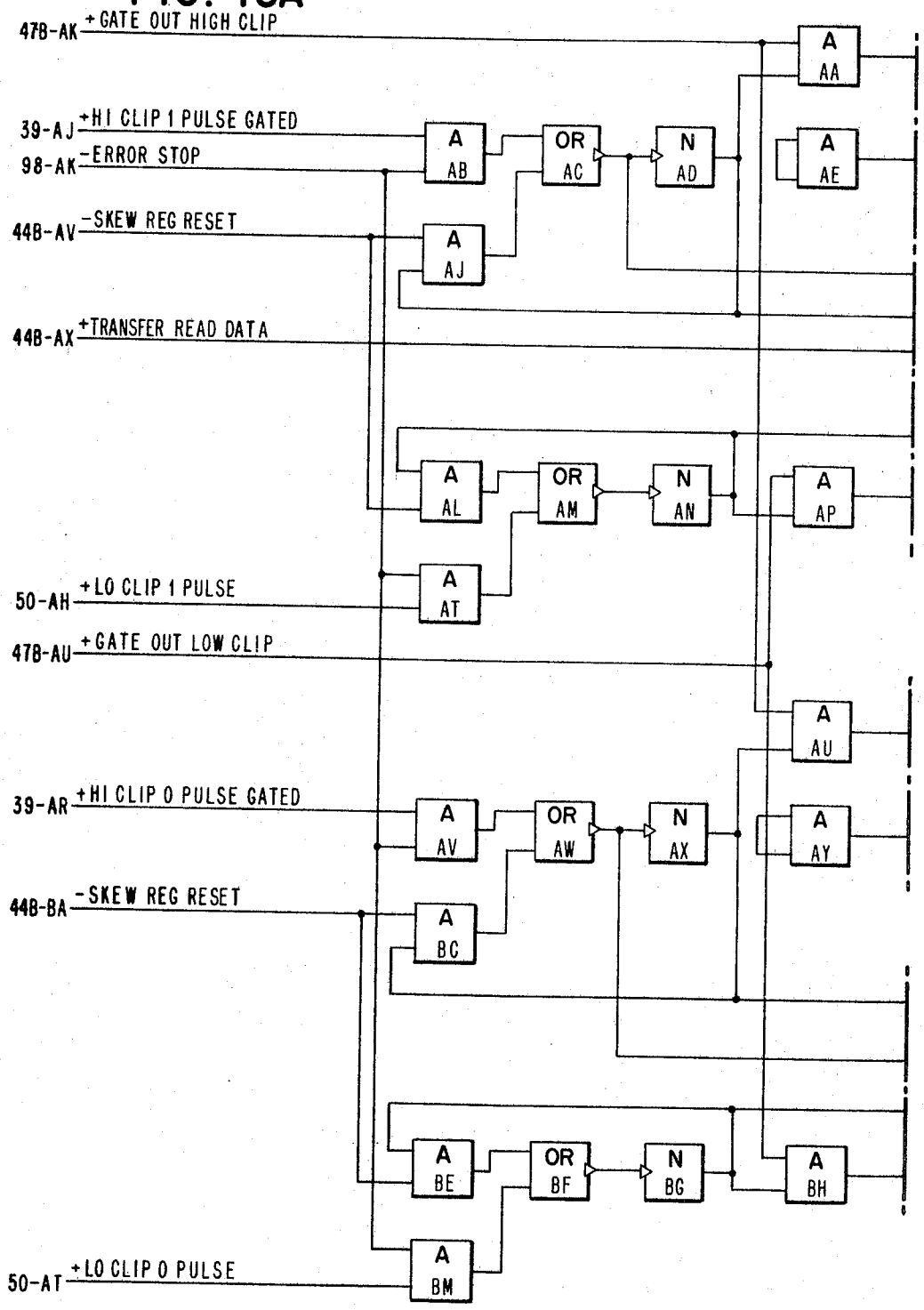

47B-AK +GATE OUT HIGH CLIP
39-AJ +HI CLIP 1 PULSE GATED
98-AK -ERROR STOP
44B-AV -SKEW REG RESET
44B-AX +TRANSFER READ DATA
50-AH +LO CLIP 1 PULSE
47B-AU +GATE OUT LOW CLIP
39-AR +HI CLIP 0 PULSE GATED
44B-BA -SKEW REG RESET
50-AT +LO CLIP 0 PULSE
A AA
A AB
OR AC
N AD
A AE
A AJ
A AL
OR AM
N AN
A AP
A AT
A AU
A AV
OR AW
N AX
A AY
A BC
A BE
OR BF
N BG
A BH
A BM

FIG. 43B

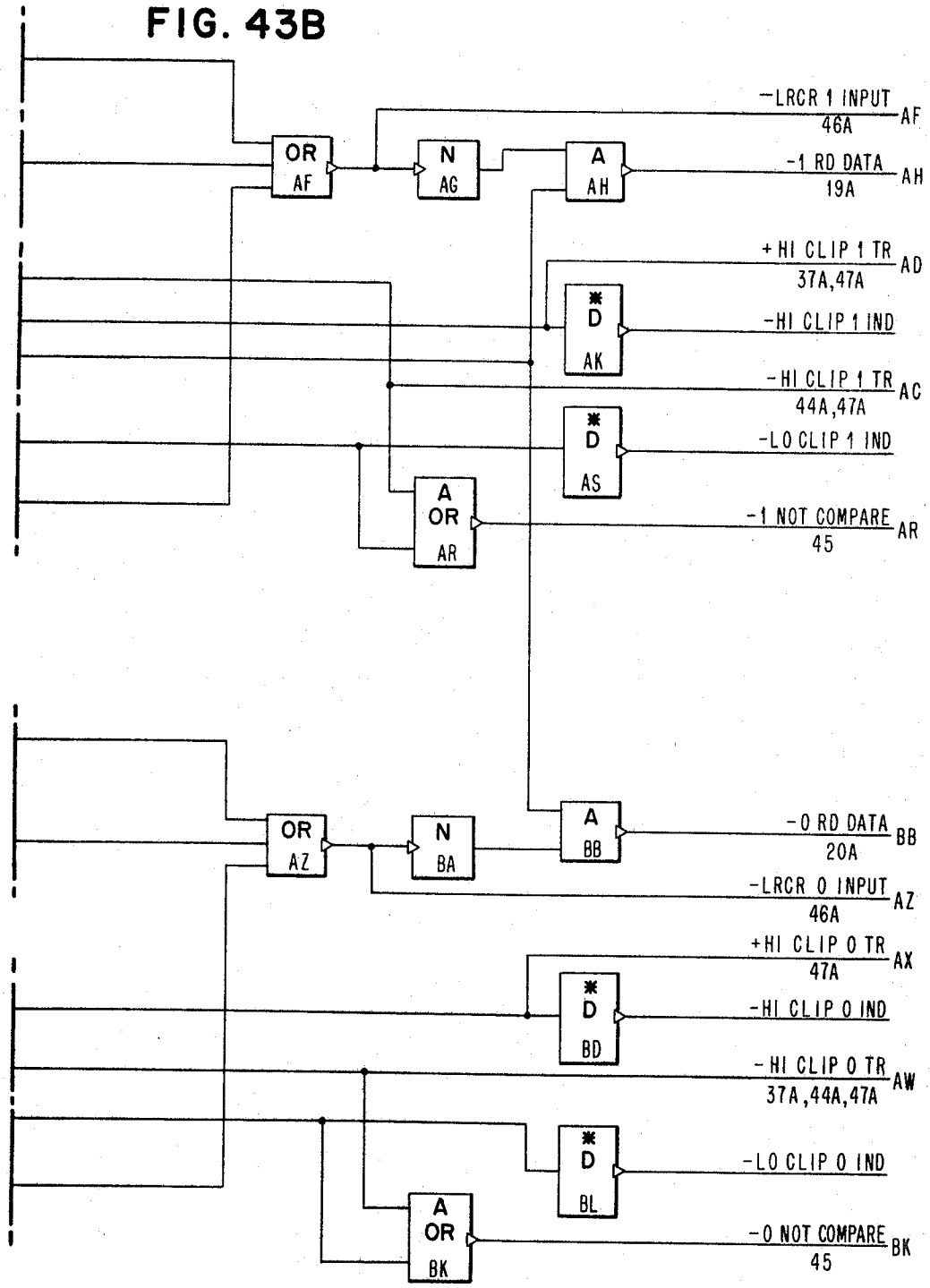

OR AF
N AG
A AH
-LRCR 1 INPUT 46A AF
-1 RD DATA 19A AH
+HI CLIP 1 TR 37A,47A AD
* D AK
-HI CLIP 1 IND
-HI CLIP 1 TR 44A,47A AC
* D AS
-LO CLIP 1 IND
A OR AR
-1 NOT COMPARE 45 AR
OR AZ
N BA
A BB
-0 RD DATA 20A BB
-LRCR 0 INPUT 46A AZ
+HI CLIP 0 TR 47A AX
* D BD
-HI CLIP 0 IND
-HI CLIP 0 TR 37A,44A,47A AW
* D BL
-LO CLIP 0 IND
A OR BK
-0 NOT COMPARE 45 BK

FIG. 44A

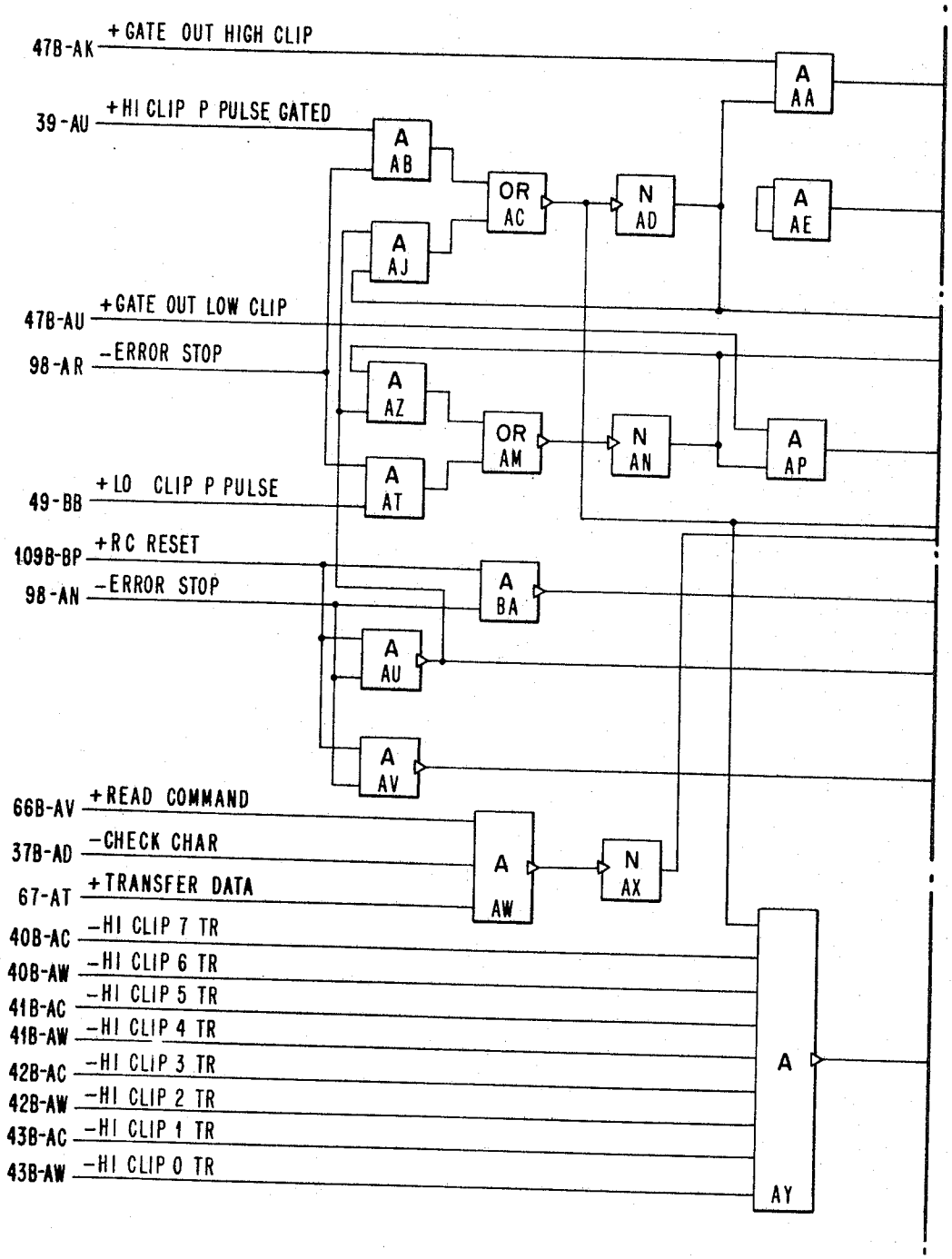

47B-AK +GATE OUT HIGH CLIP
39-AU +HI CLIP P PULSE GATED
47B-AU +GATE OUT LOW CLIP
98-AR -ERROR STOP
49-BB +LO CLIP P PULSE
109B-BP +RC RESET
98-AN -ERROR STOP
66B-AV +READ COMMAND
37B-AD -CHECK CHAR
67-AT +TRANSFER DATA
40B-AC -HI CLIP 7 TR
40B-AW -HI CLIP 6 TR
41B-AC -HI CLIP 5 TR
41B-AW -HI CLIP 4 TR
42B-AC -HI CLIP 3 TR
42B-AW -HI CLIP 2 TR
43B-AC -HI CLIP 1 TR
43B-AW -HI CLIP 0 TR
A AA
A AB
OR AC
N AD
A AE
A AJ
A AZ
OR AM
N AN
A AP
A AT
A BA
A AU
A AV
A AW
N AX
A AY

FIG. 44B

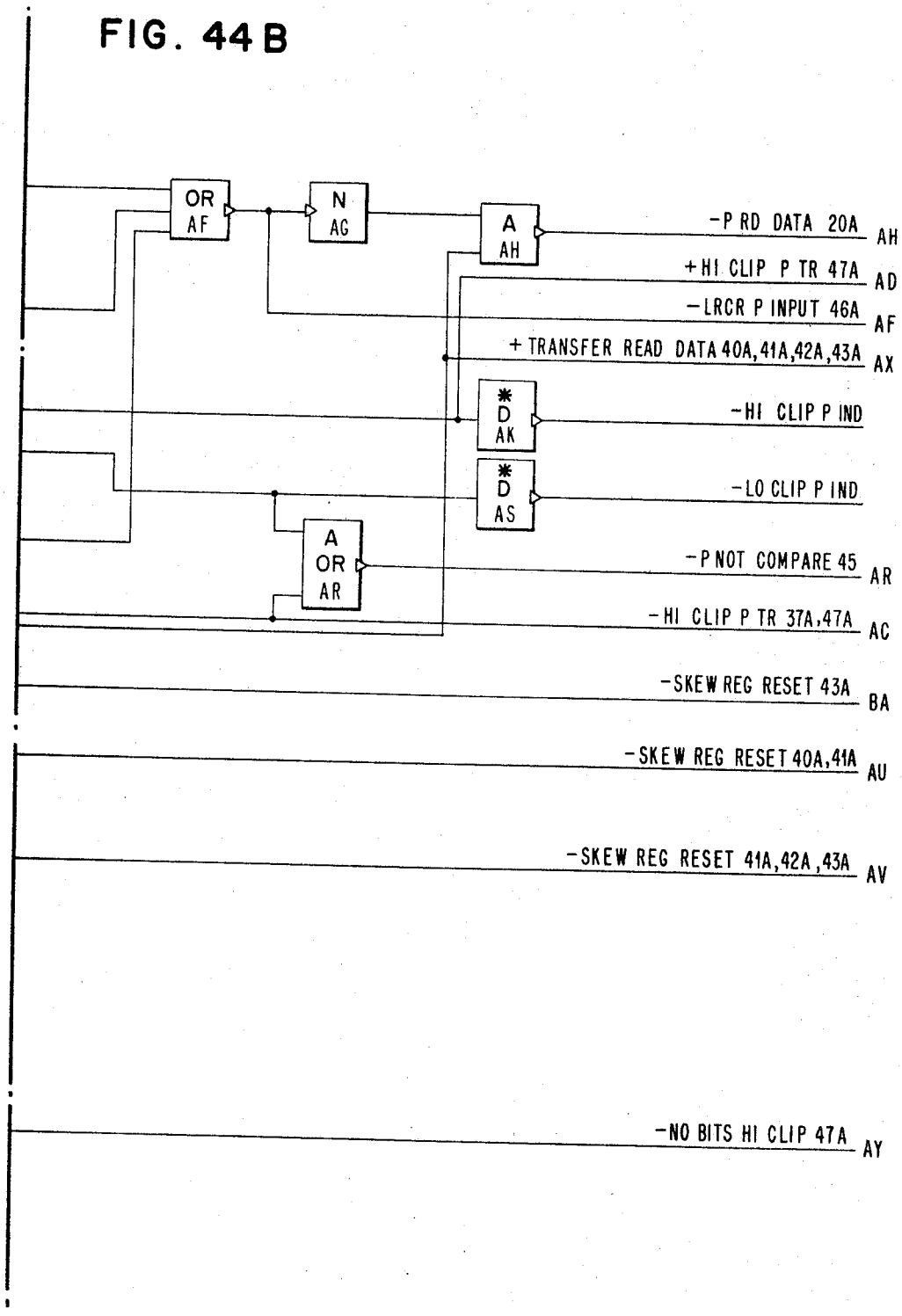

OR AF
N AG
A AH
-P RD DATA 20A AH
+HI CLIP P TR 47A AD
-LRCR P INPUT 46A AF
+TRANSFER READ DATA 40A,41A,42A,43A AX
* D AK
-HI CLIP P IND
* D AS
-LO CLIP P IND
A OR AR
-P NOT COMPARE 45 AR
-HI CLIP P TR 37A,47A AC
-SKEW REG RESET 43A BA
-SKEW REG RESET 40A,41A AU
-SKEW REG RESET 41A,42A,43A AV
-NO BITS HI CLIP 47A AY

FIG. 45

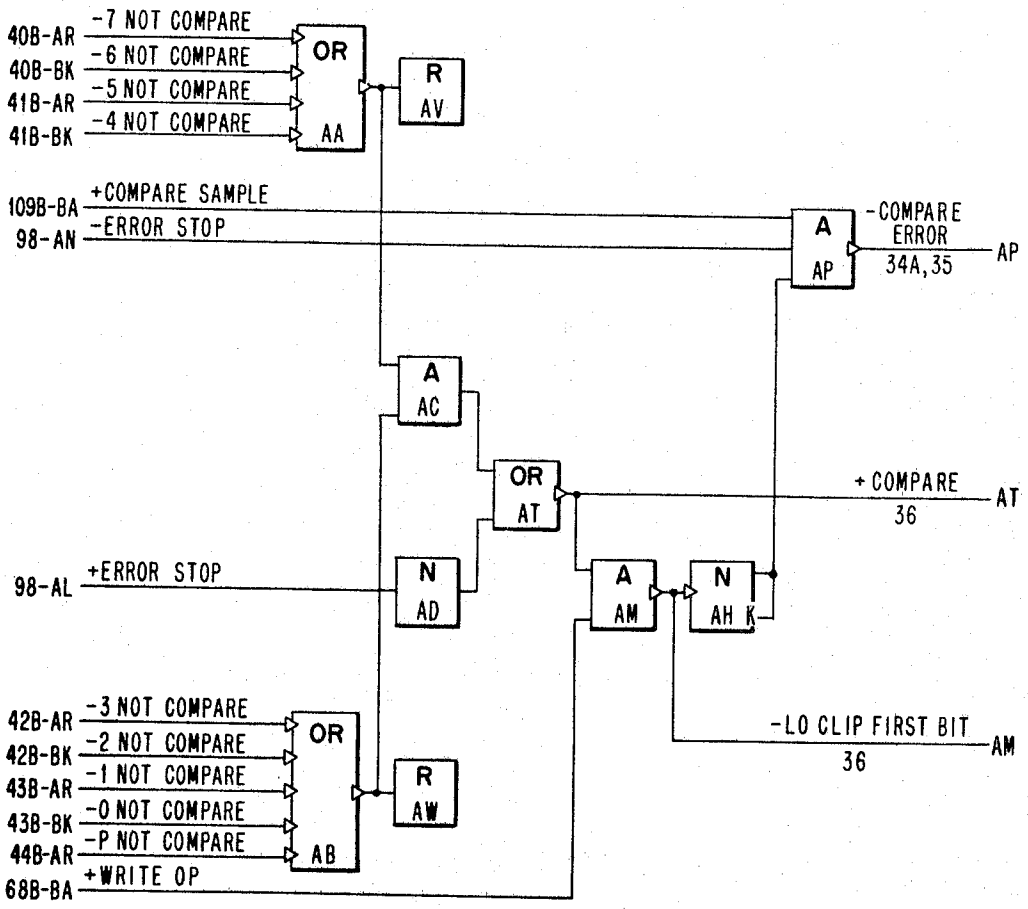

40B-AR
40B-BK
41B-AR
41B-BK
-7 NOT COMPARE
-6 NOT COMPARE
-5 NOT COMPARE
-4 NOT COMPARE
OR
AA
R
AV
109B-BA
98-AN
+COMPARE SAMPLE
-ERROR STOP
A
AP
-COMPARE ERROR
34A,35
AP
A
AC
OR
AT
+COMPARE
36
AT
98-AL
+ERROR STOP
N
AD
A
AM
N
AH K
-LO CLIP FIRST BIT
36
AM
42B-AR
42B-BK
43B-AR
43B-BK
44B-AR
68B-BA
-3 NOT COMPARE
-2 NOT COMPARE
-1 NOT COMPARE
-0 NOT COMPARE
-P NOT COMPARE
+WRITE OP
OR
AB
R
AW

FIG. 46 A

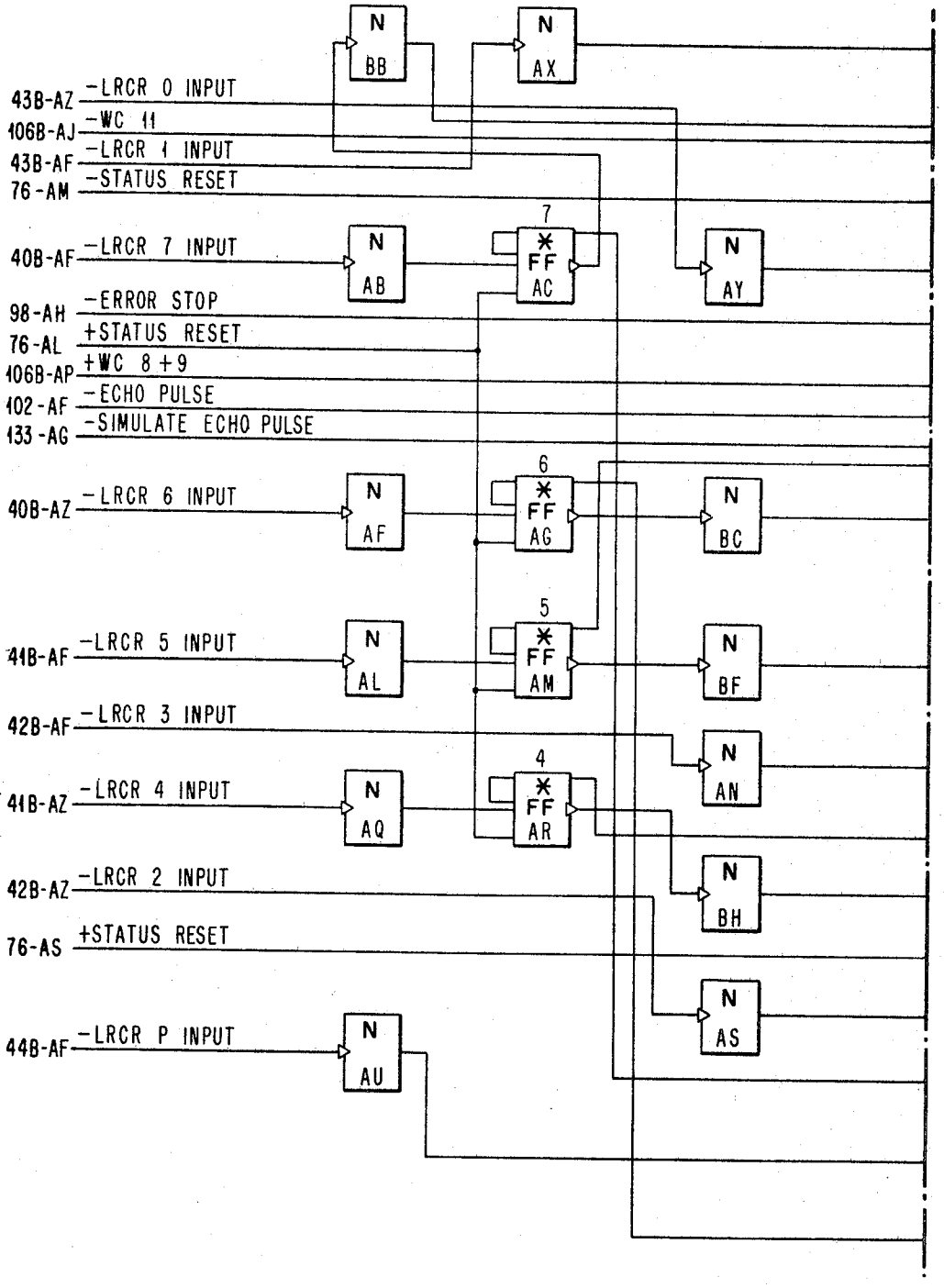

N BB
N AX
43B-AZ -LRCR 0 INPUT
106B-AJ -WC 11
43B-AF -LRCR 1 INPUT
76-AM -STATUS RESET
7
40B-AF -LRCR 7 INPUT
N AB
FF AC
N AY
98-AH -ERROR STOP
76-AL +STATUS RESET
106B-AP +WC 8+9
102-AF -ECHO PULSE
133-AG -SIMULATE ECHO PULSE
6
40B-AZ -LRCR 6 INPUT
N AF
FF AG
N BC
5
41B-AF -LRCR 5 INPUT
N AL
FF AM
N BF
42B-AF -LRCR 3 INPUT
N AN
4
41B-AZ -LRCR 4 INPUT
N AQ
FF AR
42B-AZ -LRCR 2 INPUT
N BH
76-AS +STATUS RESET
N AS
44B-AF -LRCR P INPUT
N AU

FIG.46B

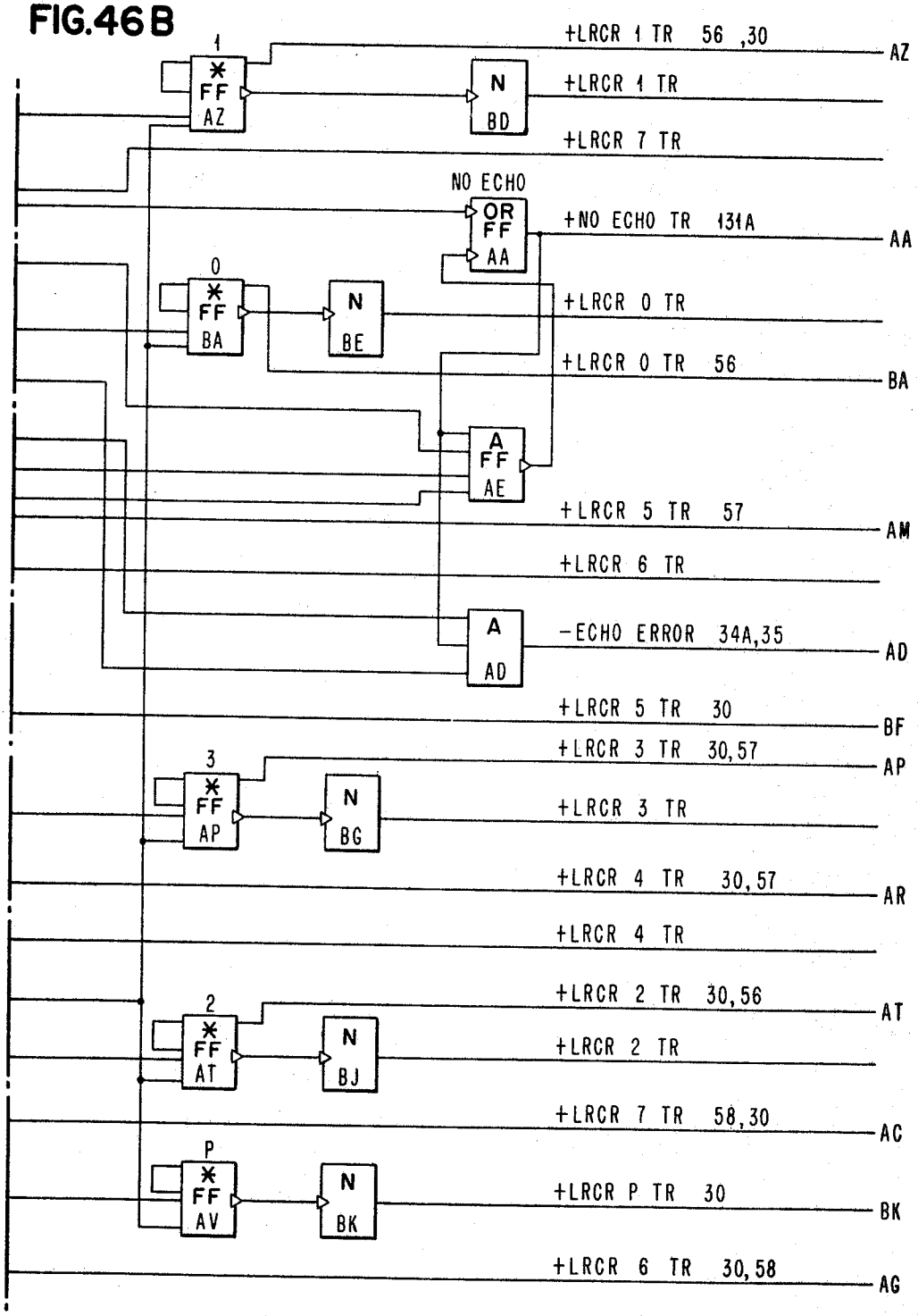

1
FF AZ
N BD
+LRCR 1 TR 56 ,30
AZ
+LRCR 1 TR
+LRCR 7 TR
NO ECHO
OR FF AA
+NO ECHO TR 131A
AA
0
FF BA
N BE
+LRCR 0 TR
+LRCR 0 TR 56
BA
A FF AE
+LRCR 5 TR 57
AM
+LRCR 6 TR
A AD
-ECHO ERROR 34A,35
AD
+LRCR 5 TR 30
BF
3
FF AP
N BG
+LRCR 3 TR 30,57
AP
+LRCR 3 TR
+LRCR 4 TR 30,57
AR
+LRCR 4 TR
2
FF AT
N BJ
+LRCR 2 TR 30,56
AT
+LRCR 2 TR
+LRCR 7 TR 58,30
AC
P
FF AV
N BK
+LRCR P TR 30
BK
+LRCR 6 TR 30,58
AG

FIG. 47A

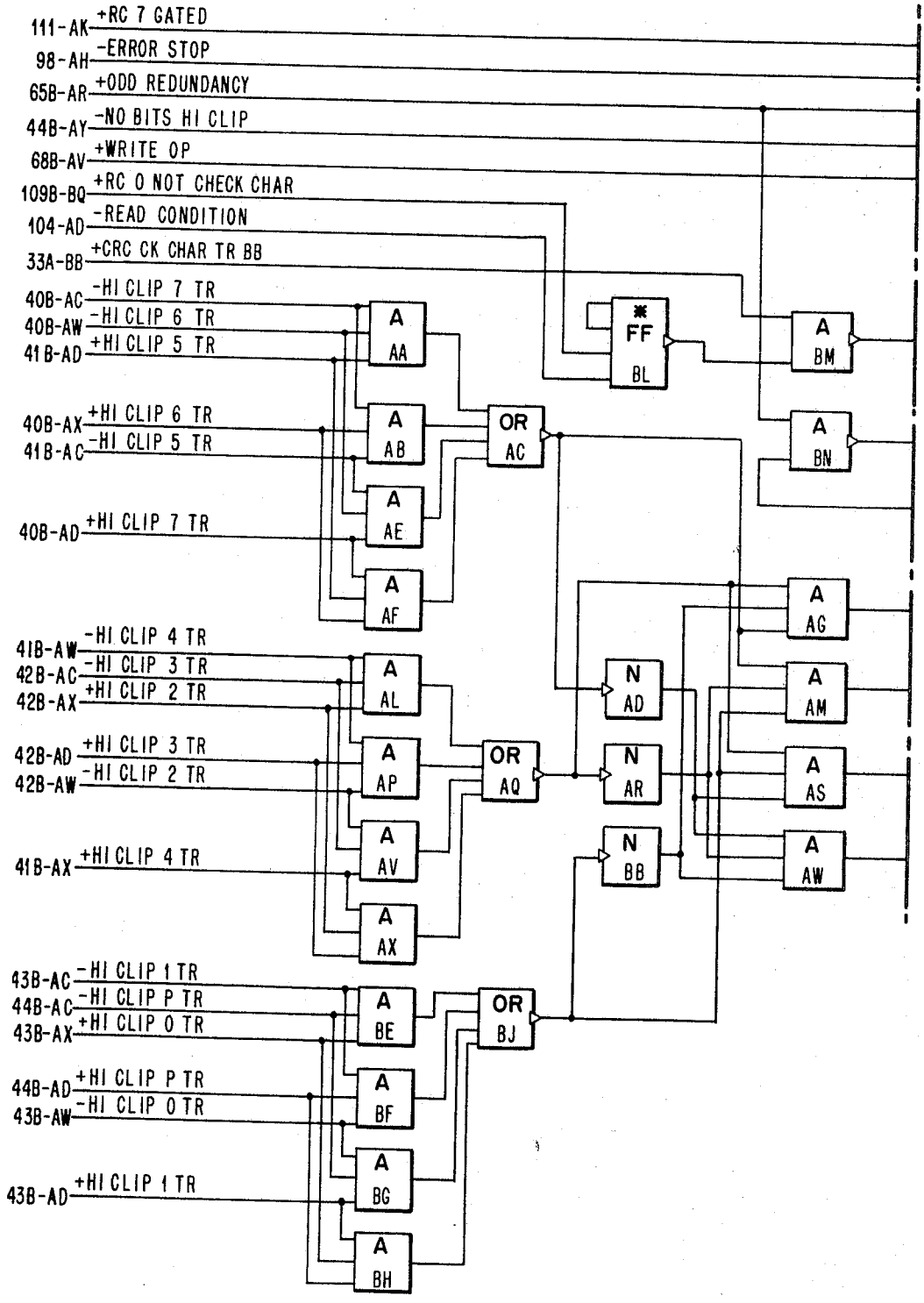

111-AK +RC 7 GATED
98-AH -ERROR STOP
65B-AR +ODD REDUNDANCY
44B-AY -NO BITS HI CLIP
68B-AV +WRITE OP
109B-BQ +RC 0 NOT CHECK CHAR
104-AD -READ CONDITION
33A-BB +CRC CK CHAR TR BB
40B-AC -HI CLIP 7 TR
40B-AW -HI CLIP 6 TR
41B-AD +HI CLIP 5 TR
40B-AX +HI CLIP 6 TR
41B-AC -HI CLIP 5 TR
40B-AD +HI CLIP 7 TR
41B-AW -HI CLIP 4 TR
42B-AC -HI CLIP 3 TR
42B-AX +HI CLIP 2 TR
42B-AD +HI CLIP 3 TR
42B-AW -HI CLIP 2 TR
41B-AX +HI CLIP 4 TR
43B-AC -HI CLIP 1 TR
44B-AC -HI CLIP P TR
43B-AX +HI CLIP 0 TR
44B-AD +HI CLIP P TR
43B-AW -HI CLIP 0 TR
43B-AD +HI CLIP 1 TR
A AA
A AB
A AE
A AF
OR AC
A AL
A AP
A AV
A AX
OR AQ
A BE
A BF
A BG
A BH
OR BJ
* FF BL
N AD
N AR
N BB
A BM
A BN
A AG
A AM
A AS
A AW

FIG. 47B

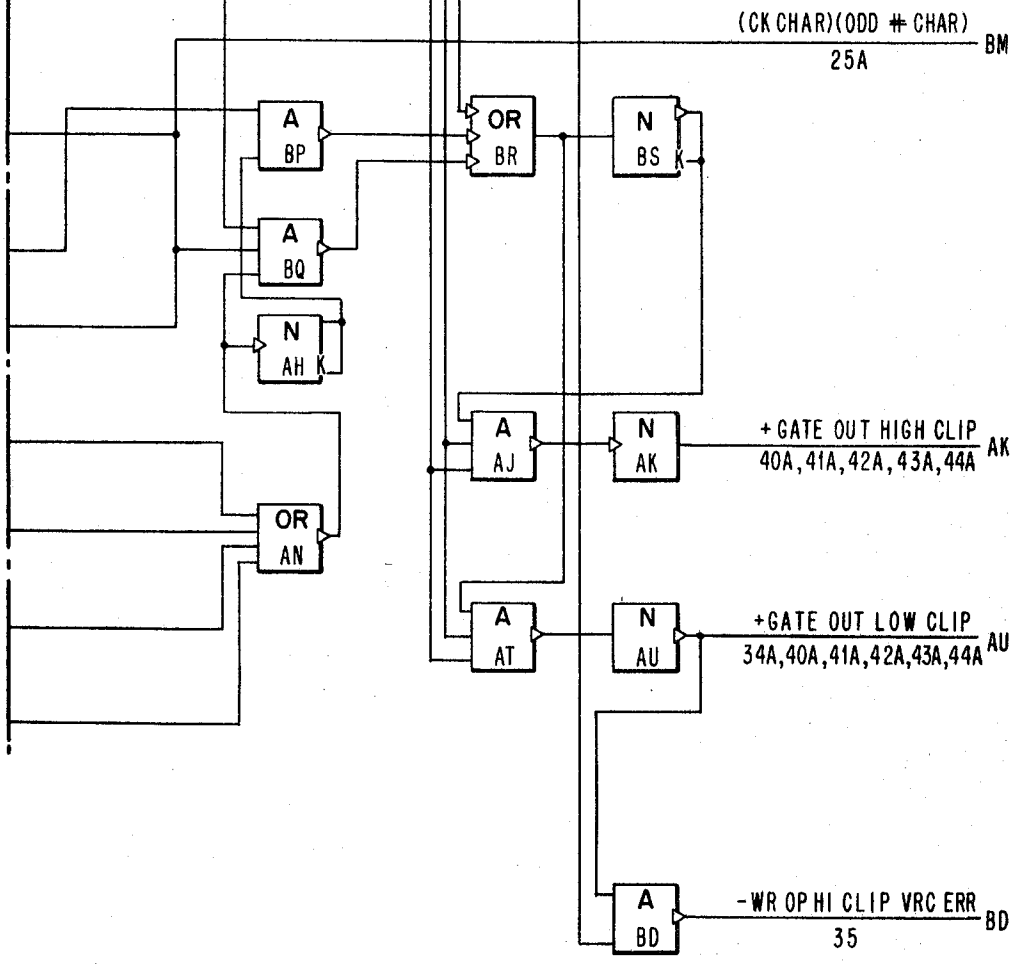

(CK CHAR)(ODD # CHAR)
25A
BM
A
BP
OR
BR
N
BS
K
A
BQ
N
AH
K
A
AJ
N
AK
+GATE OUT HIGH CLIP
40A,41A,42A,43A,44A
AK
OR
AN
A
AT
N
AU
+GATE OUT LOW CLIP
34A,40A,41A,42A,43A,44A
AU
A
BD
-WR OP HI CLIP VRC ERR
35
BD

FIG. 48

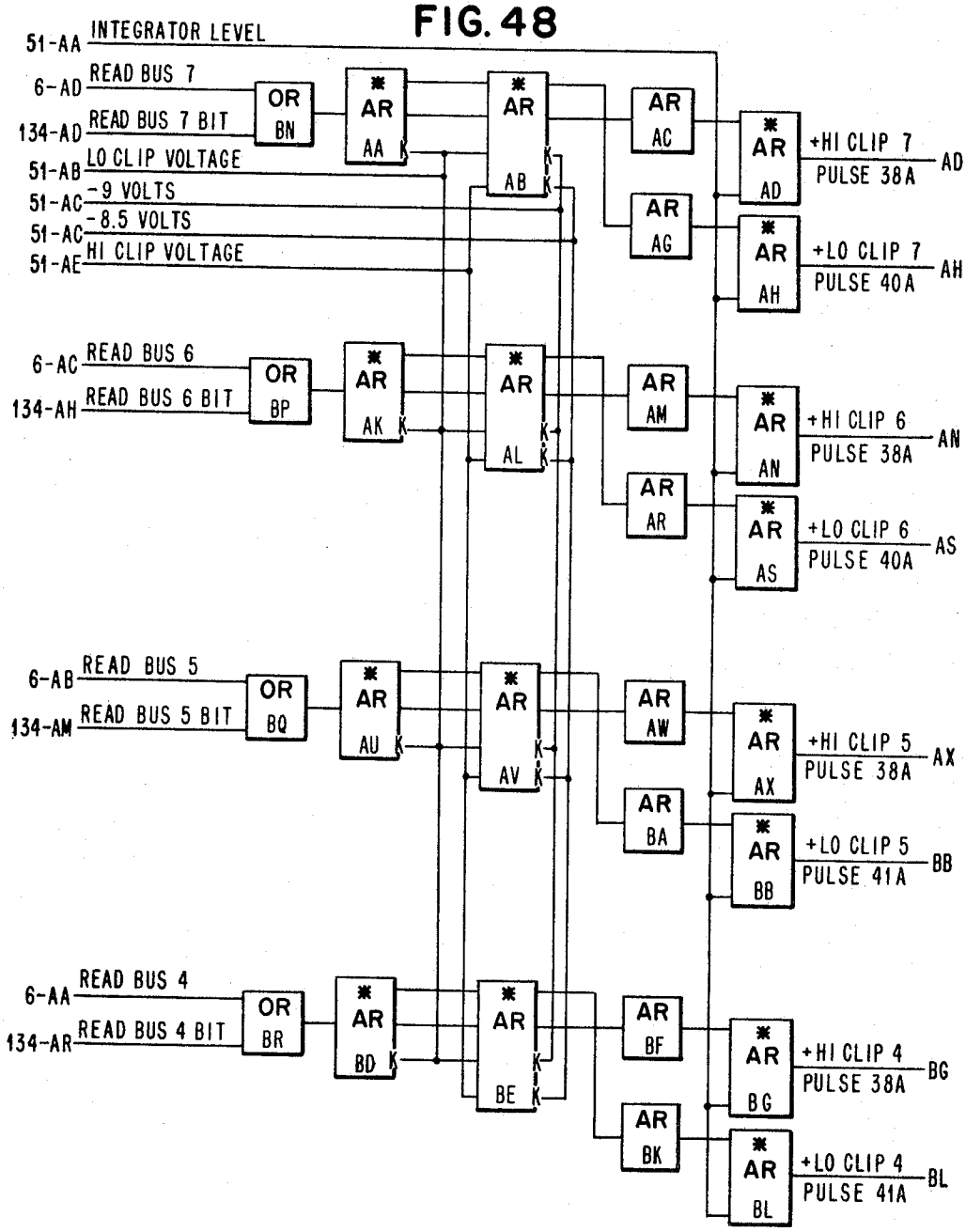

51-AA INTEGRATOR LEVEL
6-AD READ BUS 7
134-AD READ BUS 7 BIT
OR BN
51-AB LO CLIP VOLTAGE
51-AC -9 VOLTS
51-AC -8.5 VOLTS
51-AE HI CLIP VOLTAGE
AR AA
AR AB
AR AC
AR AD
+HI CLIP 7 PULSE 38A AD
AR AG
AR AH
+LO CLIP 7 PULSE 40A AH
6-AC READ BUS 6
134-AH READ BUS 6 BIT
OR BP
AR AK
AR AL
AR AM
AR AN
+HI CLIP 6 PULSE 38A AN
AR AR
AR AS
+LO CLIP 6 PULSE 40A AS
6-AB READ BUS 5
134-AM READ BUS 5 BIT
OR BQ
AR AU
AR AV
AR AW
AR AX
+HI CLIP 5 PULSE 38A AX
AR BA
AR BB
+LO CLIP 5 PULSE 41A BB
6-AA READ BUS 4
134-AR READ BUS 4 BIT
OR BR
AR BD
AR BE
AR BF
AR BG
+HI CLIP 4 PULSE 38A BG
AR BK
AR BL
+LO CLIP 4 PULSE 41A BL

FIG. 49

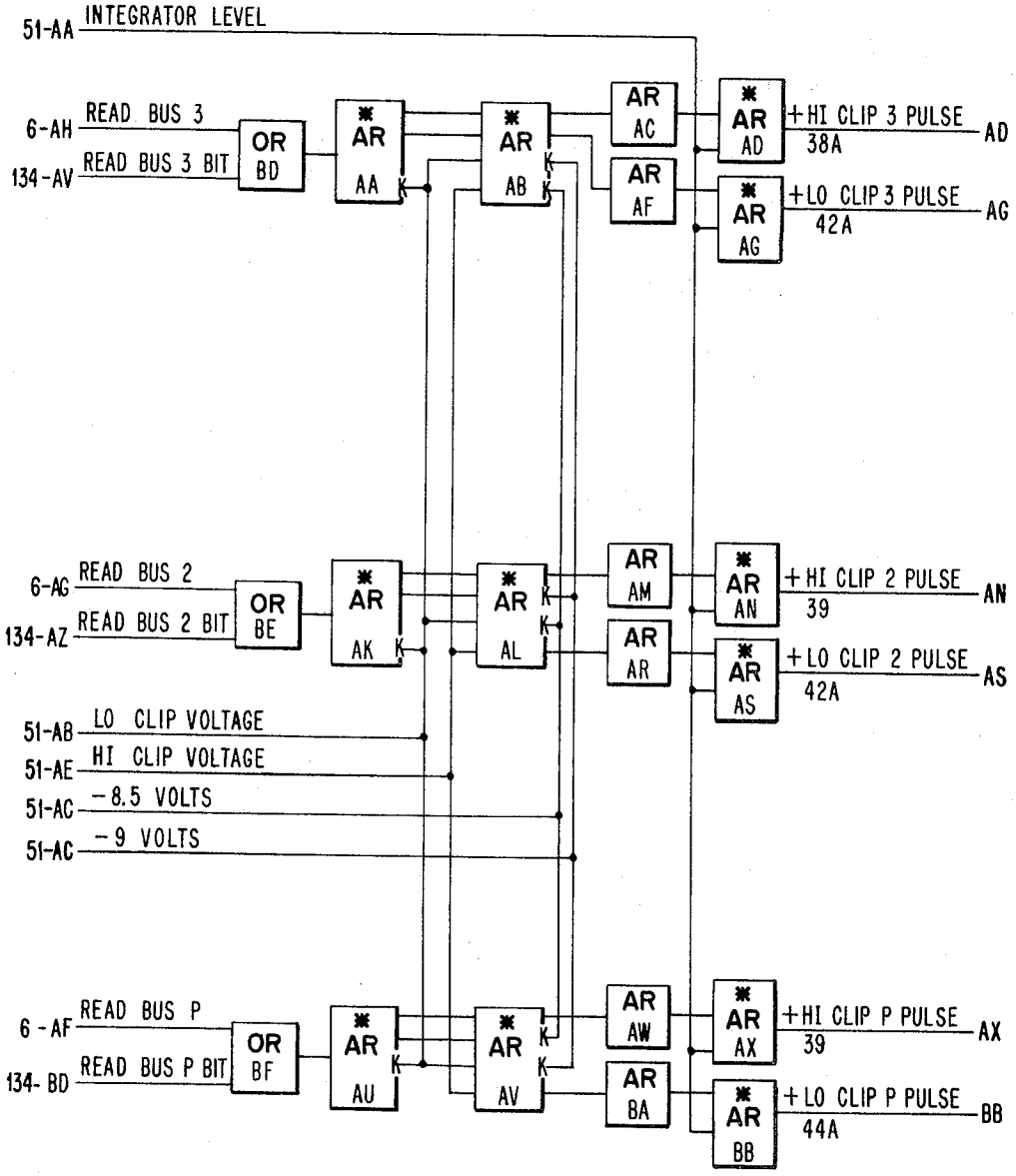

51-AA INTEGRATOR LEVEL
6-AH READ BUS 3
134-AV READ BUS 3 BIT
OR BD
AR AA
AR AB
AR AC
AR AF
AR AD
+HI CLIP 3 PULSE 38A
AD
AR AG
+LO CLIP 3 PULSE 42A
AG
6-AG READ BUS 2
134-AZ READ BUS 2 BIT
OR BE
AR AK
AR AL
AR AM
AR AR
AR AN
+HI CLIP 2 PULSE 39
AN
AR AS
+LO CLIP 2 PULSE 42A
AS
51-AB LO CLIP VOLTAGE
51-AE HI CLIP VOLTAGE
51-AC -8.5 VOLTS
51-AC -9 VOLTS
6-AF READ BUS P
134-BD READ BUS P BIT
OR BF
AR AU
AR AV
AR AW
AR BA
AR AX
+HI CLIP P PULSE 39
AX
AR BB
+LO CLIP P PULSE 44A
BB

FIG. 50

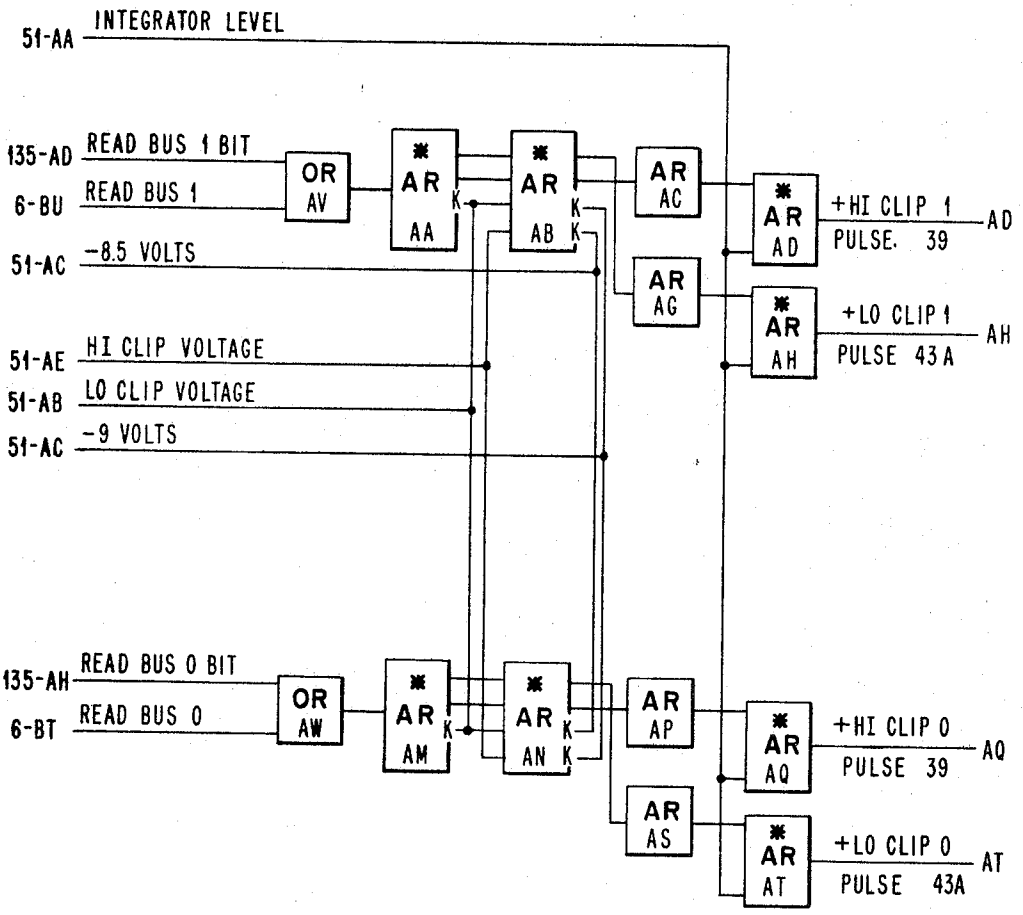

51-AA
INTEGRATOR LEVEL
135-AD
READ BUS 1 BIT
6-BU
READ BUS 1
OR AV
51-AC
-8.5 VOLTS
AR AA
AR AB
AR AC
AR AD
+HI CLIP 1
PULSE 39
AD
AR AG
AR AH
+LO CLIP 1
PULSE 43A
AH
51-AE
HI CLIP VOLTAGE
51-AB
LO CLIP VOLTAGE
51-AC
-9 VOLTS
135-AH
READ BUS 0 BIT
6-BT
READ BUS 0
OR AW
AR AM
AR AN
AR AP
AR AQ
+HI CLIP 0
PULSE 39
AQ
AR AS
AR AT
+LO CLIP 0
PULSE 43A
AT

FIG. 51

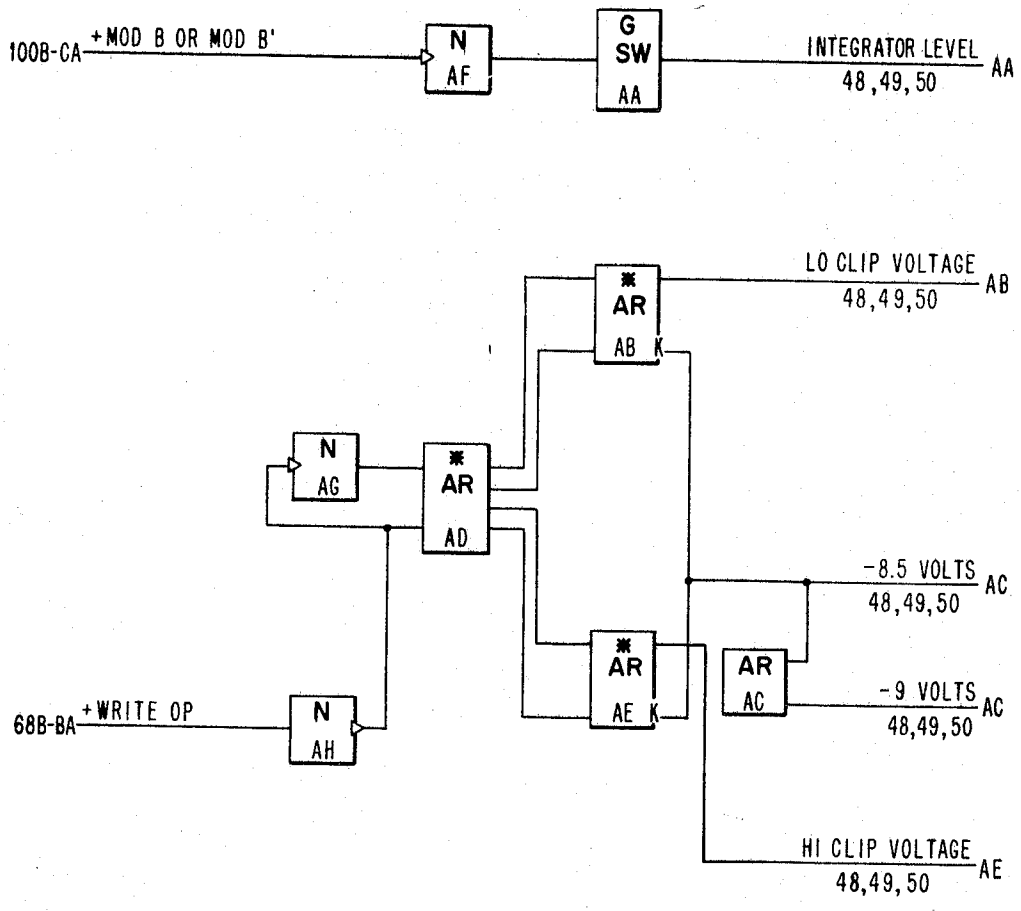

100B-CA
+MOD B OR MOD B'
N
AF
G
SW
AA
INTEGRATOR LEVEL
48,49,50
AA
LO CLIP VOLTAGE
48,49,50
AB
*
AR
AB K
N
AG
*
AR
AD
-8.5 VOLTS
48,49,50
AC
*
AR
AE K
AR
AC
-9 VOLTS
48,49,50
AC
68B-BA
+WRITE OP
N
AH
HI CLIP VOLTAGE
48,49,50
AE

FIG. 52

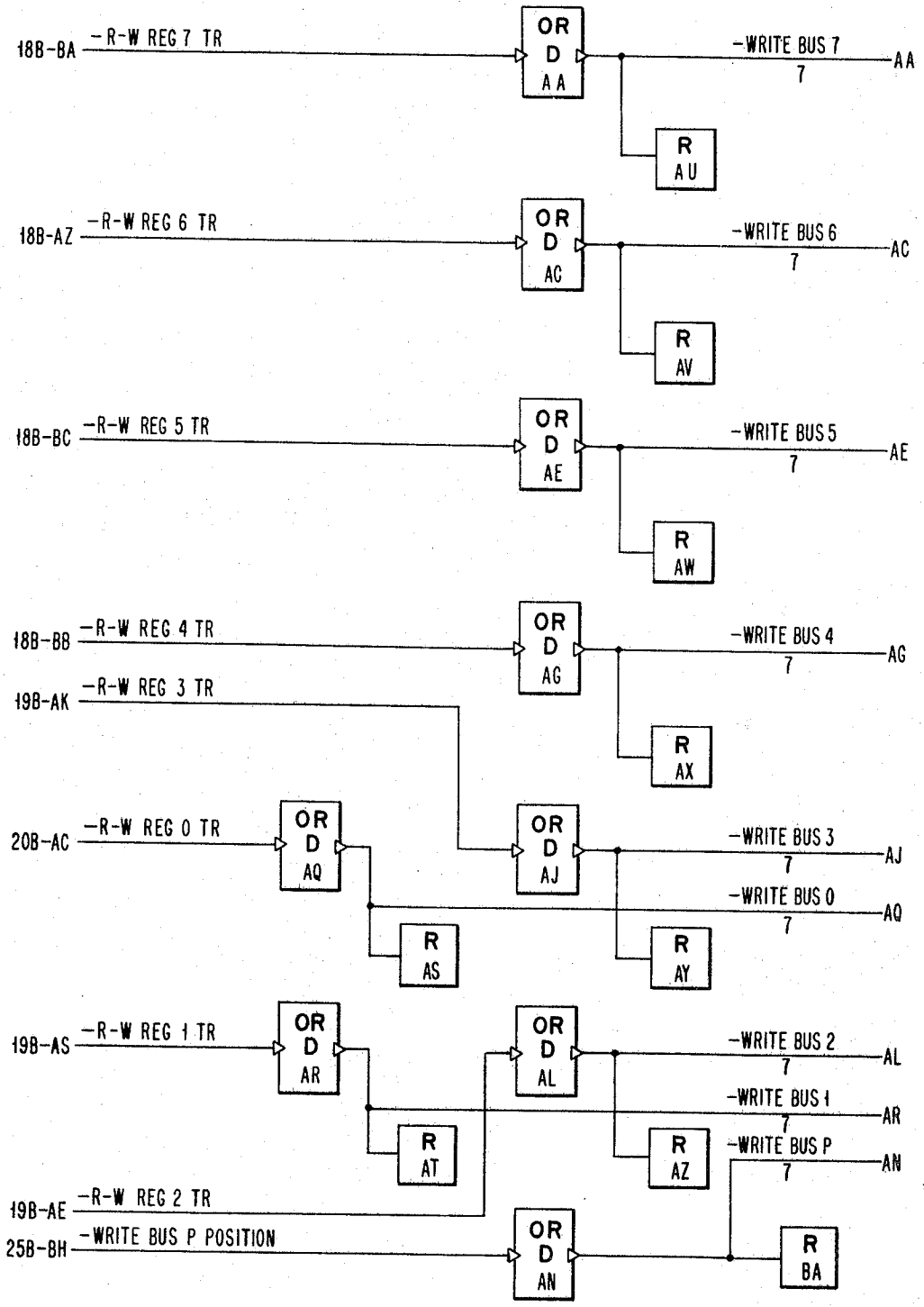

18B-BA
-R-W REG 7 TR
OR D AA
-WRITE BUS 7
7
AA
R AU
18B-AZ
-R-W REG 6 TR
OR D AC
-WRITE BUS 6
7
AC
R AV
18B-BC
-R-W REG 5 TR
OR D AE
-WRITE BUS 5
7
AE
R AW
18B-BB
-R-W REG 4 TR
OR D AG
-WRITE BUS 4
7
AG
19B-AK
-R-W REG 3 TR
R AX
20B-AC
-R-W REG 0 TR
OR D AQ
OR D AJ
-WRITE BUS 3
7
AJ
-WRITE BUS 0
7
AQ
R AS
R AY
19B-AS
-R-W REG 1 TR
OR D AR
OR D AL
-WRITE BUS 2
7
AL
-WRITE BUS 1
7
AR
R AT
R AZ
-WRITE BUS P
7
AN
19B-AE
-R-W REG 2 TR
25B-BH
-WRITE BUS P POSITION
OR D AN
R BA

FIG. 53A

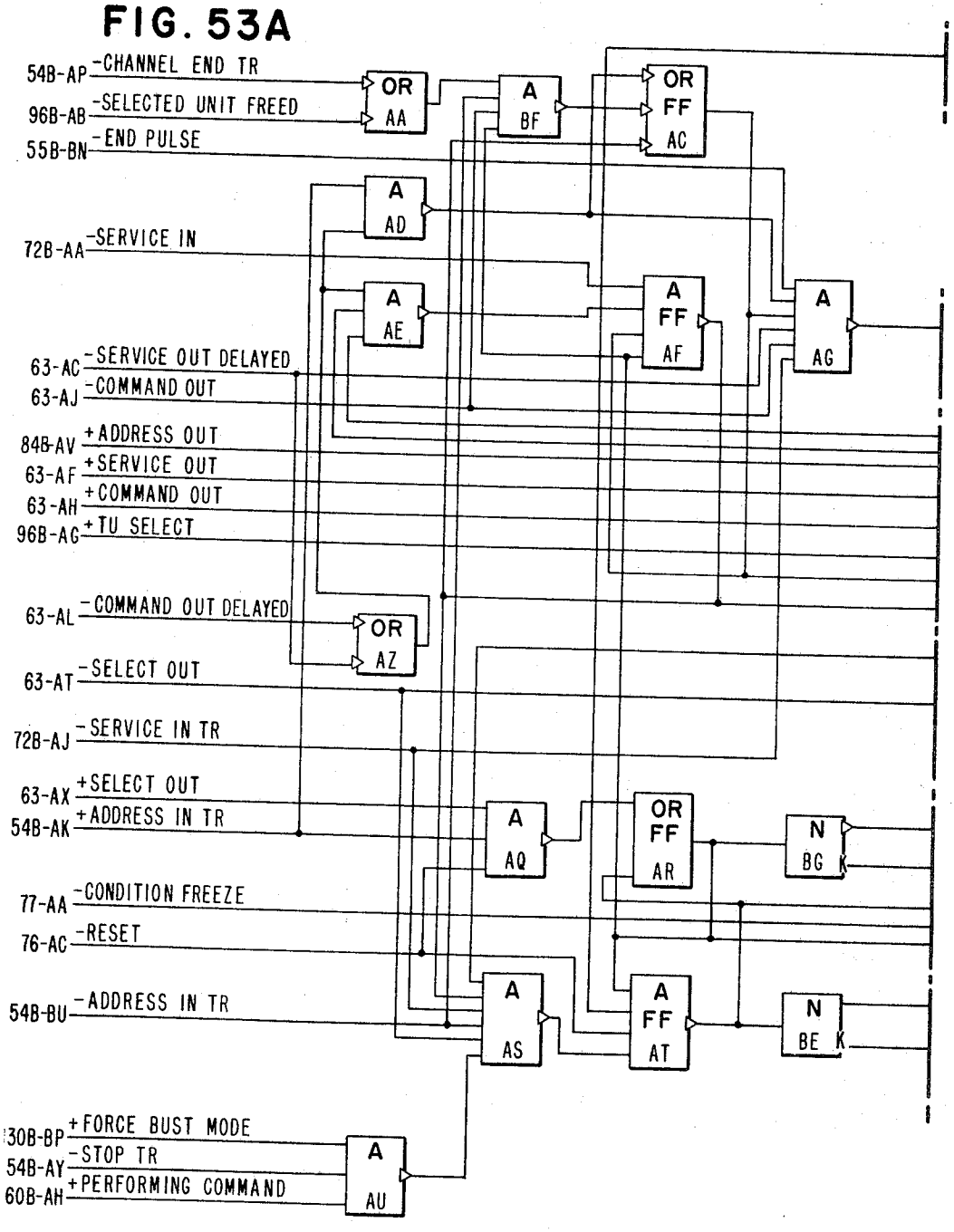

54B-AP -CHANNEL END TR
96B-AB -SELECTED UNIT FREED
55B-BN -END PULSE
OR AA
A BF
OR FF AC
A AD
72B-AA -SERVICE IN
A AE
A FF AF
A AG
63-AC -SERVICE OUT DELAYED
63-AJ -COMMAND OUT
84B-AV +ADDRESS OUT
63-AF +SERVICE OUT
63-AH +COMMAND OUT
96B-AG +TU SELECT
63-AL -COMMAND OUT DELAYED
OR AZ
63-AT -SELECT OUT
72B-AJ -SERVICE IN TR
63-AX +SELECT OUT
54B-AK +ADDRESS IN TR
A AQ
OR FF AR
N BG K
77-AA -CONDITION FREEZE
76-AC -RESET
54B-BU -ADDRESS IN TR
A AS
A FF AT
N BE K
30B-BP +FORCE BUST MODE
54B-AY -STOP TR
60B-AH +PERFORMING COMMAND
A AU

FIG.53B

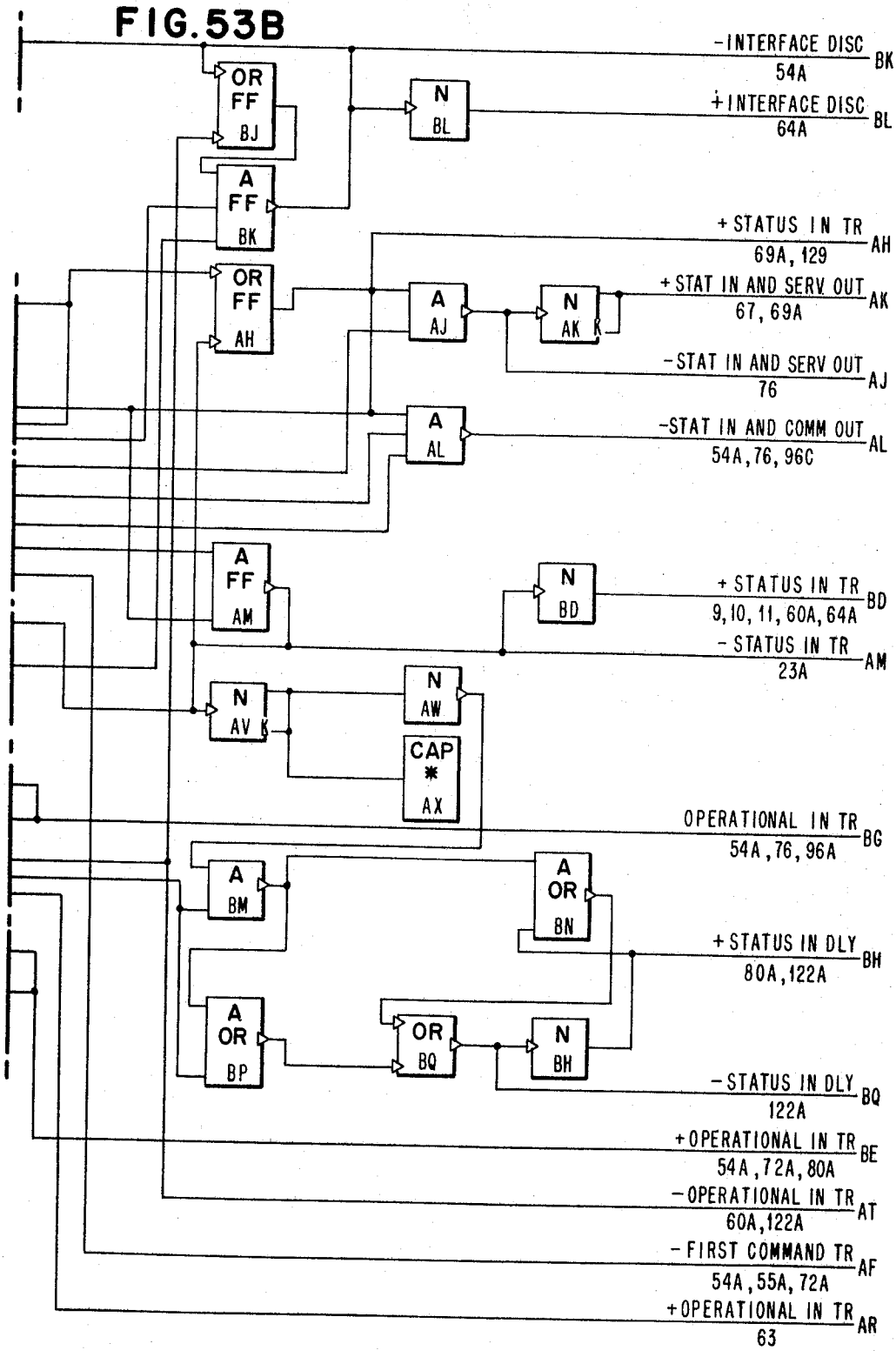

OR FF BJ
N BL
A FF BK
OR FF AH
A AJ
N AK
A AL
A FF AM
N BD
N AV
N AW
CAP * AX
A BM
A OR BN
A OR BP
OR BQ
N BH
-INTERFACE DISC 54A BK
+INTERFACE DISC 64A BL
+STATUS IN TR 69A, 129 AH
+STAT IN AND SERV OUT 67, 69A AK
-STAT IN AND SERV OUT 76 AJ
-STAT IN AND COMM OUT 54A, 76, 96C AL
+STATUS IN TR 9, 10, 11, 60A, 64A BD
-STATUS IN TR 23A AM
OPERATIONAL IN TR 54A, 76, 96A BG
+STATUS IN DLY 80A, 122A BH
-STATUS IN DLY 122A BQ
+OPERATIONAL IN TR 54A, 72A, 80A BE
-OPERATIONAL IN TR 60A, 122A AT
-FIRST COMMAND TR 54A, 55A, 72A AF
+OPERATIONAL IN TR 63 AR

FIG. 54A

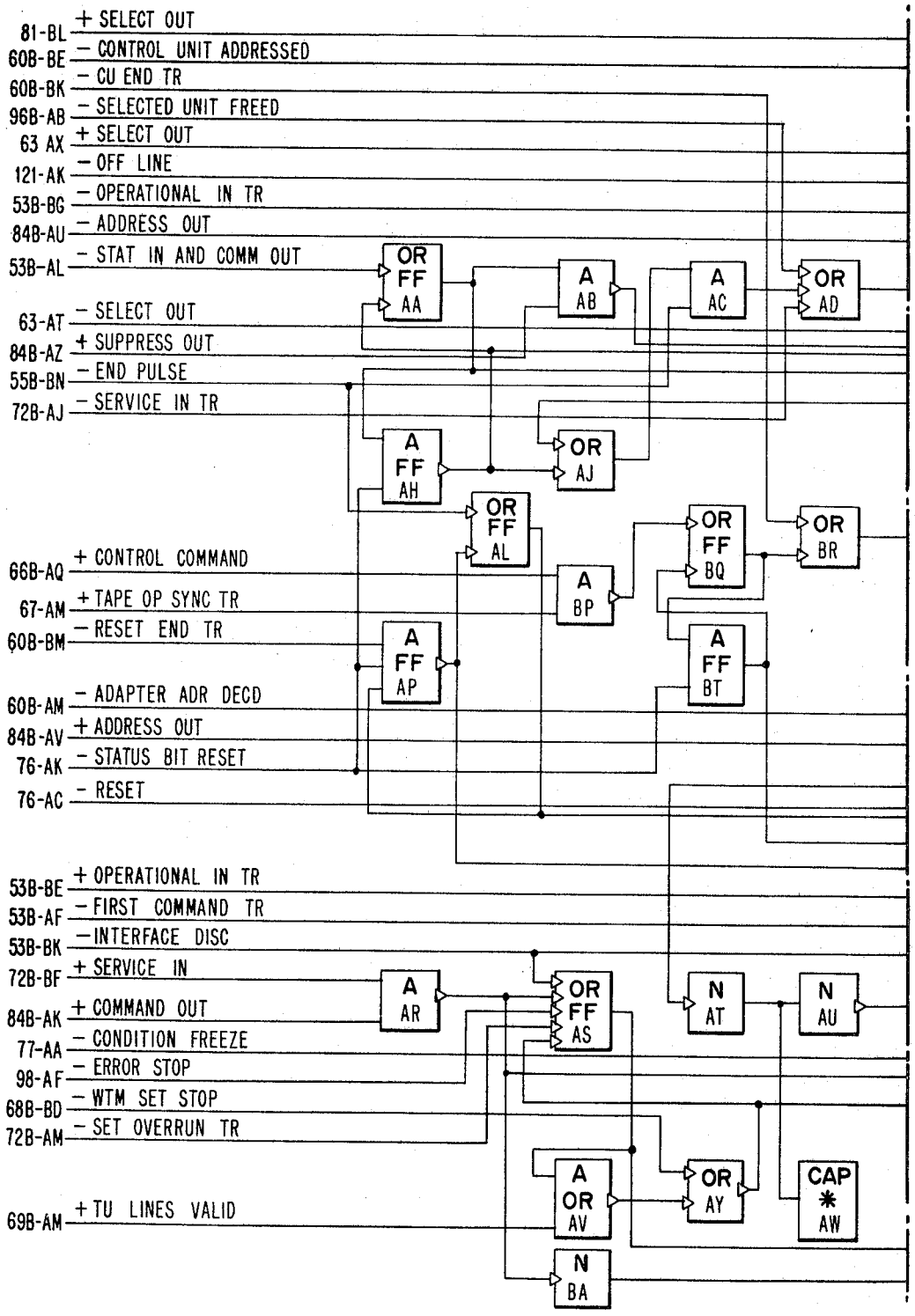

81-BL + SELECT OUT
60B-BE − CONTROL UNIT ADDRESSED
60B-BK − CU END TR
96B-AB − SELECTED UNIT FREED
63 AX + SELECT OUT
121-AK − OFF LINE
53B-BG − OPERATIONAL IN TR
84B-AU − ADDRESS OUT
53B-AL − STAT IN AND COMM OUT
63-AT − SELECT OUT
84B-AZ + SUPPRESS OUT
55B-BN − END PULSE
72B-AJ − SERVICE IN TR
66B-AQ + CONTROL COMMAND
67-AM + TAPE OP SYNC TR
60B-BM − RESET END TR
60B-AM − ADAPTER ADR DECD
84B-AV + ADDRESS OUT
76-AK − STATUS BIT RESET
76-AC − RESET
53B-BE + OPERATIONAL IN TR
53B-AF − FIRST COMMAND TR
53B-BK − INTERFACE DISC
72B-BF + SERVICE IN
84B-AK + COMMAND OUT
77-AA − CONDITION FREEZE
98-AF − ERROR STOP
68B-BD − WTM SET STOP
72B-AM − SET OVERRUN TR
69B-AM + TU LINES VALID
OR FF AA
A AB
A AC
OR AD
A FF AH
OR AJ
OR FF AL
A BP
OR FF BQ
OR BR
A FF AP
A FF BT
A AR
OR FF AS
N AT
N AU
A OR AV
OR AY
CAP * AW
N BA

FIG. 54B

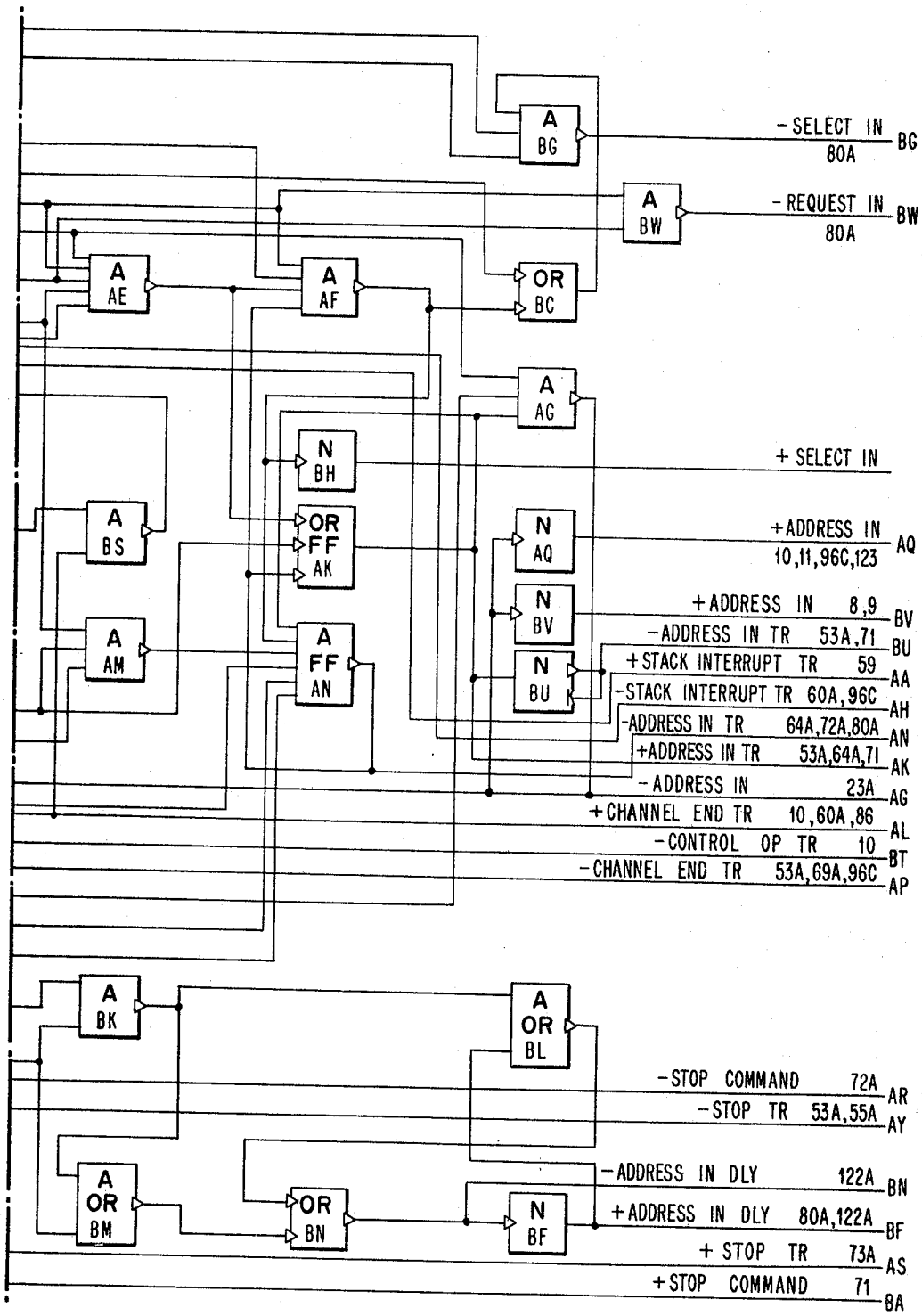

A BG
-SELECT IN 80A BG
A BW
-REQUEST IN 80A BW
A AE
A AF
OR BC
A AG
N BH
+SELECT IN
A BS
OR FF AK
N AQ
+ADDRESS IN 10,11,96C,123 AQ
N BV
+ADDRESS IN 8,9 BV
-ADDRESS IN TR 53A,71 BU
+STACK INTERRUPT TR 59 AA
-STACK INTERRUPT TR 60A,96C AH
-ADDRESS IN TR 64A,72A,80A AN
+ADDRESS IN TR 53A,64A,71 AK
-ADDRESS IN 23A AG
+CHANNEL END TR 10,60A,86 AL
-CONTROL OP TR 10 BT
-CHANNEL END TR 53A,69A,96C AP
A AM
A FF AN
N BU
A BK
A OR BL
-STOP COMMAND 72A AR
-STOP TR 53A,55A AY
A OR BM
OR BN
N BF
-ADDRESS IN DLY 122A BN
+ADDRESS IN DLY 80A,122A BF
+STOP TR 73A AS
+STOP COMMAND 71 BA

FIG. 55A

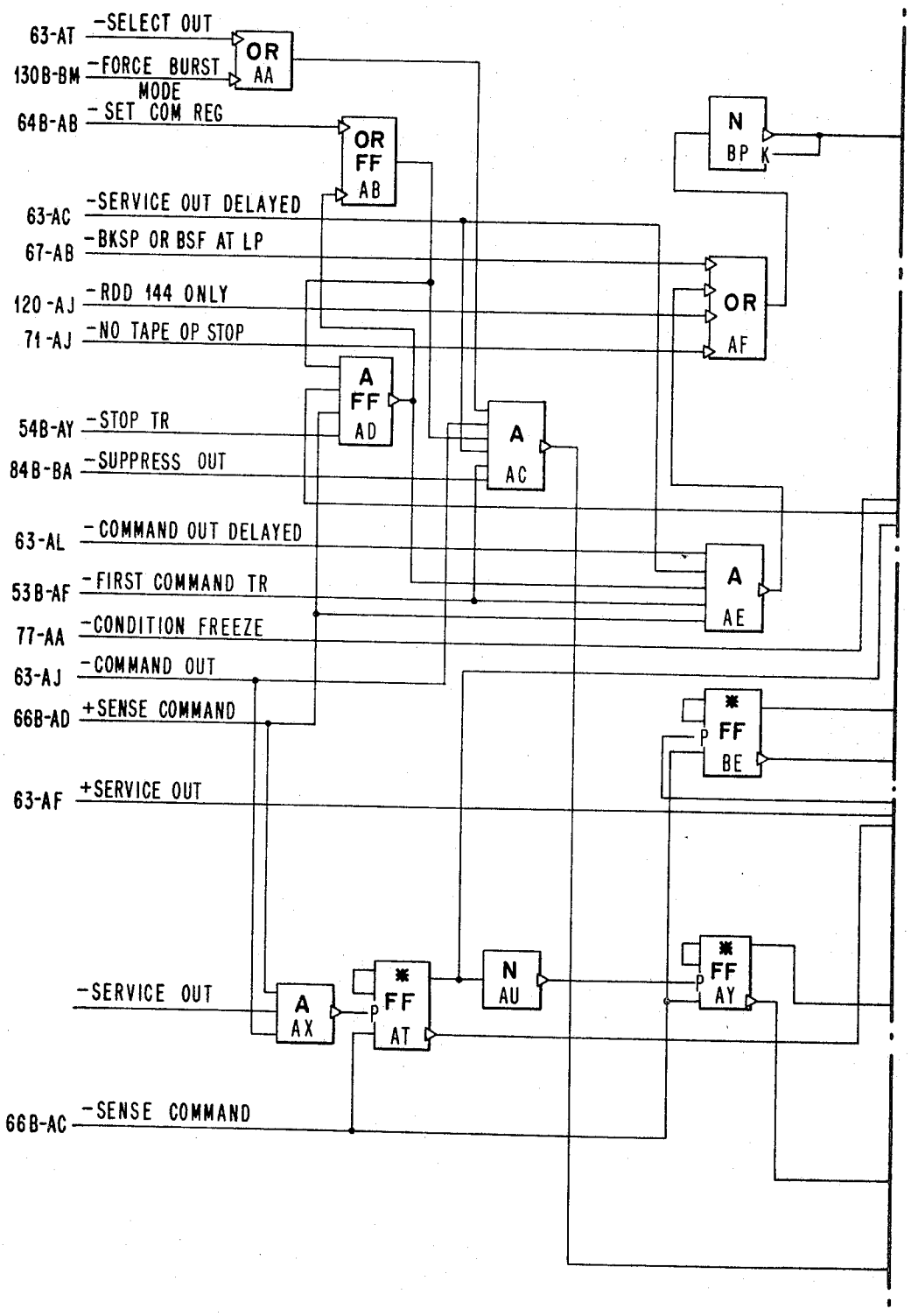

63-AT
-SELECT OUT
130B-BM
-FORCE BURST MODE
OR AA
64B-AB
-SET COM REG
OR FF AB
63-AC
-SERVICE OUT DELAYED
67-AB
-BKSP OR BSF AT LP
120-AJ
-RDD 144 ONLY
71-AJ
-NO TAPE OP STOP
A FF AD
54B-AY
-STOP TR
84B-BA
-SUPPRESS OUT
A AC
63-AL
-COMMAND OUT DELAYED
53B-AF
-FIRST COMMAND TR
77-AA
-CONDITION FREEZE
63-AJ
-COMMAND OUT
66B-AD
+SENSE COMMAND
63-AF
+SERVICE OUT
N BP K
OR AF
A AE
* FF BE P
-SERVICE OUT
A AX
* FF AT P
N AU
* FF AY P
66B-AC
-SENSE COMMAND

FIG. 55B

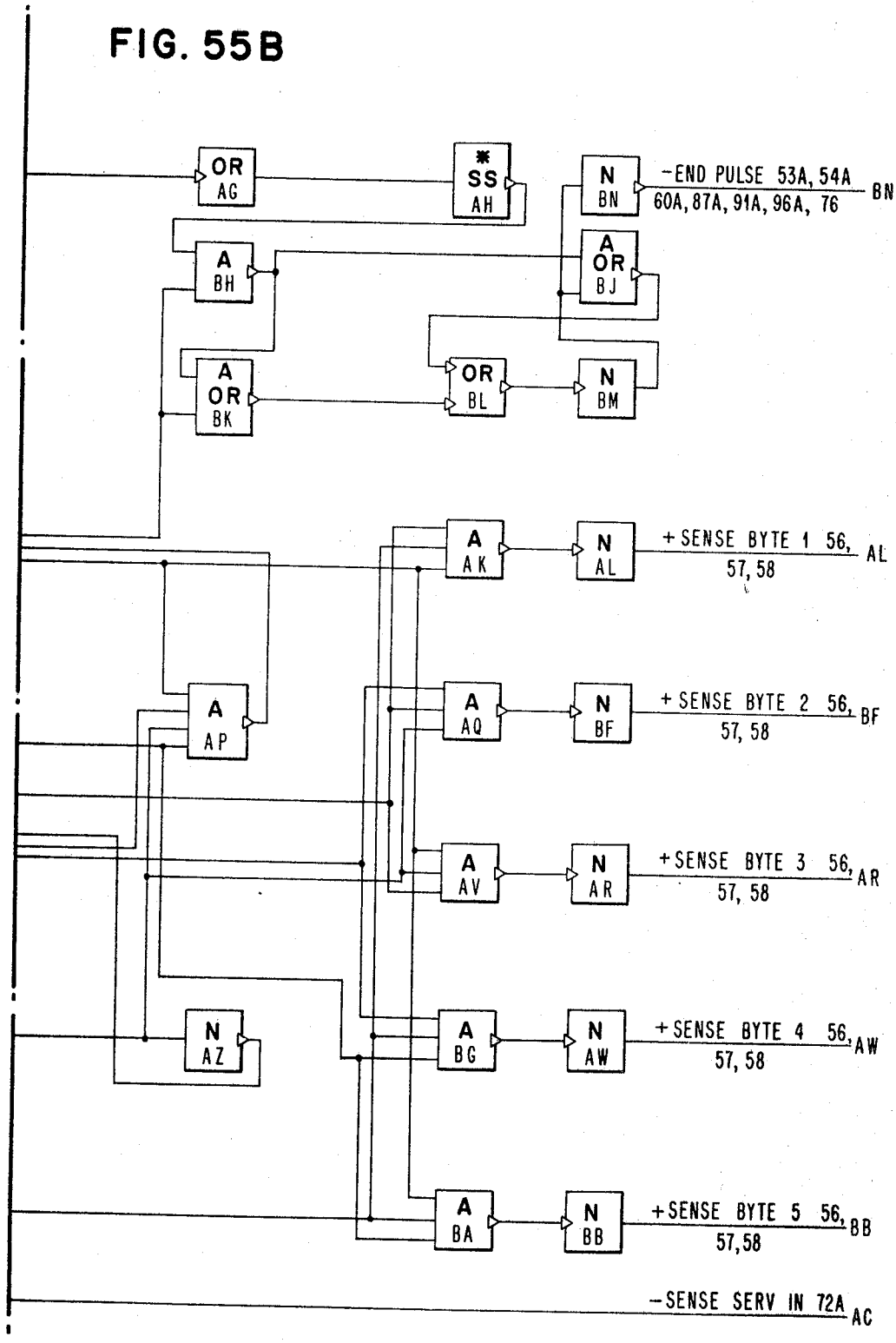

OR AG
* SS AH
N BN
-END PULSE 53A, 54A 60A, 87A, 91A, 96A, 76
BN
A BH
A OR BJ
A OR BK
OR BL
N BM
A AK
N AL
+SENSE BYTE 1 56, 57, 58
AL
A AP
A AQ
N BF
+SENSE BYTE 2 56, 57, 58
BF
A AV
N AR
+SENSE BYTE 3 56, 57, 58
AR
N AZ
A BG
N AW
+SENSE BYTE 4 56, 57, 58
AW
A BA
N BB
+SENSE BYTE 5 56, 57, 58
BB
-SENSE SERV IN 72A
AC

FIG. 56

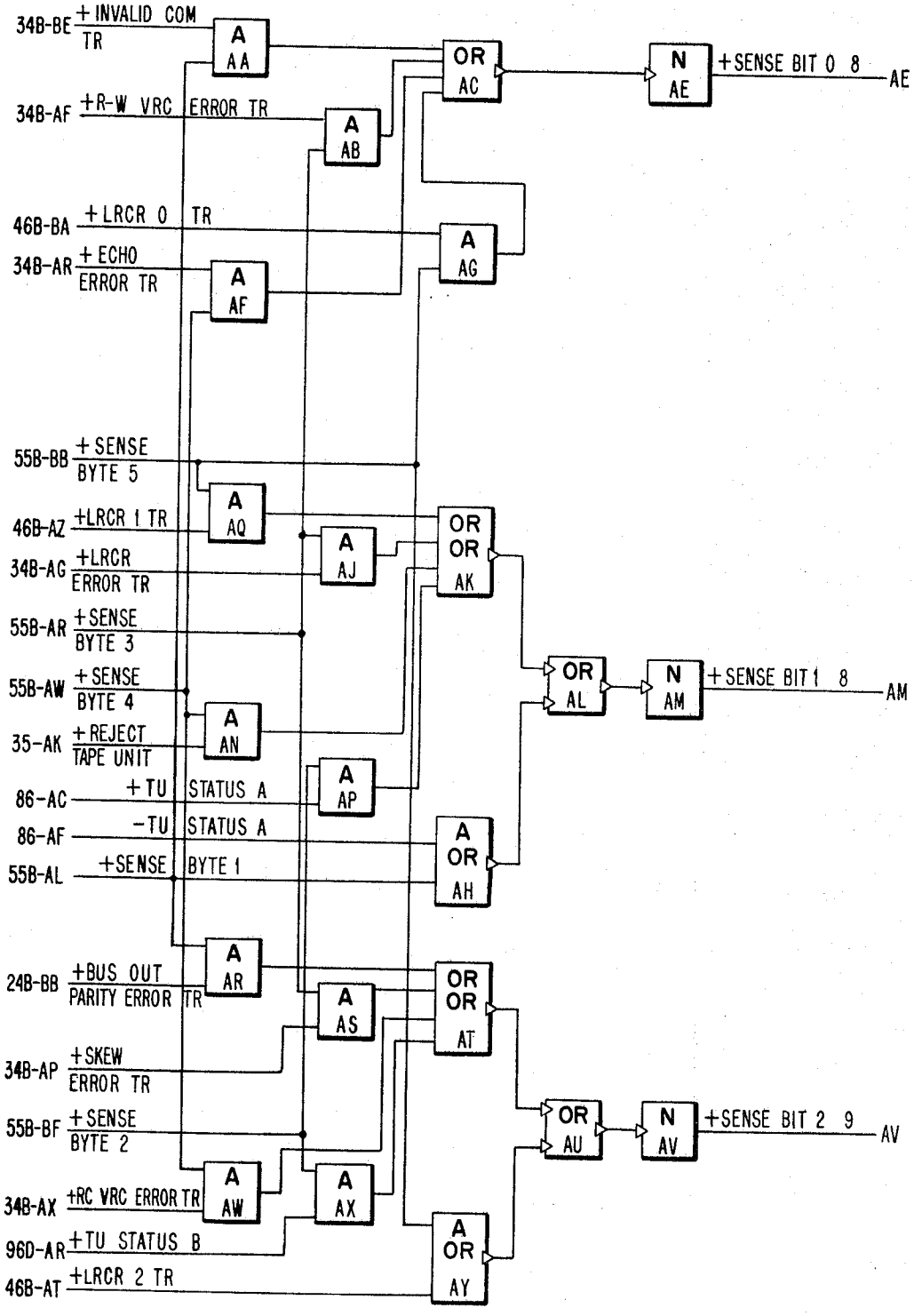

34B-BE +INVALID COM TR
A AA
34B-AF +R-W VRC ERROR TR
A AB
OR AC
N AE
+SENSE BIT 0 8
AE
46B-BA +LRCR 0 TR
34B-AR +ECHO ERROR TR
A AF
A AG
55B-BB +SENSE BYTE 5
A AQ
46B-AZ +LRCR 1 TR
34B-AG +LRCR ERROR TR
A AJ
OR OR AK
55B-AR +SENSE BYTE 3
55B-AW +SENSE BYTE 4
35-AK +REJECT TAPE UNIT
A AN
OR AL
N AM
+SENSE BIT 1 8
AM
86-AC +TU STATUS A
86-AF -TU STATUS A
55B-AL +SENSE BYTE 1
A AP
A OR AH
24B-BB +BUS OUT PARITY ERROR TR
A AR
A AS
OR OR AT
34B-AP +SKEW ERROR TR
55B-BF +SENSE BYTE 2
OR AU
N AV
+SENSE BIT 2 9
AV
34B-AX +RC VRC ERROR TR
96D-AR +TU STATUS B
A AW
A AX
A OR AY
46B-AT +LRCR 2 TR

FIG. 57

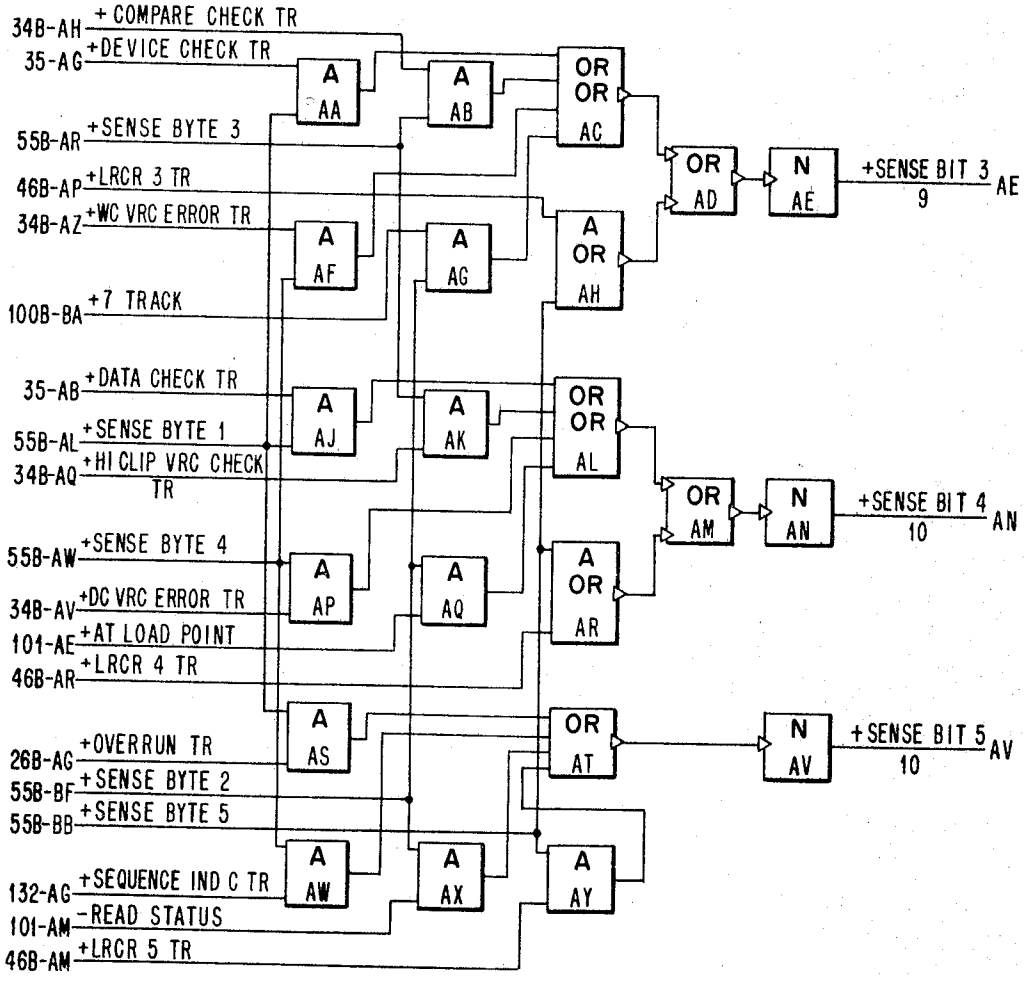

34B-AH +COMPARE CHECK TR
35-AG +DEVICE CHECK TR
55B-AR +SENSE BYTE 3
46B-AP +LRCR 3 TR
34B-AZ +WC VRC ERROR TR
100B-BA +7 TRACK
35-AB +DATA CHECK TR
55B-AL +SENSE BYTE 1
34B-AQ +HI CLIP VRC CHECK TR
55B-AW +SENSE BYTE 4
34B-AV +DC VRC ERROR TR
101-AE +AT LOAD POINT
46B-AR +LRCR 4 TR
26B-AG +OVERRUN TR
55B-BF +SENSE BYTE 2
55B-BB +SENSE BYTE 5
132-AG +SEQUENCE IND C TR
101-AM -READ STATUS
46B-AM +LRCR 5 TR
A AA
A AB
OR OR AC
A AF
A AG
A OR AH
A AJ
A AK
OR OR AL
A AP
A AQ
A OR AR
A AS
OR AT
A AW
A AX
A AY
OR AD
OR AM
N AE
N AN
N AV
+SENSE BIT 3 9 AE
+SENSE BIT 4 10 AN
+SENSE BIT 5 10 AV

FIG. 58

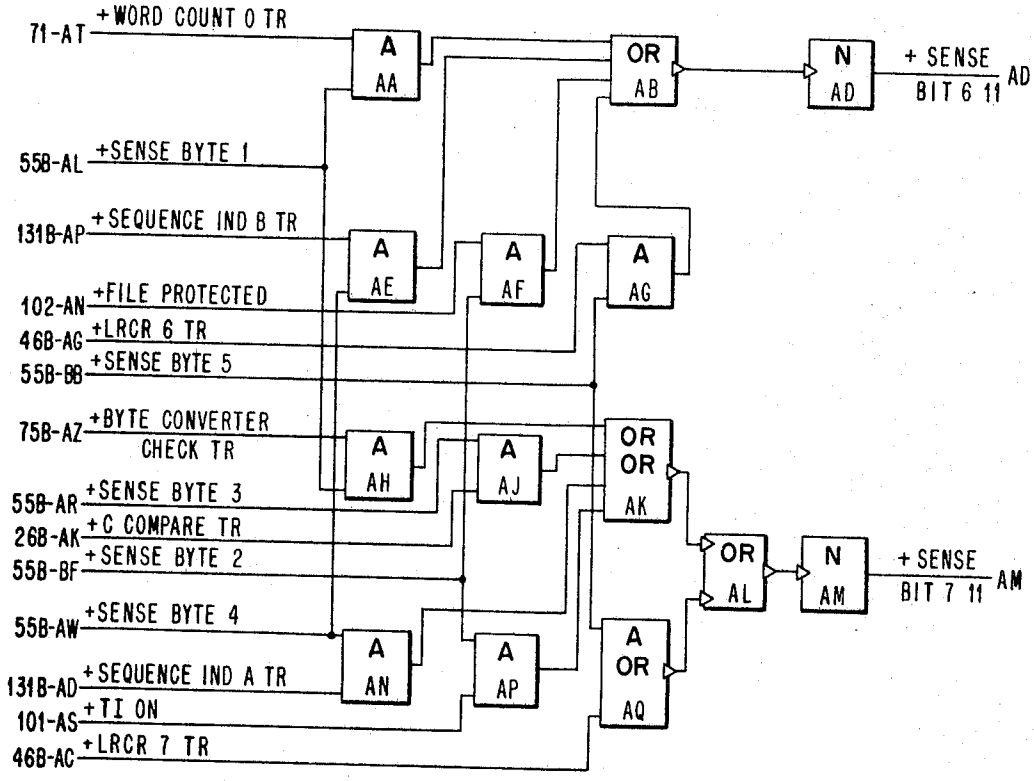

71-AT +WORD COUNT 0 TR
55B-AL +SENSE BYTE 1
131B-AP +SEQUENCE IND B TR
102-AN +FILE PROTECTED
46B-AG +LRCR 6 TR
55B-BB +SENSE BYTE 5
75B-AZ +BYTE CONVERTER CHECK TR
55B-AR +SENSE BYTE 3
26B-AK +C COMPARE TR
55B-BF +SENSE BYTE 2
55B-AW +SENSE BYTE 4
131B-AD +SEQUENCE IND A TR
101-AS +TI ON
46B-AC +LRCR 7 TR
A AA
A AE
A AF
A AG
A AH
A AJ
A AN
A AP
OR AB
OR OR AK
A OR AQ
OR AL
N AD
+ SENSE BIT 6 11 AD
N AM
+ SENSE BIT 7 11 AM

FIG. 59

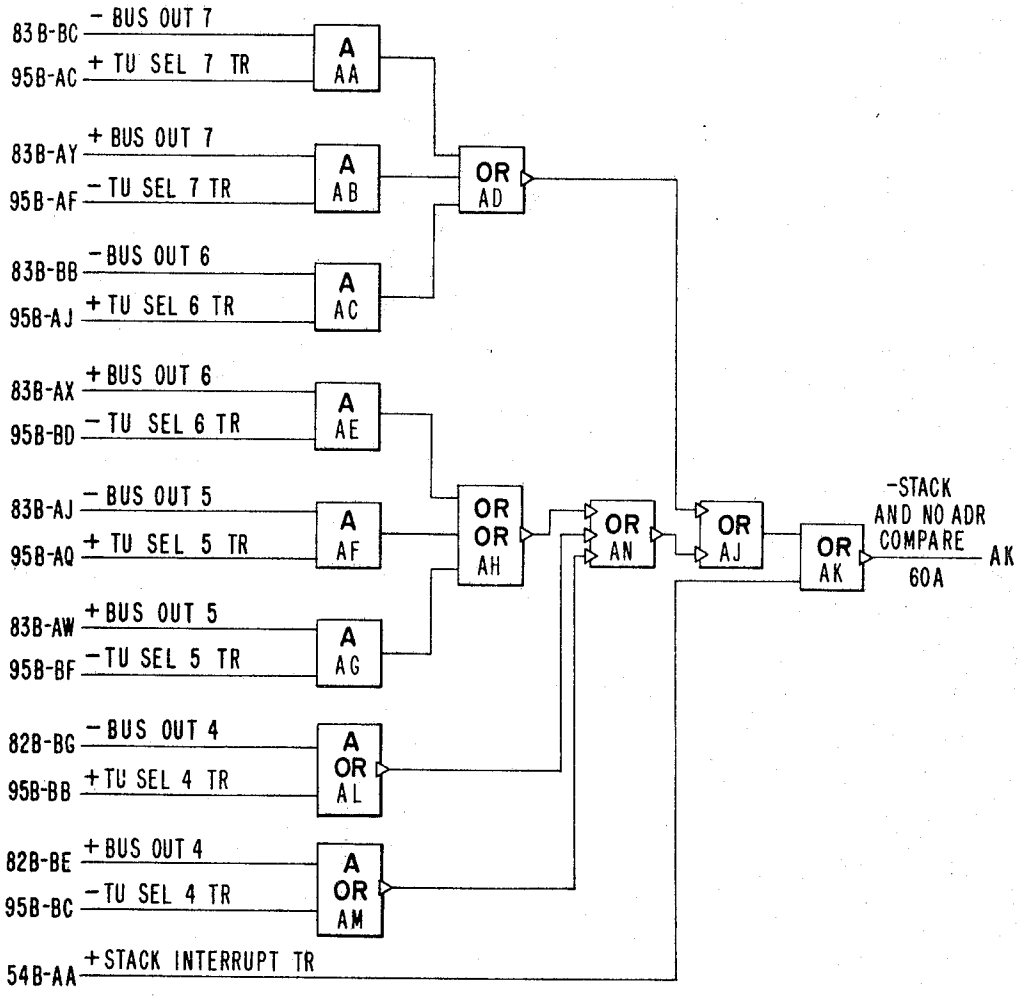

83B-BC
- BUS OUT 7
95B-AC
+ TU SEL 7 TR
A AA
83B-AY
+ BUS OUT 7
95B-AF
- TU SEL 7 TR
A AB
OR AD
83B-BB
- BUS OUT 6
95B-AJ
+ TU SEL 6 TR
A AC
83B-AX
+ BUS OUT 6
95B-BD
- TU SEL 6 TR
A AE
83B-AJ
- BUS OUT 5
95B-AQ
+ TU SEL 5 TR
A AF
OR OR AH
83B-AW
+ BUS OUT 5
95B-BF
- TU SEL 5 TR
A AG
82B-BG
- BUS OUT 4
95B-BB
+ TU SEL 4 TR
A OR AL
82B-BE
+ BUS OUT 4
95B-BC
- TU SEL 4 TR
A OR AM
54B-AA
+ STACK INTERRUPT TR
OR AN
OR AJ
OR AK
- STACK AND NO ADR COMPARE
60A
AK

FIG. 60A

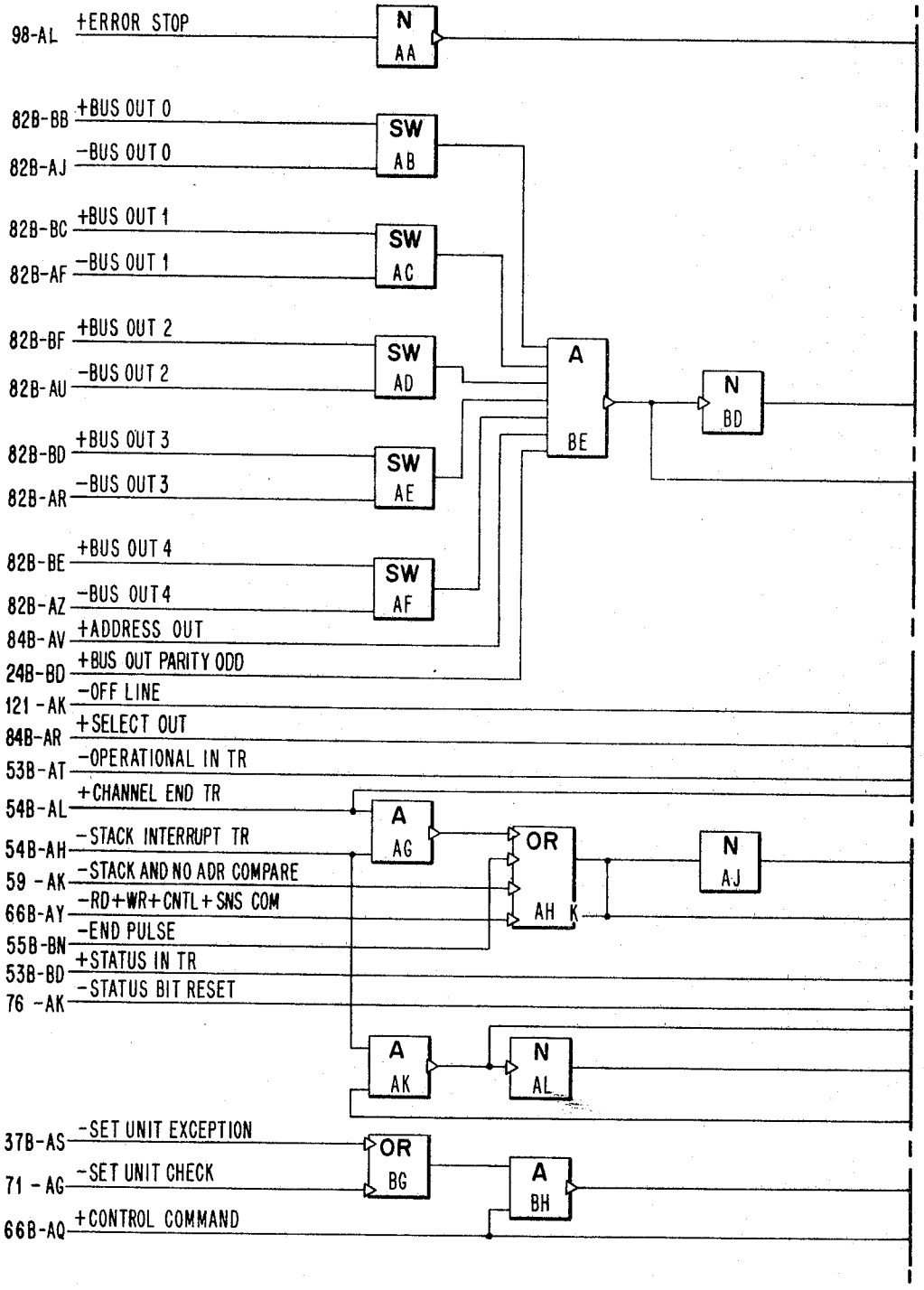

98-AL +ERROR STOP
N AA
82B-BB +BUS OUT 0
82B-AJ -BUS OUT 0
SW AB
82B-BC +BUS OUT 1
82B-AF -BUS OUT 1
SW AC
82B-BF +BUS OUT 2
82B-AU -BUS OUT 2
SW AD
82B-BD +BUS OUT 3
82B-AR -BUS OUT 3
SW AE
82B-BE +BUS OUT 4
82B-AZ -BUS OUT 4
SW AF
84B-AV +ADDRESS OUT
24B-BD +BUS OUT PARITY ODD
A BE
N BD
121 -AK -OFF LINE
84B-AR +SELECT OUT
53B-AT -OPERATIONAL IN TR
54B-AL +CHANNEL END TR
54B-AH -STACK INTERRUPT TR
59 -AK -STACK AND NO ADR COMPARE
66B-AY -RD+WR+CNTL+SNS COM
55B-BN -END PULSE
53B-BD +STATUS IN TR
76 -AK -STATUS BIT RESET
A AG
OR AH K
N AJ
A AK
N AL
37B-AS -SET UNIT EXCEPTION
71 -AG -SET UNIT CHECK
OR BG
66B-AQ +CONTROL COMMAND
A BH

FIG. 60B

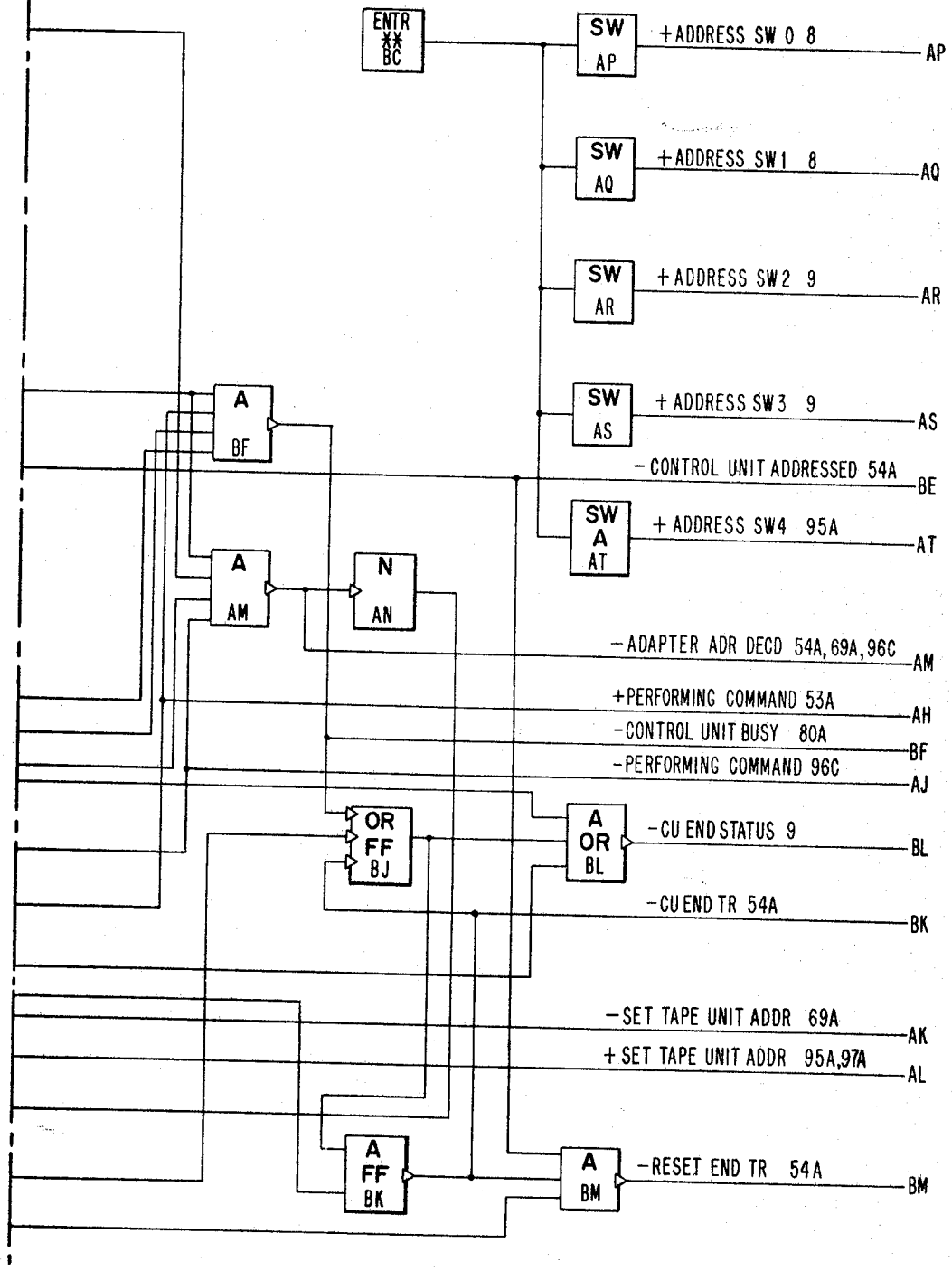

ENTR ** BC
SW AP
+ADDRESS SW 0 8
AP
SW AQ
+ADDRESS SW1 8
AQ
SW AR
+ADDRESS SW2 9
AR
SW AS
+ADDRESS SW3 9
AS
-CONTROL UNIT ADDRESSED 54A
BE
SW A AT
+ADDRESS SW4 95A
AT
A BF
A AM
N AN
-ADAPTER ADR DECD 54A,69A,96C
AM
+PERFORMING COMMAND 53A
AH
-CONTROL UNIT BUSY 80A
BF
-PERFORMING COMMAND 96C
AJ
OR FF BJ
A OR BL
-CU END STATUS 9
BL
-CU END TR 54A
BK
-SET TAPE UNIT ADDR 69A
AK
+SET TAPE UNIT ADDR 95A,97A
AL
A FF BK
A BM
-RESET END TR 54A
BM

FIG. 61A

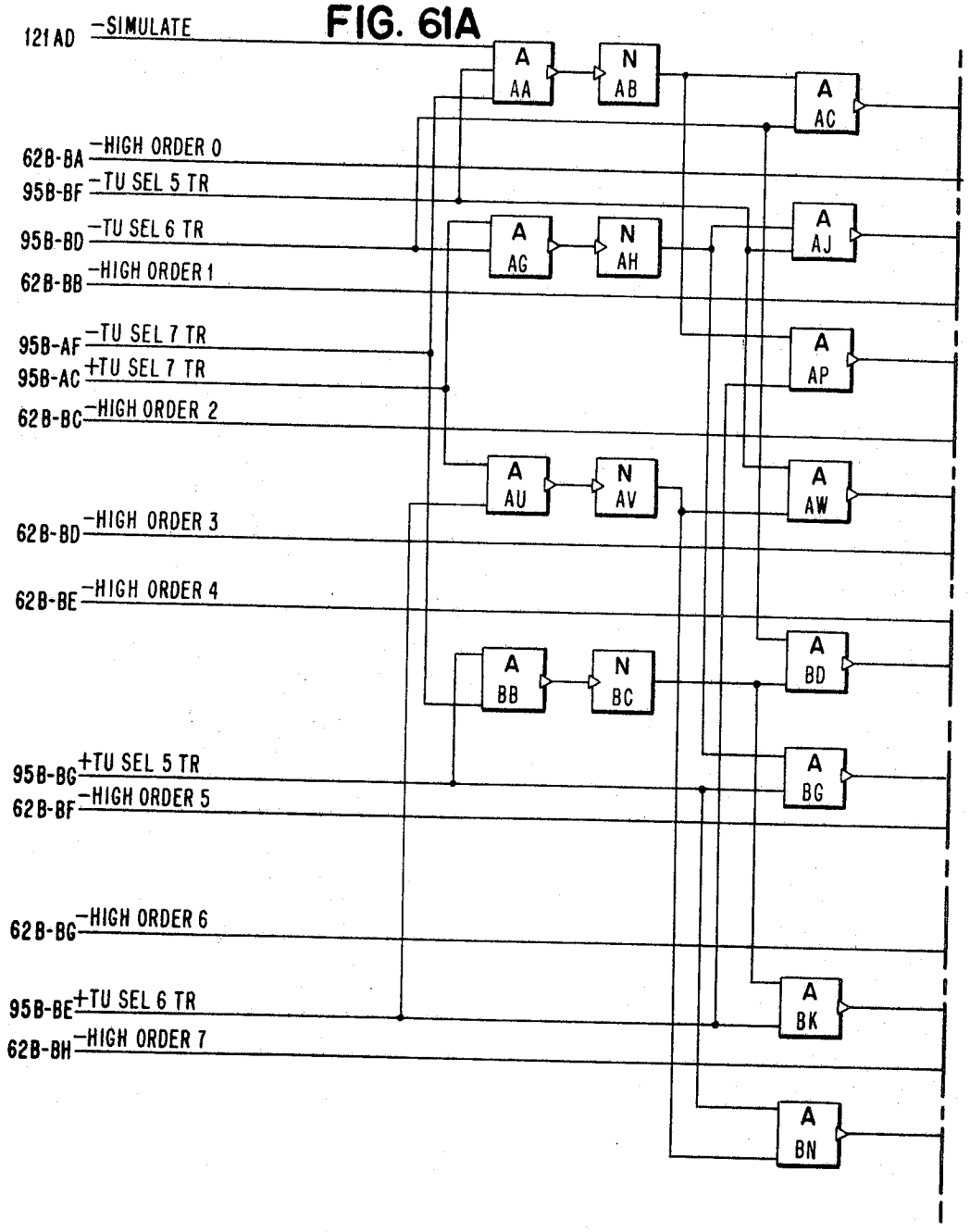

121AD -SIMULATE
62B-BA -HIGH ORDER 0
95B-BF -TU SEL 5 TR
95B-BD -TU SEL 6 TR
62B-BB -HIGH ORDER 1
95B-AF -TU SEL 7 TR
95B-AC +TU SEL 7 TR
62B-BC -HIGH ORDER 2
62B-BD -HIGH ORDER 3
62B-BE -HIGH ORDER 4
95B-BG +TU SEL 5 TR
62B-BF -HIGH ORDER 5
62B-BG -HIGH ORDER 6
95B-BE +TU SEL 6 TR
62B-BH -HIGH ORDER 7
A AA
N AB
A AC
A AG
N AH
A AJ
A AP
A AU
N AV
A AW
A BB
N BC
A BD
A BG
A BK
A BN

FIG. 61B

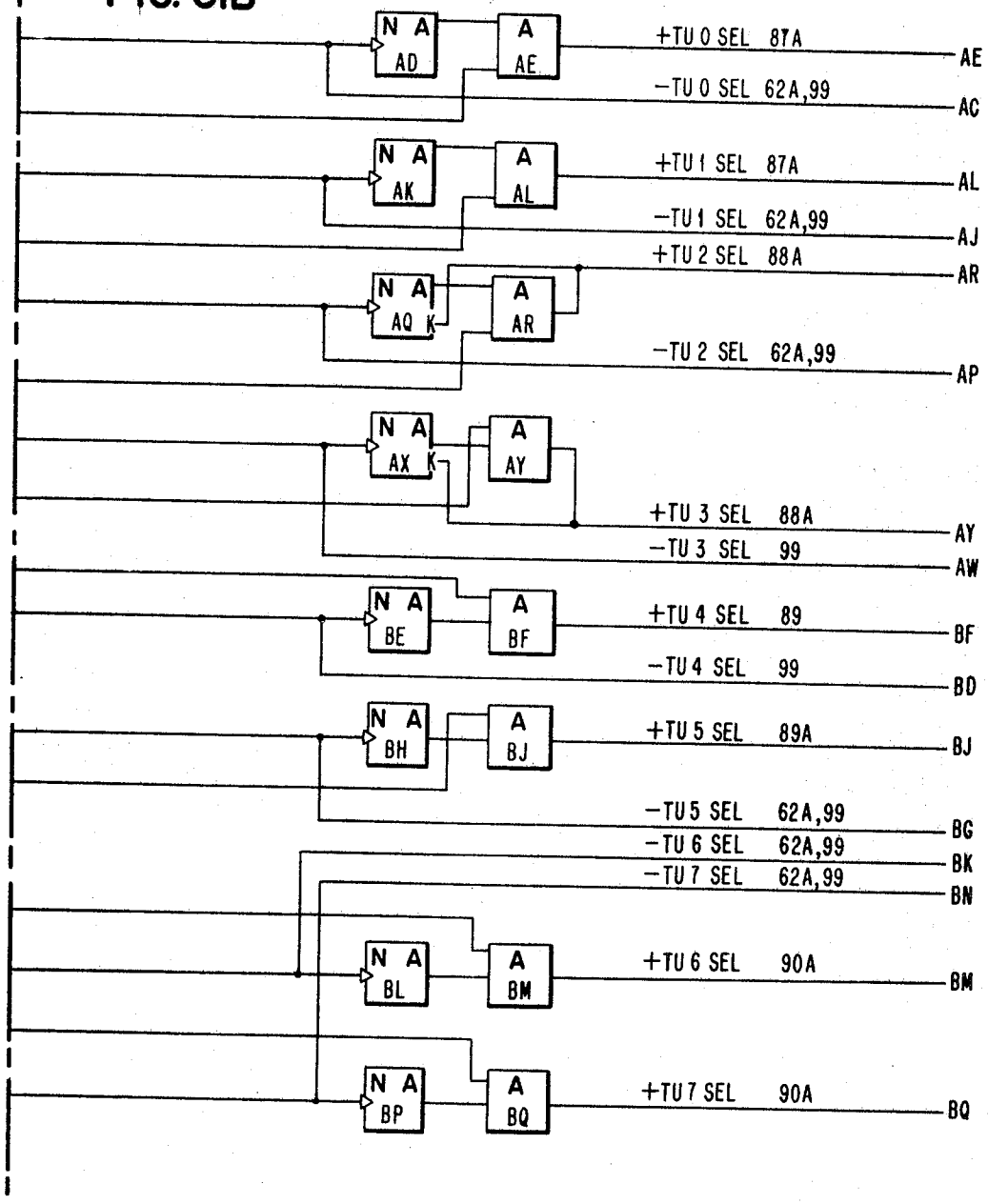
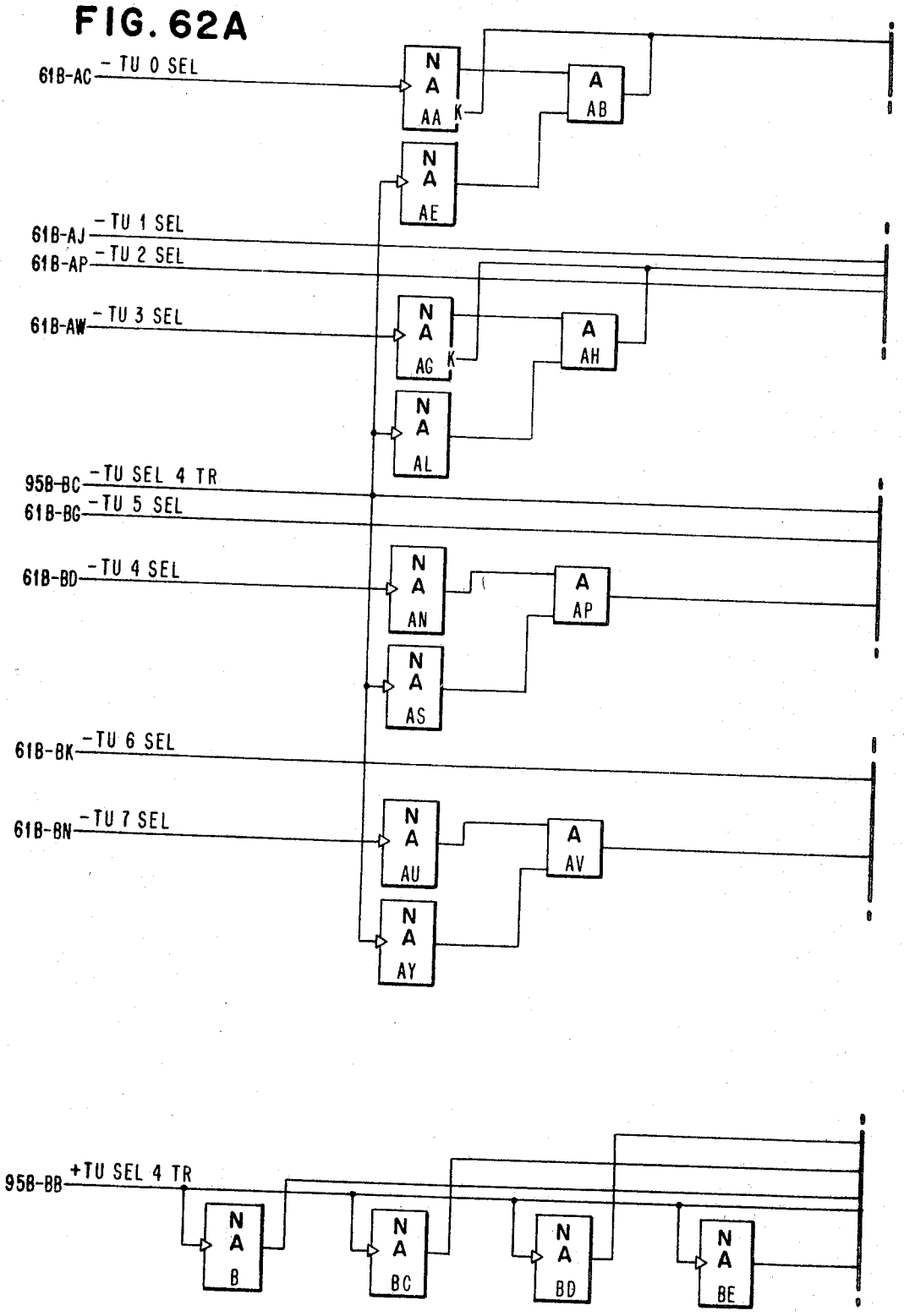

N A AD
A AE
+TU 0 SEL 87A
AE
−TU 0 SEL 62A,99
AC
N A AK
A AL
+TU 1 SEL 87A
AL
−TU 1 SEL 62A,99
AJ
N A AQ K
A AR
+TU 2 SEL 88A
AR
−TU 2 SEL 62A,99
AP
N A AX K
A AY
+TU 3 SEL 88A
AY
−TU 3 SEL 99
AW
N A BE
A BF
+TU 4 SEL 89
BF
−TU 4 SEL 99
BD
N A BH
A BJ
+TU 5 SEL 89A
BJ
−TU 5 SEL 62A,99
BG
−TU 6 SEL 62A,99
BK
−TU 7 SEL 62A,99
BN
N A BL
A BM
+TU 6 SEL 90A
BM
N A BP
A BQ
+TU 7 SEL 90A
BQ

FIG. 62A


61B-AC -TU 0 SEL
NA AA K
A AB
NA AE
61B-AJ -TU 1 SEL
61B-AP -TU 2 SEL
61B-AW -TU 3 SEL
NA AG K
A AH
NA AL
95B-BC -TU SEL 4 TR
61B-BG -TU 5 SEL
61B-BD -TU 4 SEL
NA AN
A AP
NA AS
61B-BK -TU 6 SEL
61B-BN -TU 7 SEL
NA AU
A AV
NA AY
95B-BB +TU SEL 4 TR
NA B
NA BC
NA BD
NA BE

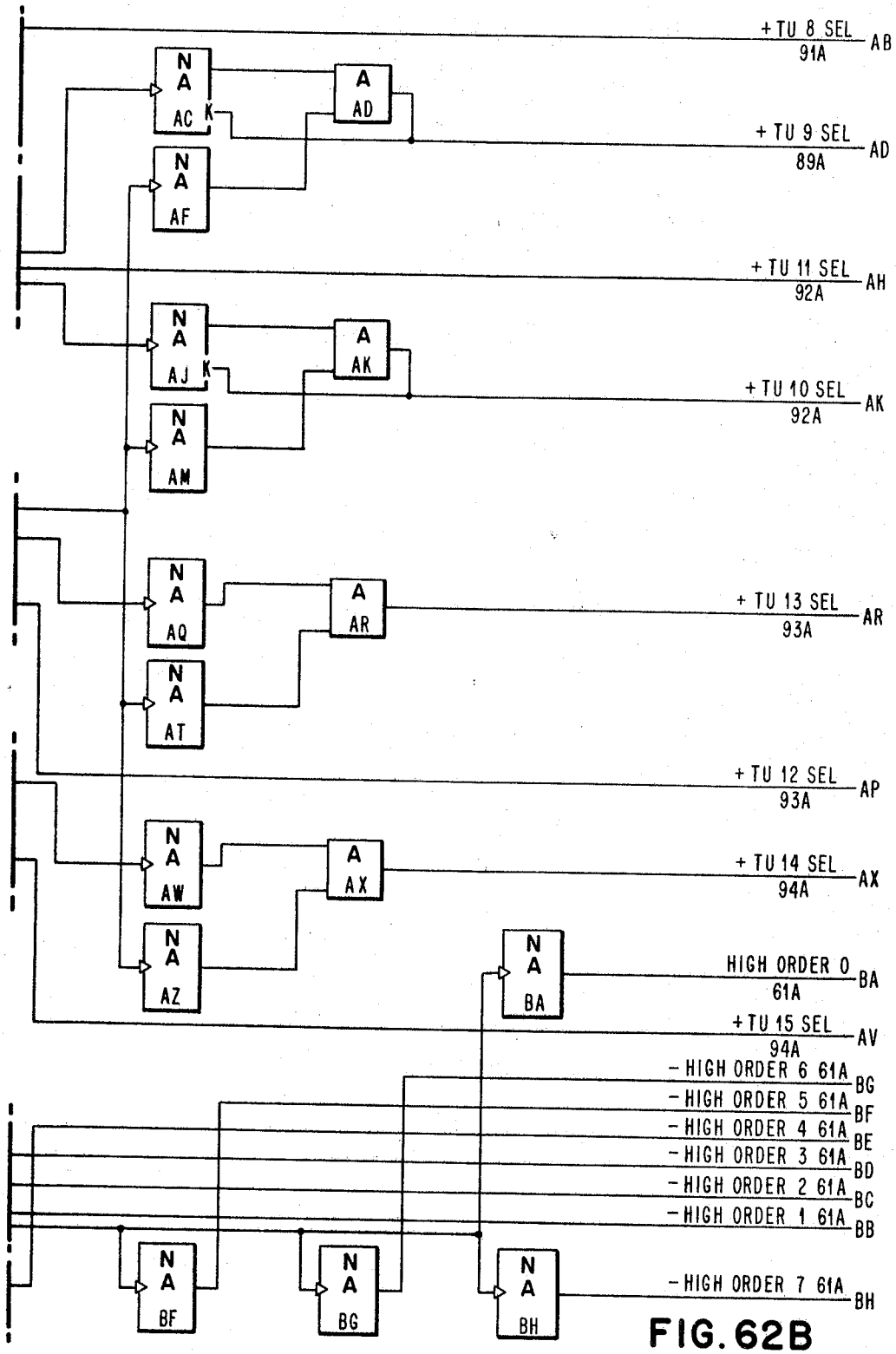

+TU 8 SEL
91A
AB
+TU 9 SEL
89A
AD
+TU 11 SEL
92A
AH
+TU 10 SEL
92A
AK
+TU 13 SEL
93A
AR
+TU 12 SEL
93A
AP
+TU 14 SEL
94A
AX
HIGH ORDER 0
61A
BA
+TU 15 SEL
94A
AV
-HIGH ORDER 6 61A
BG
-HIGH ORDER 5 61A
BF
-HIGH ORDER 4 61A
BE
-HIGH ORDER 3 61A
BD
-HIGH ORDER 2 61A
BC
-HIGH ORDER 1 61A
BB
-HIGH ORDER 7 61A
BH
NA AC
A AD
NA AF
NA AJ
A AK
NA AM
NA AQ
A AR
NA AT
NA AW
A AX
NA AZ
NA BA
NA BF
NA BG
NA BH

FIG. 62B

FIG.63

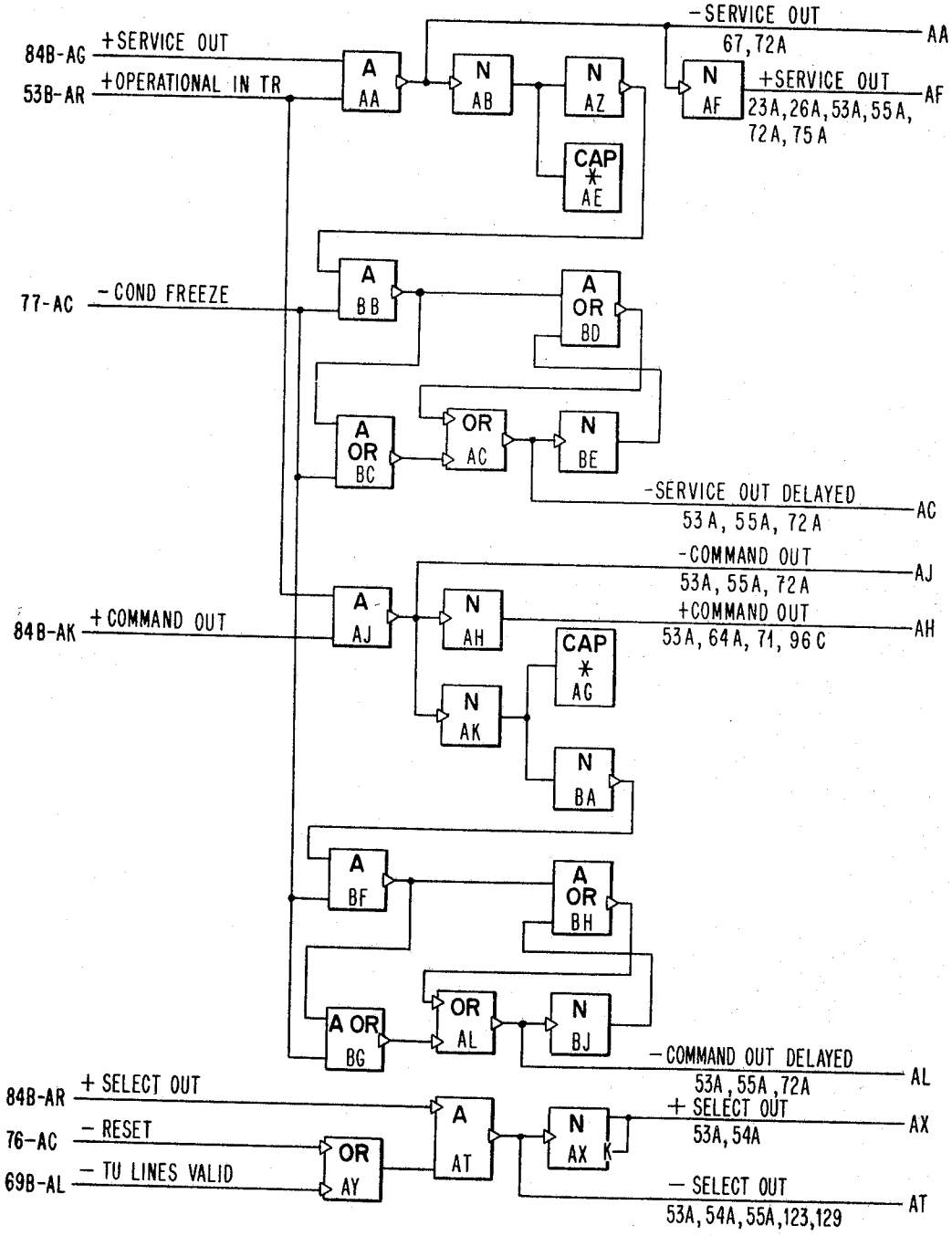

84B-AG +SERVICE OUT
53B-AR +OPERATIONAL IN TR
A AA
N AB
N AZ
CAP * AE
N AF
-SERVICE OUT 67,72A
AA
+SERVICE OUT 23A,26A,53A,55A, 72A,75A
AF
77-AC -COND FREEZE
A BB
A OR BD
A OR BC
OR AC
N BE
-SERVICE OUT DELAYED 53A,55A,72A
AC
-COMMAND OUT 53A,55A,72A
AJ
84B-AK +COMMAND OUT
A AJ
N AH
+COMMAND OUT 53A,64A,71,96C
AH
CAP * AG
N AK
N BA
A BF
A OR BH
A OR BG
OR AL
N BJ
-COMMAND OUT DELAYED 53A,55A,72A
AL
84B-AR +SELECT OUT
76-AC -RESET
69B-AL -TU LINES VALID
OR AY
A AT
N AX K
+SELECT OUT 53A,54A
AX
-SELECT OUT 53A,54A,55A,123,129
AT

FIG. 64A

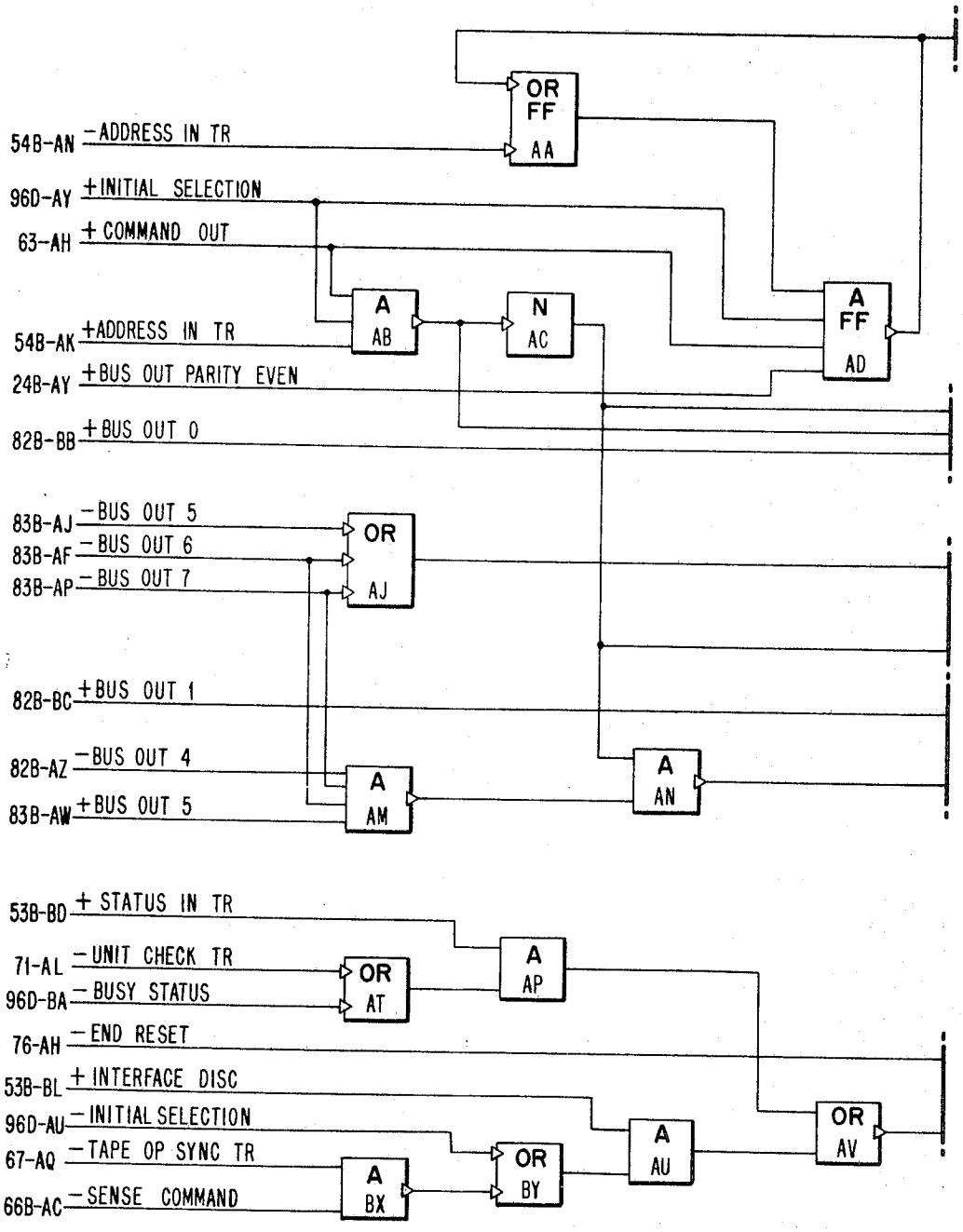

54B-AN -ADDRESS IN TR
96D-AY +INITIAL SELECTION
63-AH +COMMAND OUT
54B-AK +ADDRESS IN TR
24B-AY +BUS OUT PARITY EVEN
82B-BB +BUS OUT 0
83B-AJ -BUS OUT 5
83B-AF -BUS OUT 6
83B-AP -BUS OUT 7
82B-BC +BUS OUT 1
82B-AZ -BUS OUT 4
83B-AW +BUS OUT 5
53B-BD +STATUS IN TR
71-AL -UNIT CHECK TR
96D-BA -BUSY STATUS
76-AH -END RESET
53B-BL +INTERFACE DISC
96D-AU -INITIAL SELECTION
67-AQ -TAPE OP SYNC TR
66B-AC -SENSE COMMAND
OR FF AA
A AB
N AC
A FF AD
OR AJ
A AM
A AN
A AP
OR AT
OR BY
A AU
A BX
OR AV

FIG. 64B

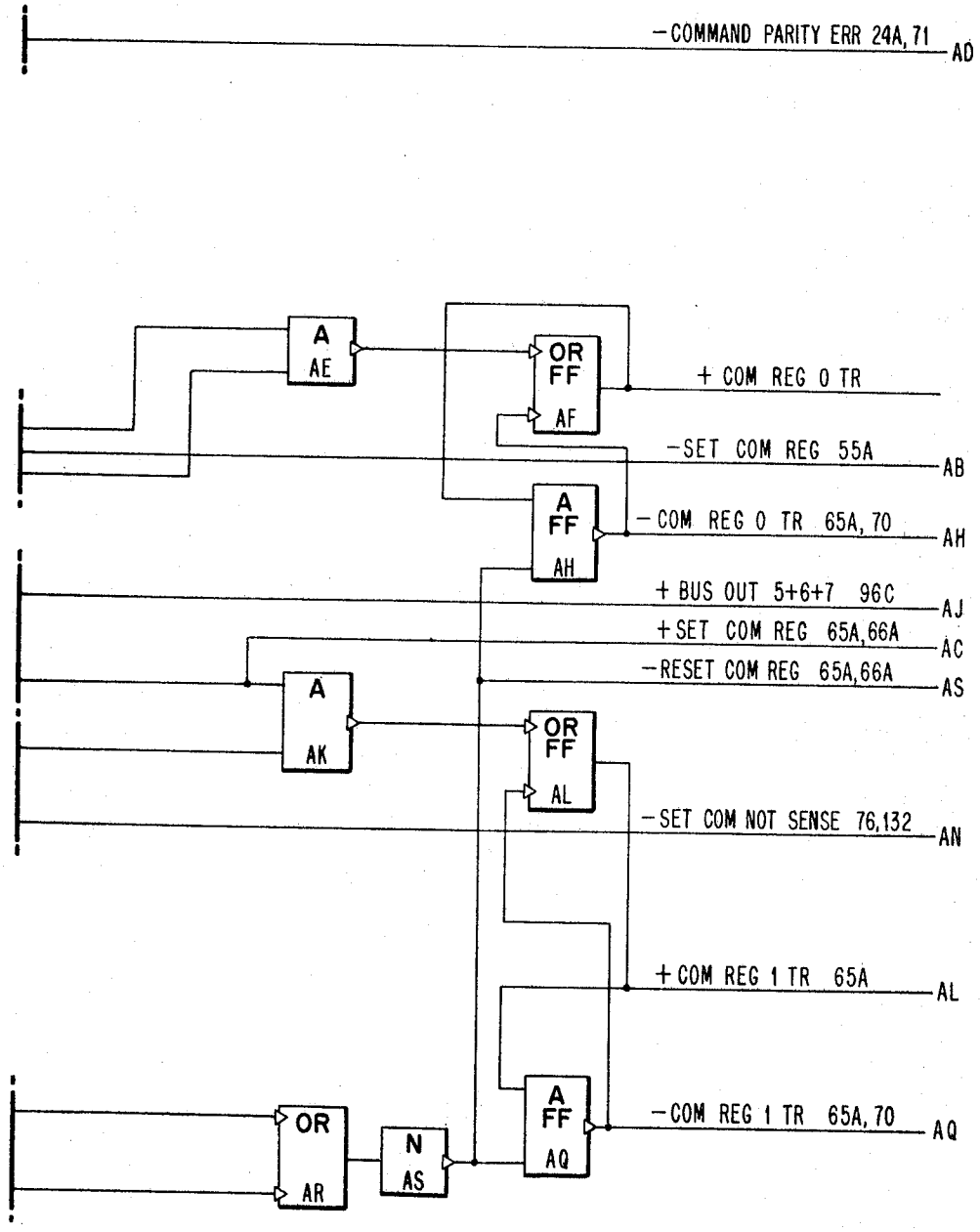

-COMMAND PARITY ERR 24A, 71
AD
A
AE
OR
FF
AF
+ COM REG 0 TR
-SET COM REG 55A
AB
A
FF
AH
- COM REG 0 TR 65A, 70
AH
+ BUS OUT 5+6+7 96C
AJ
+SET COM REG 65A, 66A
AC
-RESET COM REG 65A, 66A
AS
A
AK
OR
FF
AL
-SET COM NOT SENSE 76, 132
AN
+COM REG 1 TR 65A
AL
OR
AR
N
AS
A
FF
AQ
-COM REG 1 TR 65A, 70
AQ

FIG. 65A

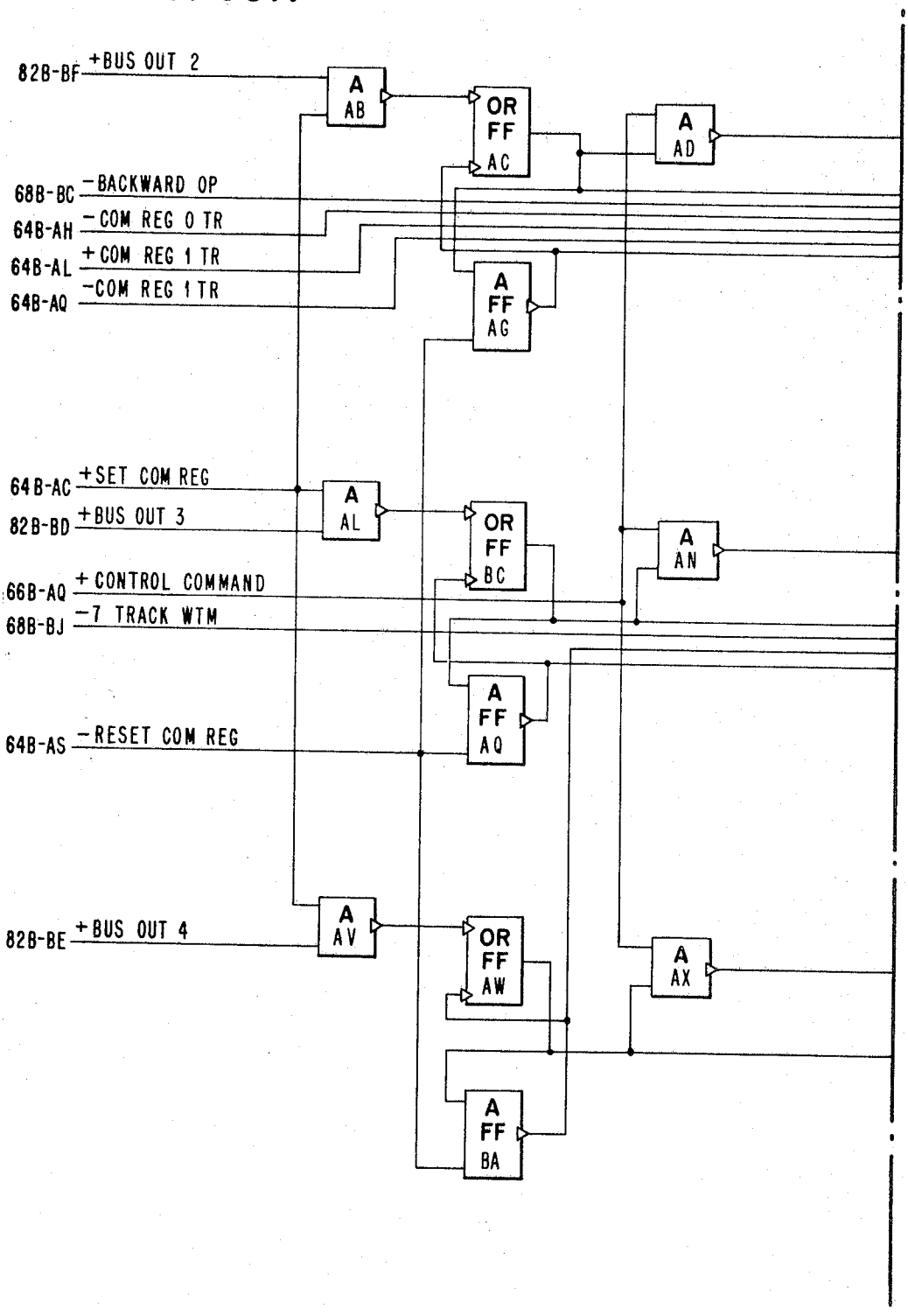

82B-BF +BUS OUT 2
A AB
OR FF AC
A AD
68B-BC -BACKWARD OP
64B-AH -COM REG 0 TR
64B-AL +COM REG 1 TR
64B-AQ -COM REG 1 TR
A FF AG
64B-AC +SET COM REG
82B-BD +BUS OUT 3
A AL
OR FF BC
A AN
66B-AQ +CONTROL COMMAND
68B-BJ -7 TRACK WTM
A FF AQ
64B-AS -RESET COM REG
82B-BE +BUS OUT 4
A AV
OR FF AW
A AX
A FF BA

FIG. 65B

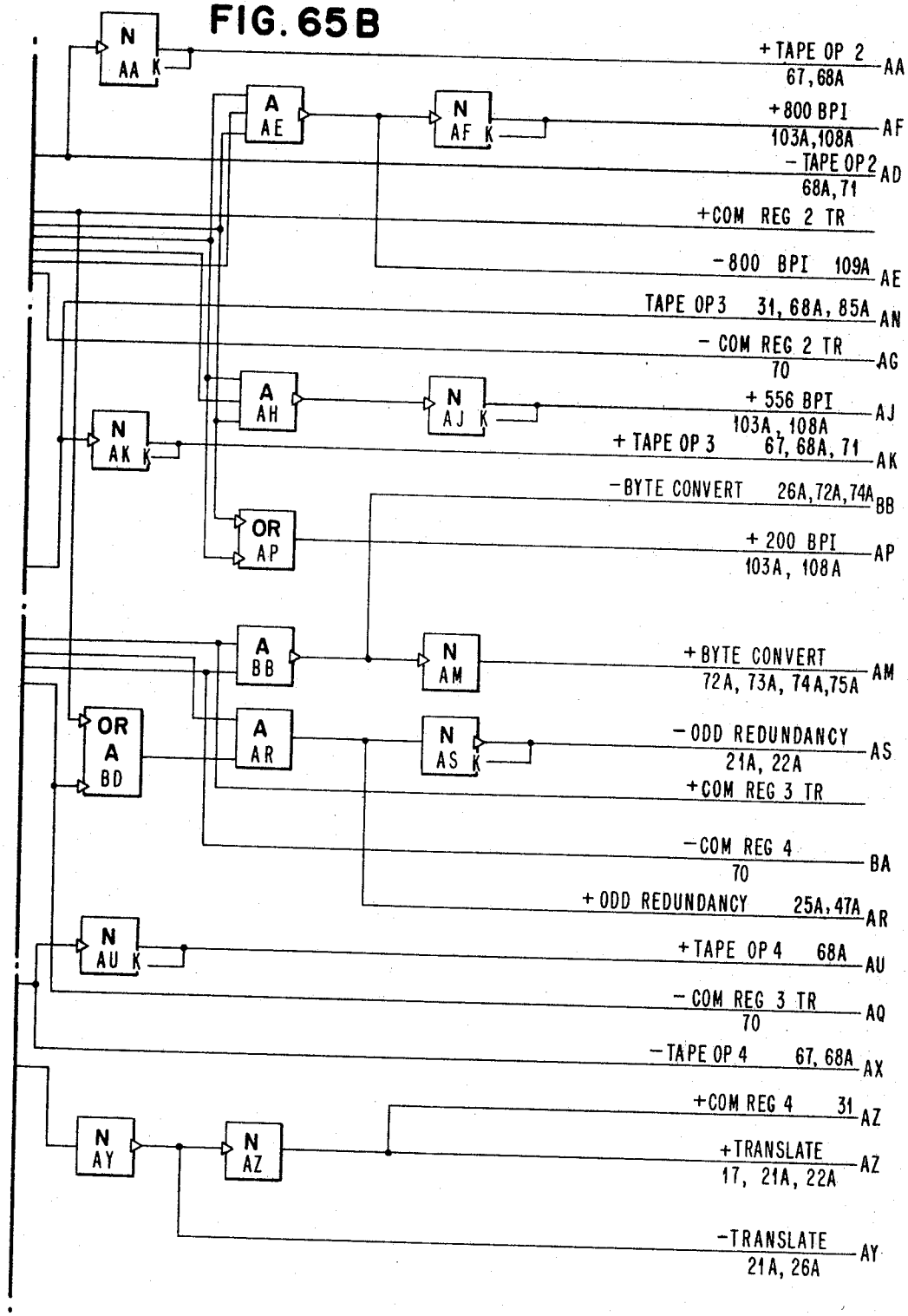

N AA K
+TAPE OP 2 67,68A AA
A AE
N AF K
+800 BPI 103A,108A AF
-TAPE OP2 68A,71 AD
+COM REG 2 TR
-800 BPI 109A AE
TAPE OP3 31, 68A, 85A AN
- COM REG 2 TR 70 AG
A AH
N AJ K
+ 556 BPI 103A, 108A AJ
N AK K
+ TAPE OP 3 67, 68A, 71 AK
-BYTE CONVERT 26A,72A,74A BB
OR AP
+ 200 BPI 103A, 108A AP
A BB
N AM
+BYTE CONVERT 72A, 73A, 74A,75A AM
OR A BD
A AR
N AS K
-ODD REDUNDANCY 21A, 22A AS
+COM REG 3 TR
-COM REG 4 70 BA
+ODD REDUNDANCY 25A, 47A AR
N AU K
+TAPE OP 4 68A AU
- COM REG 3 TR 70 AQ
-TAPE OP 4 67, 68A AX
N AY
N AZ
+COM REG 4 31 AZ
+TRANSLATE 17, 21A, 22A AZ
-TRANSLATE 21A, 26A AY

FIG. 66A

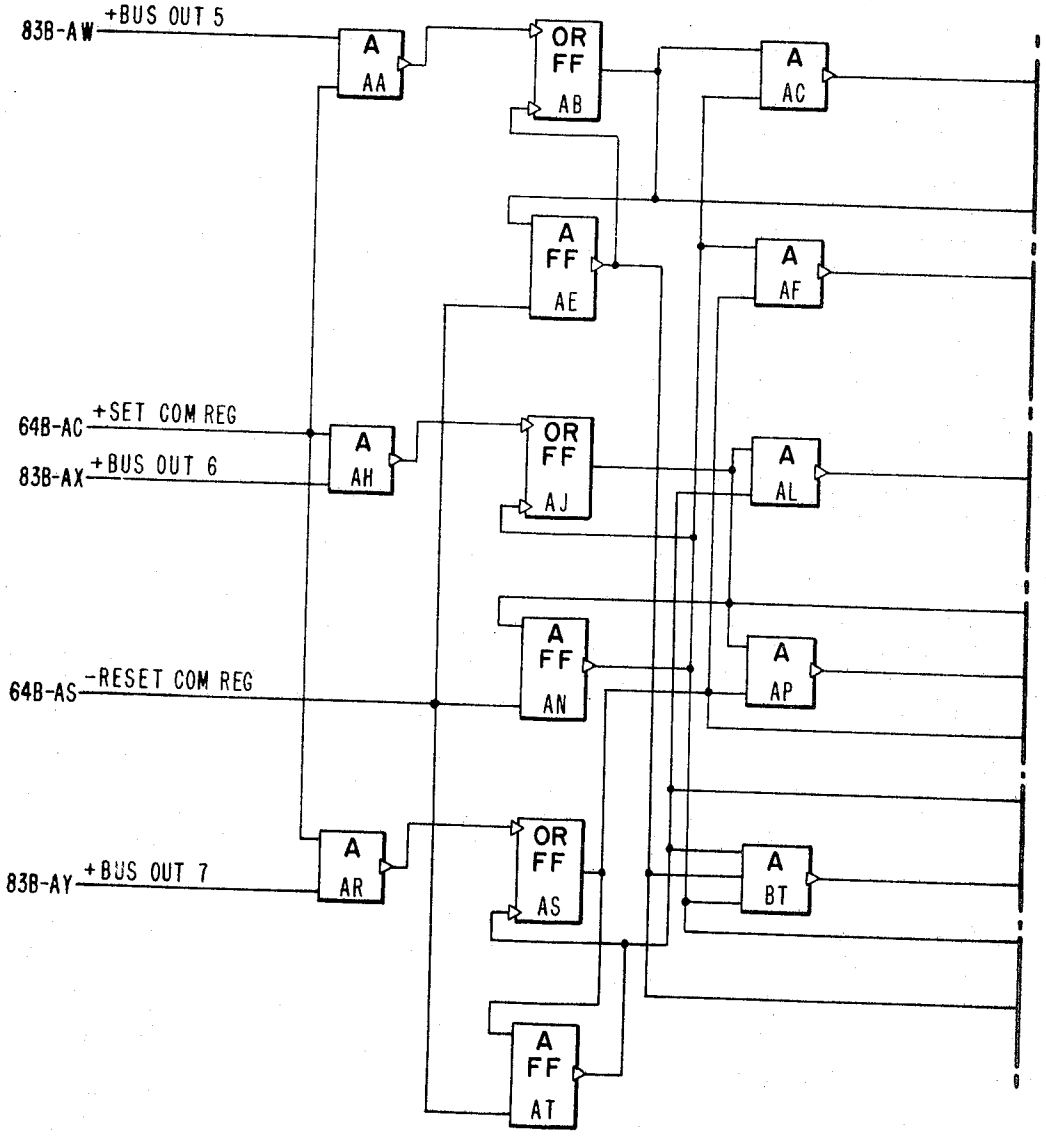

83B-AW
+BUS OUT 5
A AA
OR FF AB
A AC
A FF AE
A AF
64B-AC
+SET COM REG
83B-AX
+BUS OUT 6
A AH
OR FF AJ
A AL
A FF AN
64B-AS
-RESET COM REG
A AP
83B-AY
+BUS OUT 7
A AR
OR FF AS
A BT
A FF AT

FIG. 66B

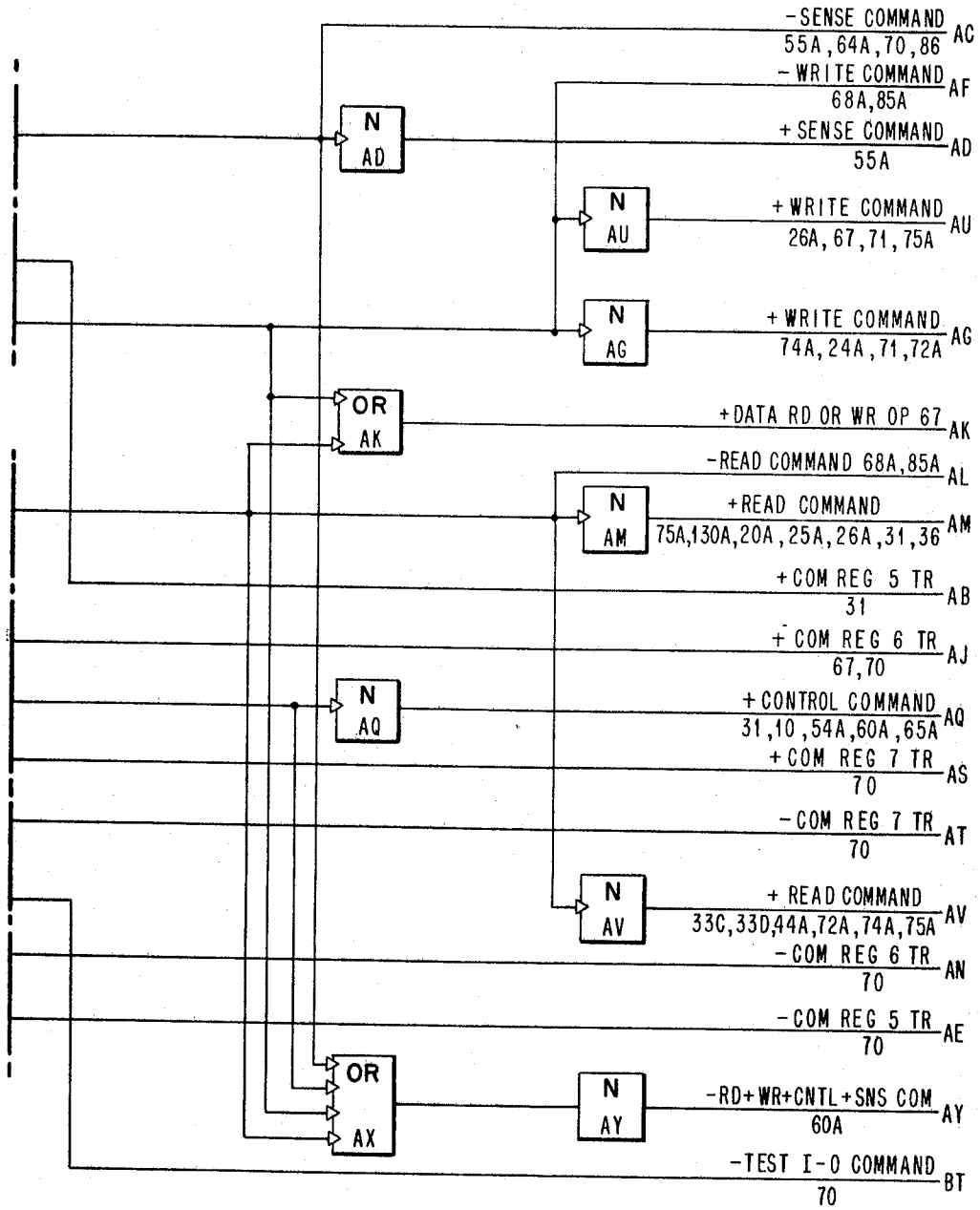

N AD
N AU
N AG
OR AK
N AM
N AQ
N AV
OR AX
N AY
-SENSE COMMAND 55A,64A,70,86 AC
-WRITE COMMAND 68A,85A AF
+SENSE COMMAND 55A AD
+WRITE COMMAND 26A,67,71,75A AU
+WRITE COMMAND 74A,24A,71,72A AG
+DATA RD OR WR OP 67 AK
-READ COMMAND 68A,85A AL
+READ COMMAND 75A,130A,20A,25A,26A,31,36 AM
+COM REG 5 TR 31 AB
+COM REG 6 TR 67,70 AJ
+CONTROL COMMAND 31,10,54A,60A,65A AQ
+COM REG 7 TR 70 AS
-COM REG 7 TR 70 AT
+READ COMMAND 33C,33D,44A,72A,74A,75A AV
-COM REG 6 TR 70 AN
-COM REG 5 TR 70 AE
-RD+WR+CNTL+SNS COM 60A AY
-TEST I-O COMMAND 70 BT

FIG. 67

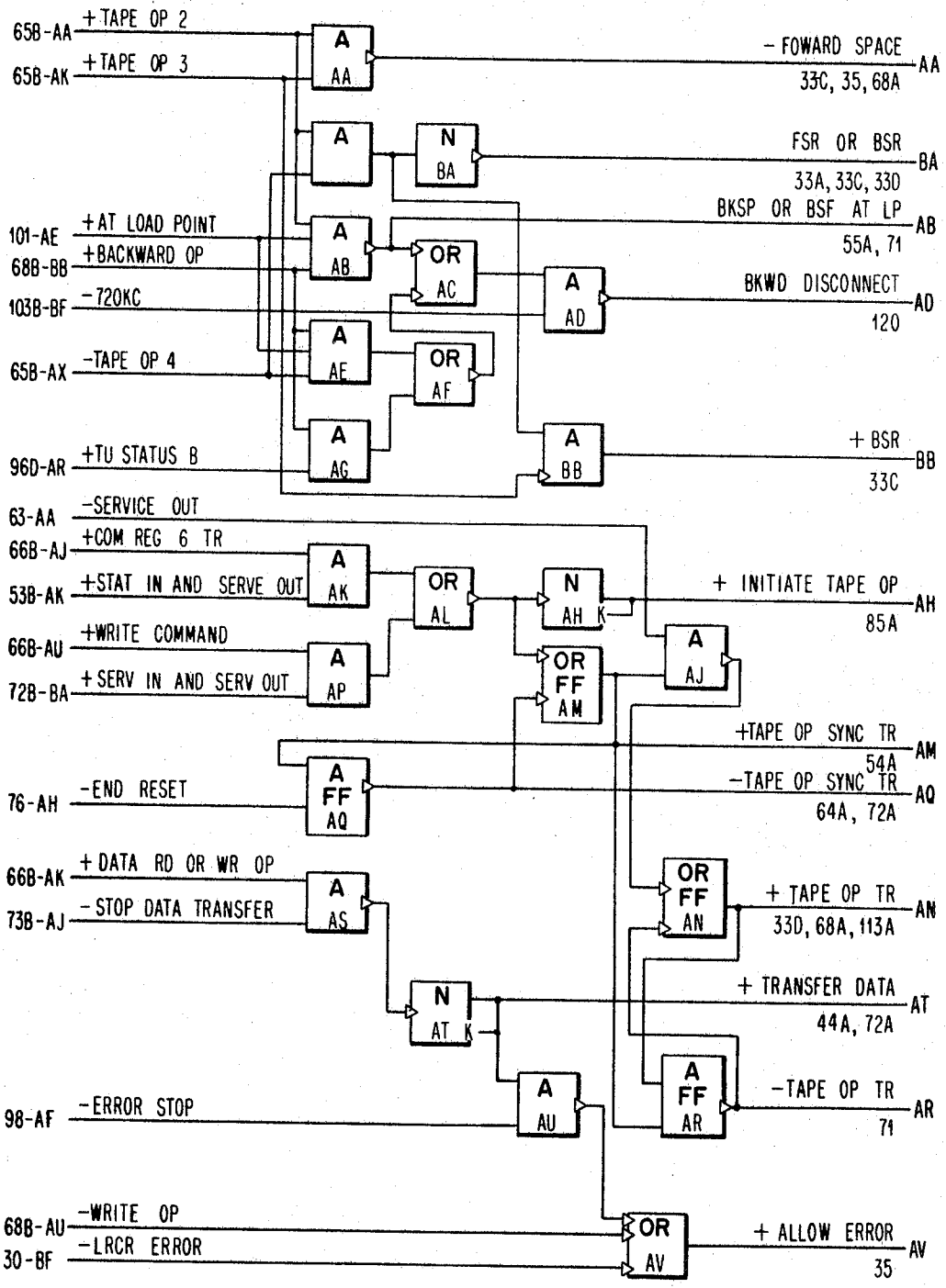

65B-AA
+TAPE OP 2
65B-AK
+TAPE OP 3
A AA
- FOWARD SPACE
33C, 35, 68A
AA
A
N BA
FSR OR BSR
33A, 33C, 33D
BA
101-AE
+AT LOAD POINT
68B-BB
+BACKWARD OP
A AB
OR AC
BKSP OR BSF AT LP
55A, 71
AB
103B-BF
-720KC
A AD
BKWD DISCONNECT
120
AD
A AE
65B-AX
-TAPE OP 4
OR AF
A AG
96D-AR
+TU STATUS B
A BB
+ BSR
33C
BB
63-AA
-SERVICE OUT
66B-AJ
+COM REG 6 TR
53B-AK
+STAT IN AND SERVE OUT
A AK
OR AL
N AH K
+ INITIATE TAPE OP
85A
AH
66B-AU
+WRITE COMMAND
72B-BA
+SERV IN AND SERV OUT
A AP
OR FF AM
A AJ
+TAPE OP SYNC TR
54A
AM
A FF AQ
76-AH
-END RESET
-TAPE OP SYNC TR
64A, 72A
AQ
66B-AK
+DATA RD OR WR OP
73B-AJ
-STOP DATA TRANSFER
A AS
OR FF AN
+ TAPE OP TR
33D, 68A, 113A
AN
N AT K
+ TRANSFER DATA
44A, 72A
AT
A AU
A FF AR
-TAPE OP TR
71
AR
98-AF
-ERROR STOP
68B-AU
-WRITE OP
30-BF
-LRCR ERROR
OR AV
+ ALLOW ERROR
35
AV

FIG. 68A

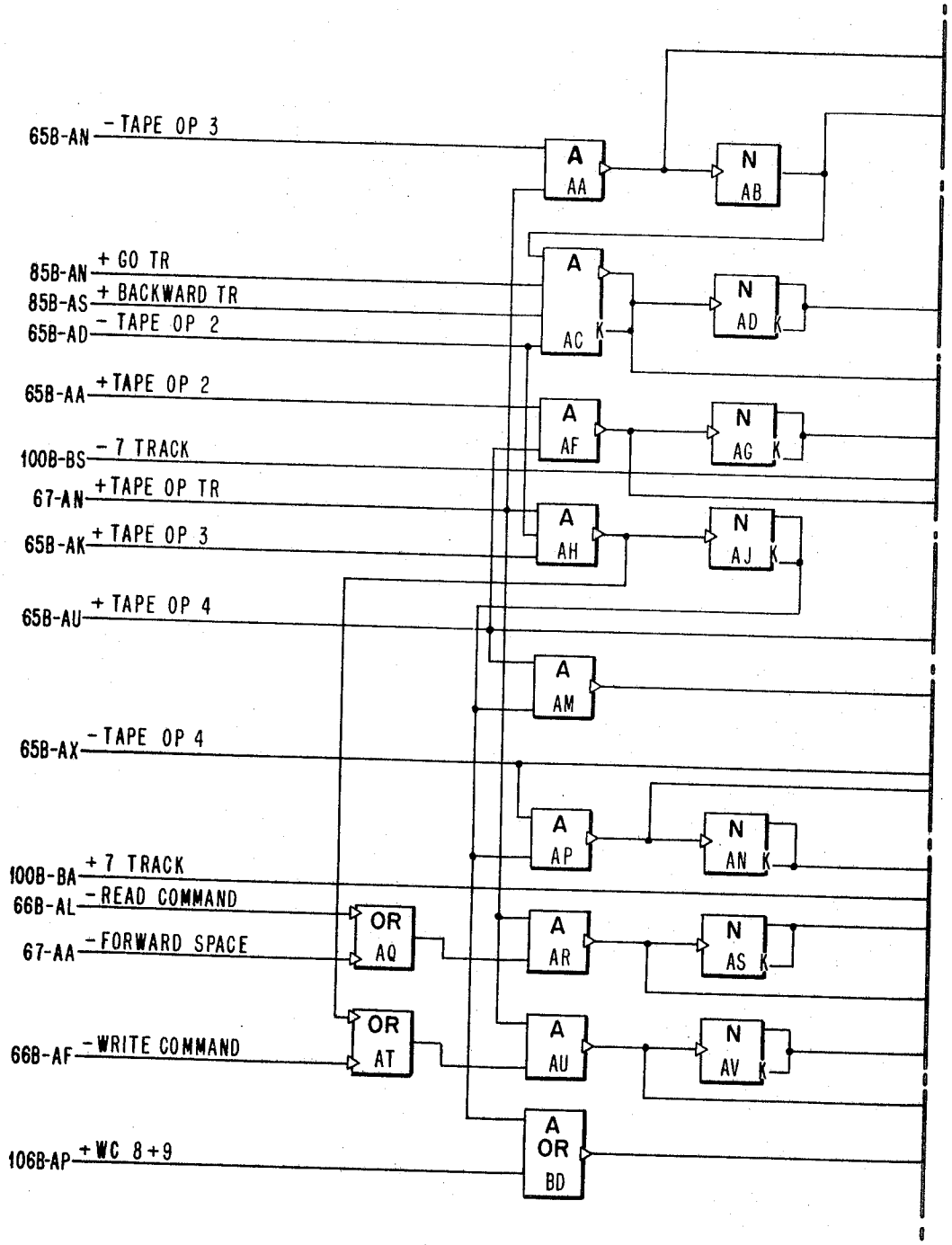

65B-AN -TAPE OP 3
85B-AN +GO TR
85B-AS +BACKWARD TR
65B-AD -TAPE OP 2
65B-AA +TAPE OP 2
100B-BS -7 TRACK
67-AN +TAPE OP TR
65B-AK +TAPE OP 3
65B-AU +TAPE OP 4
65B-AX -TAPE OP 4
100B-BA +7 TRACK
66B-AL -READ COMMAND
67-AA -FORWARD SPACE
66B-AF -WRITE COMMAND
106B-AP +WC 8+9
A AA
N AB
A AC K
N AD K
A AF
N AG K
A AH
N AJ K
A AM
A AP
N AN K
OR AQ
A AR
N AS K
OR AT
A AU
N AV K
A OR BD

FIG. 68B

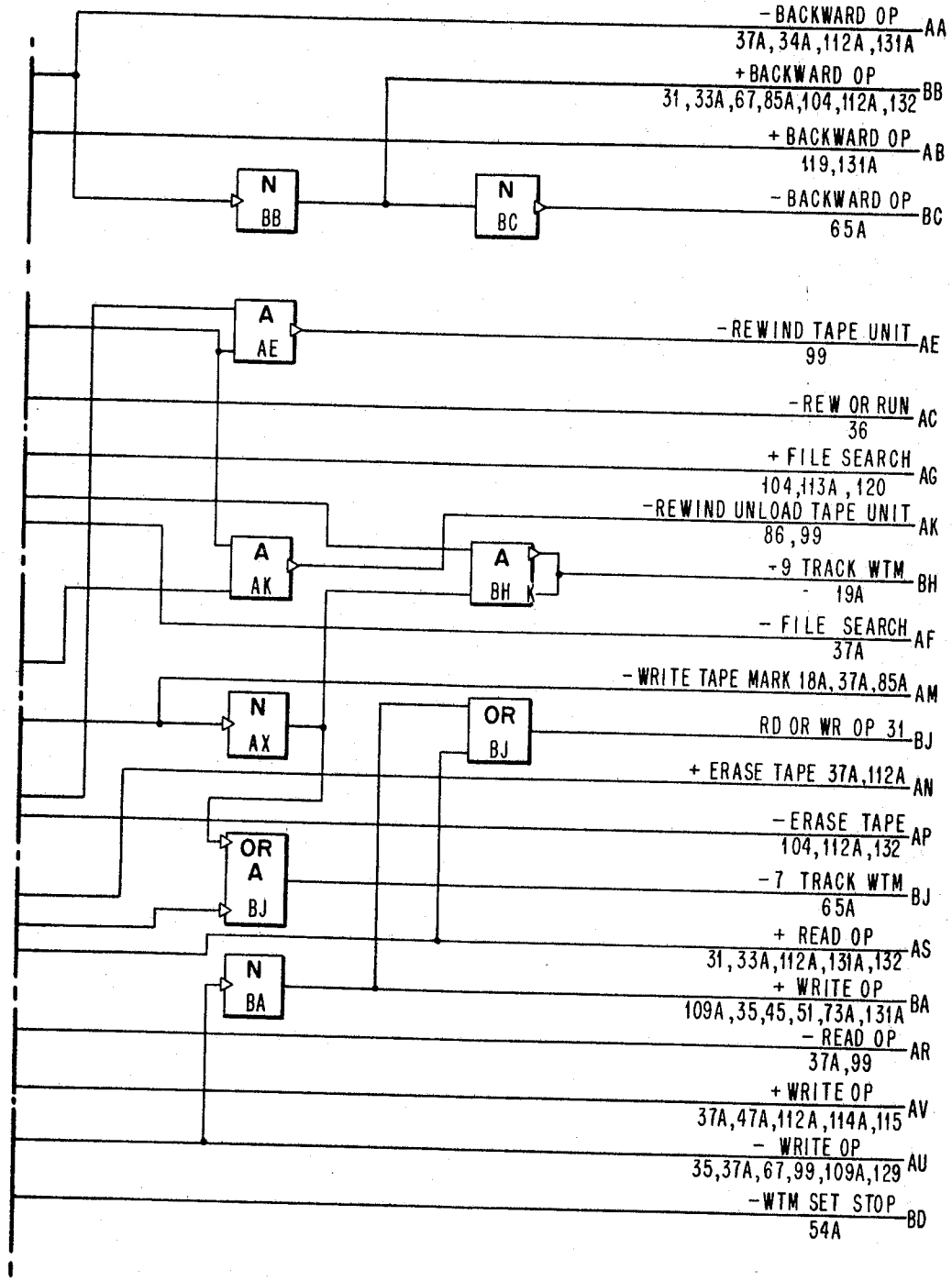

-BACKWARD OP 37A,34A,112A,131A AA
+BACKWARD OP 31,33A,67,85A,104,112A,132 BB
+BACKWARD OP 119,131A AB
N BB
N BC
-BACKWARD OP 65A BC
A AE
-REWIND TAPE UNIT 99 AE
-REW OR RUN 36 AC
+FILE SEARCH 104,113A,120 AG
-REWIND UNLOAD TAPE UNIT 86,99 AK
A AK
A BH K
+9 TRACK WTM 19A BH
-FILE SEARCH 37A AF
-WRITE TAPE MARK 18A,37A,85A AM
N AX
OR BJ
RD OR WR OP 31 BJ
+ERASE TAPE 37A,112A AN
-ERASE TAPE 104,112A,132 AP
OR A BJ
-7 TRACK WTM 65A BJ
+READ OP 31,33A,112A,131A,132 AS
N BA
+WRITE OP 109A,35,45,51,73A,131A BA
-READ OP 37A,99 AR
+WRITE OP 37A,47A,112A,114A,115 AV
-WRITE OP 35,37A,67,99,109A,129 AU
-WTM SET STOP 54A BD

FIG. 69A

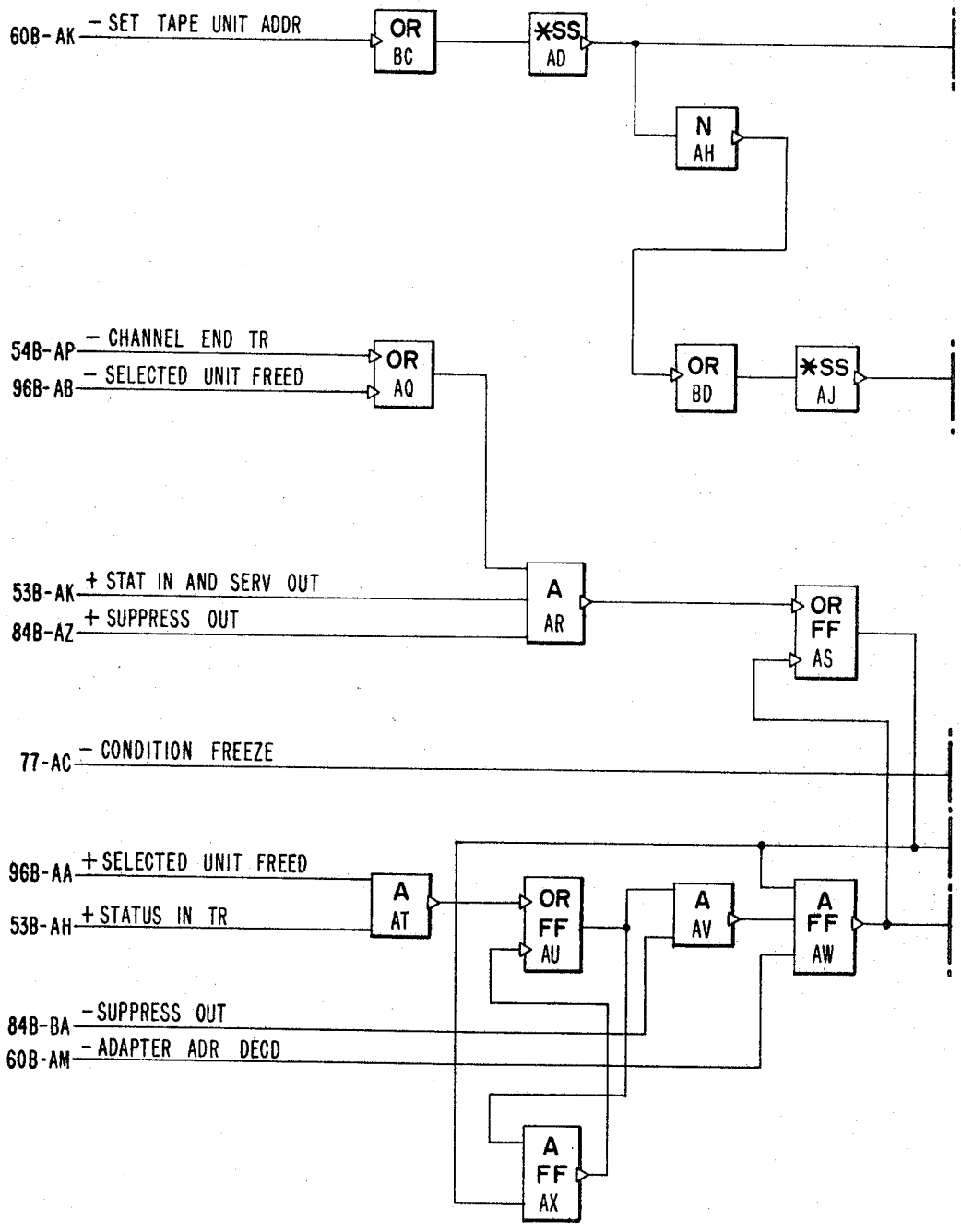

60B-AK
– SET TAPE UNIT ADDR
OR BC
*SS AD
N AH
54B-AP
– CHANNEL END TR
96B-AB
– SELECTED UNIT FREED
OR AQ
OR BD
*SS AJ
53B-AK
+ STAT IN AND SERV OUT
84B-AZ
+ SUPPRESS OUT
A AR
OR FF AS
77-AC
– CONDITION FREEZE
96B-AA
+ SELECTED UNIT FREED
53B-AH
+ STATUS IN TR
A AT
OR FF AU
A AV
A FF AW
84B-BA
– SUPPRESS OUT
60B-AM
– ADAPTER ADR DECD
A FF AX

FIG. 69B

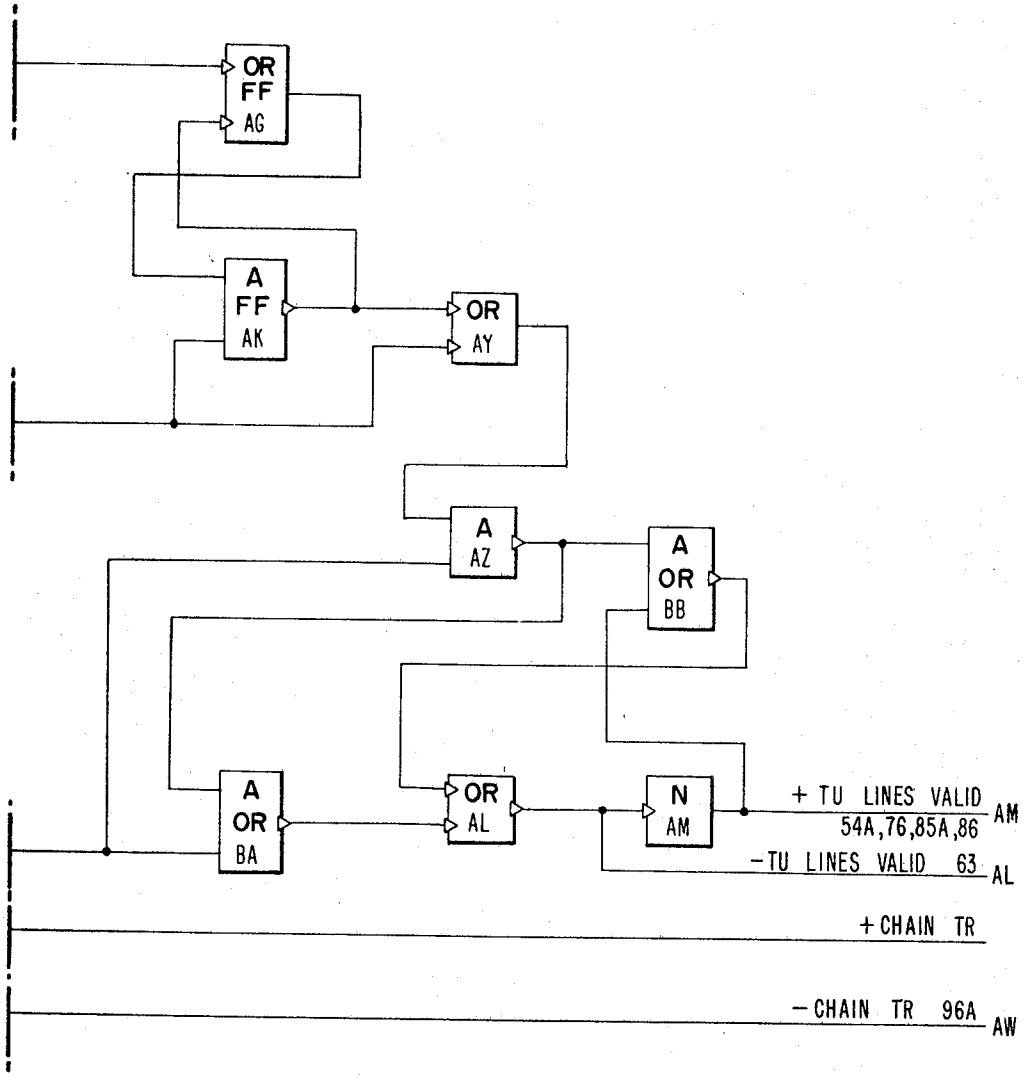

OR FF AG
A FF AK
OR AY
A AZ
A OR BB
A OR BA
OR AL
N AM
+ TU LINES VALID 54A,76,85A,86 AM
-TU LINES VALID 63 AL
+CHAIN TR
-CHAIN TR 96A AW

FIG. 70

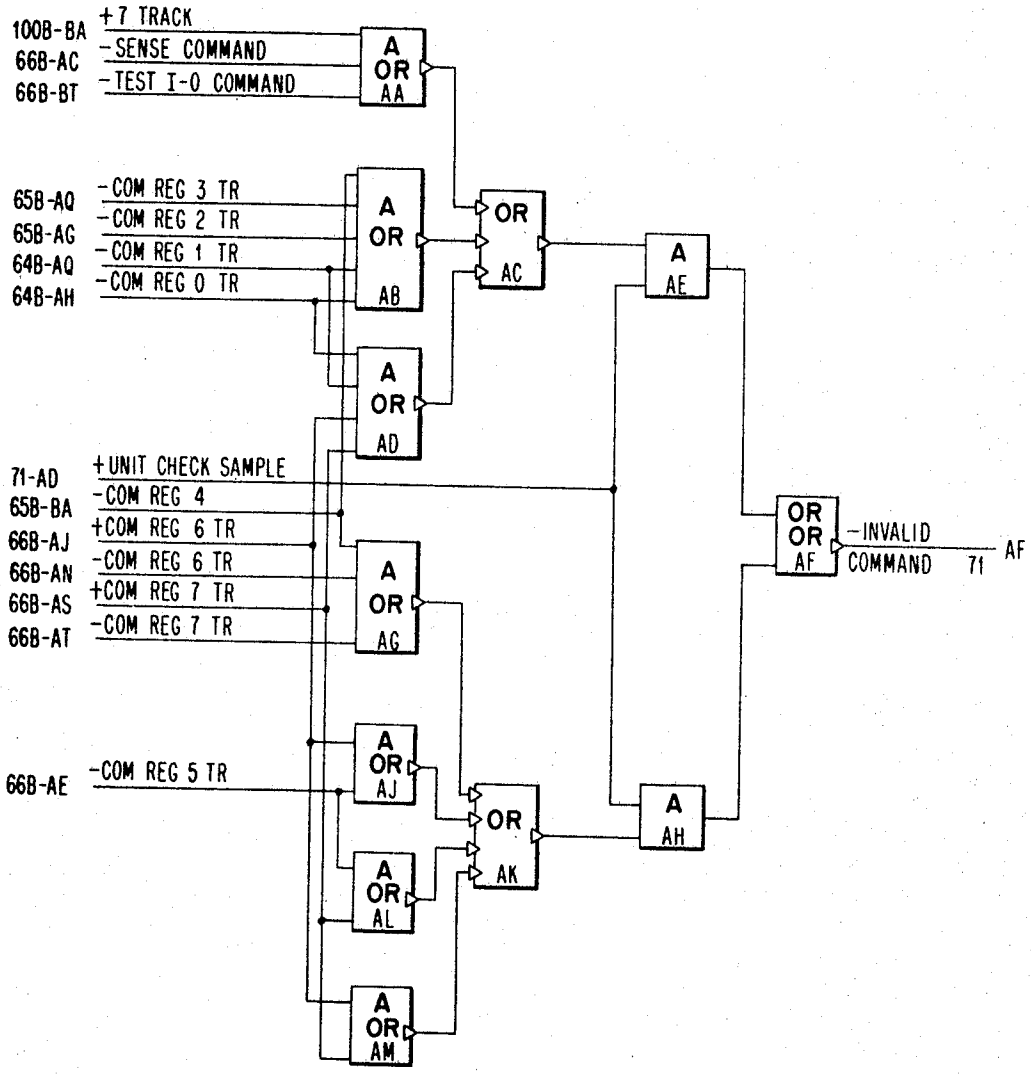

100B-BA
66B-AC
66B-BT
+7 TRACK
-SENSE COMMAND
-TEST I-O COMMAND
A OR AA
65B-AQ
65B-AG
64B-AQ
64B-AH
-COM REG 3 TR
-COM REG 2 TR
-COM REG 1 TR
-COM REG 0 TR
A OR AB
A OR AD
OR AC
A AE
71-AD
65B-BA
66B-AJ
66B-AN
66B-AS
66B-AT
+UNIT CHECK SAMPLE
-COM REG 4
+COM REG 6 TR
-COM REG 6 TR
+COM REG 7 TR
-COM REG 7 TR
A OR AG
66B-AE
-COM REG 5 TR
A OR AJ
A OR AL
A OR AM
OR AK
A AH
OR OR AF
-INVALID COMMAND 71
AF

FIG. 71

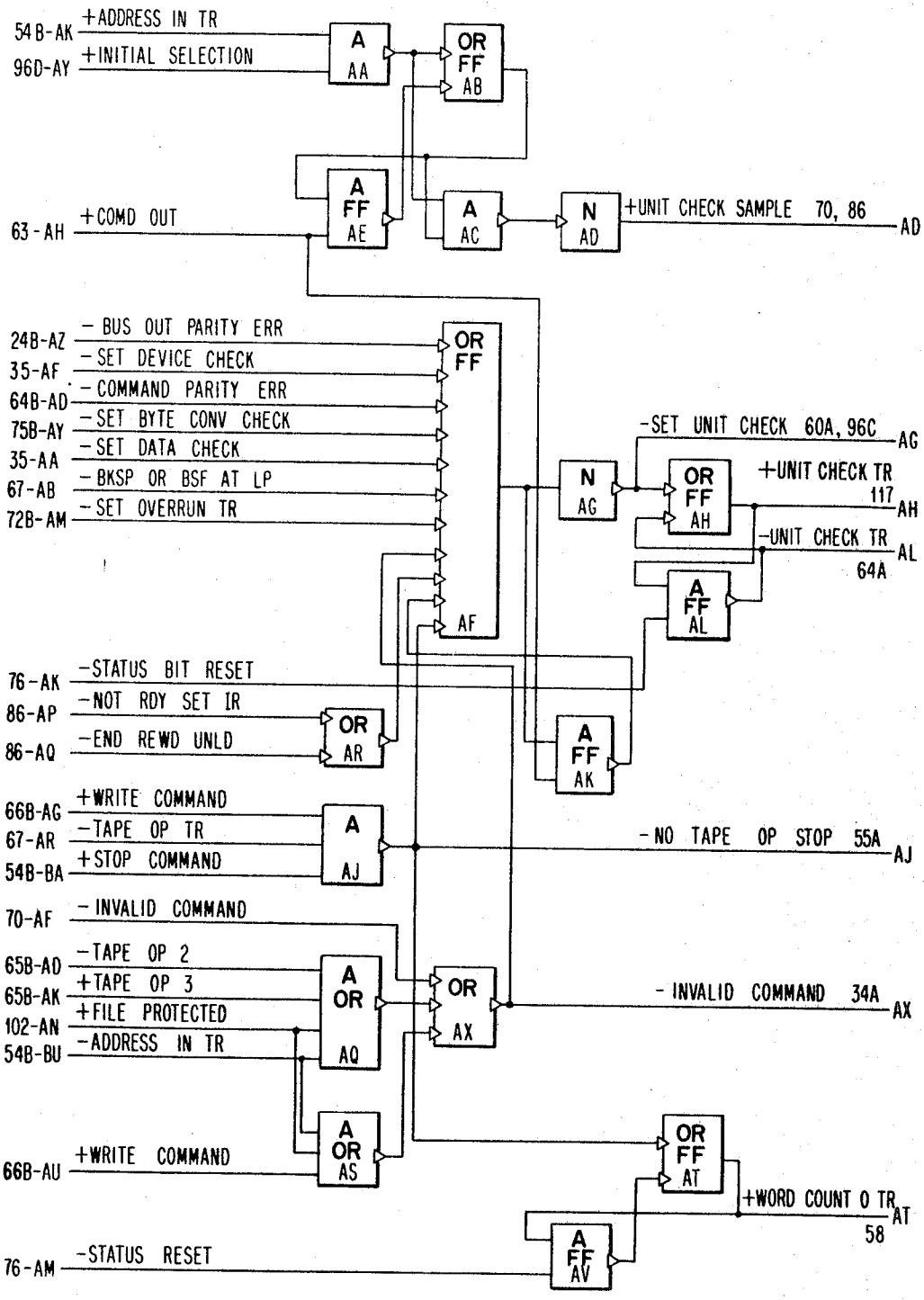

54B-AK
+ADDRESS IN TR
96D-AY
+INITIAL SELECTION
A AA
OR FF AB
A FF AE
A AC
N AD
+UNIT CHECK SAMPLE 70, 86
AD
63-AH
+COMD OUT
24B-AZ
-BUS OUT PARITY ERR
35-AF
-SET DEVICE CHECK
64B-AD
-COMMAND PARITY ERR
75B-AY
-SET BYTE CONV CHECK
35-AA
-SET DATA CHECK
67-AB
-BKSP OR BSF AT LP
72B-AM
-SET OVERRUN TR
OR FF
AF
-SET UNIT CHECK 60A, 96C
AG
N AG
OR FF AH
+UNIT CHECK TR
117
AH
-UNIT CHECK TR
64A
AL
A FF AL
76-AK
-STATUS BIT RESET
86-AP
-NOT RDY SET IR
86-AQ
-END REWD UNLD
OR AR
A FF AK
66B-AG
+WRITE COMMAND
67-AR
-TAPE OP TR
54B-BA
+STOP COMMAND
A AJ
-NO TAPE OP STOP 55A
AJ
70-AF
-INVALID COMMAND
65B-AD
-TAPE OP 2
65B-AK
+TAPE OP 3
102-AN
+FILE PROTECTED
54B-BU
-ADDRESS IN TR
A OR AQ
OR AX
-INVALID COMMAND 34A
AX
66B-AU
+WRITE COMMAND
A OR AS
OR FF AT
+WORD COUNT 0 TR
58
AT
76-AM
-STATUS RESET
A FF AV

FIG. 72A

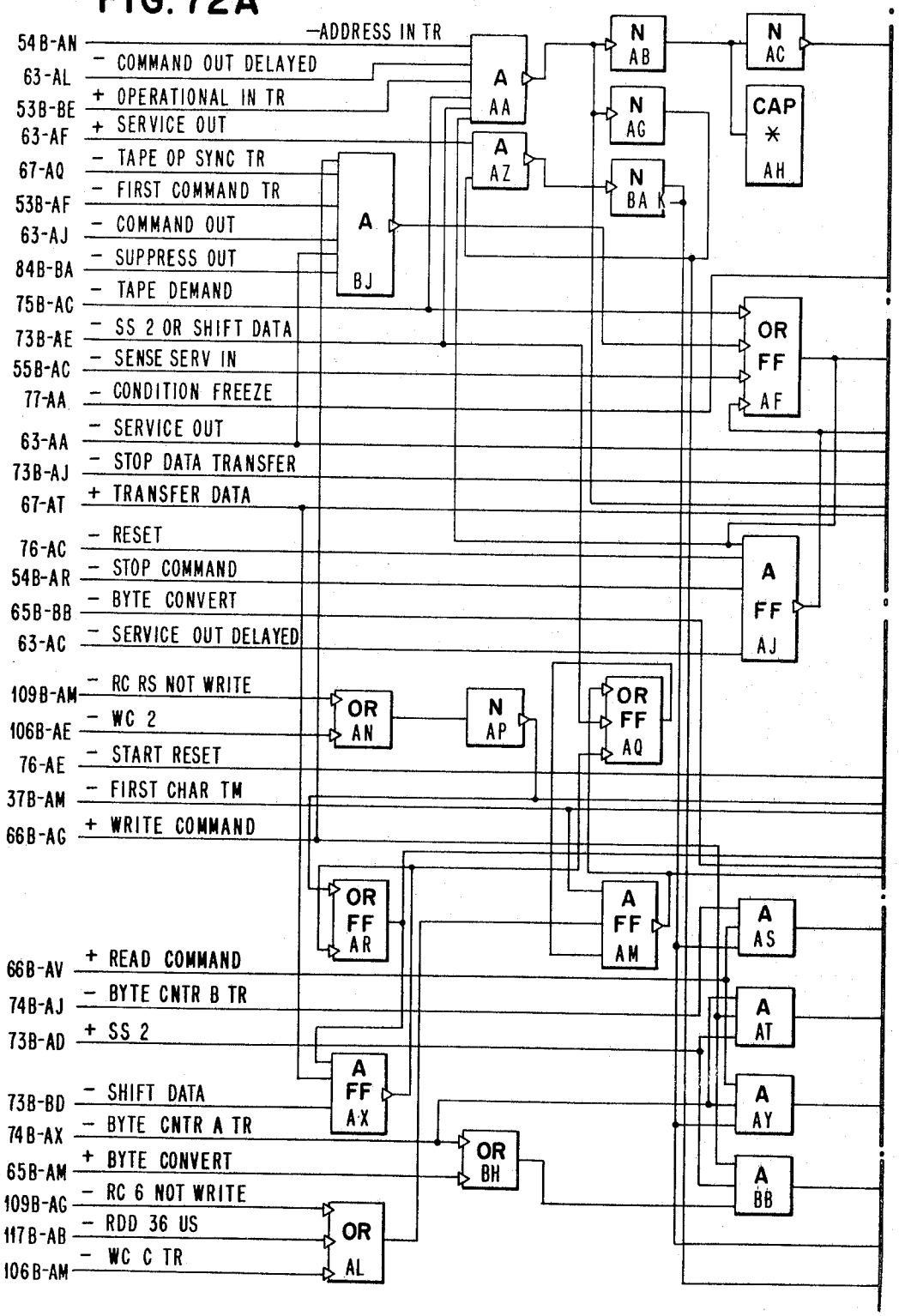

54B-AN
63-AL
53B-BE
63-AF
67-AQ
53B-AF
63-AJ
84B-BA
75B-AC
73B-AE
55B-AC
77-AA
63-AA
73B-AJ
67-AT
76-AC
54B-AR
65B-BB
63-AC
109B-AM
106B-AE
76-AE
37B-AM
66B-AC
66B-AV
74B-AJ
73B-AD
73B-BD
74B-AX
65B-AM
109B-AC
117B-AB
106B-AM
ADDRESS IN TR
- COMMAND OUT DELAYED
+ OPERATIONAL IN TR
+ SERVICE OUT
- TAPE OP SYNC TR
- FIRST COMMAND TR
- COMMAND OUT
- SUPPRESS OUT
- TAPE DEMAND
- SS 2 OR SHIFT DATA
- SENSE SERV IN
- CONDITION FREEZE
- SERVICE OUT
- STOP DATA TRANSFER
+ TRANSFER DATA
- RESET
- STOP COMMAND
- BYTE CONVERT
- SERVICE OUT DELAYED
- RC RS NOT WRITE
- WC 2
- START RESET
- FIRST CHAR TM
+ WRITE COMMAND
+ READ COMMAND
- BYTE CNTR B TR
+ SS 2
- SHIFT DATA
- BYTE CNTR A TR
+ BYTE CONVERT
- RC 6 NOT WRITE
- RDD 36 US
- WC C TR
A AA
A AZ
A BJ
N AB
N AG
N BA K
N AC
CAP * AH
OR FF AF
A FF AJ
OR AN
N AP
OR FF AQ
OR FF AR
A FF AM
A FF AX
OR BH
OR AL
A AS
A AT
A AY
A BB

FIG. 72 B

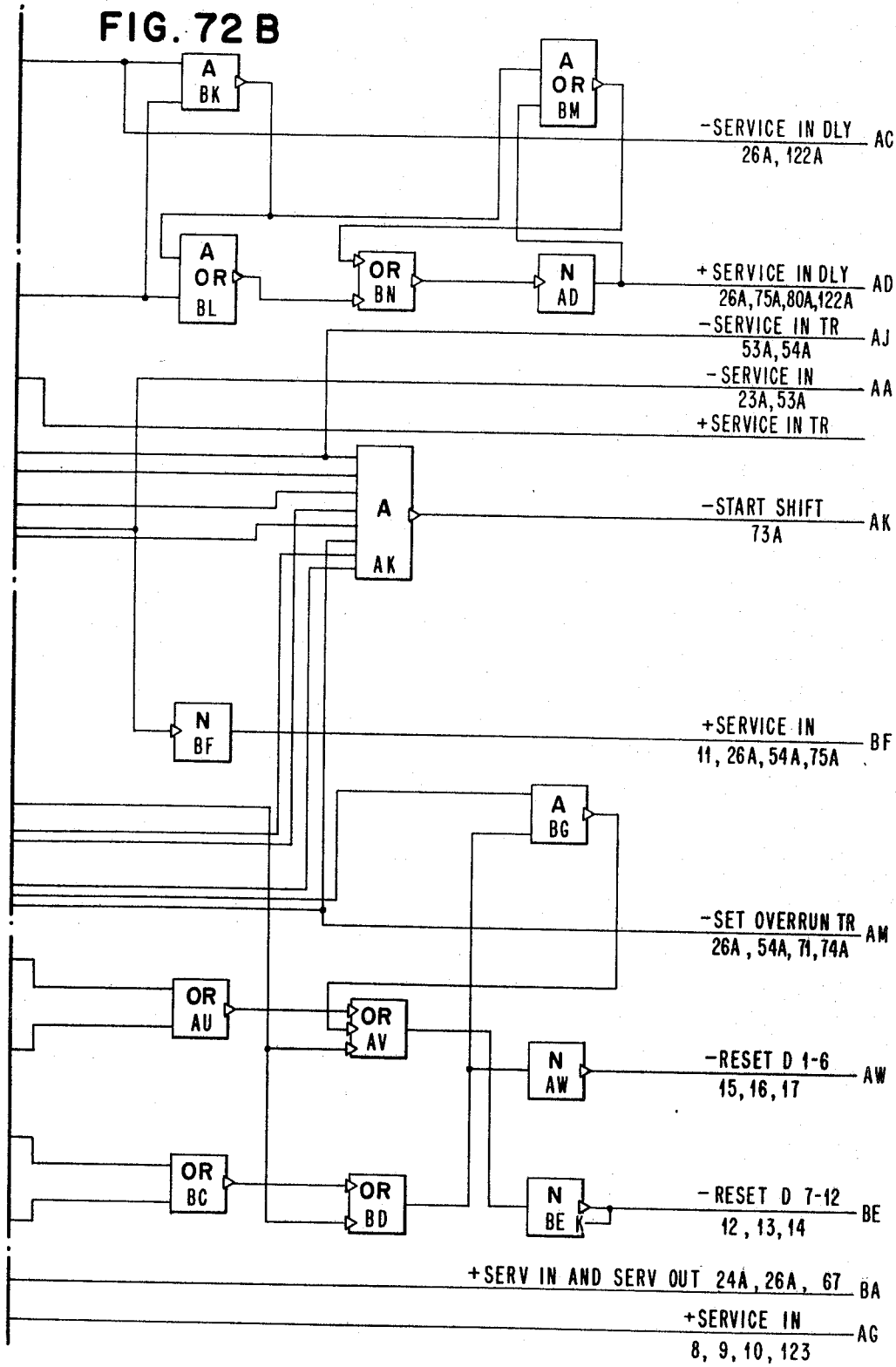

A BK
A OR BM
-SERVICE IN DLY 26A, 122A AC
A OR BL
OR BN
N AD
+SERVICE IN DLY 26A,75A,80A,122A AD
-SERVICE IN TR 53A,54A AJ
-SERVICE IN 23A,53A AA
+SERVICE IN TR
A AK
-START SHIFT 73A AK
N BF
+SERVICE IN 11, 26A, 54A, 75A BF
A BG
-SET OVERRUN TR 26A, 54A, 71, 74A AM
OR AU
OR AV
N AW
-RESET D 1-6 15, 16, 17 AW
OR BC
OR BD
N BE K
-RESET D 7-12 12, 13, 14 BE
+SERV IN AND SERV OUT 24A, 26A, 67 BA
+SERVICE IN 8, 9, 10, 123 AG

FIG. 73A

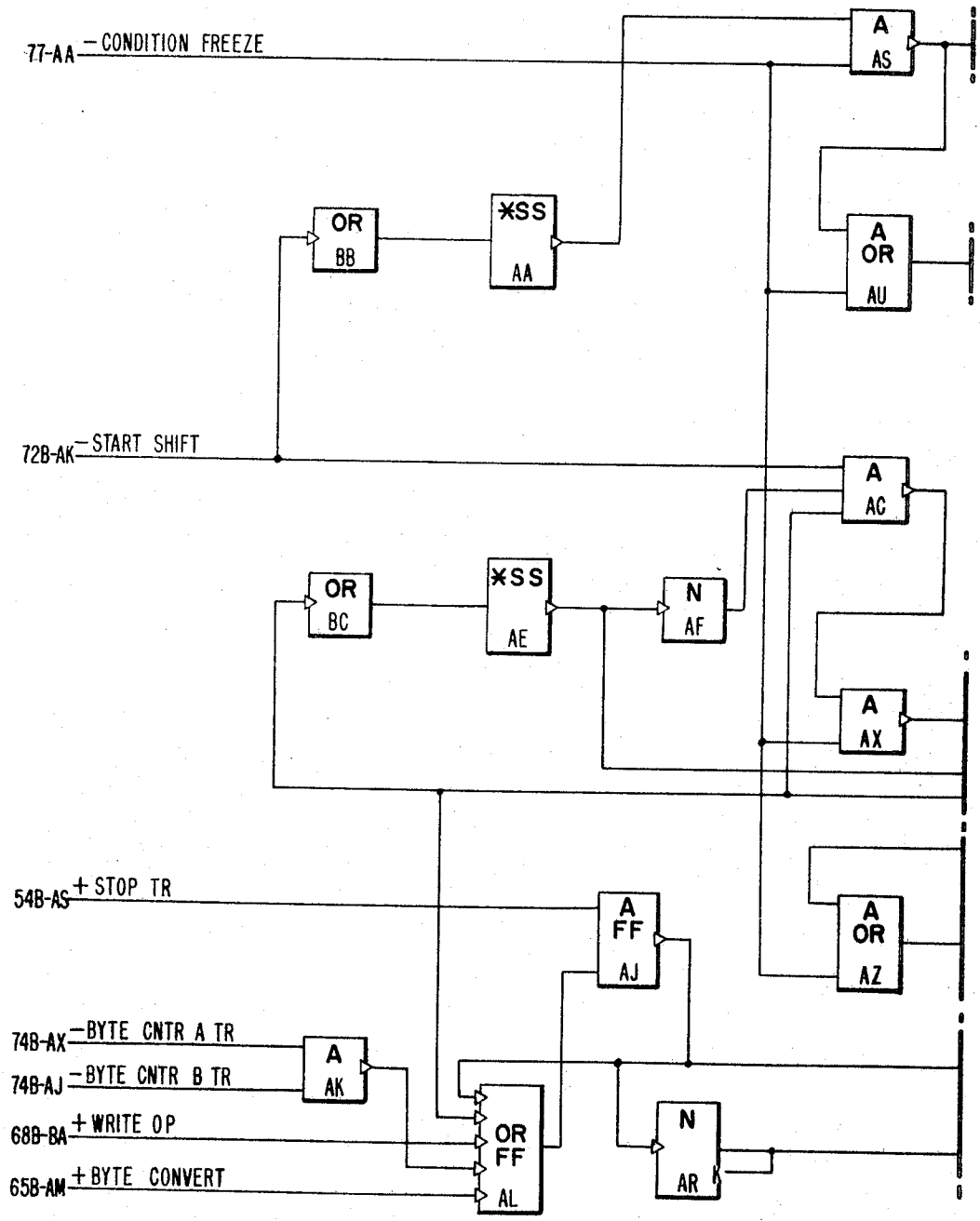

77-AA −CONDITION FREEZE
A AS
OR BB
*SS AA
A OR AU
72B-AK −START SHIFT
A AC
OR BC
*SS AE
N AF
A AX
54B-AS +STOP TR
A FF AJ
A OR AZ
74B-AX −BYTE CNTR A TR
74B-AJ −BYTE CNTR B TR
68B-BA +WRITE OP
65B-AM +BYTE CONVERT
A AK
OR FF AL
N AR

FIG. 73B

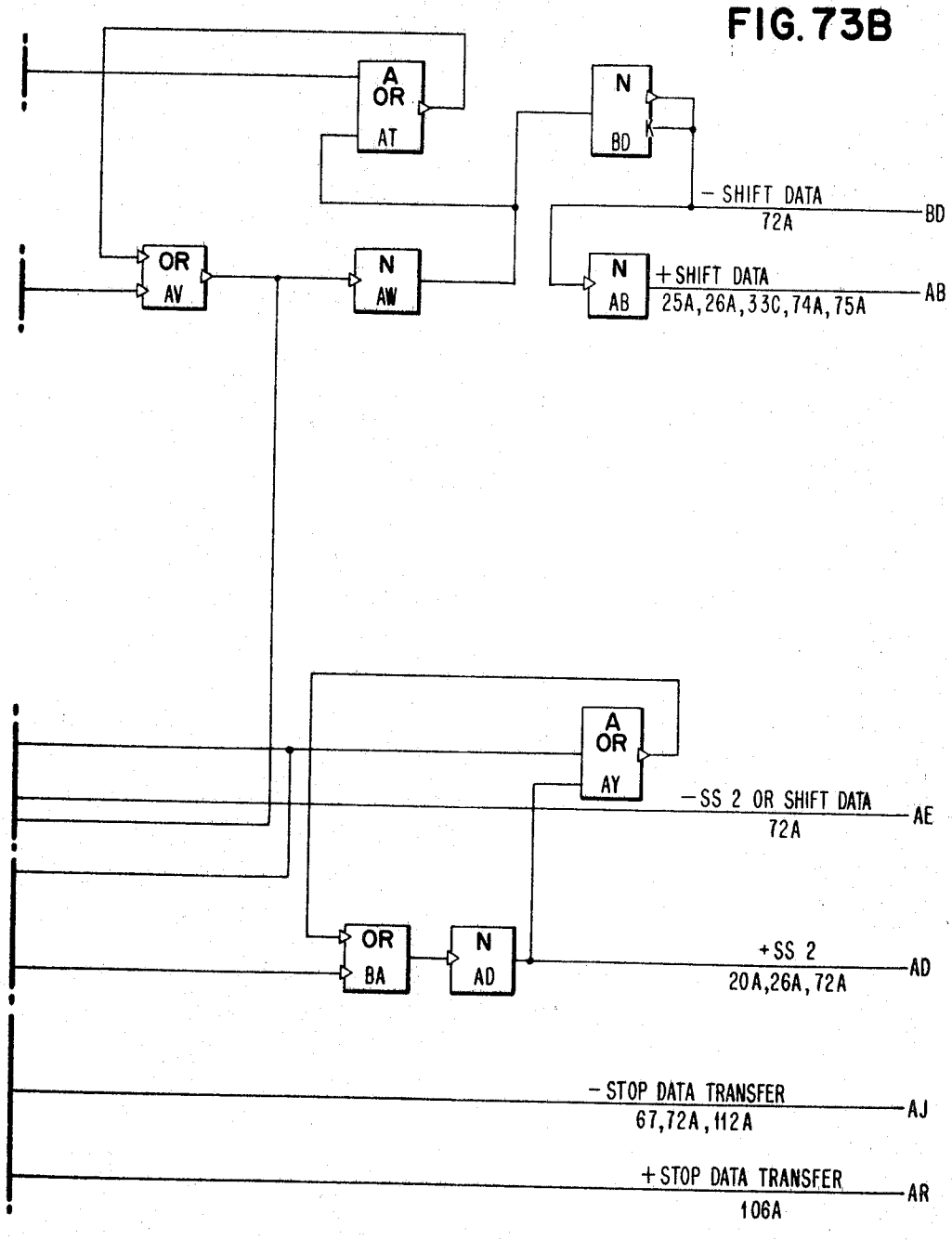

A OR AT
N BD
- SHIFT DATA 72A
BD
OR AV
N AW
N AB
+ SHIFT DATA 25A,26A,33C,74A,75A
AB
A OR AY
- SS 2 OR SHIFT DATA 72A
AE
OR BA
N AD
+ SS 2 20A,26A,72A
AD
- STOP DATA TRANSFER 67,72A,112A
AJ
+ STOP DATA TRANSFER 106A
AR

FIG. 74A

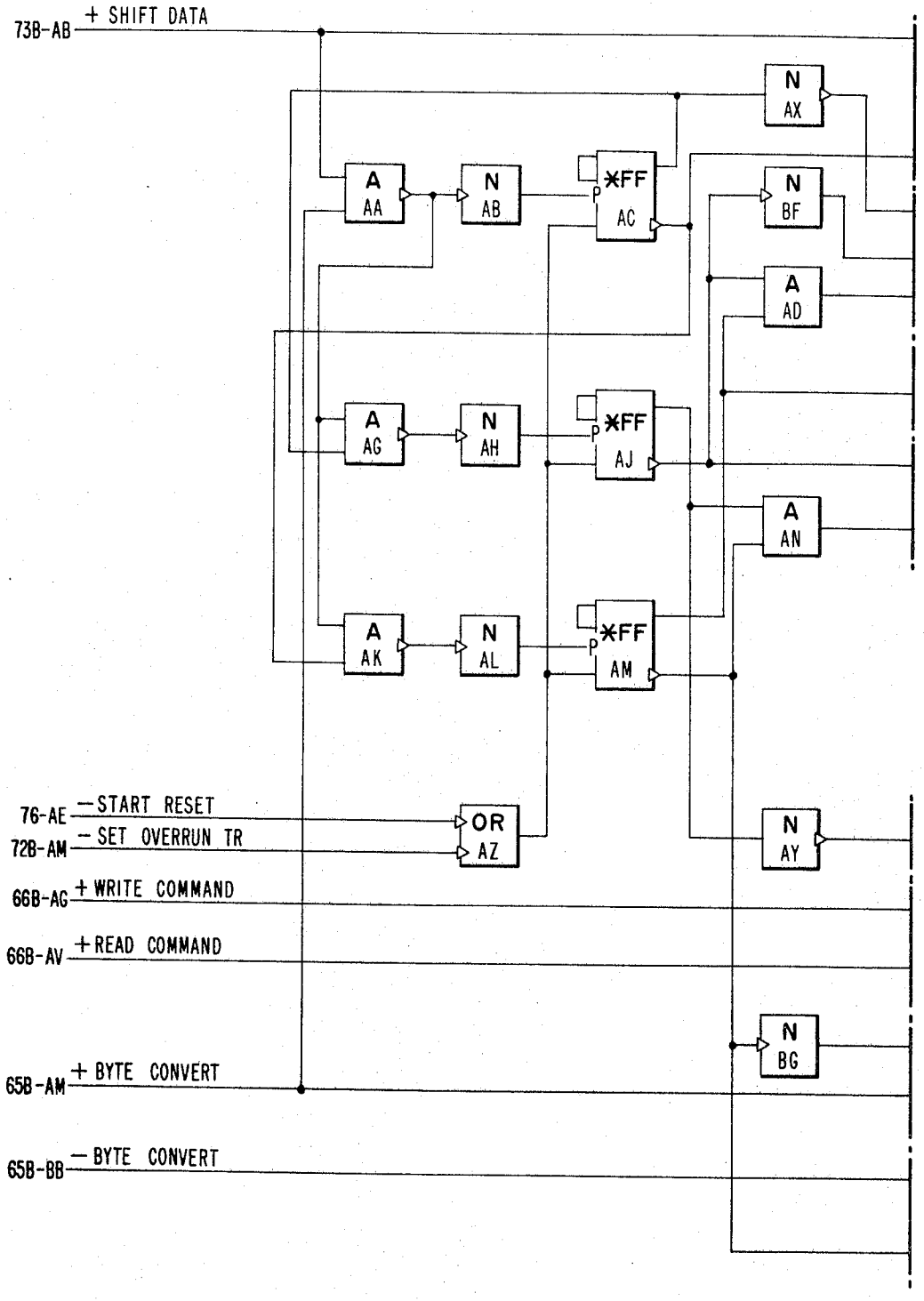

73B-AB + SHIFT DATA
N AX
A AA
N AB
*FF AC
P
N BF
A AD
A AG
N AH
*FF AJ
P
A AN
A AK
N AL
*FF AM
P
76-AE — START RESET
72B-AM — SET OVERRUN TR
OR AZ
N AY
66B-AG + WRITE COMMAND
66B-AV + READ COMMAND
N BG
65B-AM + BYTE CONVERT
65B-BB — BYTE CONVERT

FIG. 74B

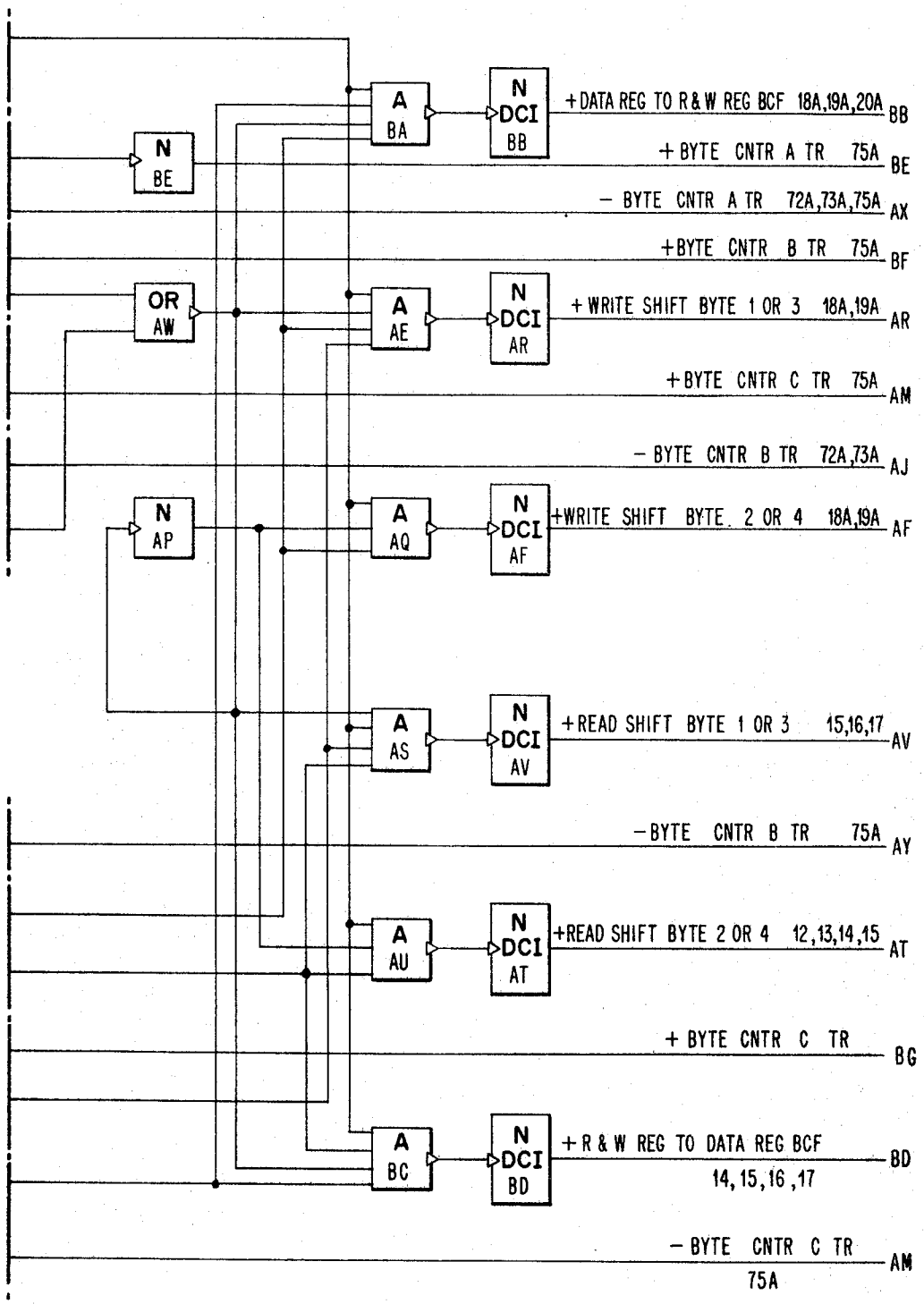

N BE
OR AW
N AP
A BA
A AE
A AQ
A AS
A AU
A BC
N DCI BB
N DCI AR
N DCI AF
N DCI AV
N DCI AT
N DCI BD
+DATA REG TO R & W REG BCF 18A,19A,20A BB
+ BYTE CNTR A TR 75A BE
- BYTE CNTR A TR 72A,73A,75A AX
+BYTE CNTR B TR 75A BF
+ WRITE SHIFT BYTE 1 OR 3 18A,19A AR
+BYTE CNTR C TR 75A AM
- BYTE CNTR B TR 72A,73A AJ
+WRITE SHIFT BYTE 2 OR 4 18A,19A AF
+READ SHIFT BYTE 1 OR 3 15,16,17 AV
-BYTE CNTR B TR 75A AY
+READ SHIFT BYTE 2 OR 4 12,13,14,15 AT
+ BYTE CNTR C TR BG
+ R & W REG TO DATA REG BCF 14,15,16,17 BD
- BYTE CNTR C TR 75A AM

FIG. 75A

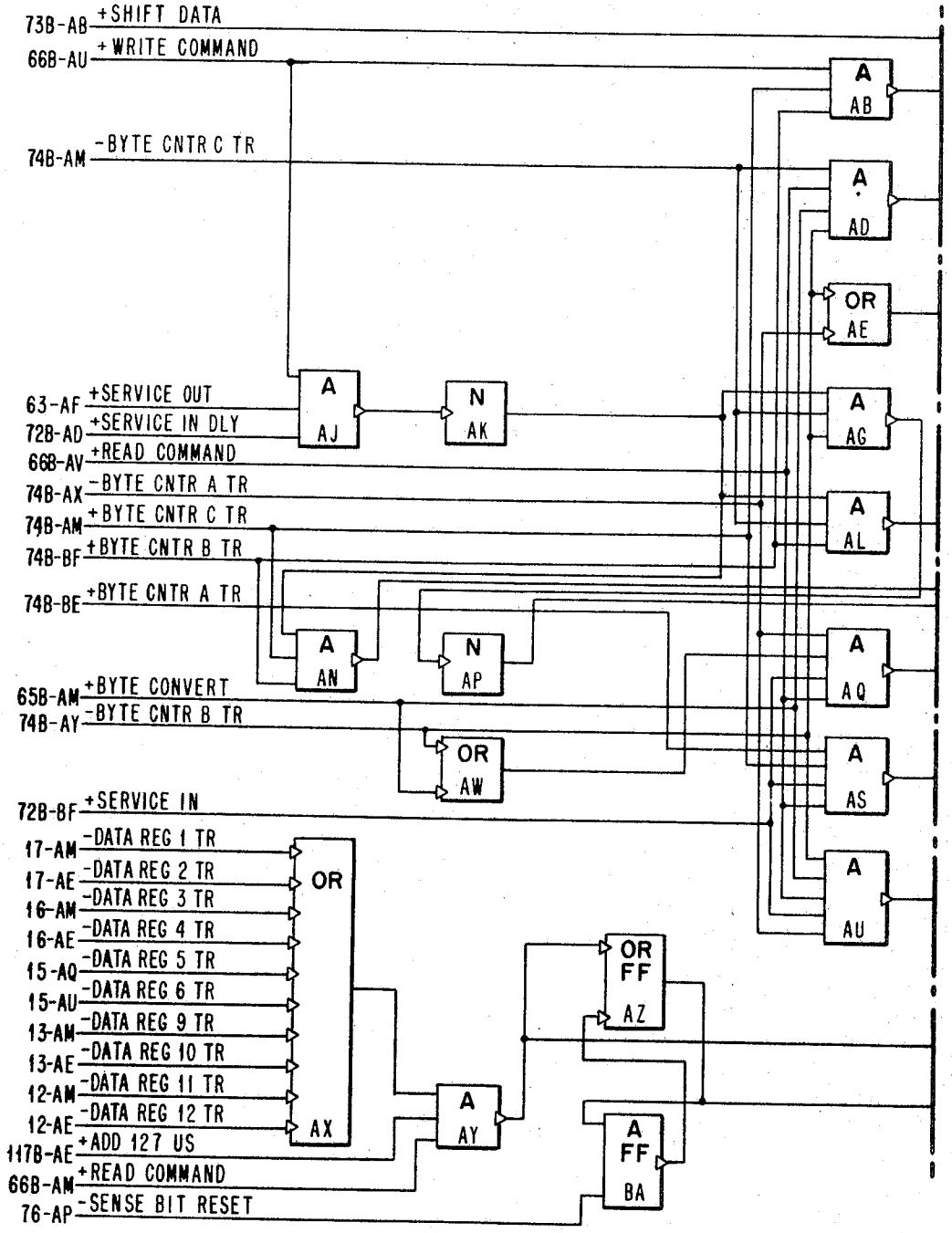

73B-AB +SHIFT DATA
66B-AU +WRITE COMMAND
74B-AM -BYTE CNTR C TR
63-AF +SERVICE OUT
72B-AD +SERVICE IN DLY
66B-AV +READ COMMAND
74B-AX -BYTE CNTR A TR
74B-AM +BYTE CNTR C TR
74B-BF +BYTE CNTR B TR
74B-BE +BYTE CNTR A TR
65B-AM +BYTE CONVERT
74B-AY -BYTE CNTR B TR
72B-BF +SERVICE IN
17-AM -DATA REG 1 TR
17-AE -DATA REG 2 TR
16-AM -DATA REG 3 TR
16-AE -DATA REG 4 TR
15-AQ -DATA REG 5 TR
15-AU -DATA REG 6 TR
13-AM -DATA REG 9 TR
13-AE -DATA REG 10 TR
12-AM -DATA REG 11 TR
12-AE -DATA REG 12 TR
117B-AE +ADD 127 US
66B-AM +READ COMMAND
76-AP -SENSE BIT RESET
A AJ
N AK
A AN
N AP
OR AW
OR AX
A AY
OR FF AZ
A FF BA
A AB
A AD
OR AE
A AG
A AL
A AQ
A AS
A AU

FIG. 75B

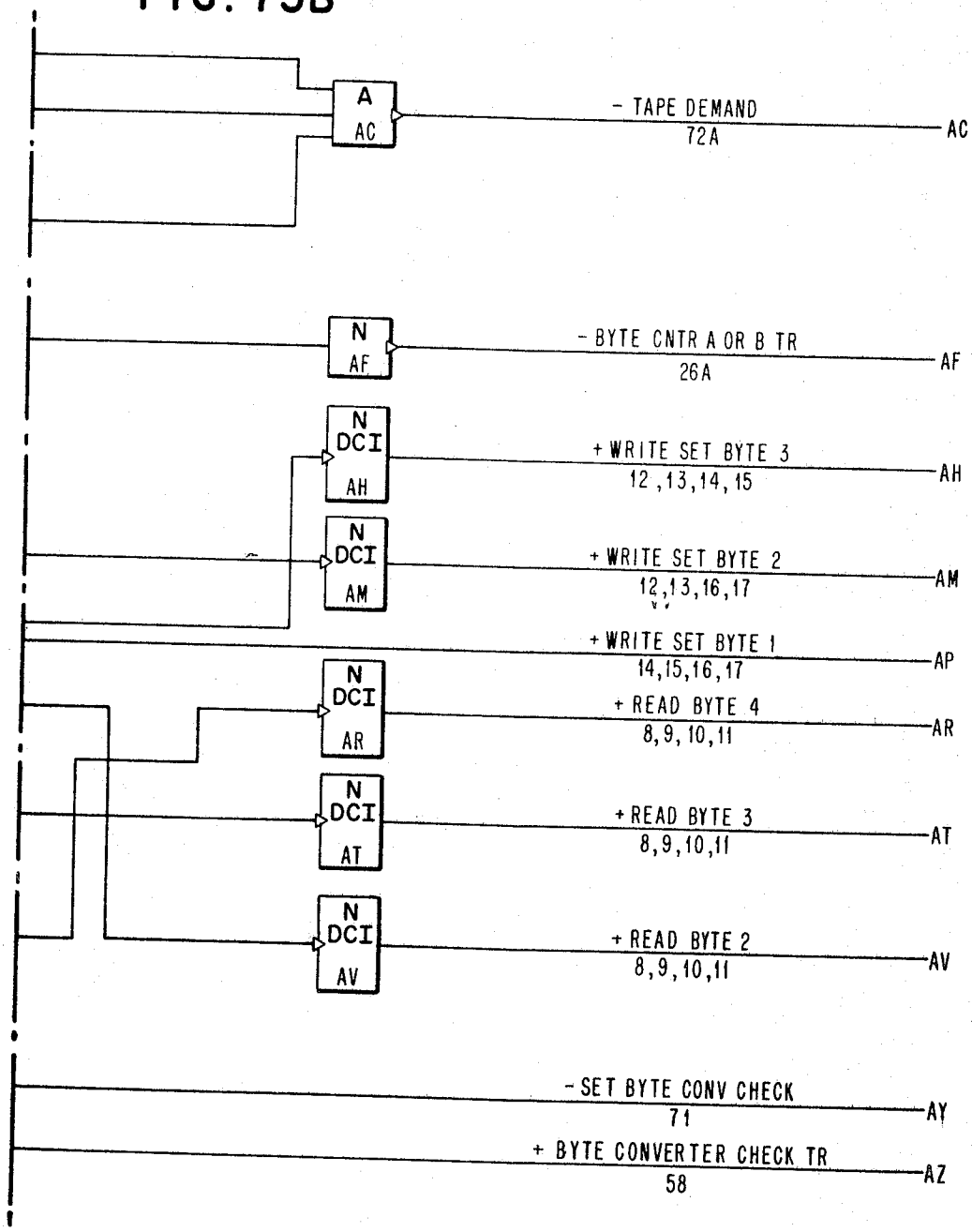

A
AC
- TAPE DEMAND
72A
AC
N
AF
- BYTE CNTR A OR B TR
26A
AF
N
DCI
AH
+ WRITE SET BYTE 3
12,13,14,15
AH
N
DCI
AM
+ WRITE SET BYTE 2
12,13,16,17
AM
+ WRITE SET BYTE 1
14,15,16,17
AP
N
DCI
AR
+ READ BYTE 4
8,9,10,11
AR
N
DCI
AT
+ READ BYTE 3
8,9,10,11
AT
N
DCI
AV
+ READ BYTE 2
8,9,10,11
AV
- SET BYTE CONV CHECK
71
AY
+ BYTE CONVERTER CHECK TR
58
AZ

FIG. 76

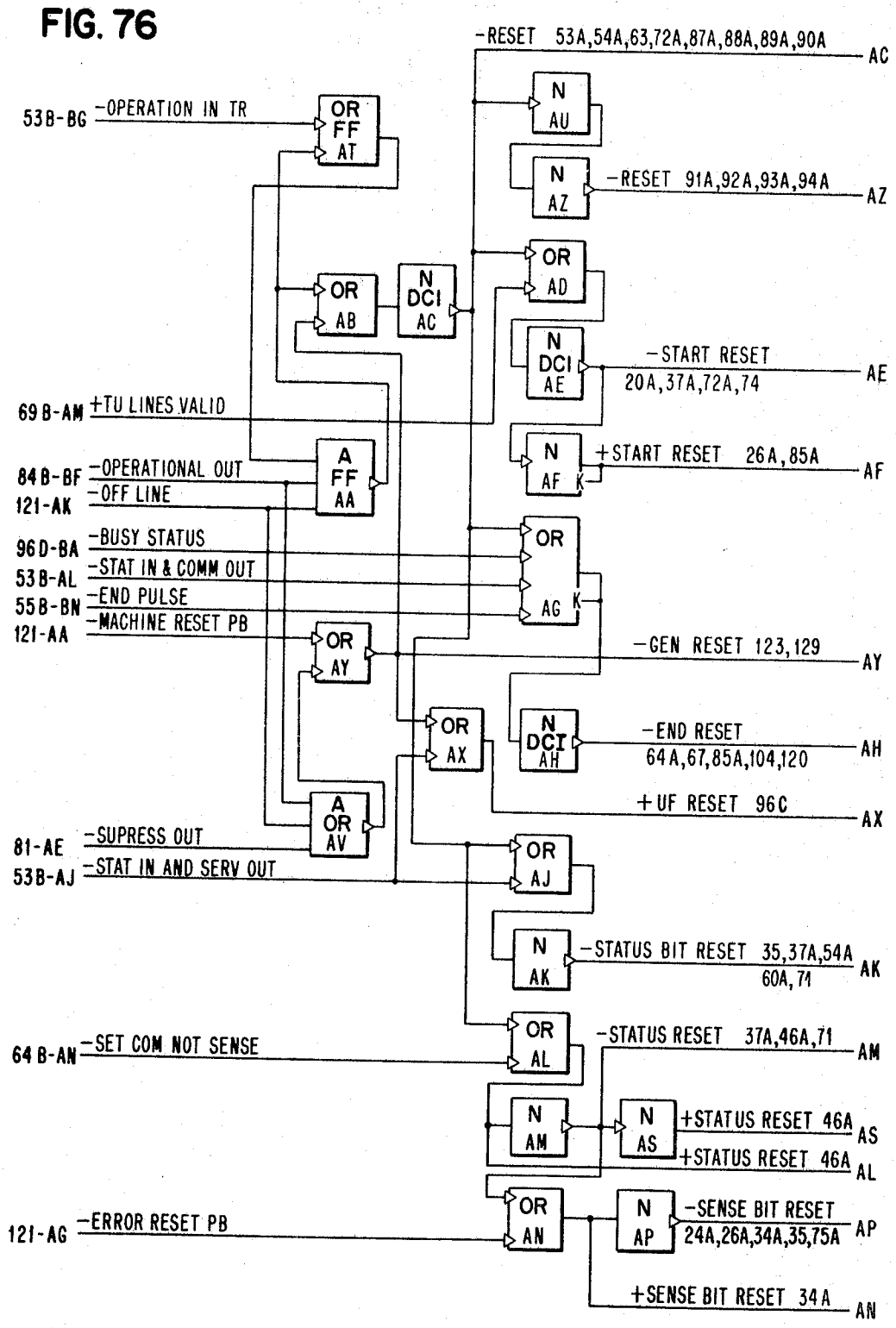

-RESET 53A,54A,63,72A,87A,88A,89A,90A
AC
N AU
N AZ
-RESET 91A,92A,93A,94A
AZ
53B-BG -OPERATION IN TR
OR FF AT
OR AB
N DCI AC
OR AD
N DCI AE
-START RESET 20A,37A,72A,74
AE
69B-AM +TU LINES VALID
84B-BF -OPERATIONAL OUT
121-AK -OFF LINE
A FF AA
N AF K
+START RESET 26A,85A
AF
96D-BA -BUSY STATUS
53B-AL -STAT IN & COMM OUT
55B-BN -END PULSE
121-AA -MACHINE RESET PB
OR AG K
OR AY
-GEN RESET 123,129
AY
OR AX
N DCI AH
-END RESET 64A,67,85A,104,120
AH
+UF RESET 96C
AX
A OR AV
81-AE -SUPRESS OUT
53B-AJ -STAT IN AND SERV OUT
OR AJ
N AK
-STATUS BIT RESET 35,37A,54A 60A,71
AK
64B-AN -SET COM NOT SENSE
OR AL
-STATUS RESET 37A,46A,71
AM
N AM
N AS
+STATUS RESET 46A
AS
+STATUS RESET 46A
AL
121-AG -ERROR RESET PB
OR AN
N AP
-SENSE BIT RESET 24A,26A,34A,35,75A
AP
+SENSE BIT RESET 34A
AN

FIG. 77

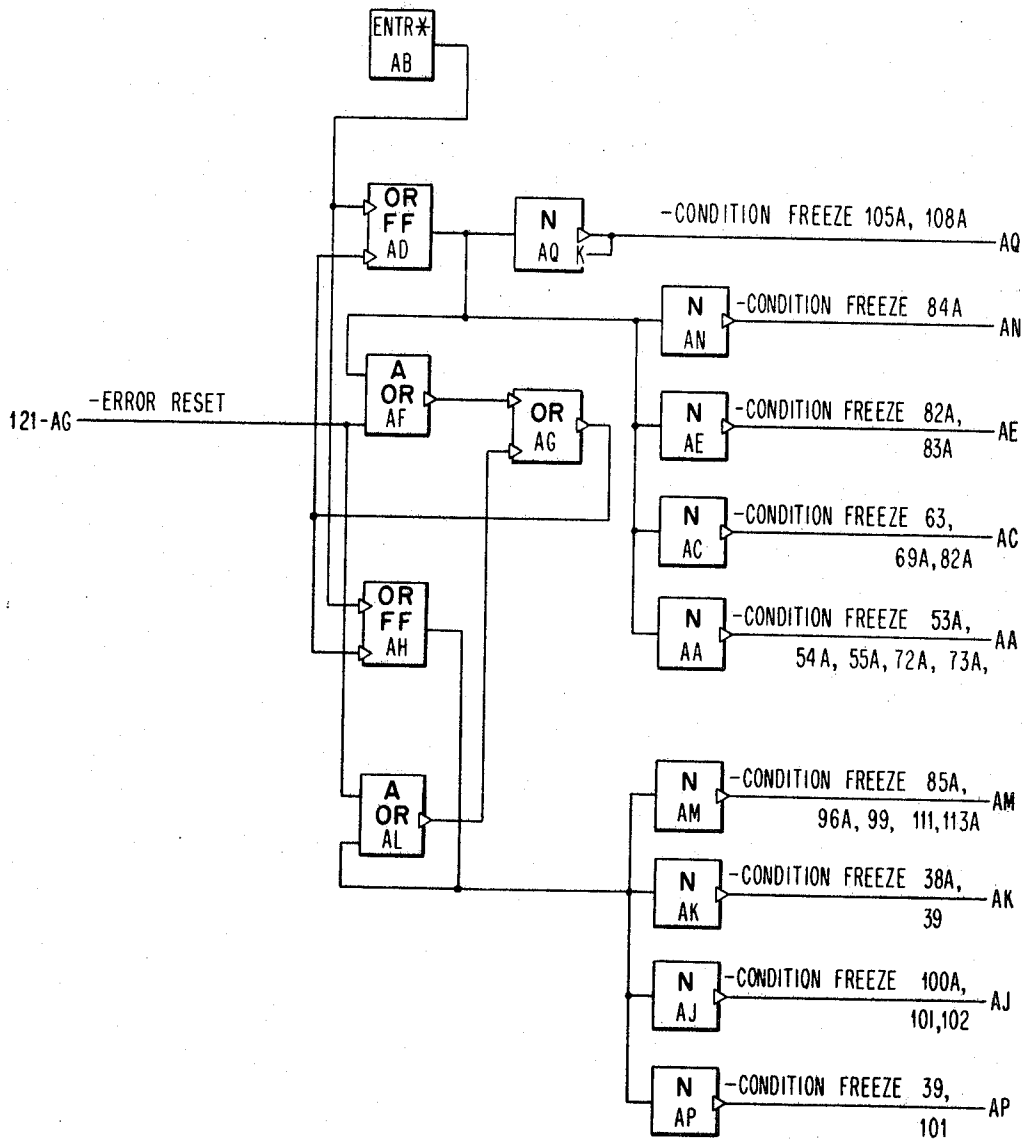

ENTR*
AB
OR FF AD
N AQ K
-CONDITION FREEZE 105A, 108A
AQ
N AN
-CONDITION FREEZE 84A
AN
A OR AF
OR AG
121-AG
-ERROR RESET
N AE
-CONDITION FREEZE 82A, 83A
AE
N AC
-CONDITION FREEZE 63, 69A, 82A
AC
N AA
-CONDITION FREEZE 53A, 54A, 55A, 72A, 73A,
AA
OR FF AH
A OR AL
N AM
-CONDITION FREEZE 85A, 96A, 99, 111, 113A
AM
N AK
-CONDITION FREEZE 38A, 39
AK
N AJ
-CONDITION FREEZE 100A, 101, 102
AJ
N AP
-CONDITION FREEZE 39, 101
AP

FIG. 78

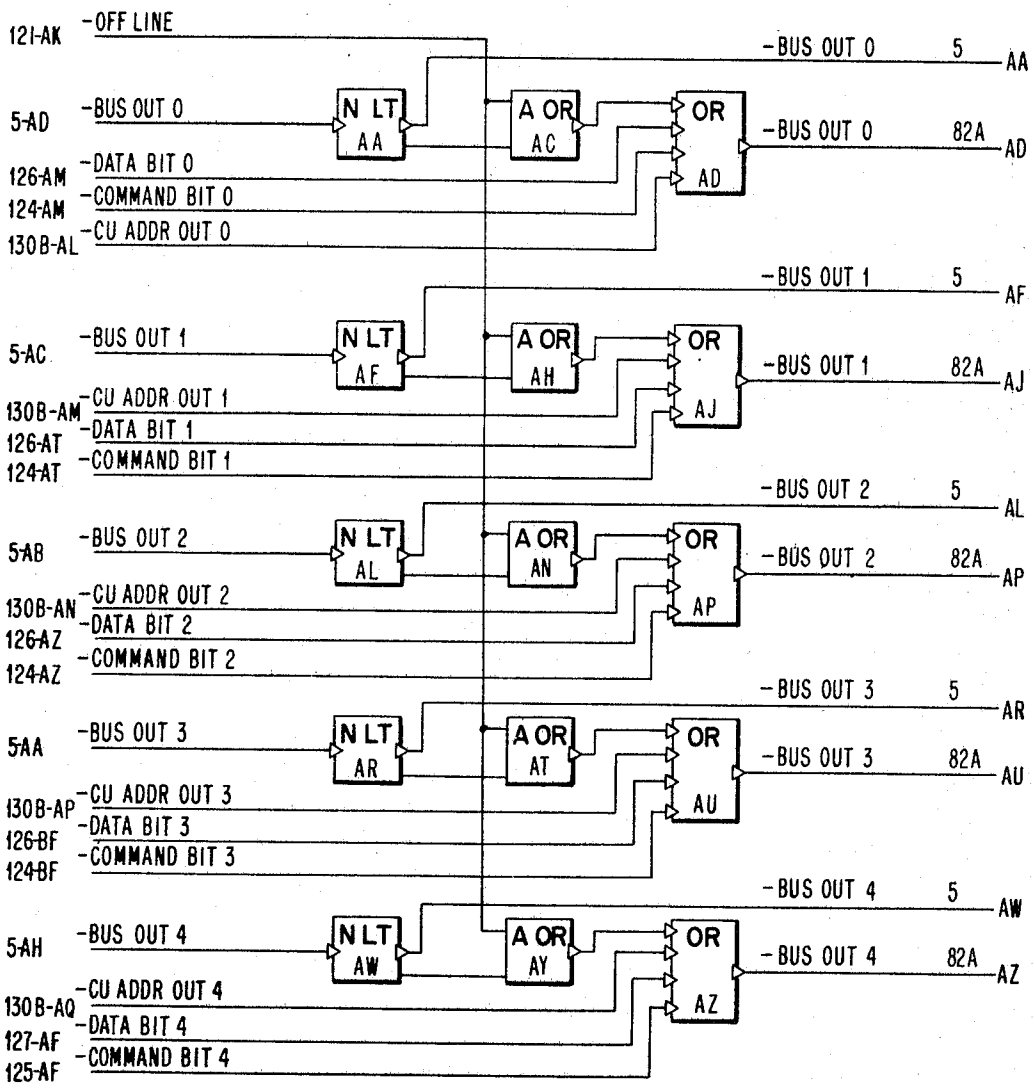

121-AK
-OFF LINE
-BUS OUT 0
5
AA
5-AD
-BUS OUT 0
N LT
AA
A OR
AC
OR
AD
-BUS OUT 0
82A
AD
126-AM
-DATA BIT 0
124-AM
-COMMAND BIT 0
130B-AL
-CU ADDR OUT 0
-BUS OUT 1
5
AF
5-AC
-BUS OUT 1
N LT
AF
A OR
AH
OR
AJ
-BUS OUT 1
82A
AJ
130B-AM
-CU ADDR OUT 1
126-AT
-DATA BIT 1
124-AT
-COMMAND BIT 1
-BUS OUT 2
5
AL
5-AB
-BUS OUT 2
N LT
AL
A OR
AN
OR
AP
-BUS OUT 2
82A
AP
130B-AN
-CU ADDR OUT 2
126-AZ
-DATA BIT 2
124-AZ
-COMMAND BIT 2
-BUS OUT 3
5
AR
5-AA
-BUS OUT 3
N LT
AR
A OR
AT
OR
AU
-BUS OUT 3
82A
AU
130B-AP
-CU ADDR OUT 3
126-BF
-DATA BIT 3
124-BF
-COMMAND BIT 3
-BUS OUT 4
5
AW
5-AH
-BUS OUT 4
N LT
AW
A OR
AY
OR
AZ
-BUS OUT 4
82A
AZ
130B-AQ
-CU ADDR OUT 4
127-AF
-DATA BIT 4
125-AF
-COMMAND BIT 4

FIG. 79

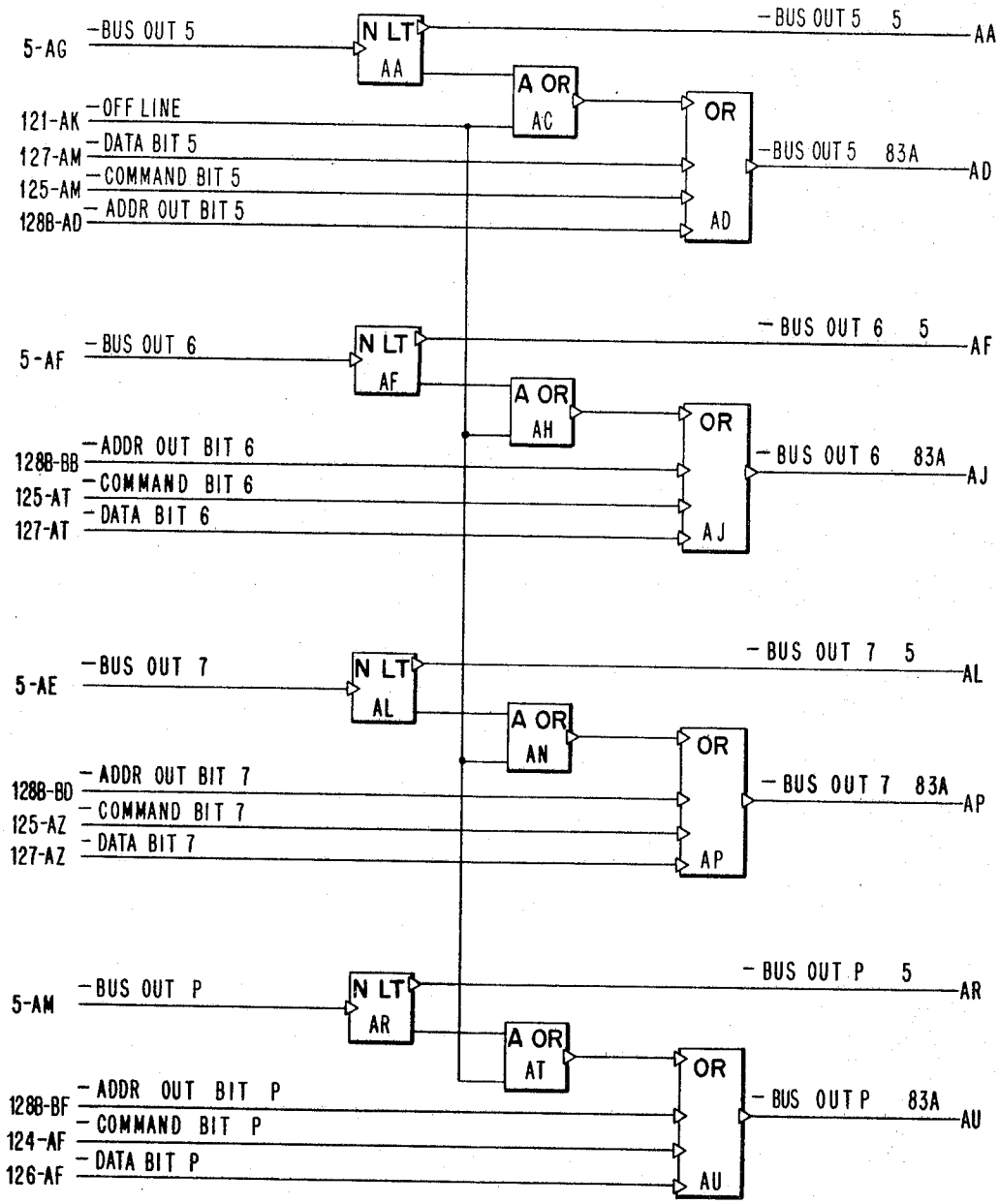

5-AG
-BUS OUT 5
N LT AA
-BUS OUT 5 5
AA
A OR AC
121-AK
-OFF LINE
127-AM
-DATA BIT 5
125-AM
-COMMAND BIT 5
128B-AD
-ADDR OUT BIT 5
OR AD
-BUS OUT 5 83A
AD
5-AF
-BUS OUT 6
N LT AF
-BUS OUT 6 5
AF
A OR AH
128B-BB
-ADDR OUT BIT 6
125-AT
-COMMAND BIT 6
127-AT
-DATA BIT 6
OR AJ
-BUS OUT 6 83A
AJ
5-AE
-BUS OUT 7
N LT AL
-BUS OUT 7 5
AL
A OR AN
128B-BD
-ADDR OUT BIT 7
125-AZ
-COMMAND BIT 7
127-AZ
-DATA BIT 7
OR AP
-BUS OUT 7 83A
AP
5-AM
-BUS OUT P
N LT AR
-BUS OUT P 5
AR
A OR AT
128B-BF
-ADDR OUT BIT P
124-AF
-COMMAND BIT P
126-AF
-DATA BIT P
OR AU
-BUS OUT P 83A
AU

FIG. 80A

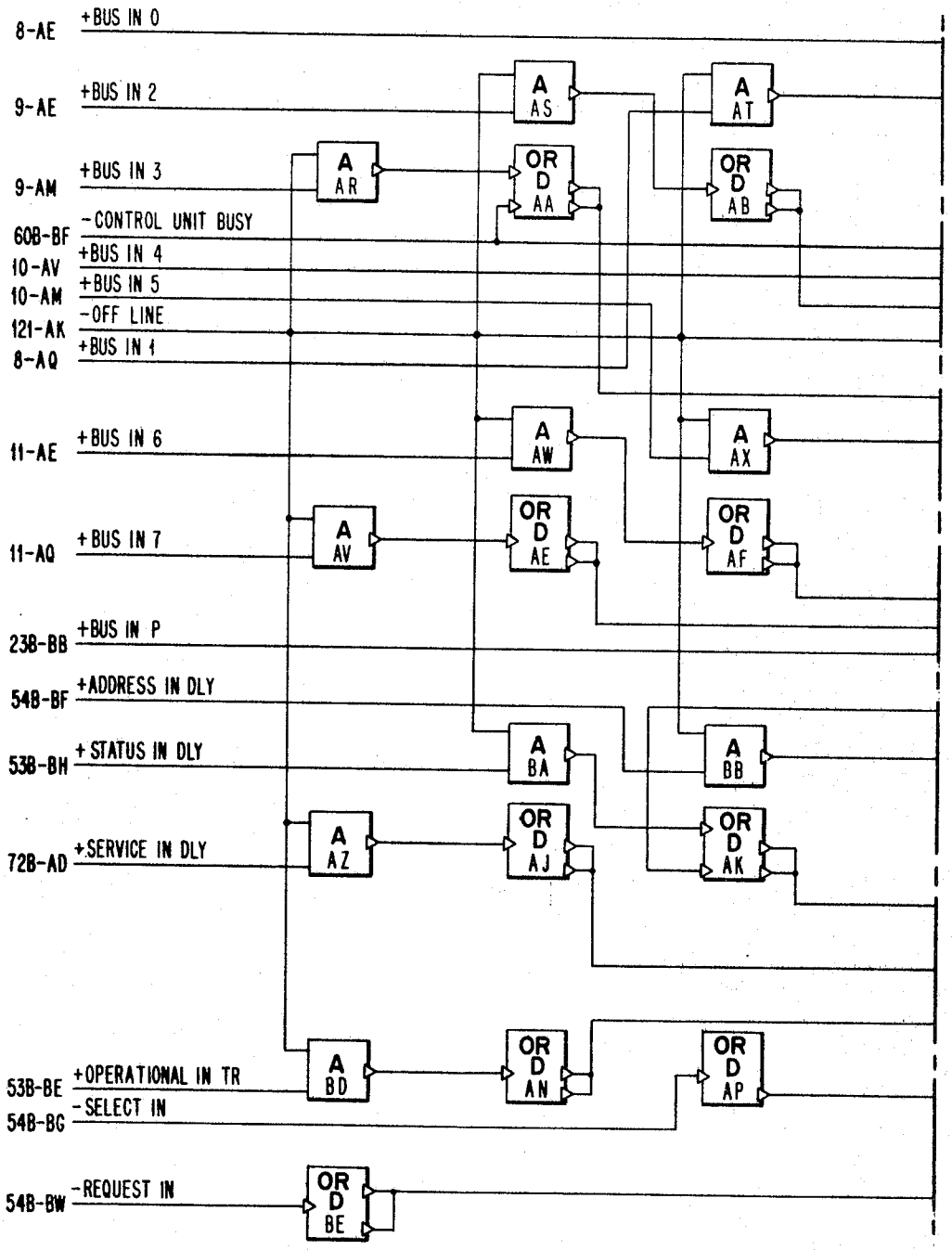

8-AE +BUS IN 0
9-AE +BUS IN 2
9-AM +BUS IN 3
60B-BF -CONTROL UNIT BUSY
10-AV +BUS IN 4
10-AM +BUS IN 5
121-AK -OFF LINE
8-AQ +BUS IN 1
11-AE +BUS IN 6
11-AQ +BUS IN 7
23B-BB +BUS IN P
54B-BF +ADDRESS IN DLY
53B-BH +STATUS IN DLY
72B-AD +SERVICE IN DLY
53B-BE +OPERATIONAL IN TR
54B-BG -SELECT IN
54B-BW -REQUEST IN
A AR
A AS
A AT
OR D AA
OR D AB
A AW
A AX
A AV
OR D AE
OR D AF
A BA
A BB
A AZ
OR D AJ
OR D AK
A BD
OR D AN
OR D AP
OR D BE

FIG. 80B

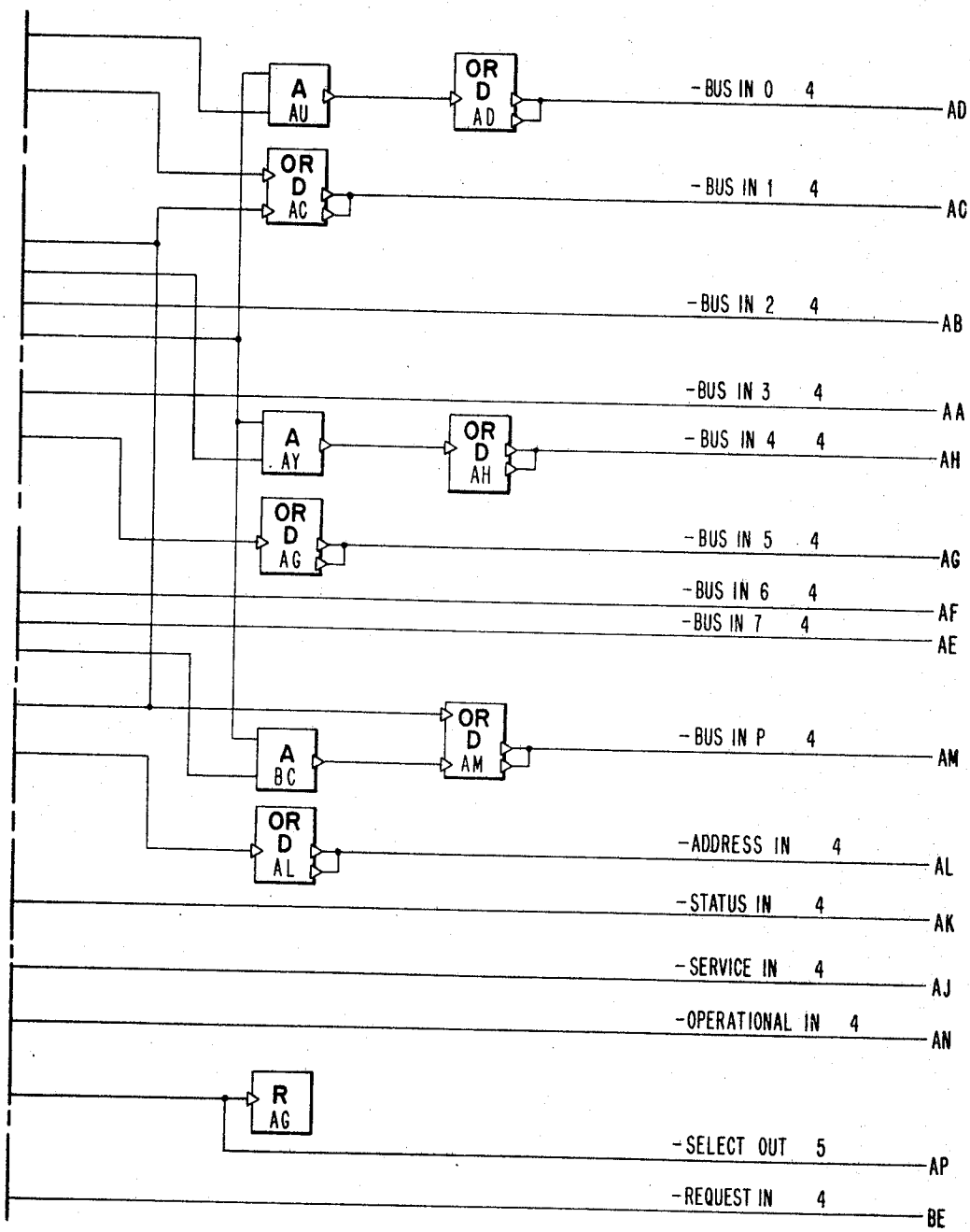

A AU
OR D AD
-BUS IN 0 4
AD
OR D AC
-BUS IN 1 4
AC
-BUS IN 2 4
AB
-BUS IN 3 4
AA
A AY
OR D AH
-BUS IN 4 4
AH
OR D AG
-BUS IN 5 4
AG
-BUS IN 6 4
AF
-BUS IN 7 4
AE
A BC
OR D AM
-BUS IN P 4
AM
OR D AL
-ADDRESS IN 4
AL
-STATUS IN 4
AK
-SERVICE IN 4
AJ
-OPERATIONAL IN 4
AN
R AG
-SELECT OUT 5
AP
-REQUEST IN 4
BE

FIG.81

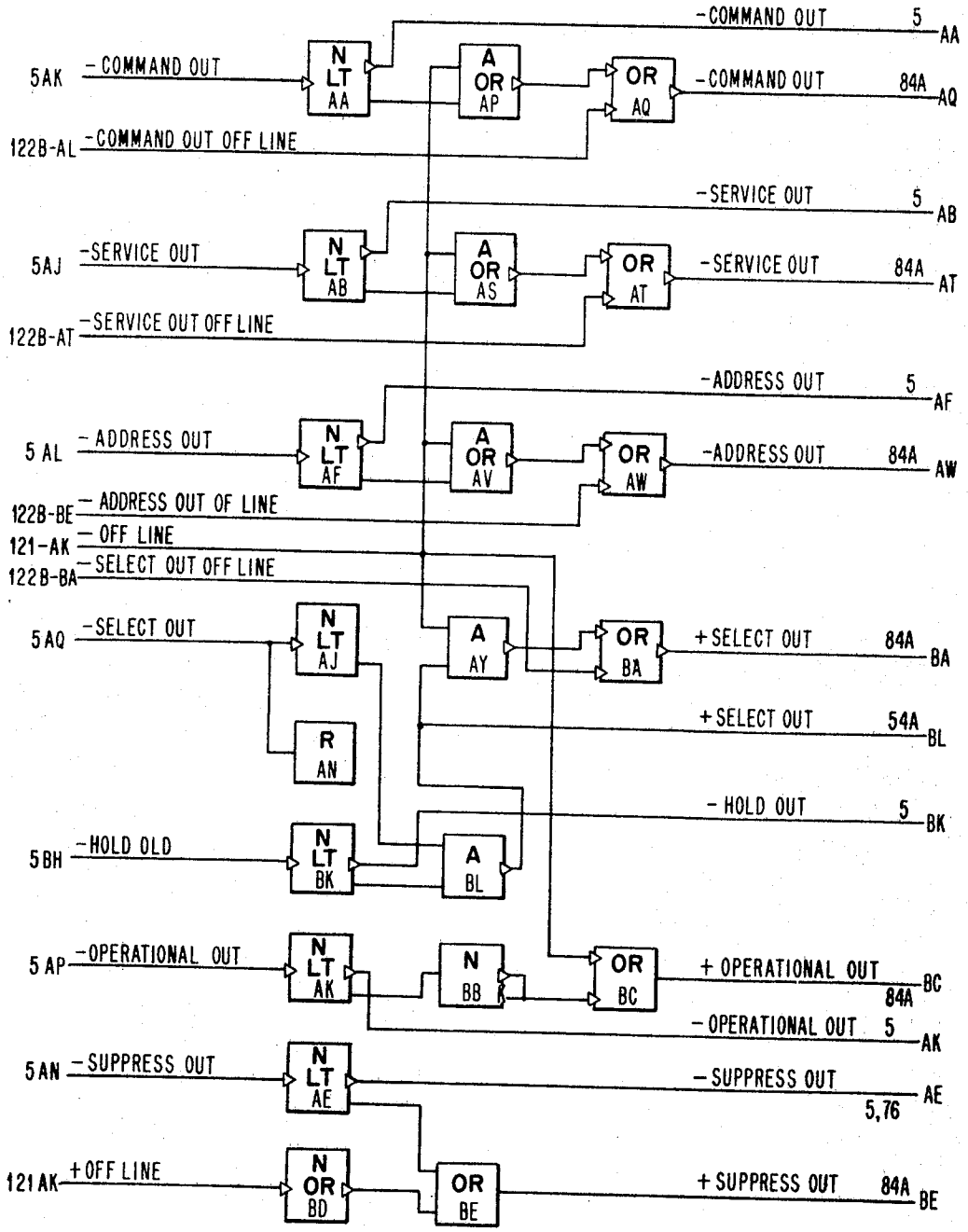

-COMMAND OUT 5 AA
5AK -COMMAND OUT
N LT AA
A OR AP
OR AQ
-COMMAND OUT 84A AQ
122B-AL -COMMAND OUT OFF LINE
-SERVICE OUT 5 AB
5AJ -SERVICE OUT
N LT AB
A OR AS
OR AT
-SERVICE OUT 84A AT
122B-AT -SERVICE OUT OFF LINE
-ADDRESS OUT 5 AF
5 AL -ADDRESS OUT
N LT AF
A OR AV
OR AW
-ADDRESS OUT 84A AW
122B-BE -ADDRESS OUT OF LINE
121-AK -OFF LINE
122B-BA -SELECT OUT OFF LINE
5AQ -SELECT OUT
N LT AJ
A AY
OR BA
+SELECT OUT 84A BA
R AN
+SELECT OUT 54A BL
-HOLD OUT 5 BK
5BH -HOLD OLD
N LT BK
A BL
5AP -OPERATIONAL OUT
N LT AK
N BB
OR BC
+OPERATIONAL OUT 84A BC
-OPERATIONAL OUT 5 AK
5AN -SUPPRESS OUT
N LT AE
-SUPPRESS OUT 5,76 AE
121AK +OFF LINE
N OR BD
OR BE
+SUPPRESS OUT 84A BE

FIG. 82A

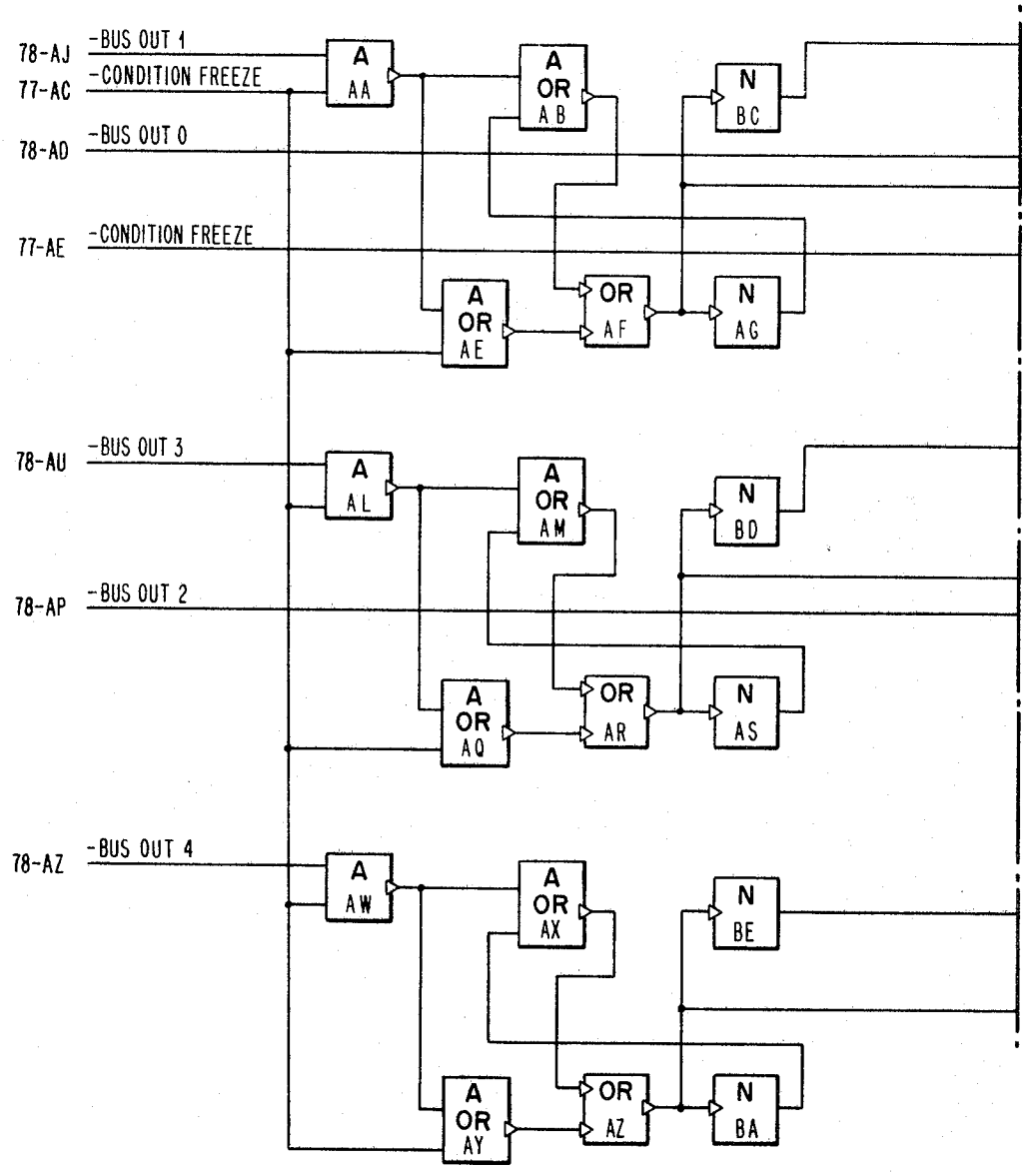

78-AJ
-BUS OUT 1
77-AC
-CONDITION FREEZE
78-AD
-BUS OUT 0
77-AE
-CONDITION FREEZE
78-AU
-BUS OUT 3
78-AP
-BUS OUT 2
78-AZ
-BUS OUT 4
A AA
A OR AB
N BC
A OR AE
OR AF
N AG
A AL
A OR AM
N BD
A OR AQ
OR AR
N AS
A AW
A OR AX
N BE
A OR AY
OR AZ
N BA

FIG. 82B

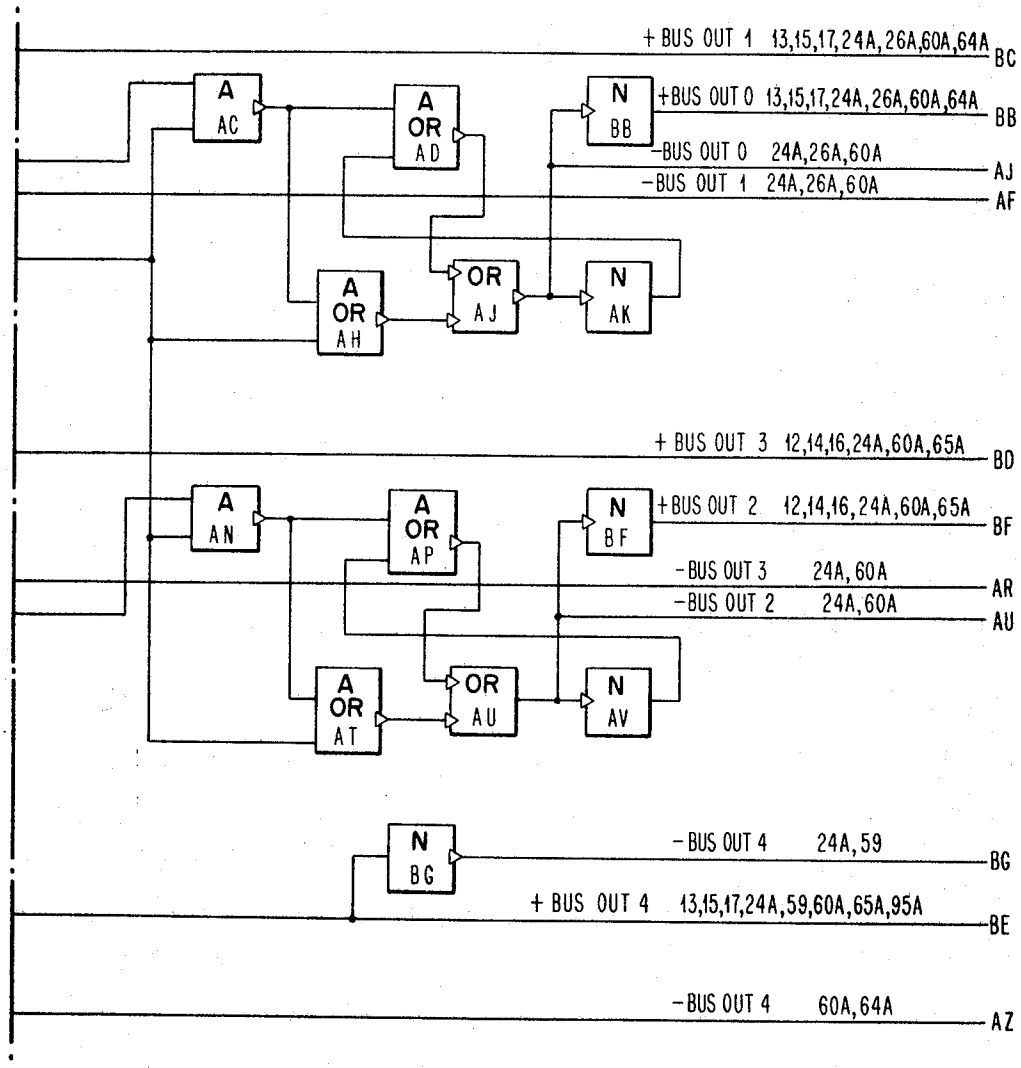

+ BUS OUT 1 13,15,17,24A,26A,60A,64A
BC
A AC
A OR AD
N BB
+BUS OUT 0 13,15,17,24A,26A,60A,64A
BB
-BUS OUT 0 24A,26A,60A
AJ
-BUS OUT 1 24A,26A,60A
AF
A OR AH
OR AJ
N AK
+ BUS OUT 3 12,14,16,24A,60A,65A
BD
A AN
A OR AP
N BF
+BUS OUT 2 12,14,16,24A,60A,65A
BF
-BUS OUT 3 24A,60A
AR
-BUS OUT 2 24A,60A
AU
A OR AT
OR AU
N AV
N BG
-BUS OUT 4 24A,59
BG
+ BUS OUT 4 13,15,17,24A,59,60A,65A,95A
BE
-BUS OUT 4 60A,64A
AZ

FIG. 83A

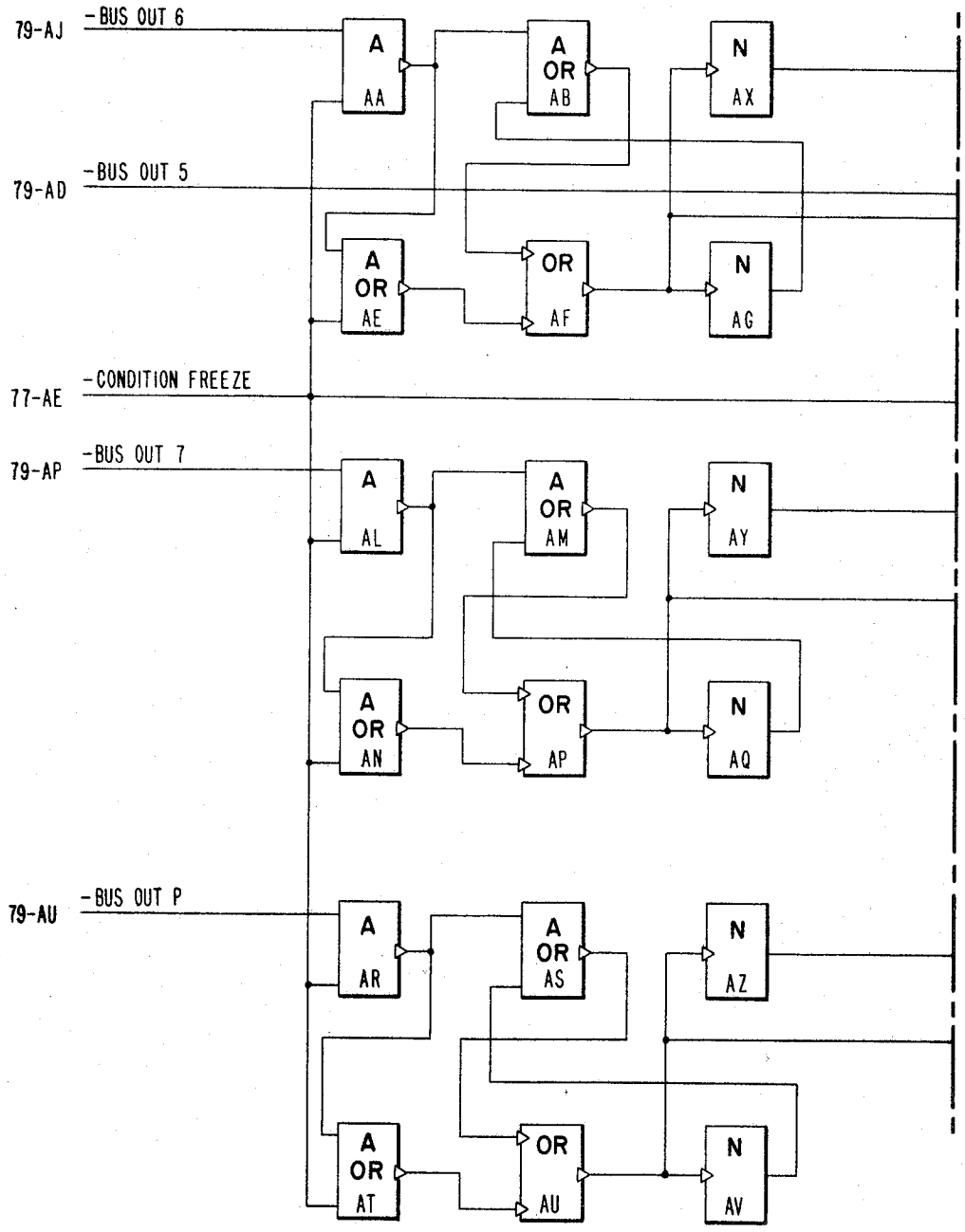

79-AJ
-BUS OUT 6
79-AD
-BUS OUT 5
77-AE
-CONDITION FREEZE
79-AP
-BUS OUT 7
79-AU
-BUS OUT P
A AA
A OR AB
N AX
A OR AE
OR AF
N AG
A AL
A OR AM
N AY
A OR AN
OR AP
N AQ
A AR
A OR AS
N AZ
A OR AT
OR AU
N AV

FIG. 83B

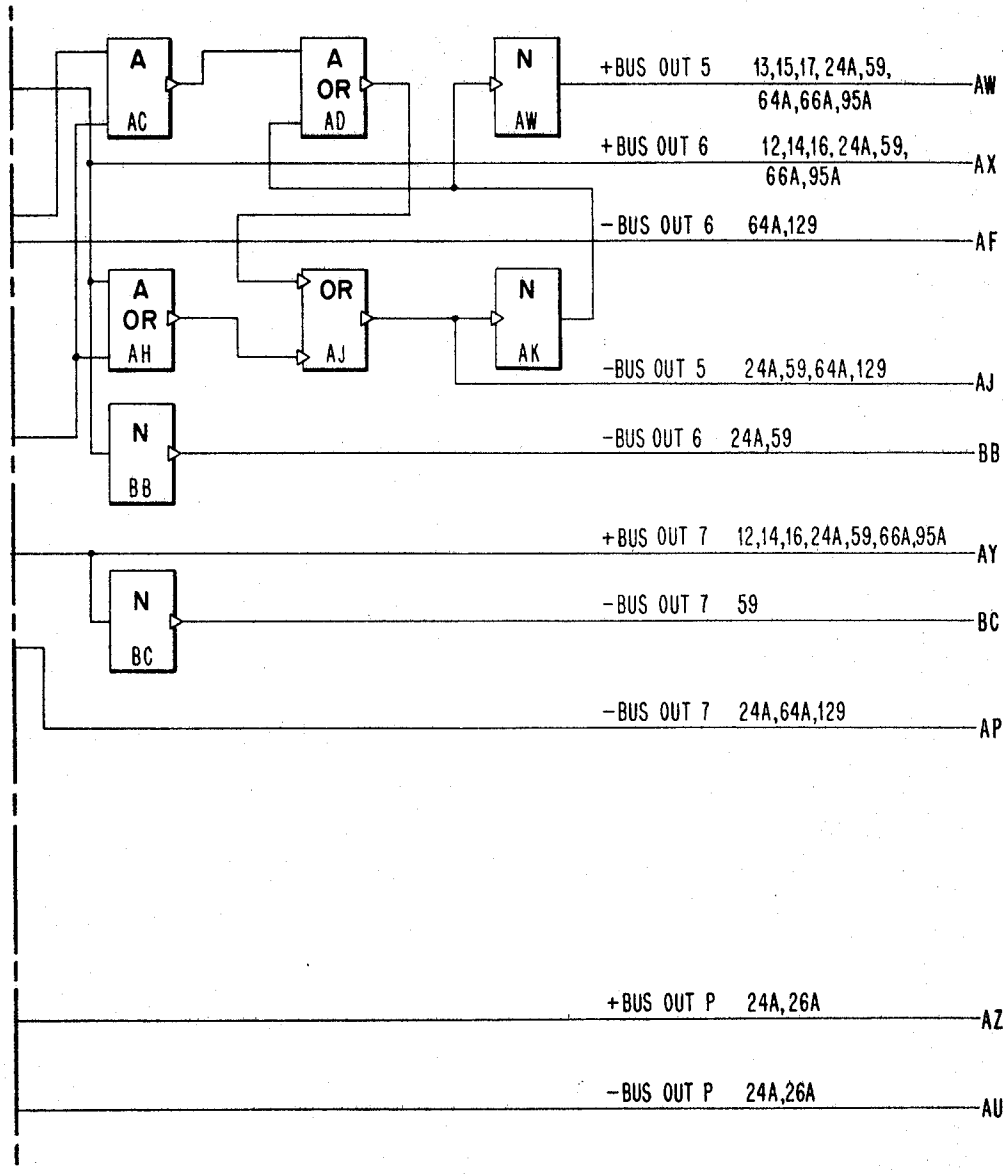

A
AC
A
OR
AD
N
AW
+BUS OUT 5 13,15,17,24A,59,
64A,66A,95A
AW
+BUS OUT 6 12,14,16,24A,59,
66A,95A
AX
-BUS OUT 6 64A,129
AF
A
OR
AH
OR
AJ
N
AK
-BUS OUT 5 24A,59,64A,129
AJ
N
BB
-BUS OUT 6 24A,59
BB
+BUS OUT 7 12,14,16,24A,59,66A,95A
AY
N
BC
-BUS OUT 7 59
BC
-BUS OUT 7 24A,64A,129
AP
+BUS OUT P 24A,26A
AZ
-BUS OUT P 24A,26A
AU

FIG. 84A

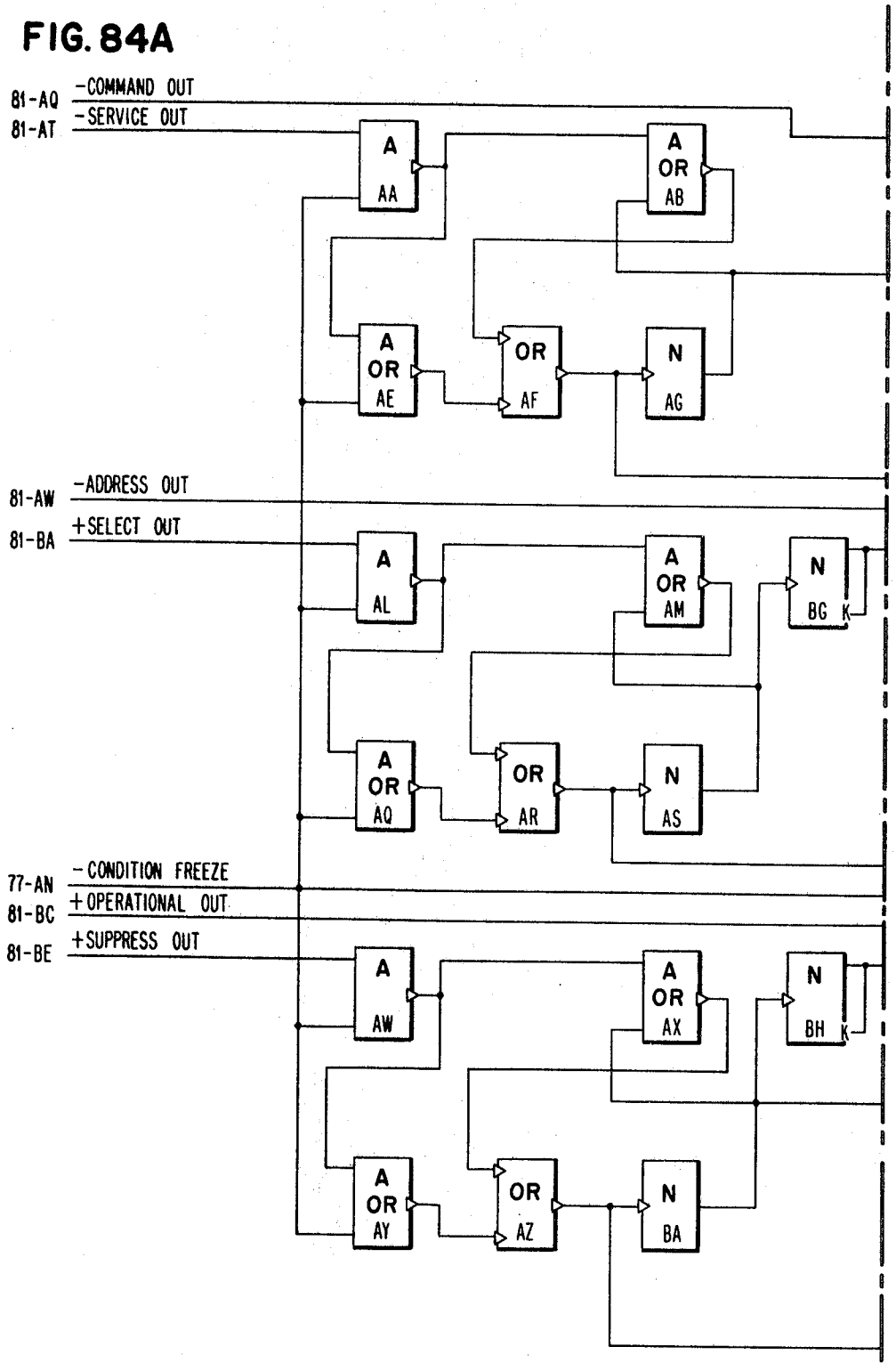

81-AQ
-COMMAND OUT
81-AT
-SERVICE OUT
A
AA
A
OR
AB
A
OR
AE
OR
AF
N
AG
81-AW
-ADDRESS OUT
81-BA
+SELECT OUT
A
AL
A
OR
AM
N
BG
K
A
OR
AQ
OR
AR
N
AS
77-AN
-CONDITION FREEZE
81-BC
+OPERATIONAL OUT
81-BE
+SUPPRESS OUT
A
AW
A
OR
AX
N
BH
K
A
OR
AY
OR
AZ
N
BA

FIG. 84B

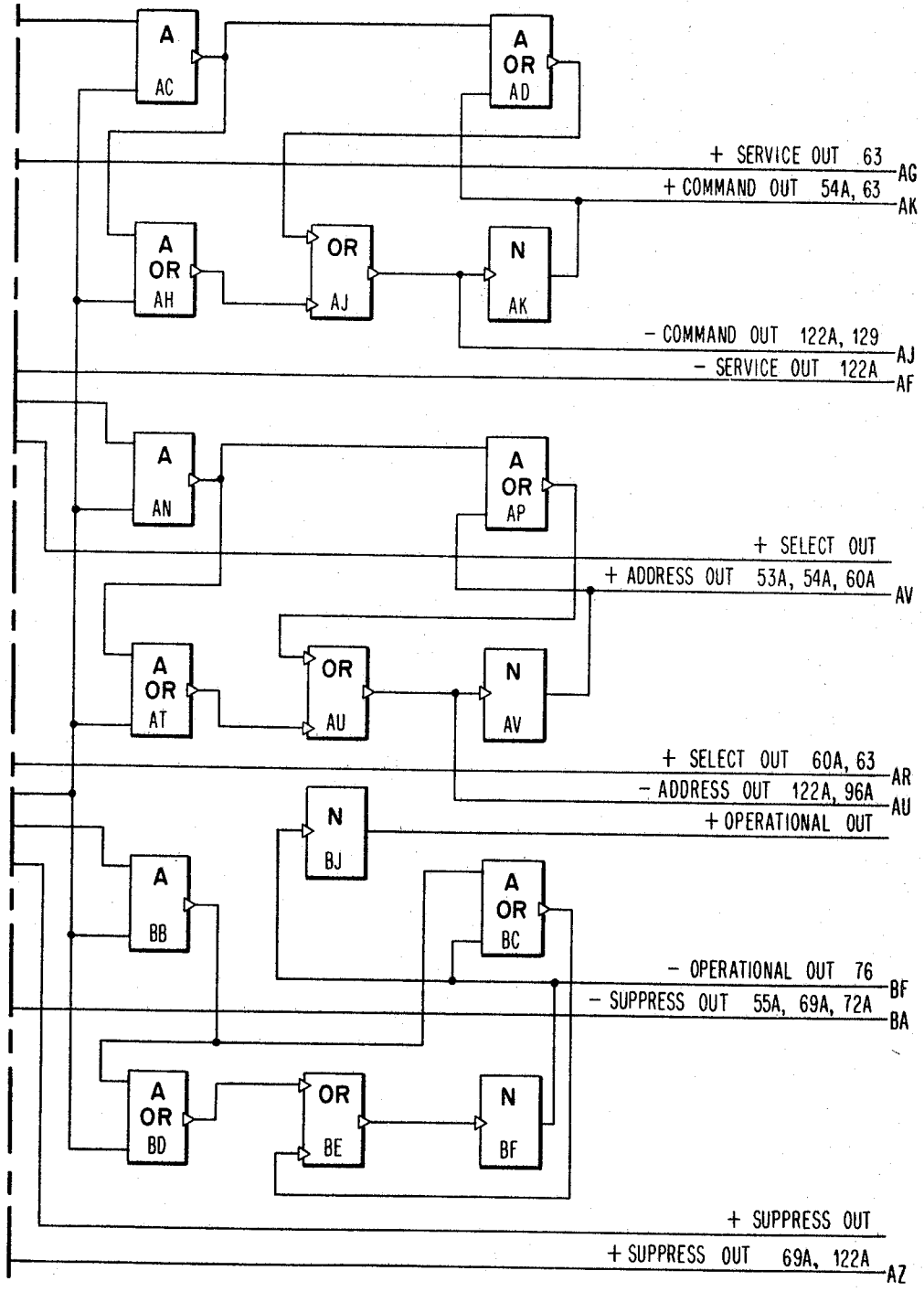

A AC
A OR AD
+ SERVICE OUT 63
AG
+ COMMAND OUT 54A, 63
AK
A OR AH
OR AJ
N AK
- COMMAND OUT 122A, 129
AJ
- SERVICE OUT 122A
AF
A AN
A OR AP
+ SELECT OUT
+ ADDRESS OUT 53A, 54A, 60A
AV
A OR AT
OR AU
N AV
+ SELECT OUT 60A, 63
AR
- ADDRESS OUT 122A, 96A
AU
+ OPERATIONAL OUT
N BJ
A BB
A OR BC
- OPERATIONAL OUT 76
BF
- SUPPRESS OUT 55A, 69A, 72A
BA
A OR BD
OR BE
N BF
+ SUPPRESS OUT
+ SUPPRESS OUT 69A, 122A
AZ

FIG. 85A

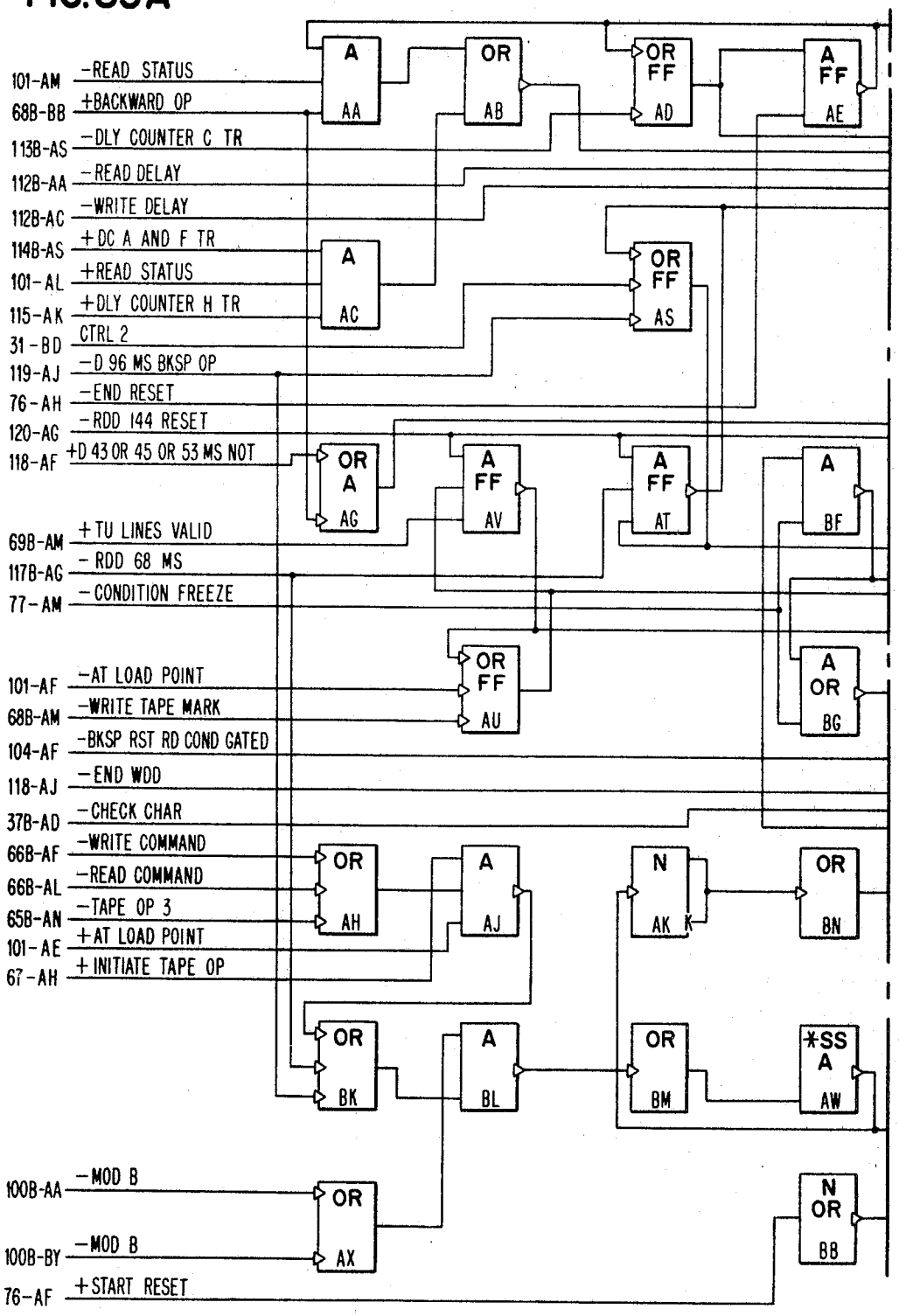

101-AM -READ STATUS
68B-BB +BACKWARD OP
113B-AS -DLY COUNTER C TR
112B-AA -READ DELAY
112B-AC -WRITE DELAY
114B-AS +DC A AND F TR
101-AL +READ STATUS
115-AK +DLY COUNTER H TR
31-BD CTRL 2
119-AJ -D 96 MS BKSP OP
76-AH -END RESET
120-AG -RDD 144 RESET
118-AF +D 43 OR 45 OR 53 MS NOT
69B-AM +TU LINES VALID
117B-AG -RDD 68 MS
77-AM -CONDITION FREEZE
101-AF -AT LOAD POINT
68B-AM -WRITE TAPE MARK
104-AF -BKSP RST RD COND GATED
118-AJ -END WDD
37B-AD -CHECK CHAR
66B-AF -WRITE COMMAND
66B-AL -READ COMMAND
65B-AN -TAPE OP 3
101-AE +AT LOAD POINT
67-AH +INITIATE TAPE OP
100B-AA -MOD B
100B-BY -MOD B
76-AF +START RESET
A AA
OR AB
OR FF AD
A FF AE
A AC
OR FF AS
OR A AG
A FF AV
A FF AT
A BF
OR FF AU
A OR BG
OR AH
A AJ
N AK
OR BN
OR BK
A BL
OR BM
*SS A AW
OR AX
N OR BB

FIG.85B

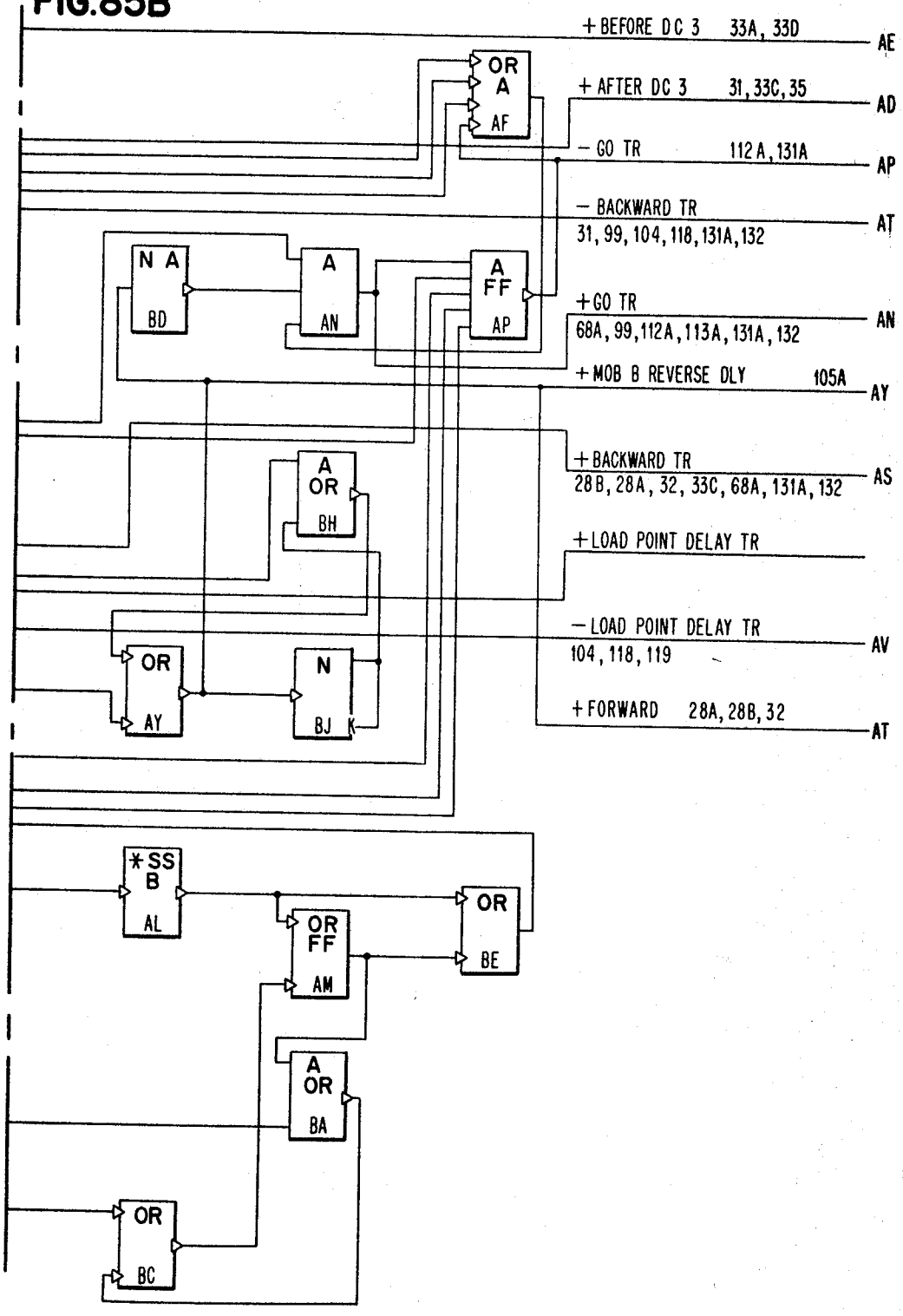

+BEFORE DC 3 33A, 33D
AE
OR A AF
+AFTER DC 3 31, 33C, 35
AD
-GO TR 112A, 131A
AP
-BACKWARD TR
31, 99, 104, 118, 131A, 132
AT
N A BD
A AN
A FF AP
+GO TR
68A, 99, 112A, 113A, 131A, 132
AN
+MOB B REVERSE DLY 105A
AY
+BACKWARD TR
28B, 28A, 32, 33C, 68A, 131A, 132
AS
A OR BH
+LOAD POINT DELAY TR
-LOAD POINT DELAY TR
104, 118, 119
AV
OR AY
N BJ K
+FORWARD 28A, 28B, 32
AT
*SS B AL
OR FF AM
OR BE
A OR BA
OR BC

FIG. 86

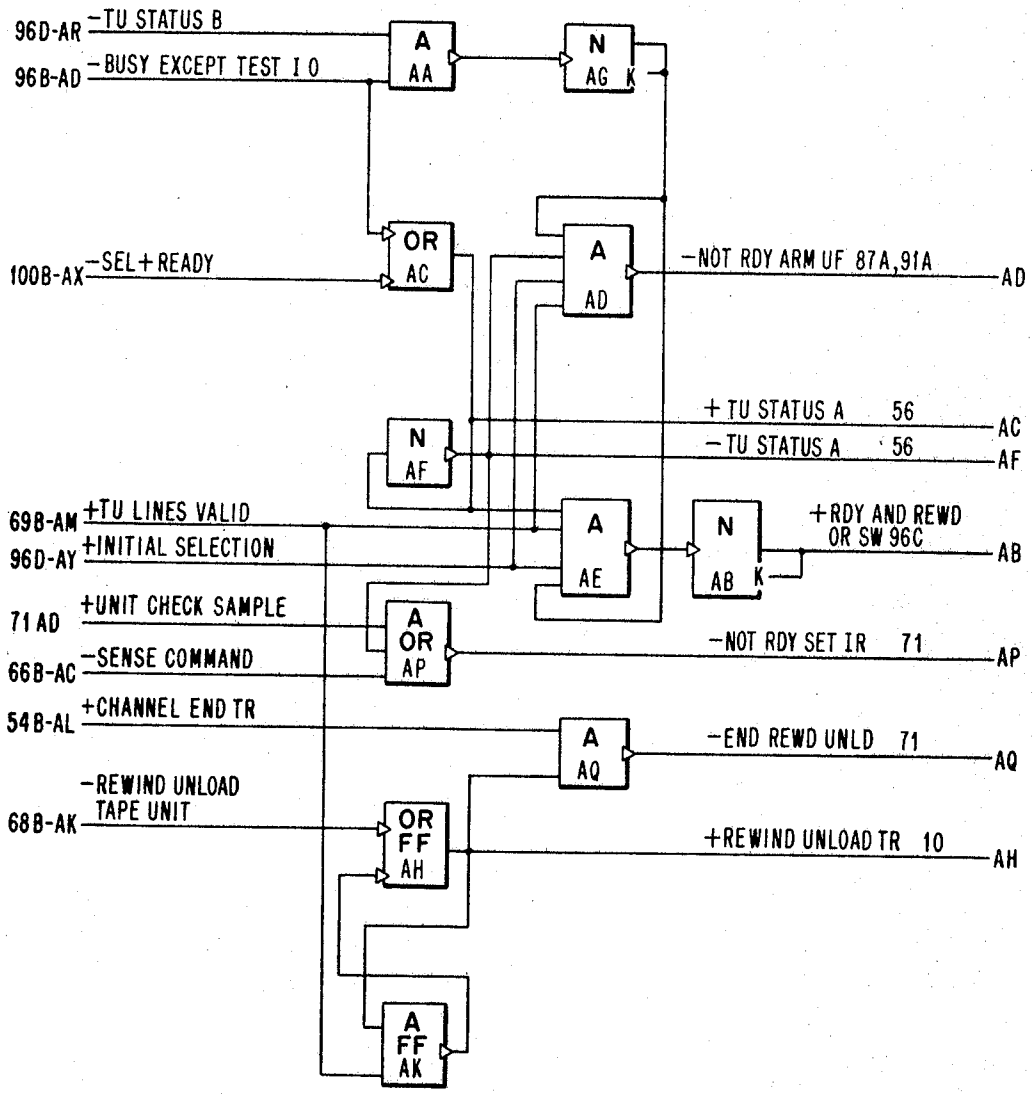

96D-AR
-TU STATUS B
96B-AD
-BUSY EXCEPT TEST I O
A AA
N AG K
OR AC
100B-AX
-SEL+READY
A AD
-NOT RDY ARM UF 87A,91A
AD
+TU STATUS A 56
AC
-TU STATUS A 56
AF
N AF
69B-AM
+TU LINES VALID
96D-AY
+INITIAL SELECTION
A AE
N AB K
+RDY AND REWD OR SW 96C
AB
71 AD
+UNIT CHECK SAMPLE
66B-AC
-SENSE COMMAND
A OR AP
-NOT RDY SET IR 71
AP
54B-AL
+CHANNEL END TR
A AQ
-END REWD UNLD 71
AQ
68B-AK
-REWIND UNLOAD TAPE UNIT
OR FF AH
+REWIND UNLOAD TR 10
AH
A FF AK

FIG. 87A

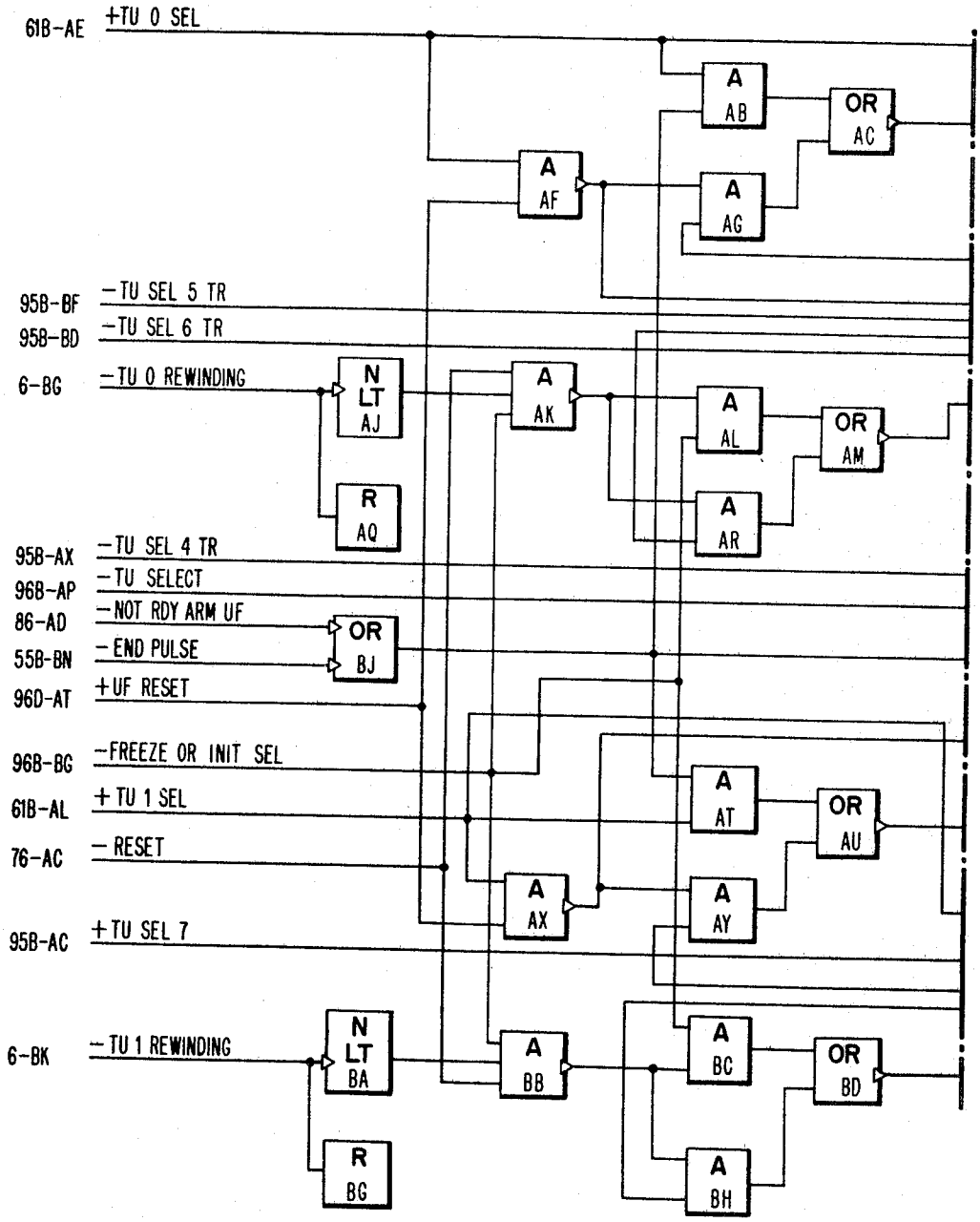

61B-AE +TU 0 SEL
95B-BF −TU SEL 5 TR
95B-BD −TU SEL 6 TR
6-BG −TU 0 REWINDING
95B-AX −TU SEL 4 TR
96B-AP −TU SELECT
86-AD −NOT RDY ARM UF
55B-BN −END PULSE
96D-AT +UF RESET
96B-BG −FREEZE OR INIT SEL
61B-AL +TU 1 SEL
76-AC −RESET
95B-AC +TU SEL 7
6-BK −TU 1 REWINDING
A AB
OR AC
A AF
A AG
N LT AJ
A AK
A AL
OR AM
R AQ
A AR
OR BJ
A AT
OR AU
A AX
A AY
N LT BA
A BB
A BC
OR BD
R BG
A BH

FIG. 87B

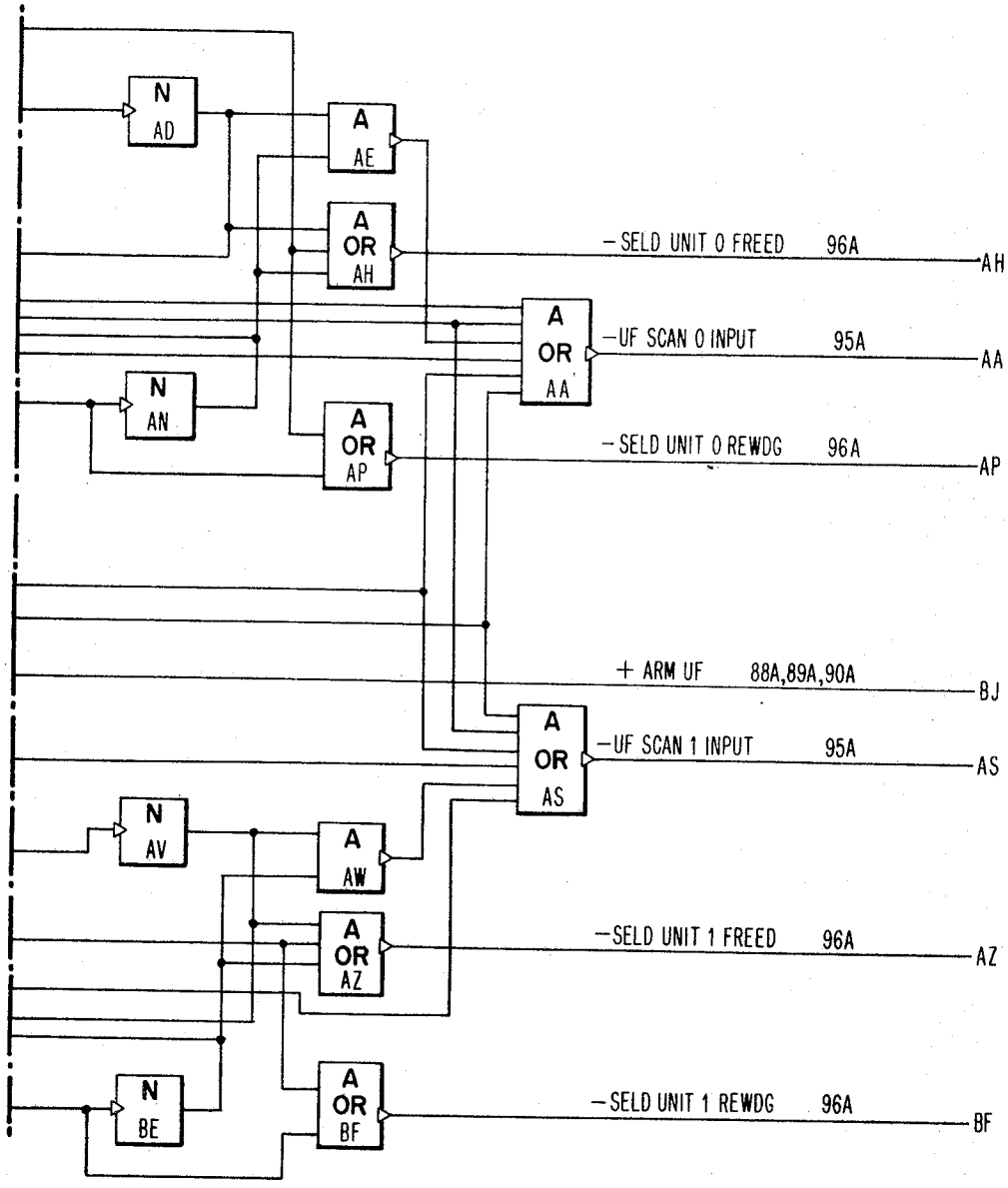

N AD
A AE
A OR AH
−SELD UNIT 0 FREED 96A
AH
A OR AA
−UF SCAN 0 INPUT 95A
AA
N AN
A OR AP
−SELD UNIT 0 REWDG 96A
AP
+ ARM UF 88A,89A,90A
BJ
A OR AS
−UF SCAN 1 INPUT 95A
AS
N AV
A AW
A OR AZ
−SELD UNIT 1 FREED 96A
AZ
N BE
A OR BF
−SELD UNIT 1 REWDG 96A
BF

FIG. 88A

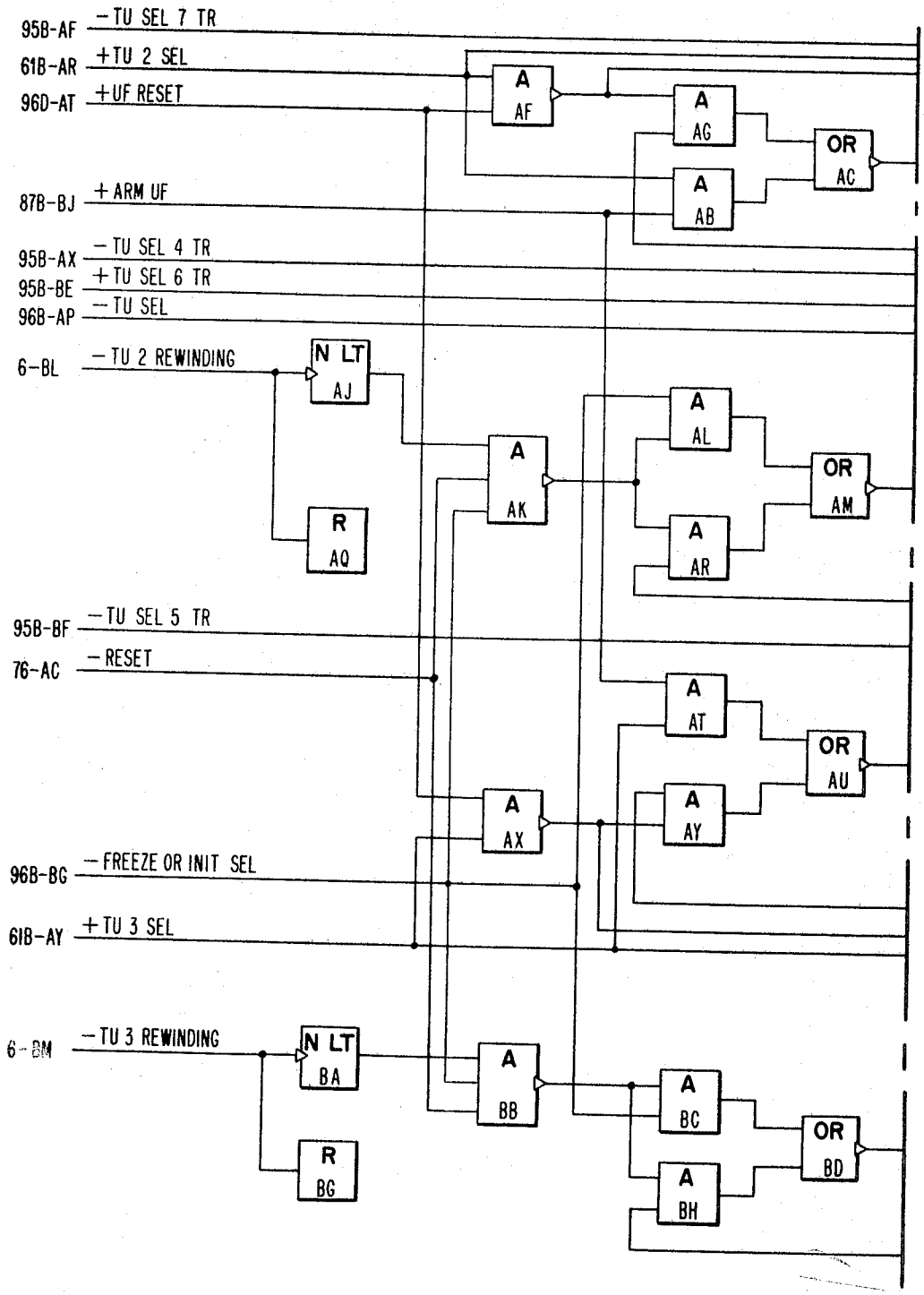

95B-AF
−TU SEL 7 TR
61B-AR
+TU 2 SEL
96D-AT
+UF RESET
87B-BJ
+ARM UF
95B-AX
−TU SEL 4 TR
95B-BE
+TU SEL 6 TR
96B-AP
−TU SEL
6-BL
−TU 2 REWINDING
N LT AJ
R AQ
A AF
A AG
A AB
OR AC
A AK
A AL
A AR
OR AM
95B-BF
−TU SEL 5 TR
76-AC
−RESET
A AT
A AY
OR AU
A AX
96B-BG
−FREEZE OR INIT SEL
61B-AY
+TU 3 SEL
6-BM
−TU 3 REWINDING
N LT BA
R BG
A BB
A BC
A BH
OR BD

FIG. 88B

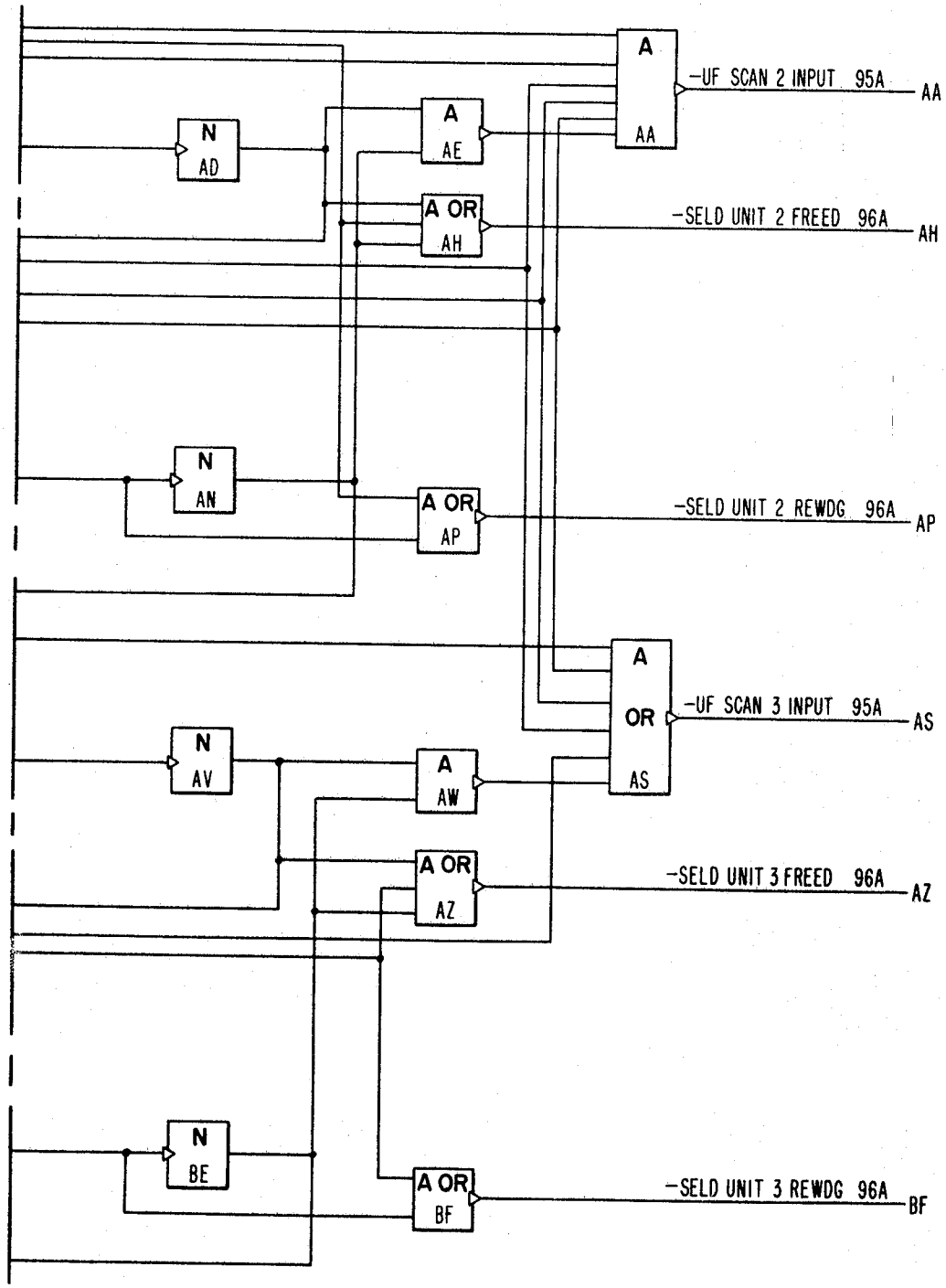

A
AA
-UF SCAN 2 INPUT 95A
AA
N
AD
A
AE
A OR
AH
-SELD UNIT 2 FREED 96A
AH
N
AN
A OR
AP
-SELD UNIT 2 REWDG 96A
AP
A
OR
AS
-UF SCAN 3 INPUT 95A
AS
N
AV
A
AW
A OR
AZ
-SELD UNIT 3 FREED 96A
AZ
N
BE
A OR
BF
-SELD UNIT 3 REWDG 96A
BF

FIG. 89A

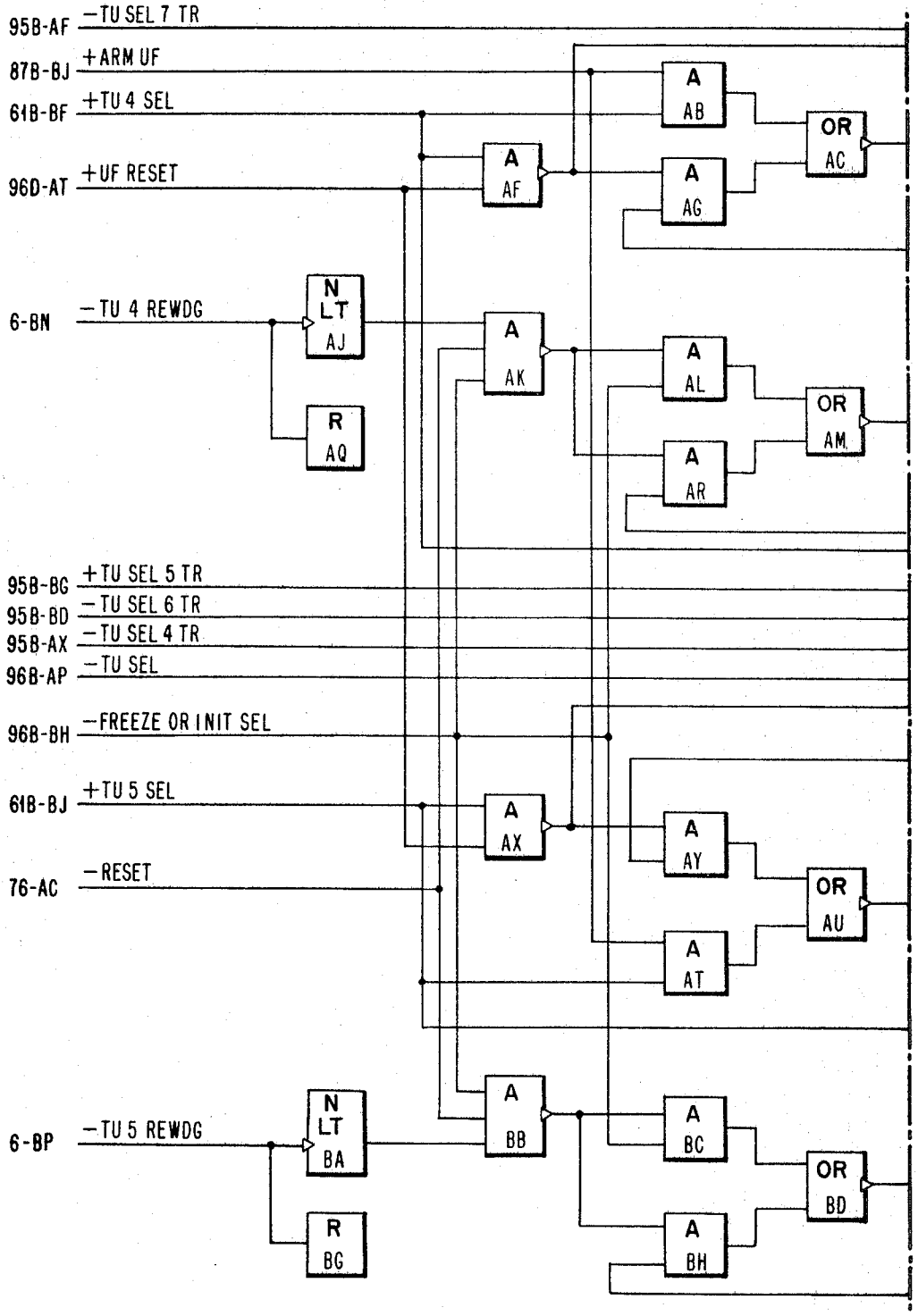

95B-AF
−TU SEL 7 TR
87B-BJ
+ARM UF
61B-BF
+TU 4 SEL
96D-AT
+UF RESET
6-BN
−TU 4 REWDG
N LT AJ
R AQ
A AF
A AK
A AL
A AR
A AB
A AG
OR AC
OR AM
95B-BG
+TU SEL 5 TR
95B-BD
−TU SEL 6 TR
95B-AX
−TU SEL 4 TR
96B-AP
−TU SEL
96B-BH
−FREEZE OR INIT SEL
61B-BJ
+TU 5 SEL
76-AC
−RESET
A AX
A AY
A AT
OR AU
6-BP
−TU 5 REWDG
N LT BA
R BG
A BB
A BC
A BH
OR BD

FIG. 89B

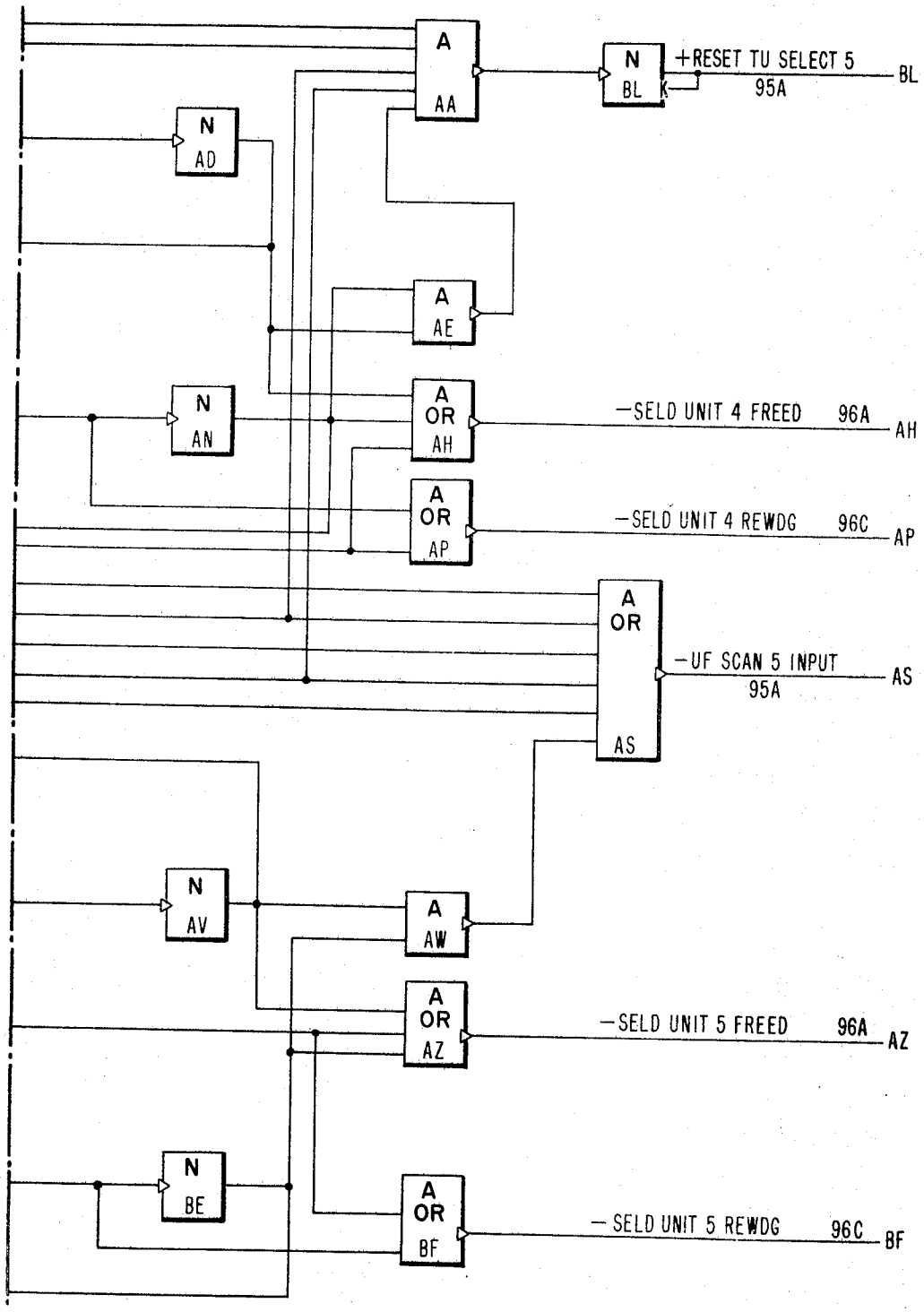

A
AA
N
BL
+RESET TU SELECT 5
95A
BL
N
AD
A
AE
N
AN
A
OR
AH
—SELD UNIT 4 FREED
96A
AH
A
OR
AP
—SELD UNIT 4 REWDG
96C
AP
A
OR
AS
—UF SCAN 5 INPUT
95A
AS
N
AV
A
AW
A
OR
AZ
—SELD UNIT 5 FREED
96A
AZ
N
BE
A
OR
BF
— SELD UNIT 5 REWDG
96C
BF

FIG. 90A

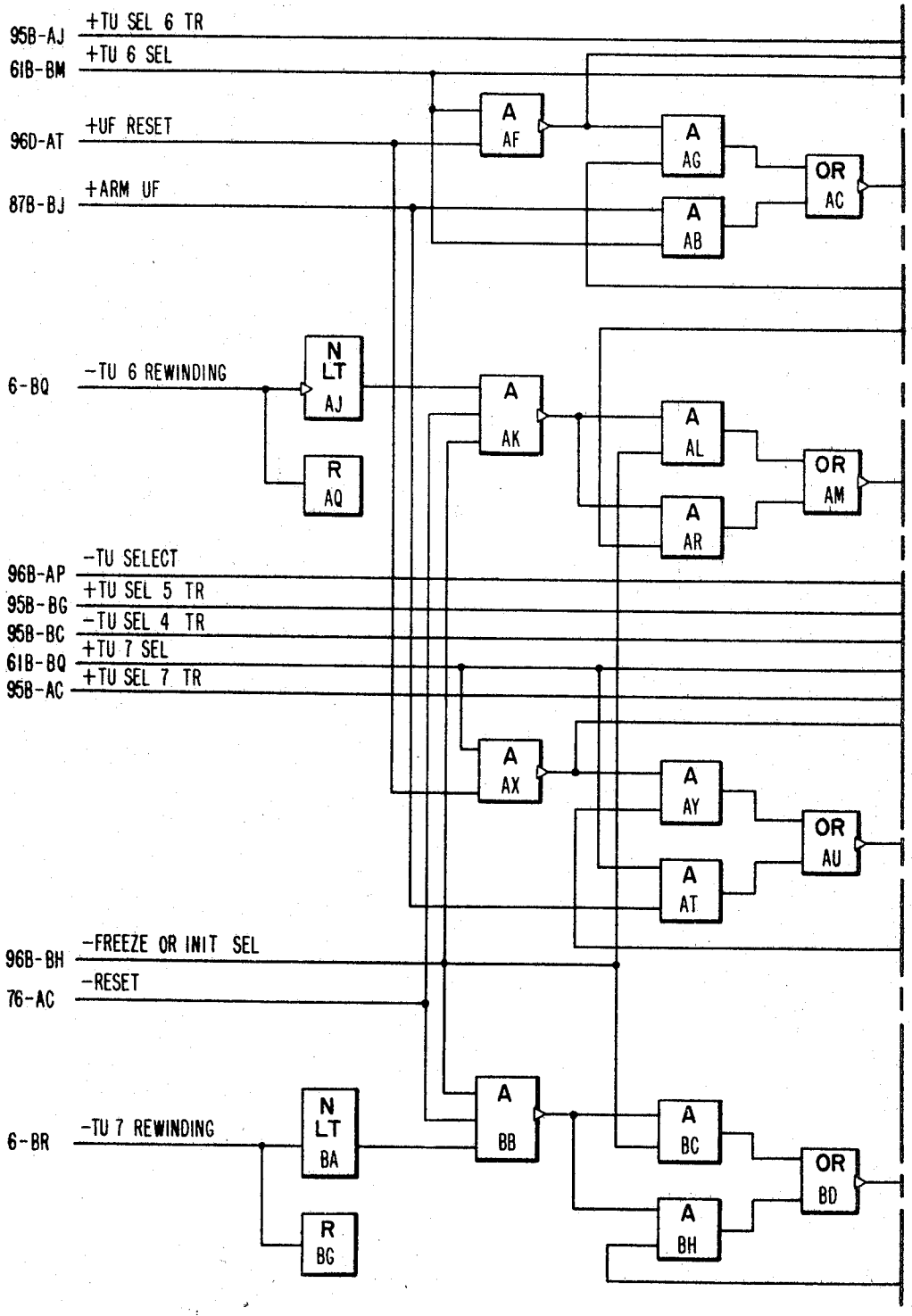

95B-AJ
+TU SEL 6 TR
61B-BM
+TU 6 SEL
96D-AT
+UF RESET
87B-BJ
+ARM UF
A AF
A AG
A AB
OR AC
6-BQ
-TU 6 REWINDING
N LT AJ
R AQ
A AK
A AL
A AR
OR AM
96B-AP
-TU SELECT
95B-BG
+TU SEL 5 TR
95B-BC
-TU SEL 4 TR
61B-BQ
+TU 7 SEL
95B-AC
+TU SEL 7 TR
A AX
A AY
A AT
OR AU
96B-BH
-FREEZE OR INIT SEL
76-AC
-RESET
6-BR
-TU 7 REWINDING
N LT BA
R BG
A BB
A BC
A BH
OR BD

FIG. 90B

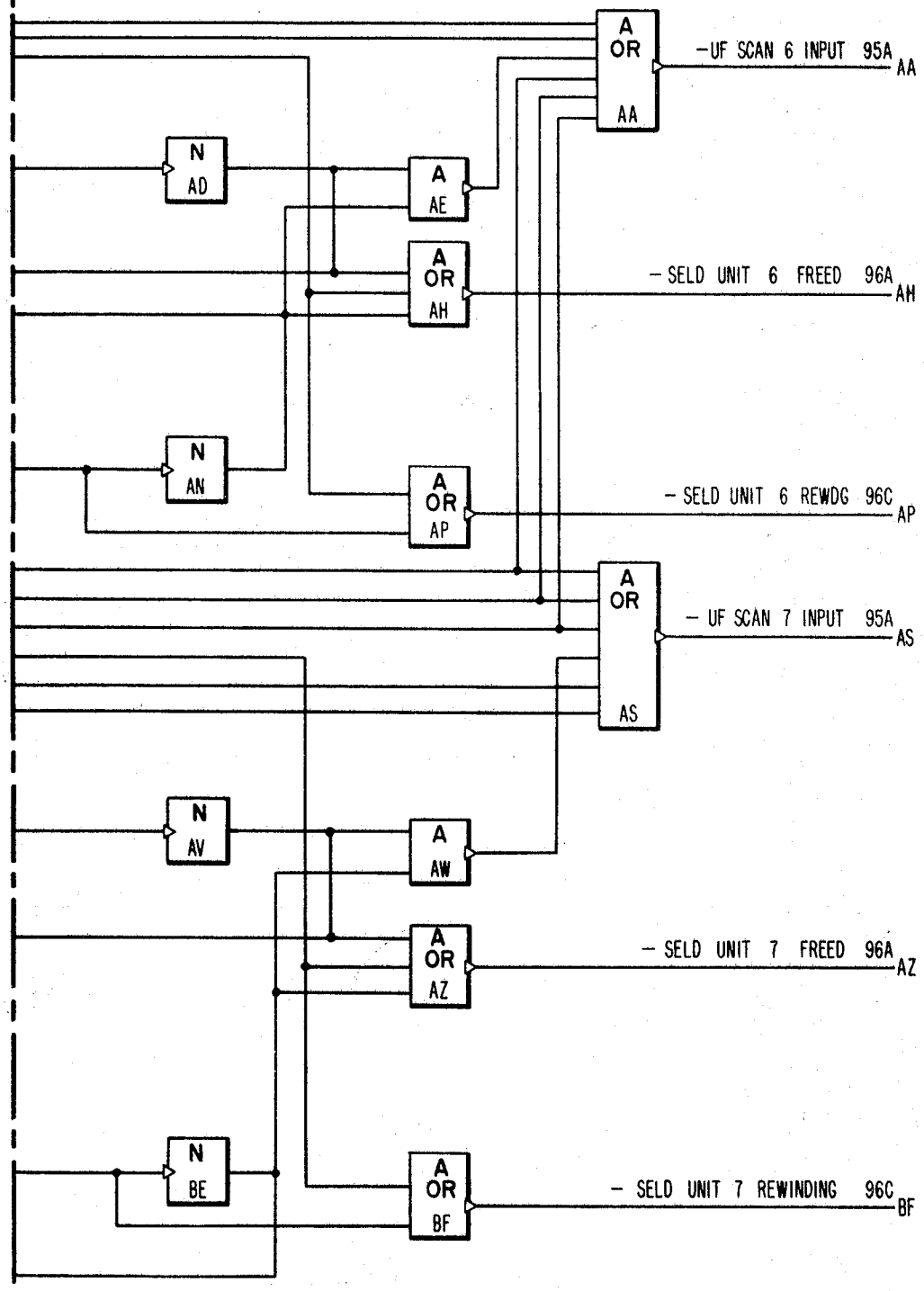

A OR AA
-UF SCAN 6 INPUT 95A
AA
N AD
A AE
A OR AH
- SELD UNIT 6 FREED 96A
AH
N AN
A OR AP
- SELD UNIT 6 REWDG 96C
AP
A OR AS
- UF SCAN 7 INPUT 95A
AS
N AV
A AW
A OR AZ
- SELD UNIT 7 FREED 96A
AZ
N BE
A OR BF
- SELD UNIT 7 REWINDING 96C
BF

FIG.91A

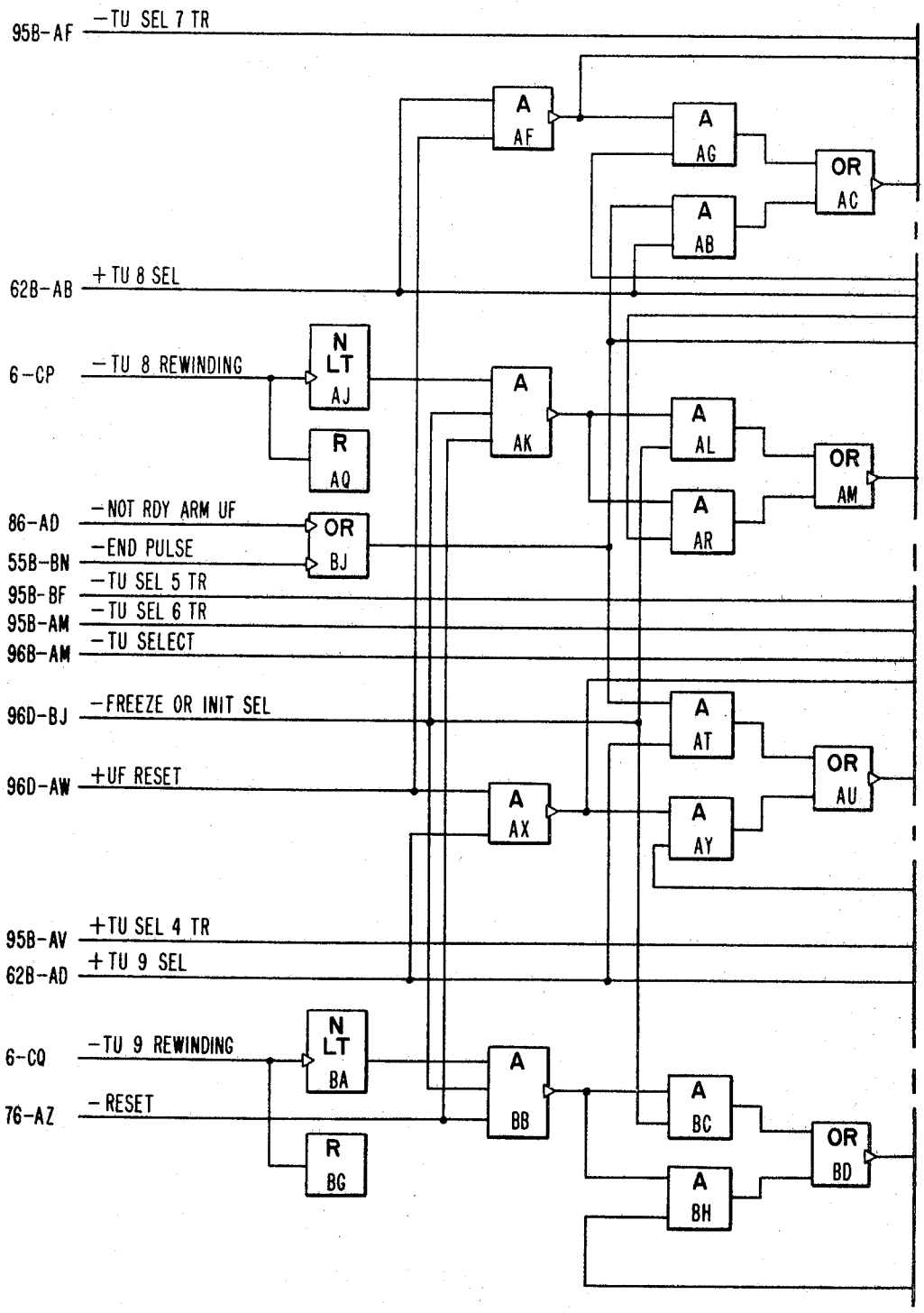

95B-AF
-TU SEL 7 TR
A AF
A AG
OR AC
A AB
62B-AB
+TU 8 SEL
6-CP
-TU 8 REWINDING
N LT AJ
R AQ
A AK
A AL
OR AM
A AR
86-AD
-NOT RDY ARM UF
55B-BN
-END PULSE
OR BJ
95B-BF
-TU SEL 5 TR
95B-AM
-TU SEL 6 TR
96B-AM
-TU SELECT
96D-BJ
-FREEZE OR INIT SEL
A AT
OR AU
96D-AW
+UF RESET
A AX
A AY
95B-AV
+TU SEL 4 TR
62B-AD
+TU 9 SEL
6-CQ
-TU 9 REWINDING
N LT BA
76-AZ
-RESET
A BB
A BC
OR BD
R BG
A BH

FIG. 91B

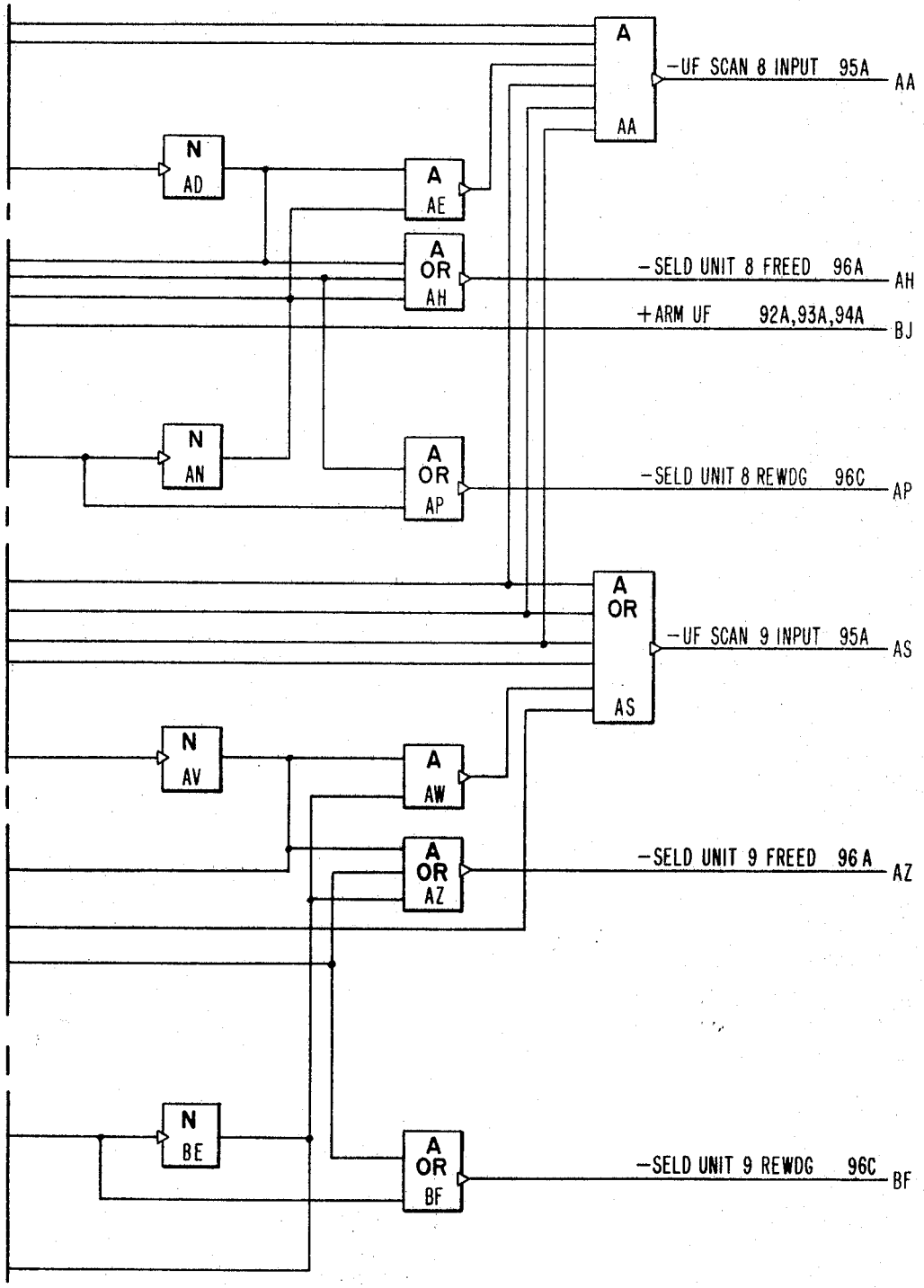

A
AA
-UF SCAN 8 INPUT 95A
AA
N
AD
A
AE
A
OR
AH
-SELD UNIT 8 FREED 96A
AH
+ARM UF 92A,93A,94A
BJ
N
AN
A
OR
AP
-SELD UNIT 8 REWDG 96C
AP
A
OR
AS
-UF SCAN 9 INPUT 95A
AS
N
AV
A
AW
A
OR
AZ
-SELD UNIT 9 FREED 96A
AZ
N
BE
A
OR
BF
-SELD UNIT 9 REWDG 96C
BF

FIG. 92A

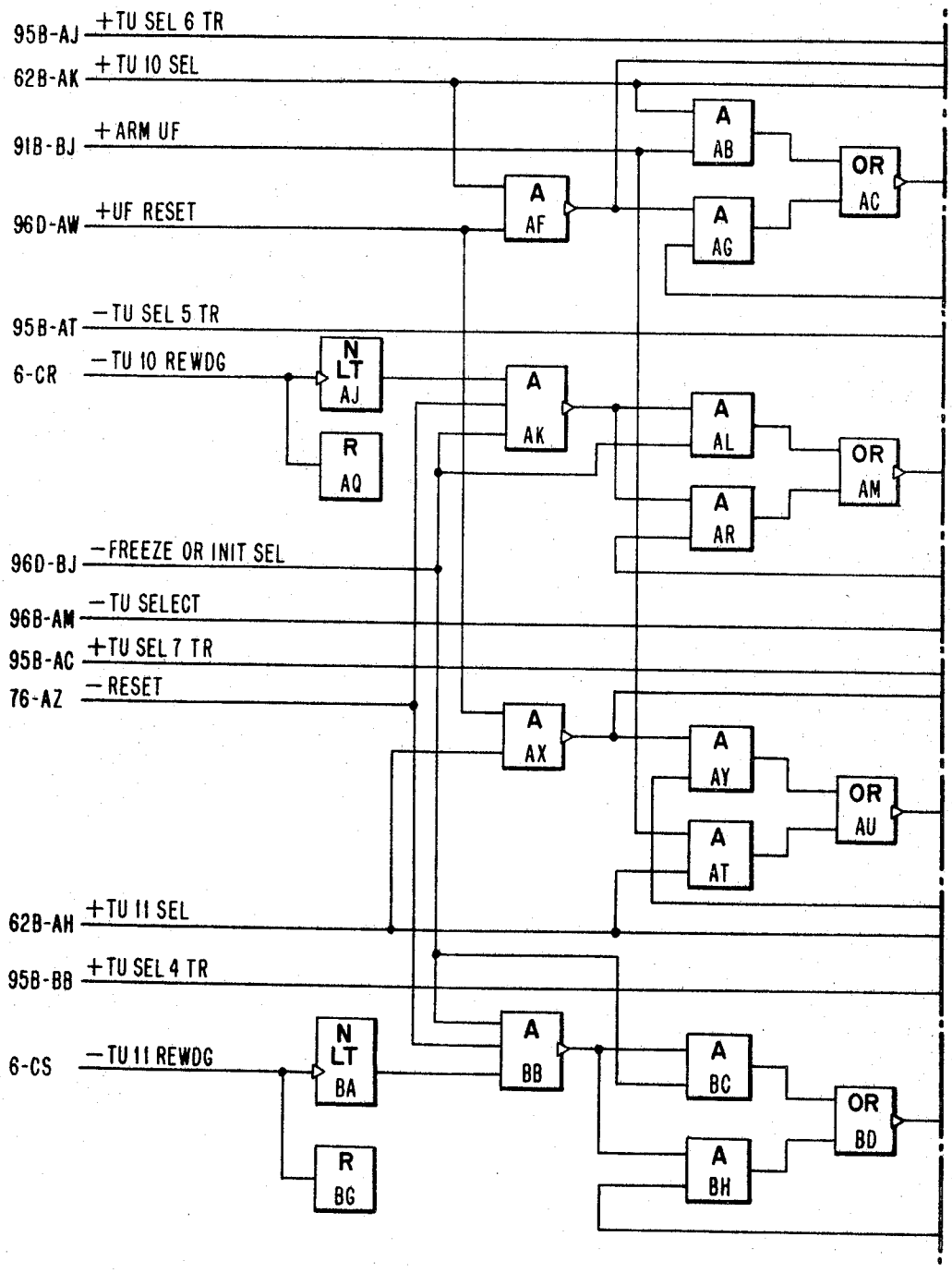

95B-AJ
+TU SEL 6 TR
62B-AK
+TU 10 SEL
91B-BJ
+ARM UF
96D-AW
+UF RESET
95B-AT
−TU SEL 5 TR
6-CR
−TU 10 REWDG
96D-BJ
−FREEZE OR INIT SEL
96B-AM
−TU SELECT
95B-AC
+TU SEL 7 TR
76-AZ
−RESET
62B-AH
+TU 11 SEL
95B-BB
+TU SEL 4 TR
6-CS
−TU 11 REWDG
A AB
A AF
A AG
OR AC
N LT AJ
R AQ
A AK
A AL
A AR
OR AM
A AX
A AY
A AT
OR AU
N LT BA
R BG
A BB
A BC
A BH
OR BD

FIG.92B

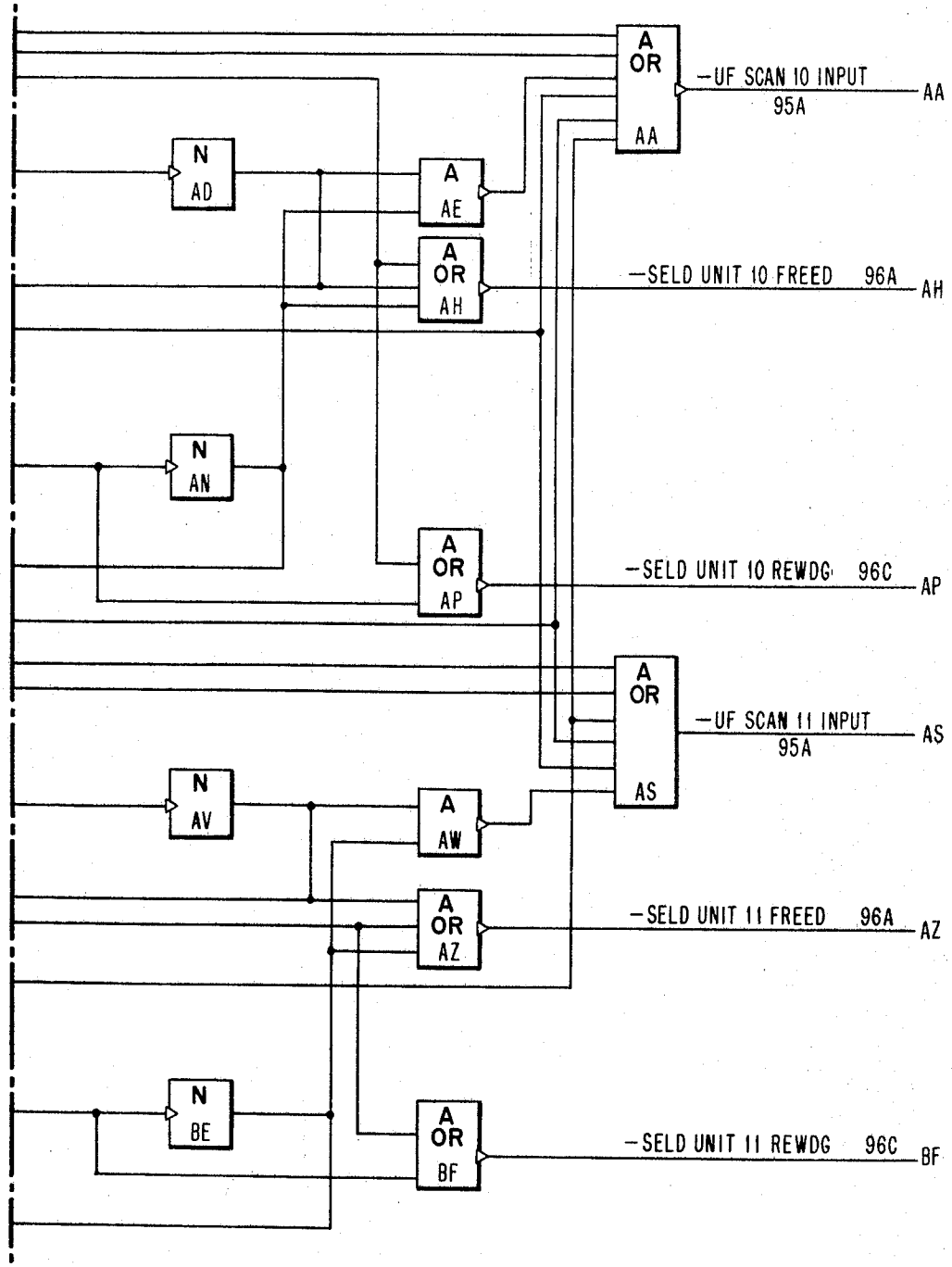

A OR AA
—UF SCAN 10 INPUT 95A
AA
N AD
A AE
A OR AH
—SELD UNIT 10 FREED 96A
AH
N AN
A OR AP
—SELD UNIT 10 REWDG 96C
AP
A OR AS
—UF SCAN 11 INPUT 95A
AS
N AV
A AW
A OR AZ
—SELD UNIT 11 FREED 96A
AZ
N BE
A OR BF
—SELD UNIT 11 REWDG 96C
BF

FIG.93A

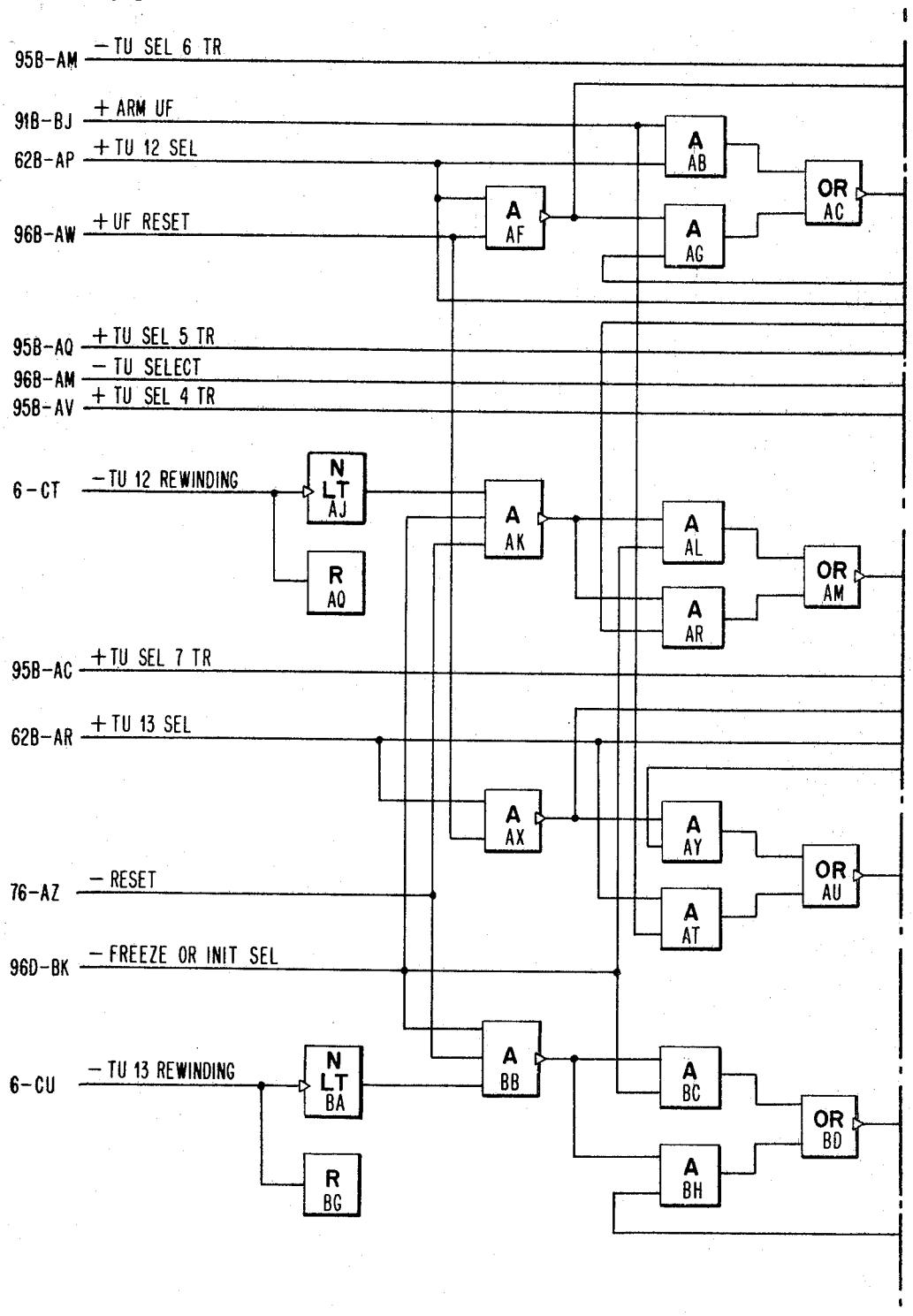

95B-AM
−TU SEL 6 TR
91B-BJ
+ARM UF
62B-AP
+TU 12 SEL
96B-AW
+UF RESET
A AF
A AB
A AG
OR AC
95B-AQ
+TU SEL 5 TR
96B-AM
−TU SELECT
95B-AV
+TU SEL 4 TR
6-CT
−TU 12 REWINDING
N LT AJ
R AQ
A AK
A AL
A AR
OR AM
95B-AC
+TU SEL 7 TR
62B-AR
+TU 13 SEL
A AX
A AY
A AT
OR AU
76-AZ
−RESET
96D-BK
−FREEZE OR INIT SEL
6-CU
−TU 13 REWINDING
N LT BA
R BG
A BB
A BC
A BH
OR BD

FIG.93B

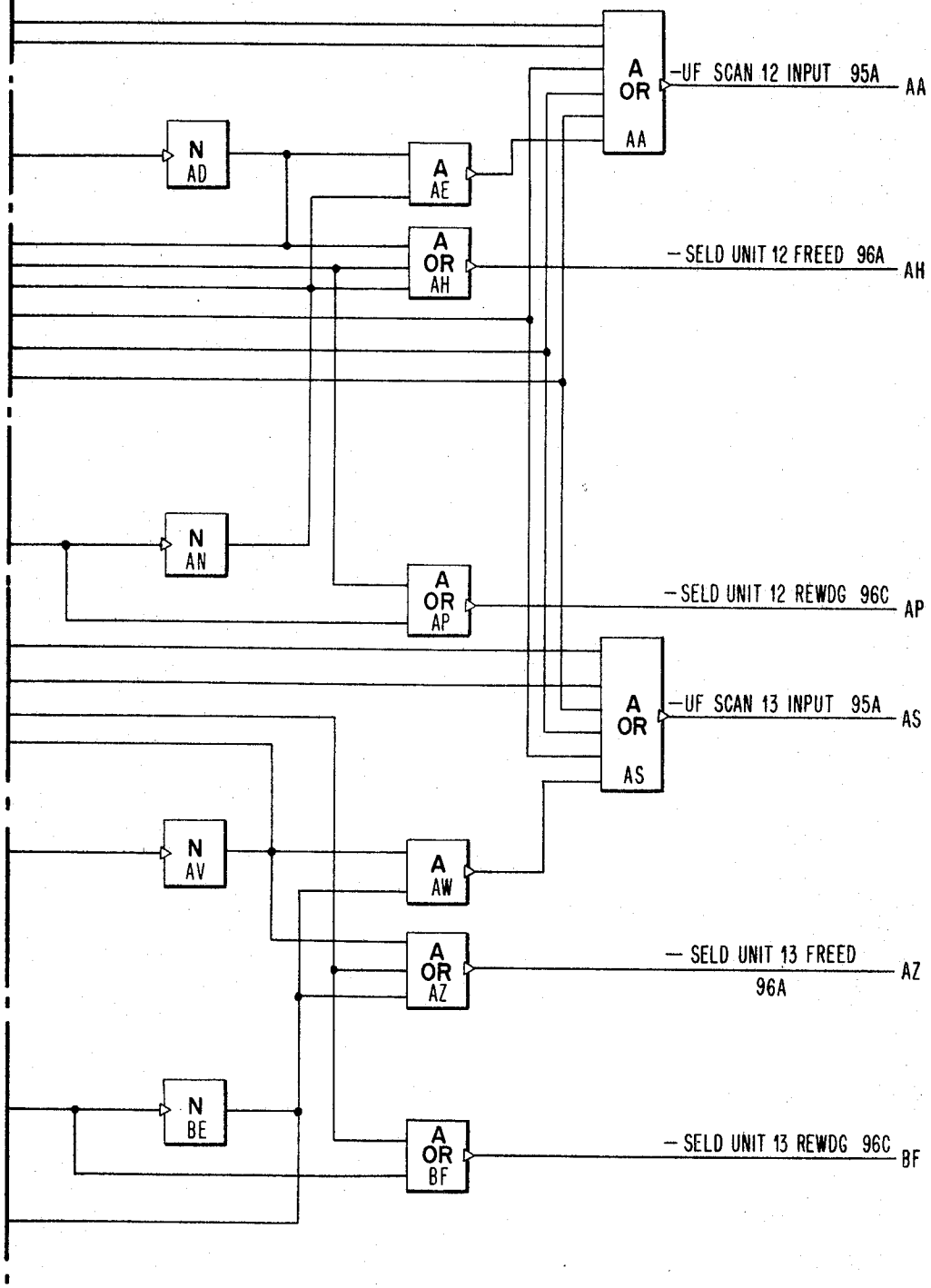

A OR AA
UF SCAN 12 INPUT 95A
AA
N AD
A AE
A OR AH
SELD UNIT 12 FREED 96A
AH
N AN
A OR AP
SELD UNIT 12 REWDG 96C
AP
A OR AS
UF SCAN 13 INPUT 95A
AS
N AV
A AW
A OR AZ
SELD UNIT 13 FREED 96A
AZ
N BE
A OR BF
SELD UNIT 13 REWDG 96C
BF

FIG. 94A

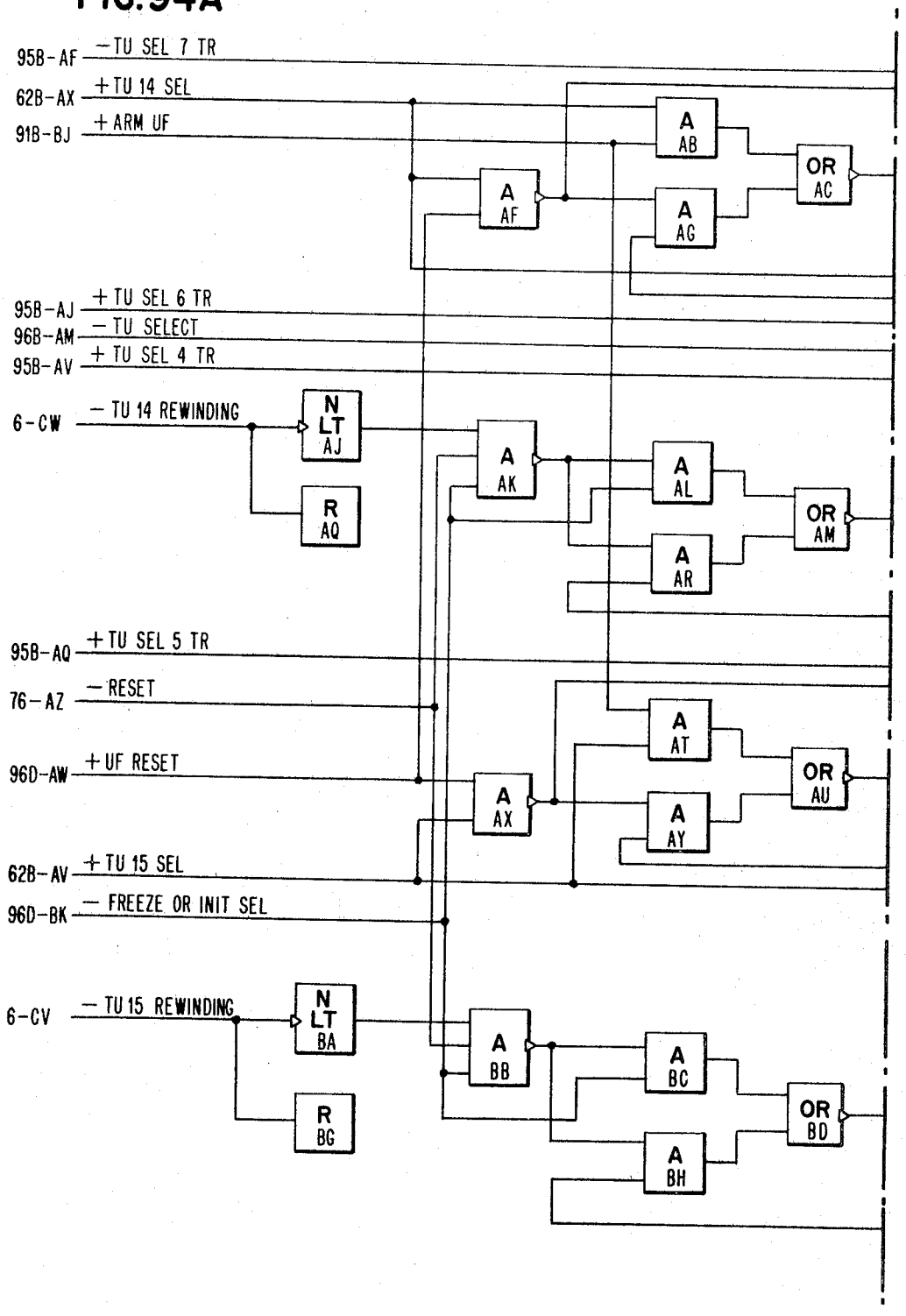

95B-AF
−TU SEL 7 TR
62B-AX
+TU 14 SEL
91B-BJ
+ARM UF
A AB
A AF
A AG
OR AC
95B-AJ
+TU SEL 6 TR
96B-AM
−TU SELECT
95B-AV
+TU SEL 4 TR
6-CW
−TU 14 REWINDING
N LT AJ
R AQ
A AK
A AL
A AR
OR AM
95B-AQ
+TU SEL 5 TR
76-AZ
−RESET
96D-AW
+UF RESET
A AT
A AX
A AY
OR AU
62B-AV
+TU 15 SEL
96D-BK
−FREEZE OR INIT SEL
6-CV
−TU 15 REWINDING
N LT BA
R BG
A BB
A BC
A BH
OR BD

FIG.94B

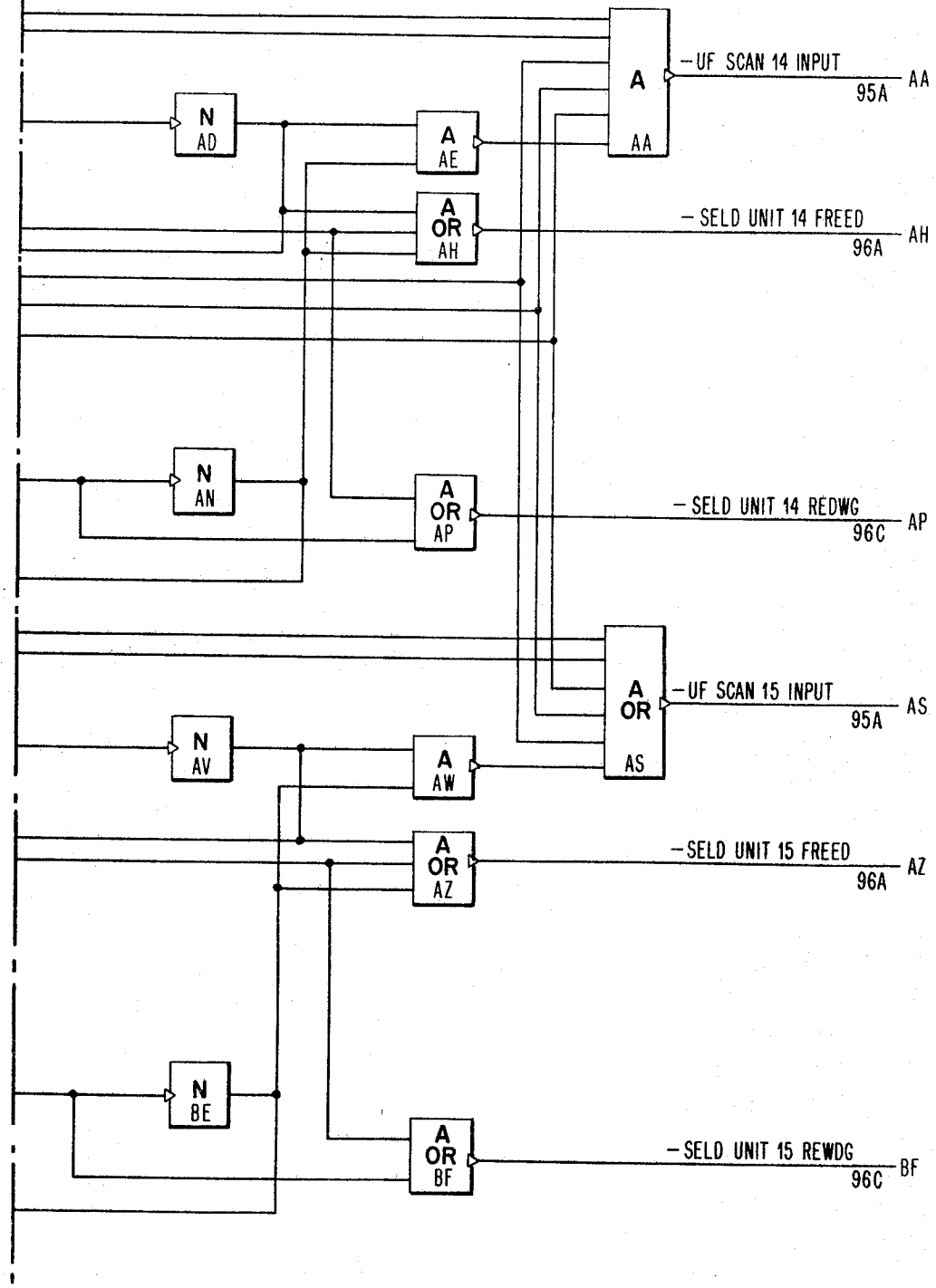

A
AA
-UF SCAN 14 INPUT
95A
AA
N
AD
A
AE
A
OR
AH
-SELD UNIT 14 FREED
96A
AH
N
AN
A
OR
AP
-SELD UNIT 14 REDWG
96C
AP
A
OR
AS
-UF SCAN 15 INPUT
95A
AS
N
AV
A
AW
A
OR
AZ
-SELD UNIT 15 FREED
96A
AZ
N
BE
A
OR
BF
-SELD UNIT 15 REWDG
96C
BF

FIG. 95A

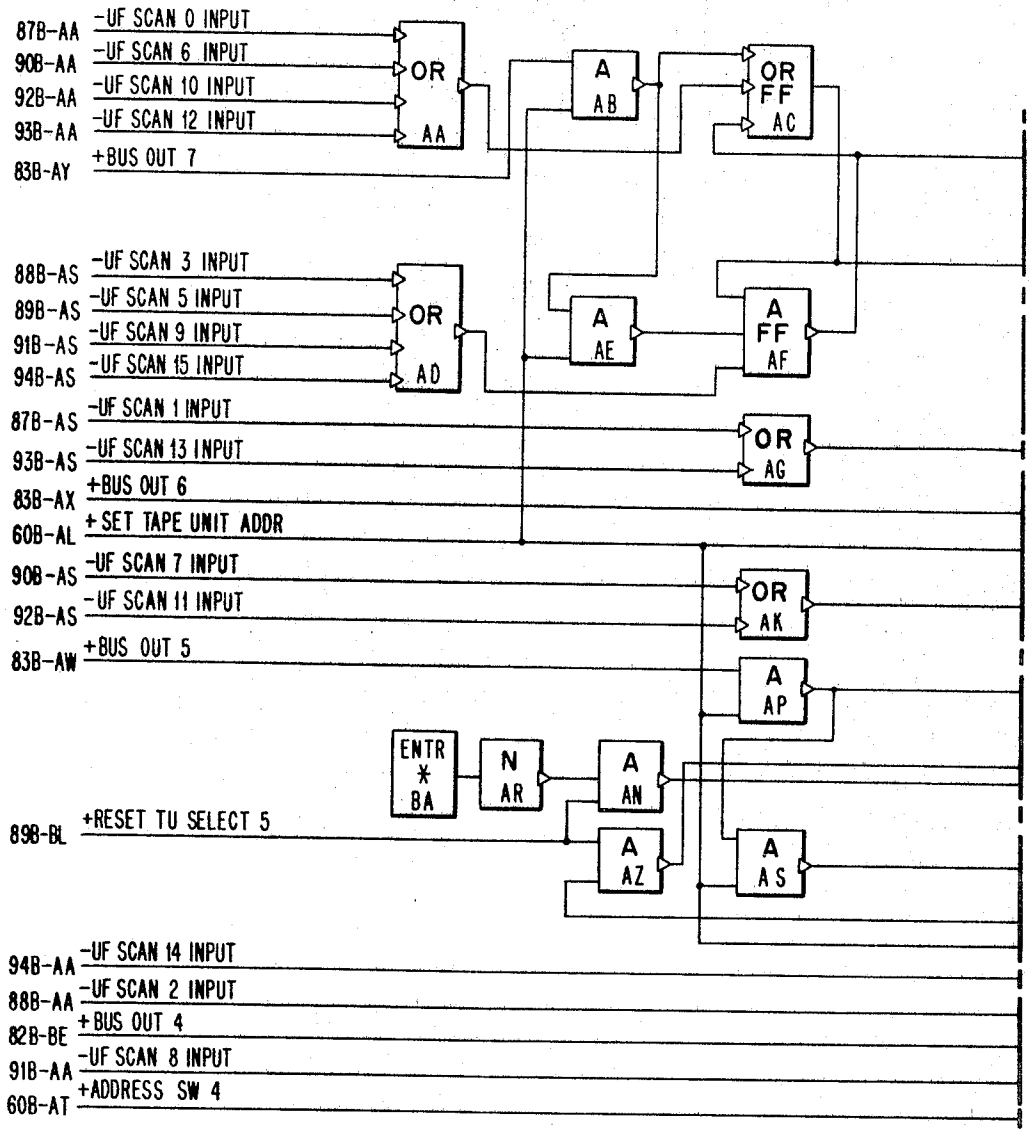

87B-AA
-UF SCAN 0 INPUT
90B-AA
-UF SCAN 6 INPUT
92B-AA
-UF SCAN 10 INPUT
93B-AA
-UF SCAN 12 INPUT
83B-AY
+BUS OUT 7
OR AA
A AB
OR FF AC
88B-AS
-UF SCAN 3 INPUT
89B-AS
-UF SCAN 5 INPUT
91B-AS
-UF SCAN 9 INPUT
94B-AS
-UF SCAN 15 INPUT
OR AD
A AE
A FF AF
87B-AS
-UF SCAN 1 INPUT
93B-AS
-UF SCAN 13 INPUT
OR AG
83B-AX
+BUS OUT 6
60B-AL
+SET TAPE UNIT ADDR
90B-AS
-UF SCAN 7 INPUT
92B-AS
-UF SCAN 11 INPUT
OR AK
83B-AW
+BUS OUT 5
A AP
ENTR * BA
N AR
A AN
89B-BL
+RESET TU SELECT 5
A AZ
A AS
94B-AA
-UF SCAN 14 INPUT
88B-AA
-UF SCAN 2 INPUT
82B-BE
+BUS OUT 4
91B-AA
-UF SCAN 8 INPUT
60B-AT
+ADDRESS SW 4

FIG. 95B

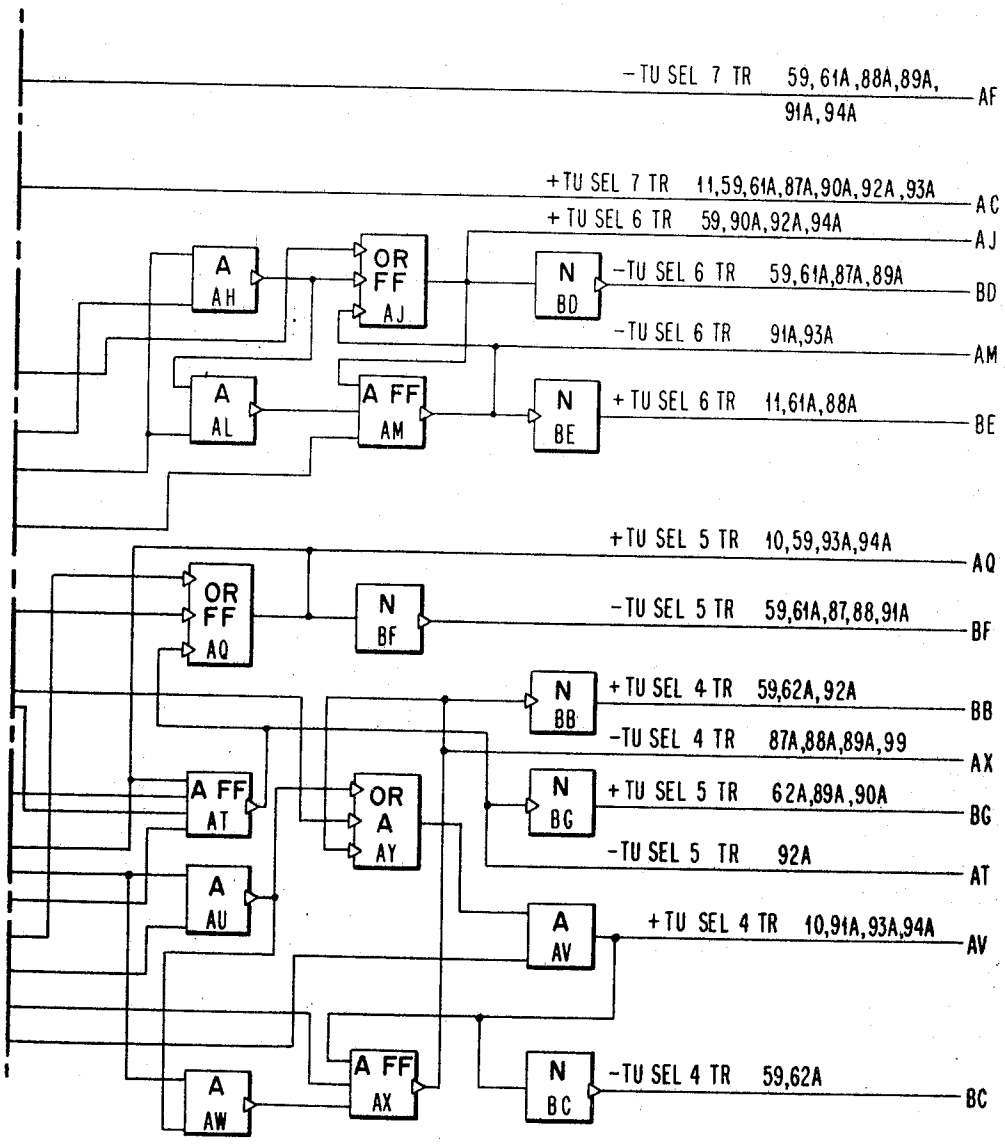

-TU SEL 7 TR 59,61A,88A,89A, 91A,94A
AF
+TU SEL 7 TR 11,59,61A,87A,90A,92A,93A
AC
+TU SEL 6 TR 59,90A,92A,94A
AJ
A AH
OR FF AJ
N BD
-TU SEL 6 TR 59,61A,87A,89A
BD
-TU SEL 6 TR 91A,93A
AM
A AL
A FF AM
N BE
+TU SEL 6 TR 11,61A,88A
BE
+TU SEL 5 TR 10,59,93A,94A
AQ
OR FF AQ
N BF
-TU SEL 5 TR 59,61A,87,88,91A
BF
N BB
+TU SEL 4 TR 59,62A,92A
BB
-TU SEL 4 TR 87A,88A,89A,99
AX
A FF AT
OR A AY
N BG
+TU SEL 5 TR 62A,89A,90A
BG
-TU SEL 5 TR 92A
AT
A AU
A AV
+TU SEL 4 TR 10,91A,93A,94A
AV
A FF AX
N BC
-TU SEL 4 TR 59,62A
BC
A AW

FIG.96A

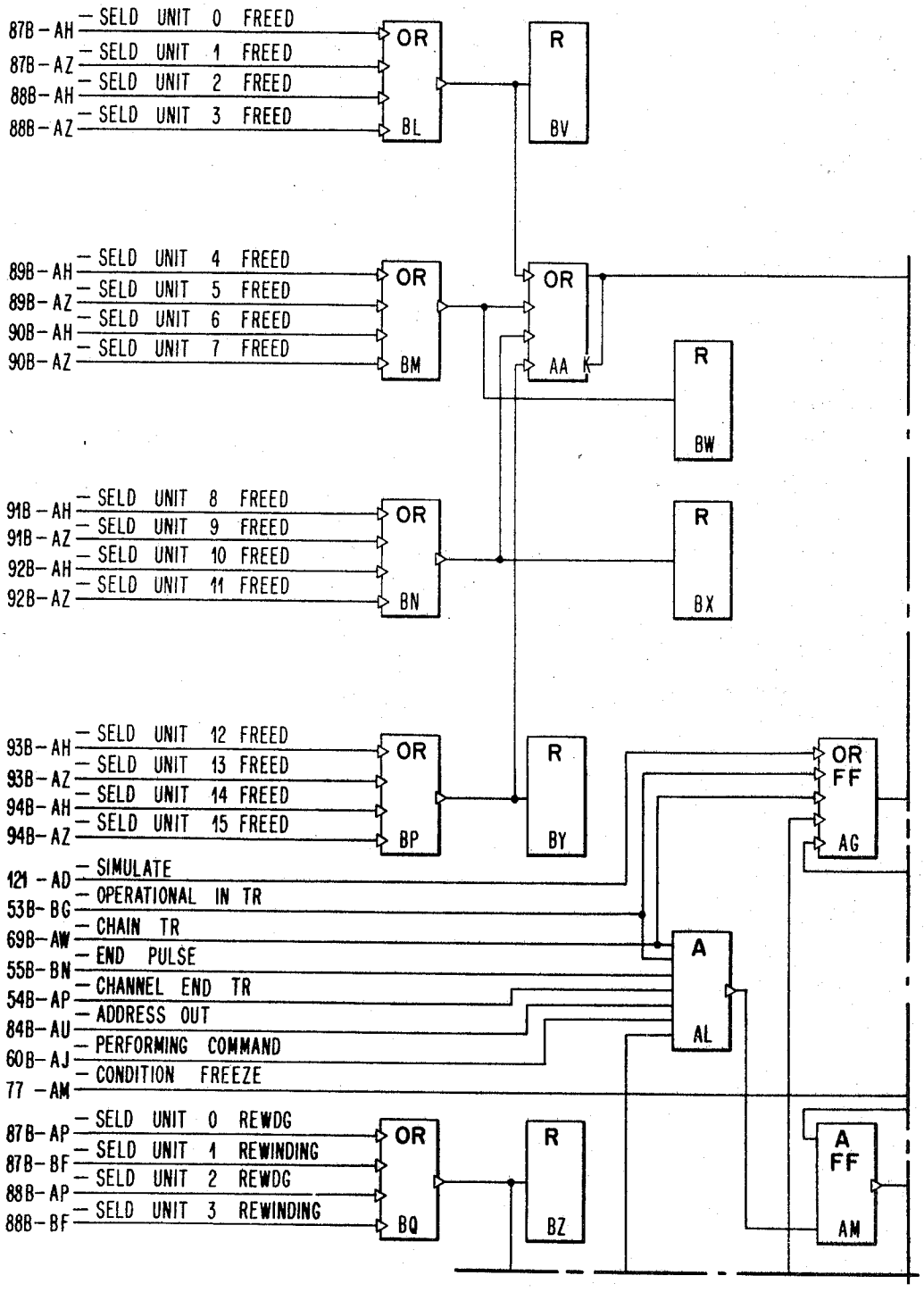

87B-AH — SELD UNIT 0 FREED
87B-AZ — SELD UNIT 1 FREED
88B-AH — SELD UNIT 2 FREED
88B-AZ — SELD UNIT 3 FREED
OR
BL
R
BV
89B-AH — SELD UNIT 4 FREED
89B-AZ — SELD UNIT 5 FREED
90B-AH — SELD UNIT 6 FREED
90B-AZ — SELD UNIT 7 FREED
OR
BM
OR
AA
K
R
BW
91B-AH — SELD UNIT 8 FREED
91B-AZ — SELD UNIT 9 FREED
92B-AH — SELD UNIT 10 FREED
92B-AZ — SELD UNIT 11 FREED
OR
BN
R
BX
93B-AH — SELD UNIT 12 FREED
93B-AZ — SELD UNIT 13 FREED
94B-AH — SELD UNIT 14 FREED
94B-AZ — SELD UNIT 15 FREED
OR
BP
R
BY
OR
FF
AG
121-AD — SIMULATE
53B-BG — OPERATIONAL IN TR
69B-AW — CHAIN TR
55B-BN — END PULSE
54B-AP — CHANNEL END TR
84B-AU — ADDRESS OUT
60B-AJ — PERFORMING COMMAND
77-AM — CONDITION FREEZE
A
AL
87B-AP — SELD UNIT 0 REWDG
87B-BF — SELD UNIT 1 REWINDING
88B-AP — SELD UNIT 2 REWDG
88B-BF — SELD UNIT 3 REWINDING
OR
BQ
R
BZ
A
FF
AM

FIG.96B

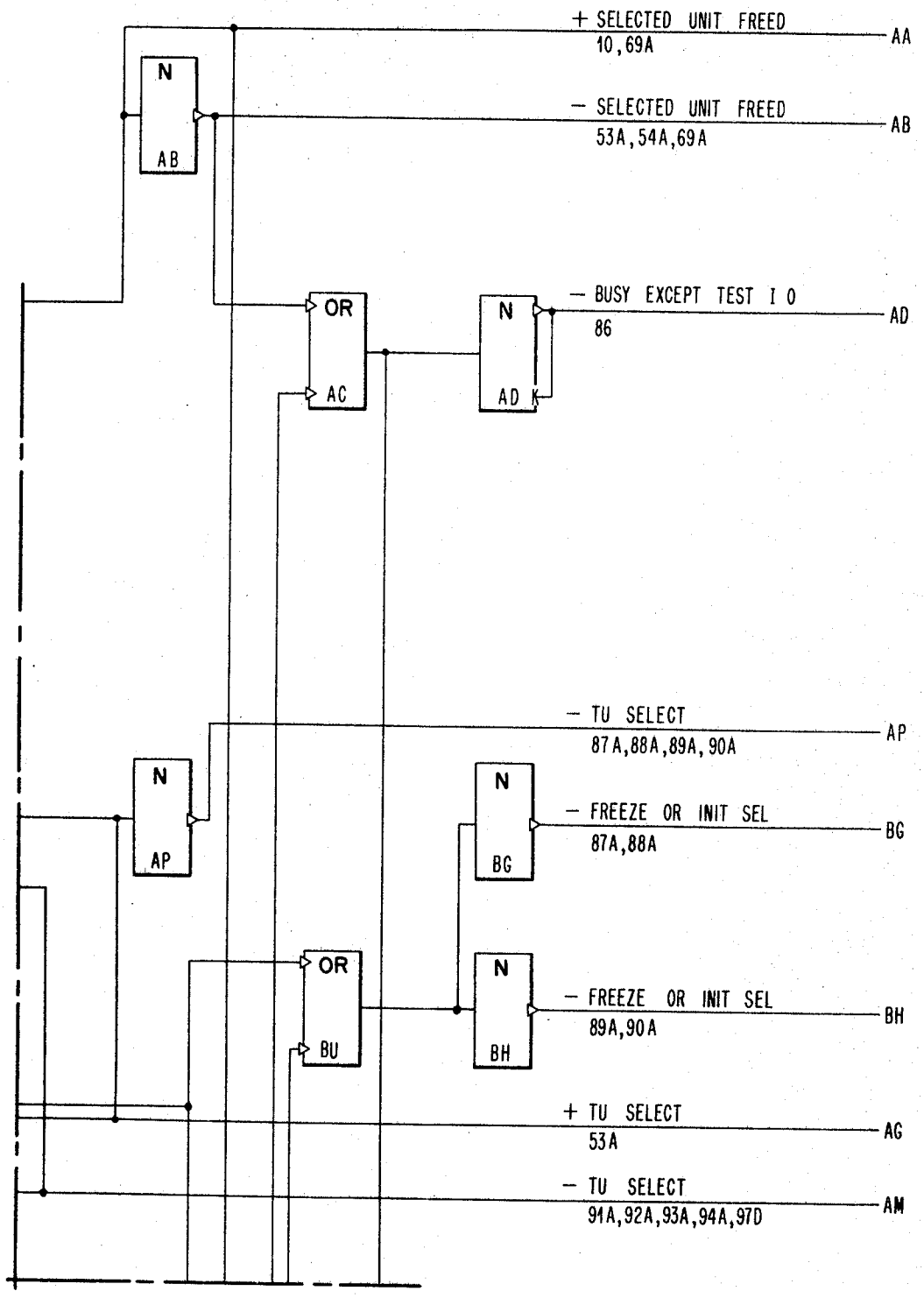

+ SELECTED UNIT FREED
10,69A
AA
N
AB
— SELECTED UNIT FREED
53A,54A,69A
AB
OR
AC
N
AD K
— BUSY EXCEPT TEST I O
86
AD
— TU SELECT
87A,88A,89A,90A
AP
N
AP
N
BG
— FREEZE OR INIT SEL
87A,88A
BG
OR
BU
N
BH
— FREEZE OR INIT SEL
89A,90A
BH
+ TU SELECT
53A
AG
— TU SELECT
91A,92A,93A,94A,97D
AM

FIG. 96C

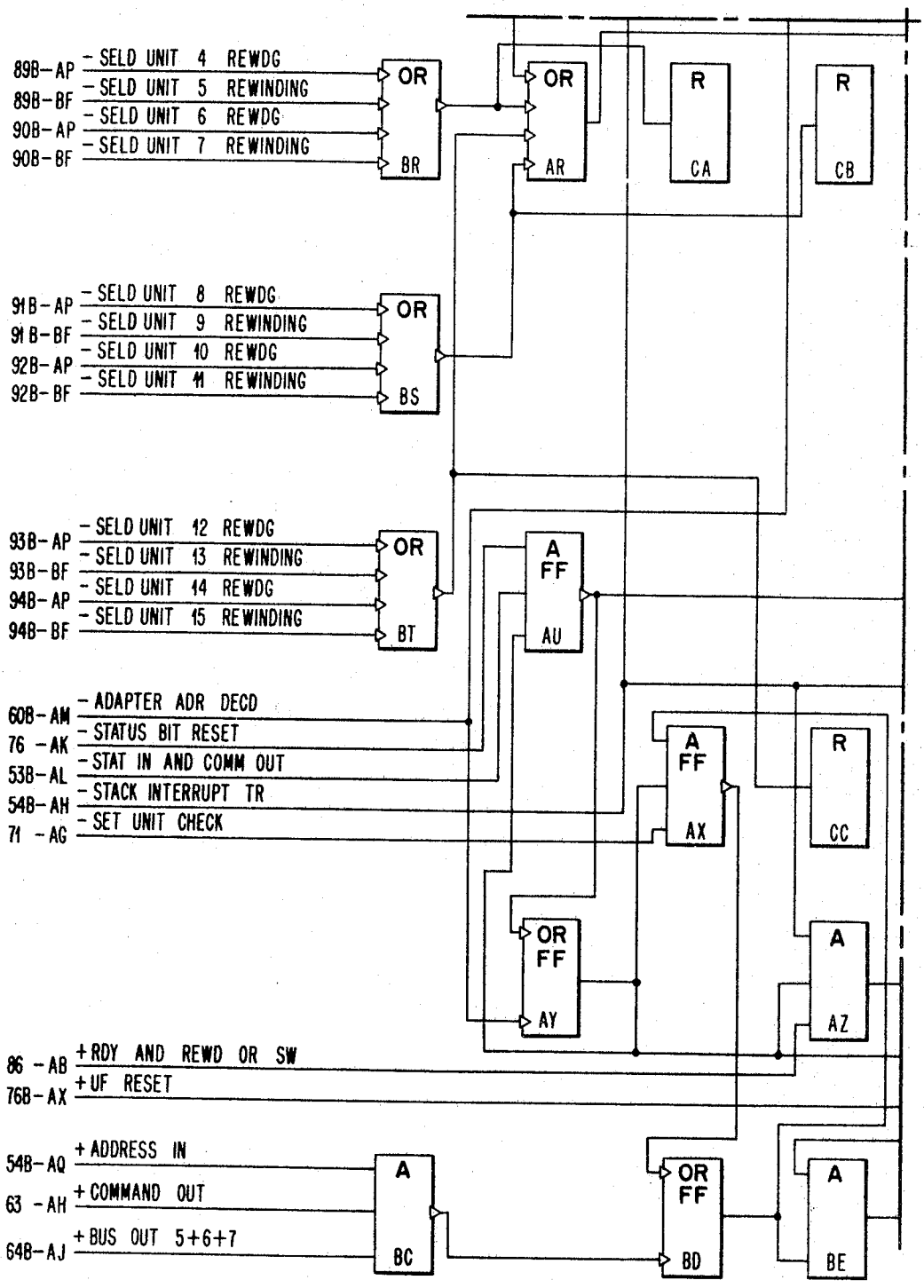
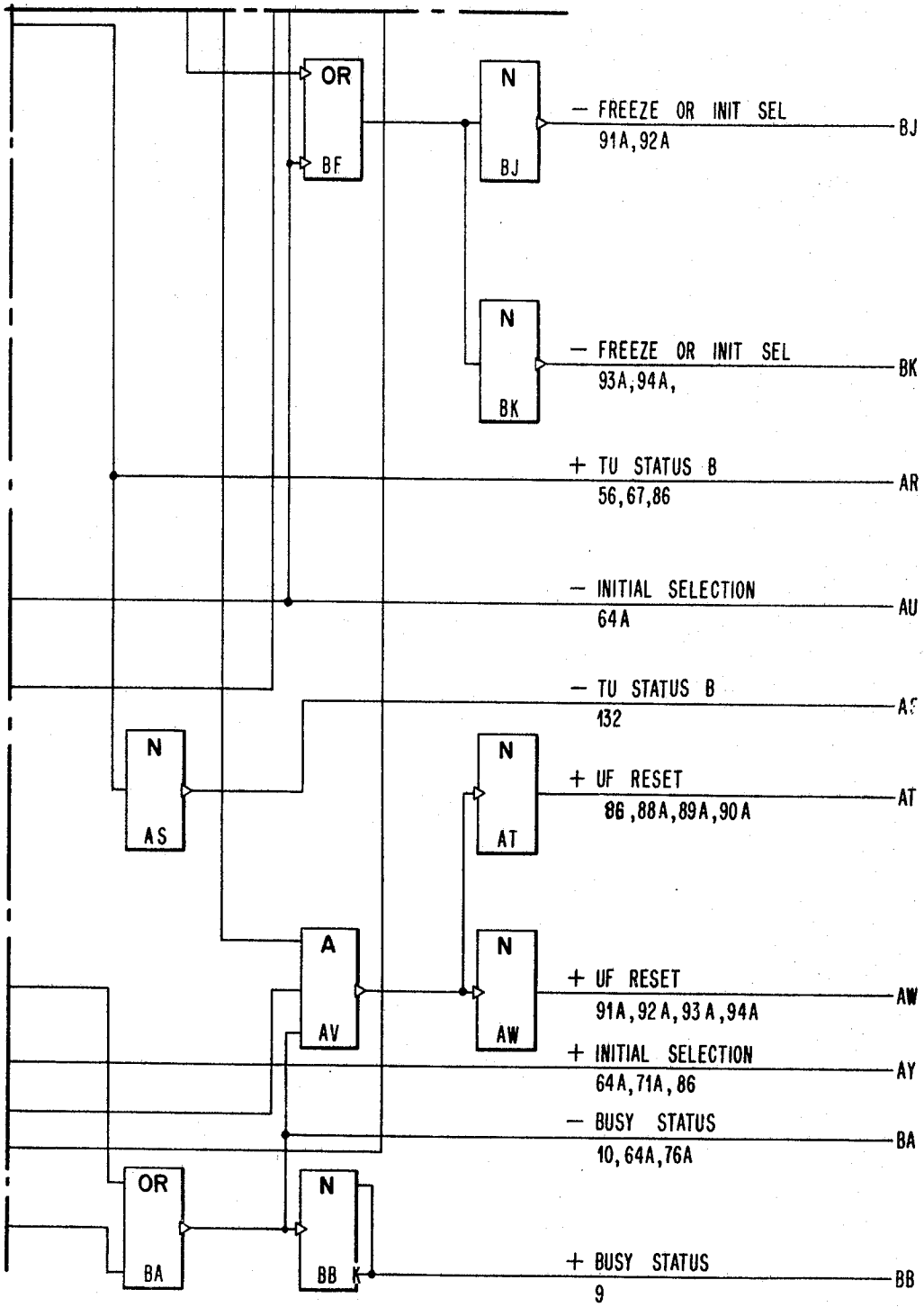

89B-AP
89B-BF
90B-AP
90B-BF
- SELD UNIT 4 REWDG
- SELD UNIT 5 REWINDING
- SELD UNIT 6 REWDG
- SELD UNIT 7 REWINDING
OR
BR
OR
AR
R
CA
R
CB
91B-AP
91B-BF
92B-AP
92B-BF
- SELD UNIT 8 REWDG
- SELD UNIT 9 REWINDING
- SELD UNIT 10 REWDG
- SELD UNIT 11 REWINDING
OR
BS
93B-AP
93B-BF
94B-AP
94B-BF
- SELD UNIT 12 REWDG
- SELD UNIT 13 REWINDING
- SELD UNIT 14 REWDG
- SELD UNIT 15 REWINDING
OR
BT
A FF
AU
60B-AM
76 -AK
53B-AL
54B-AH
71 -AG
- ADAPTER ADR DECD
- STATUS BIT RESET
- STAT IN AND COMM OUT
- STACK INTERRUPT TR
- SET UNIT CHECK
A FF
AX
R
CC
OR FF
AY
A
AZ
86 -AB
76B-AX
+ RDY AND REWD OR SW
+ UF RESET
54B-AQ
63 -AH
64B-AJ
+ ADDRESS IN
+ COMMAND OUT
+ BUS OUT 5+6+7
A
BC
OR FF
BD
A
BE

FIG.96D


OR
BF
N
BJ
— FREEZE OR INIT SEL
91A,92A
BJ
N
BK
— FREEZE OR INIT SEL
93A,94A,
BK
+ TU STATUS B
56,67,86
AR
— INITIAL SELECTION
64A
AU
— TU STATUS B
132
N
AS
N
AT
+ UF RESET
86,88A,89A,90A
AT
A
AV
N
AW
+ UF RESET
91A,92A,93A,94A
AW
+ INITIAL SELECTION
64A,71A,86
AY
— BUSY STATUS
10,64A,76A
BA
OR
BA
N
BB
+ BUSY STATUS
9
BB

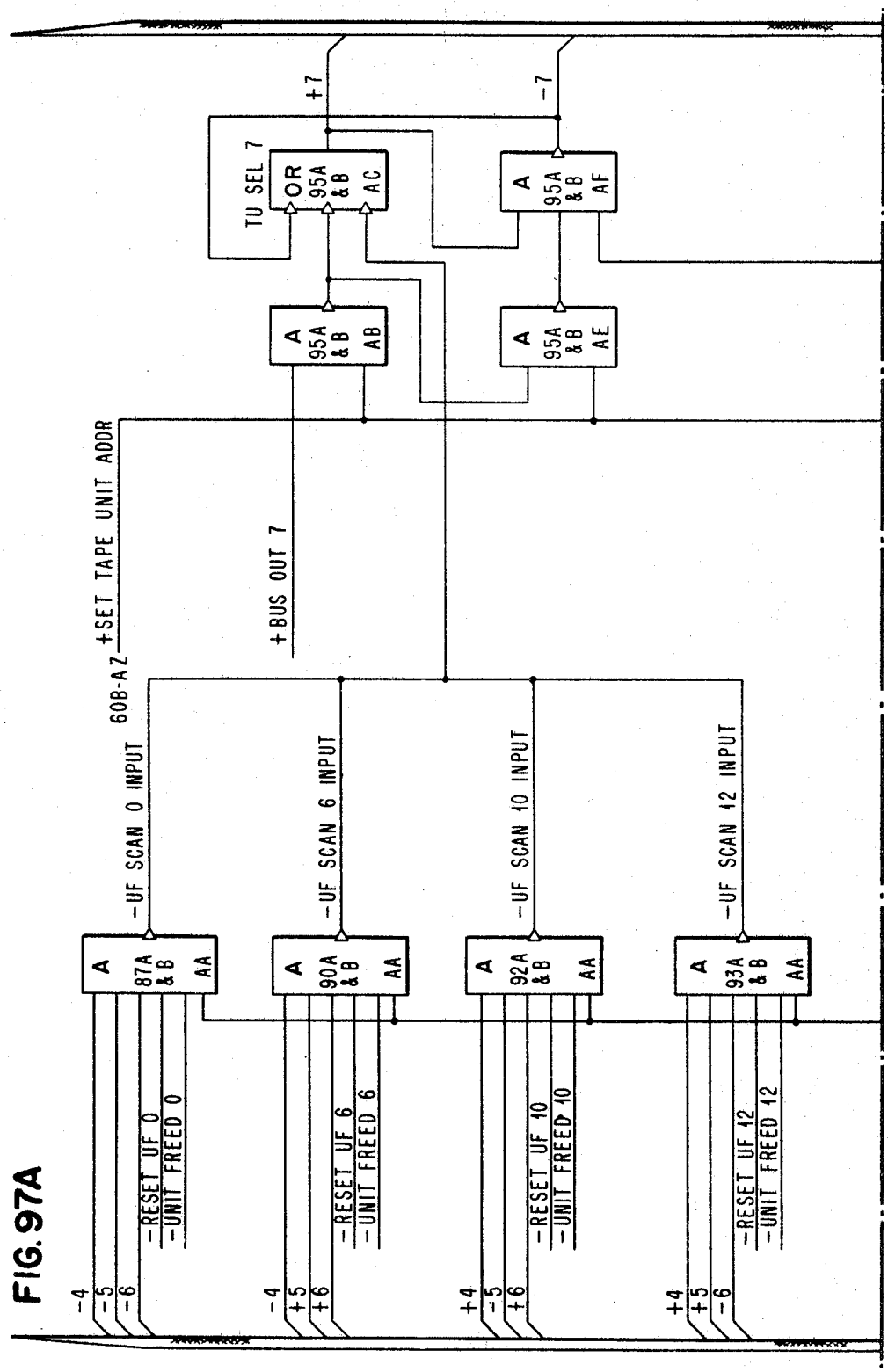

+7
-7
TU SEL 7
OR 95A &B AC
A 95A &B AF
A 95A &B AB
A 95A &B AE
+SET TAPE UNIT ADDR
60B-AZ
+BUS OUT 7
-UF SCAN 0 INPUT
-UF SCAN 6 INPUT
-UF SCAN 10 INPUT
-UF SCAN 12 INPUT
A 87A &B AA
A 90A &B AA
A 92A &B AA
A 93A &B AA
-RESET UF 0
-UNIT FREED 0
-RESET UF 6
-UNIT FREED 6
-RESET UF 10
-UNIT FREED 10
-RESET UF 12
-UNIT FREED 12
-4
-5
-6
-4
+5
+6
+4
-5
+6
+4
+5
-6

FIG. 97A

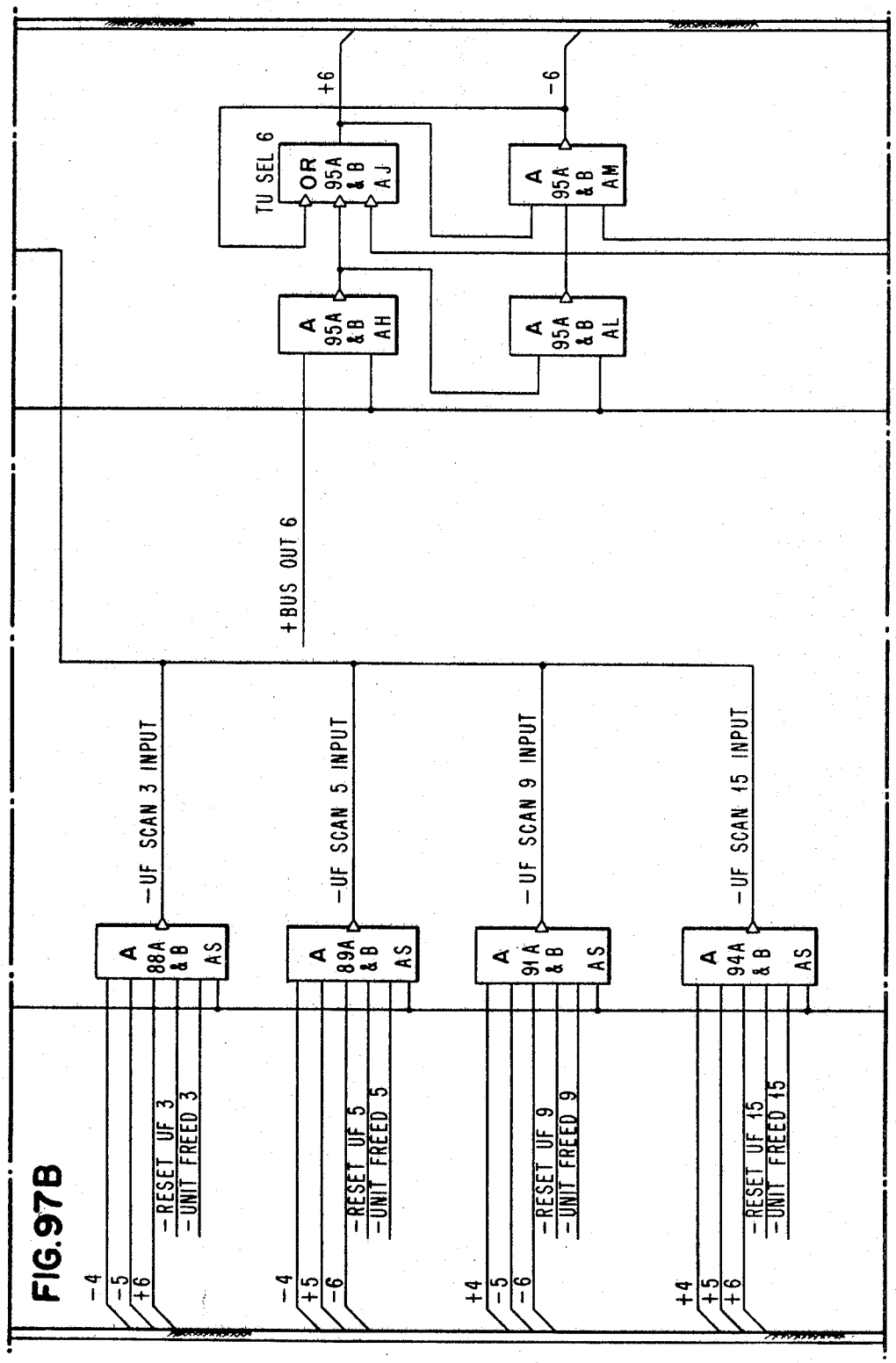

+6
-6
TU SEL 6
OR 95A & B AJ
A 95A & B AM
A 95A & B AH
A 95A & B AL
+BUS OUT 6
-UF SCAN 3 INPUT
-UF SCAN 5 INPUT
-UF SCAN 9 INPUT
-UF SCAN 15 INPUT
A 88A & B AS
A 89A & B AS
A 91A & B AS
A 94A & B AS
-RESET UF 3
-UNIT FREED 3
-RESET UF 5
-UNIT FREED 5
-RESET UF 9
-UNIT FREED 9
-RESET UF 15
-UNIT FREED 15
-4
-5
+6
-4
+5
-6
+4
-5
-6
+4
+5
+6

FIG.97B

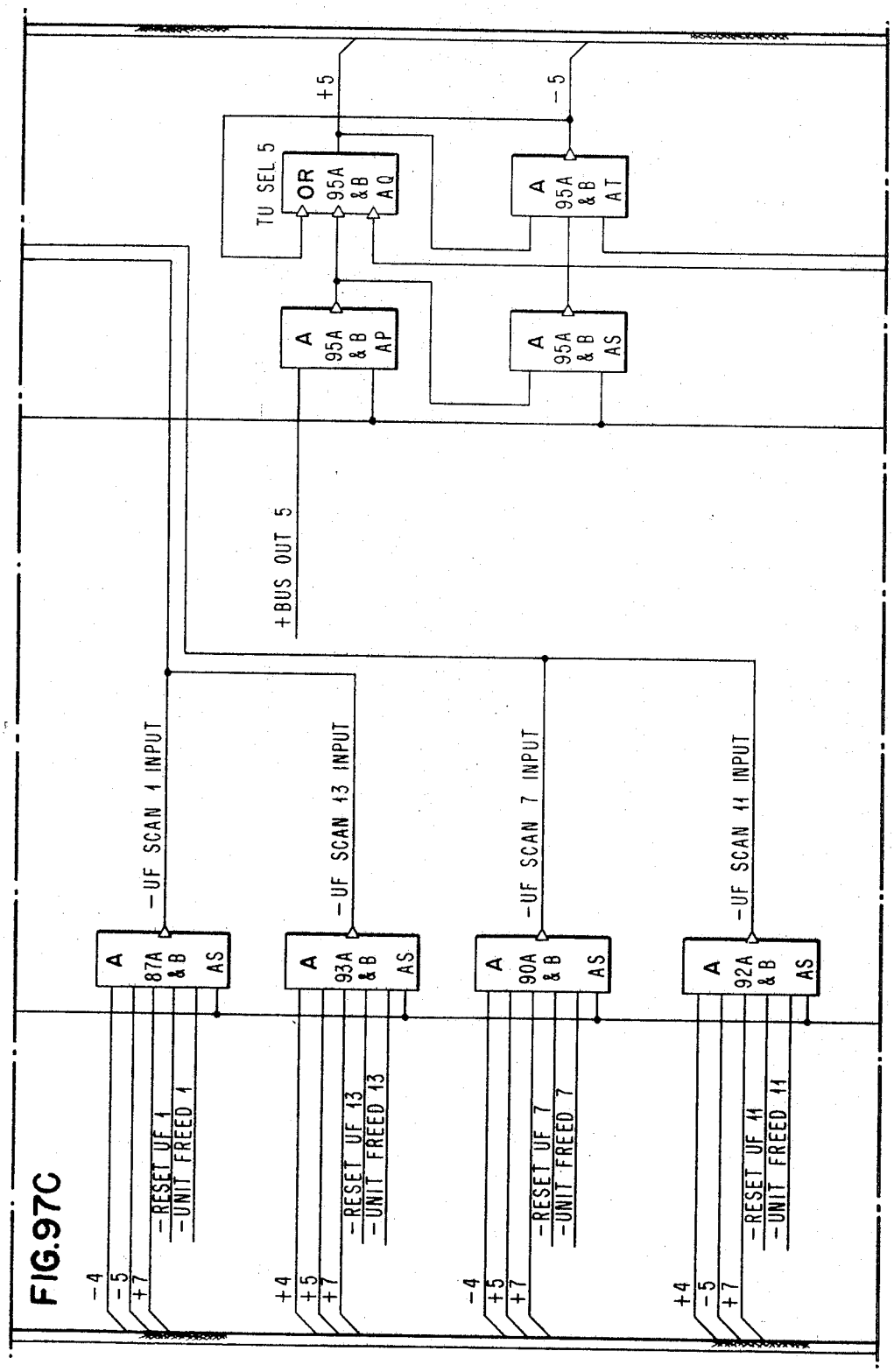

+5
-5
TU SEL 5
OR 95A &B AQ
A 95A &B AT
A 95A &B AP
A 95A &B AS
+BUS OUT 5
-UF SCAN 1 INPUT
-UF SCAN 13 INPUT
-UF SCAN 7 INPUT
-UF SCAN 11 INPUT
A 87A &B AS
A 93A &B AS
A 90A &B AS
A 92A &B AS
-RESET UF 1
-UNIT FREED 1
-RESET UF 13
-UNIT FREED 13
-RESET UF 7
-UNIT FREED 7
-RESET UF 11
-UNIT FREED 11
-4
-5
+7
+4
+5
+7
-4
+5
+7
+4
-5
+7

FIG.97C

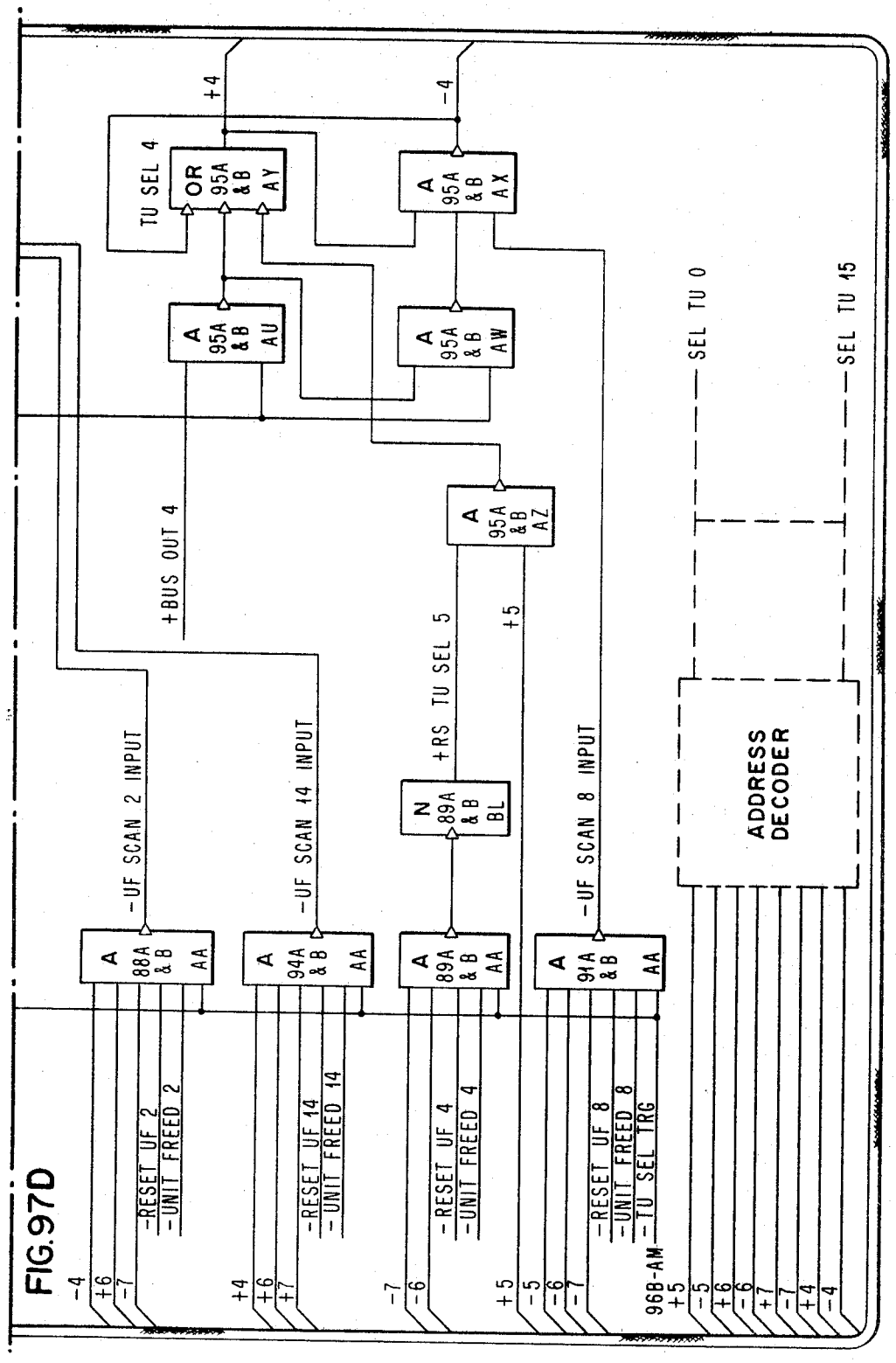

TU SEL 4
OR 95A & B AY
A 95A & B AX
A 95A & B AU
A 95A & B AW
A 95A & B AZ
+4
-4
+BUS OUT 4
+5
+RS TU SEL 5
N 89A & B BL
-UF SCAN 2 INPUT
-UF SCAN 14 INPUT
-UF SCAN 8 INPUT
A 88A & B AA
A 94A & B AA
A 89A & B AA
A 91A & B AA
-RESET UF 2
-UNIT FREED 2
-RESET UF 14
-UNIT FREED 14
-RESET UF 4
-UNIT FREED 4
-RESET UF 8
-UNIT FREED 8
-TU SEL TRG
96B-AM
SEL TU 0
SEL TU 45
ADDRESS DECODER
-4 +6 -7
+4 +6 +7
-7 -6
+5 -5 -6 -7
+5 -5 +6 -6 +7 -7 +4 -4

FIG.97D

FIG. 98

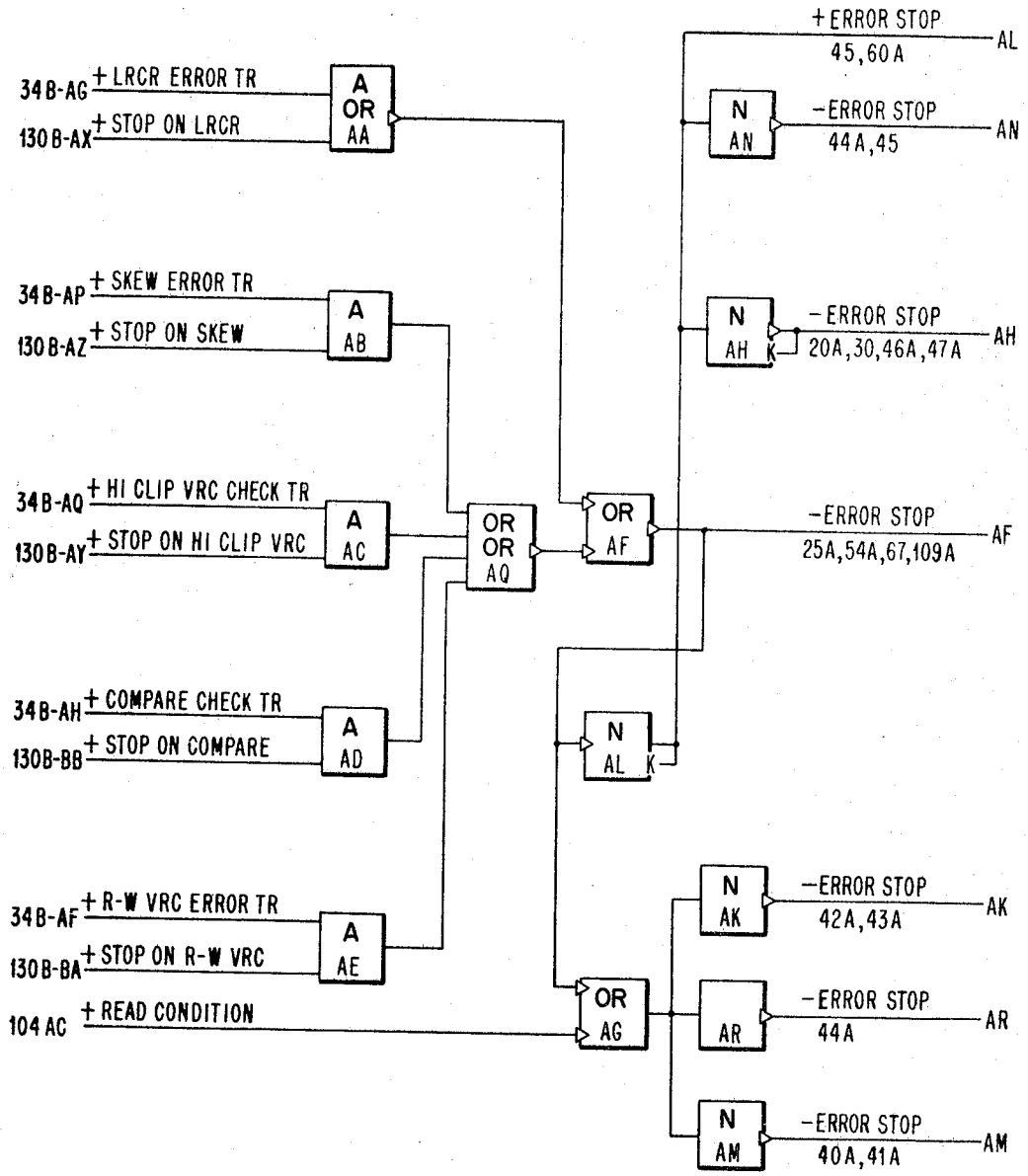

34B-AG +LRCR ERROR TR
130B-AX +STOP ON LRCR
A OR AA
34B-AP +SKEW ERROR TR
130B-AZ +STOP ON SKEW
A AB
34B-AQ +HI CLIP VRC CHECK TR
130B-AY +STOP ON HI CLIP VRC
A AC
34B-AH +COMPARE CHECK TR
130B-BB +STOP ON COMPARE
A AD
34B-AF +R-W VRC ERROR TR
130B-BA +STOP ON R-W VRC
A AE
104 AC +READ CONDITION
OR OR AQ
OR AF
N AL K
OR AG
+ERROR STOP 45,60A AL
N AN
-ERROR STOP 44A,45 AN
N AH K
-ERROR STOP 20A,30,46A,47A AH
-ERROR STOP 25A,54A,67,109A AF
N AK
-ERROR STOP 42A,43A AK
AR
-ERROR STOP 44A AR
N AM
-ERROR STOP 40A,41A AM

FIG. 99

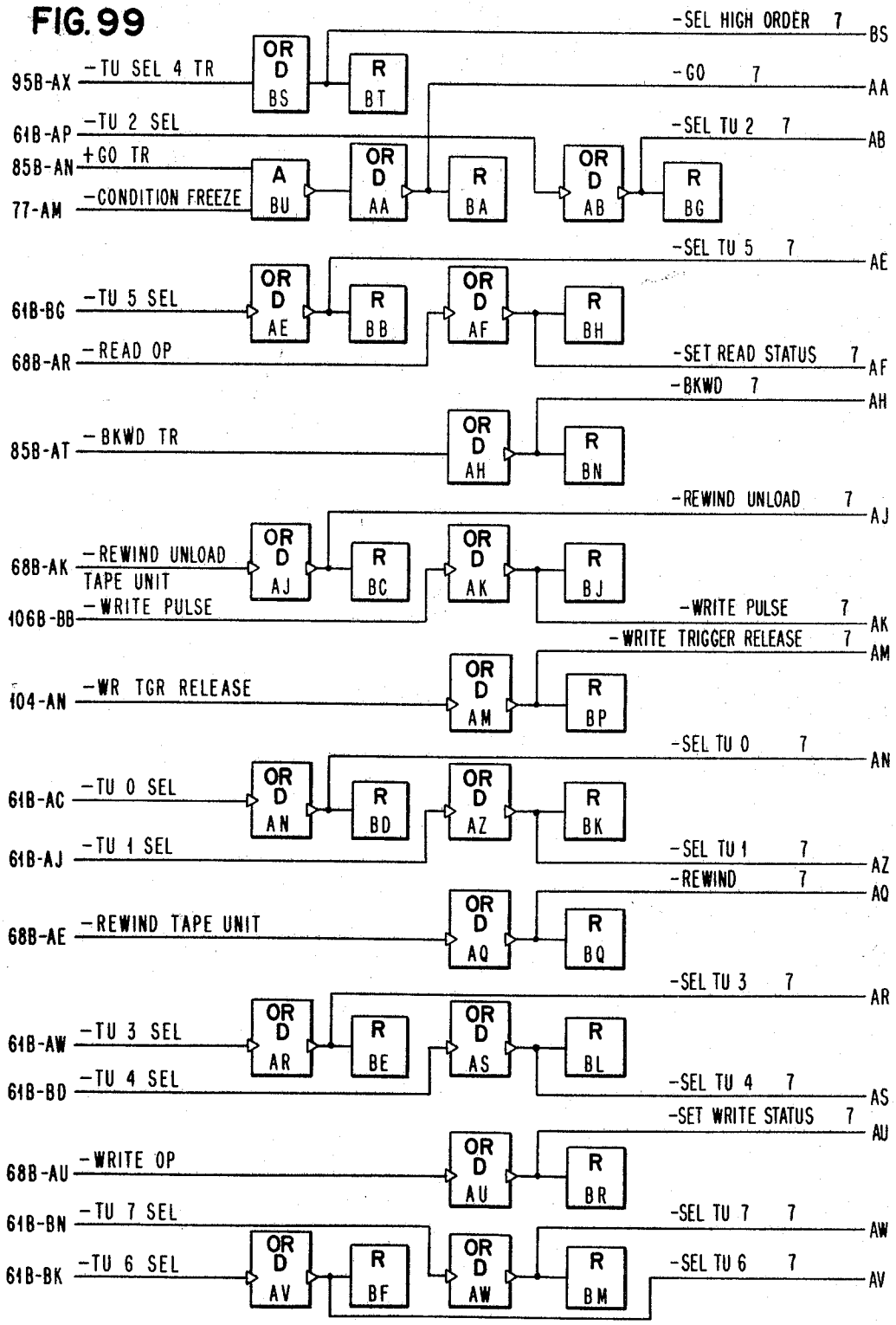

95B-AX
-TU SEL 4 TR
61B-AP
-TU 2 SEL
85B-AN
+GO TR
77-AM
-CONDITION FREEZE
61B-BG
-TU 5 SEL
68B-AR
-READ OP
85B-AT
-BKWD TR
68B-AK
-REWIND UNLOAD TAPE UNIT
106B-BB
-WRITE PULSE
104-AN
-WR TGR RELEASE
61B-AC
-TU 0 SEL
61B-AJ
-TU 1 SEL
68B-AE
-REWIND TAPE UNIT
61B-AW
-TU 3 SEL
61B-BD
-TU 4 SEL
68B-AU
-WRITE OP
61B-BN
-TU 7 SEL
61B-BK
-TU 6 SEL
OR D BS
R BT
A BU
OR D AA
R BA
OR D AB
R BG
OR D AE
R BB
OR D AF
R BH
OR D AH
R BN
OR D AJ
R BC
OR D AK
R BJ
OR D AM
R BP
OR D AN
R BD
OR D AZ
R BK
OR D AQ
R BQ
OR D AR
R BE
OR D AS
R BL
OR D AU
R BR
OR D AV
R BF
OR D AW
R BM
-SEL HIGH ORDER 7
BS
-GO 7
AA
-SEL TU 2 7
AB
-SEL TU 5 7
AE
-SET READ STATUS 7
AF
-BKWD 7
AH
-REWIND UNLOAD 7
AJ
-WRITE PULSE 7
AK
-WRITE TRIGGER RELEASE 7
AM
-SEL TU 0 7
AN
-SEL TU 1 7
AZ
-REWIND 7
AQ
-SEL TU 3 7
AR
-SEL TU 4 7
AS
-SET WRITE STATUS 7
AU
-SEL TU 7 7
AW
-SEL TU 6 7
AV

FIG.100A

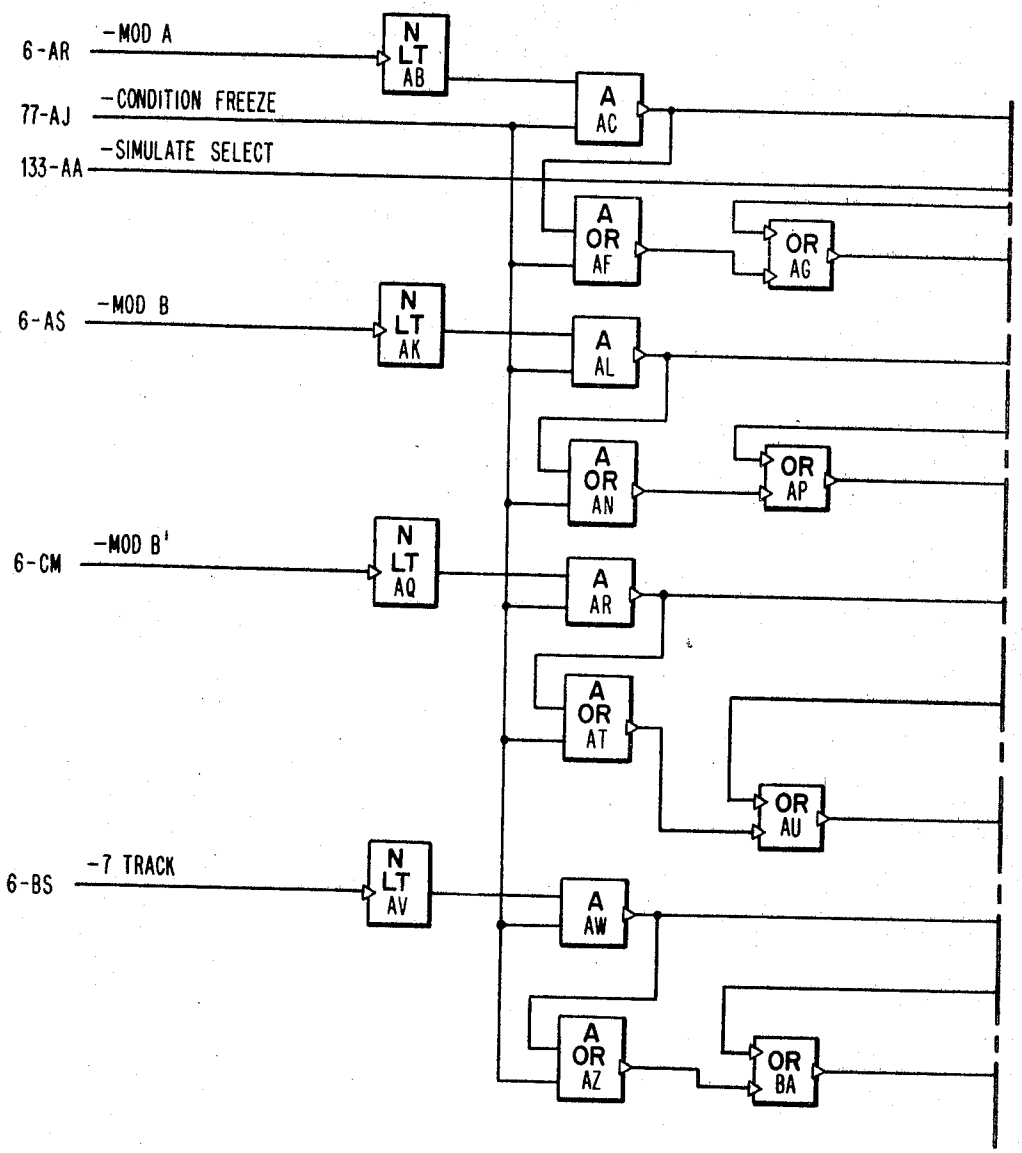

6-AR
-MOD A
77-AJ
-CONDITION FREEZE
133-AA
-SIMULATE SELECT
6-AS
-MOD B
6-CM
-MOD B'
6-BS
-7 TRACK
N LT AB
N LT AK
N LT AQ
N LT AV
A AC
A OR AF
OR AG
A AL
A OR AN
OR AP
A AR
A OR AT
OR AU
A AW
A OR AZ
OR BA

FIG.100B

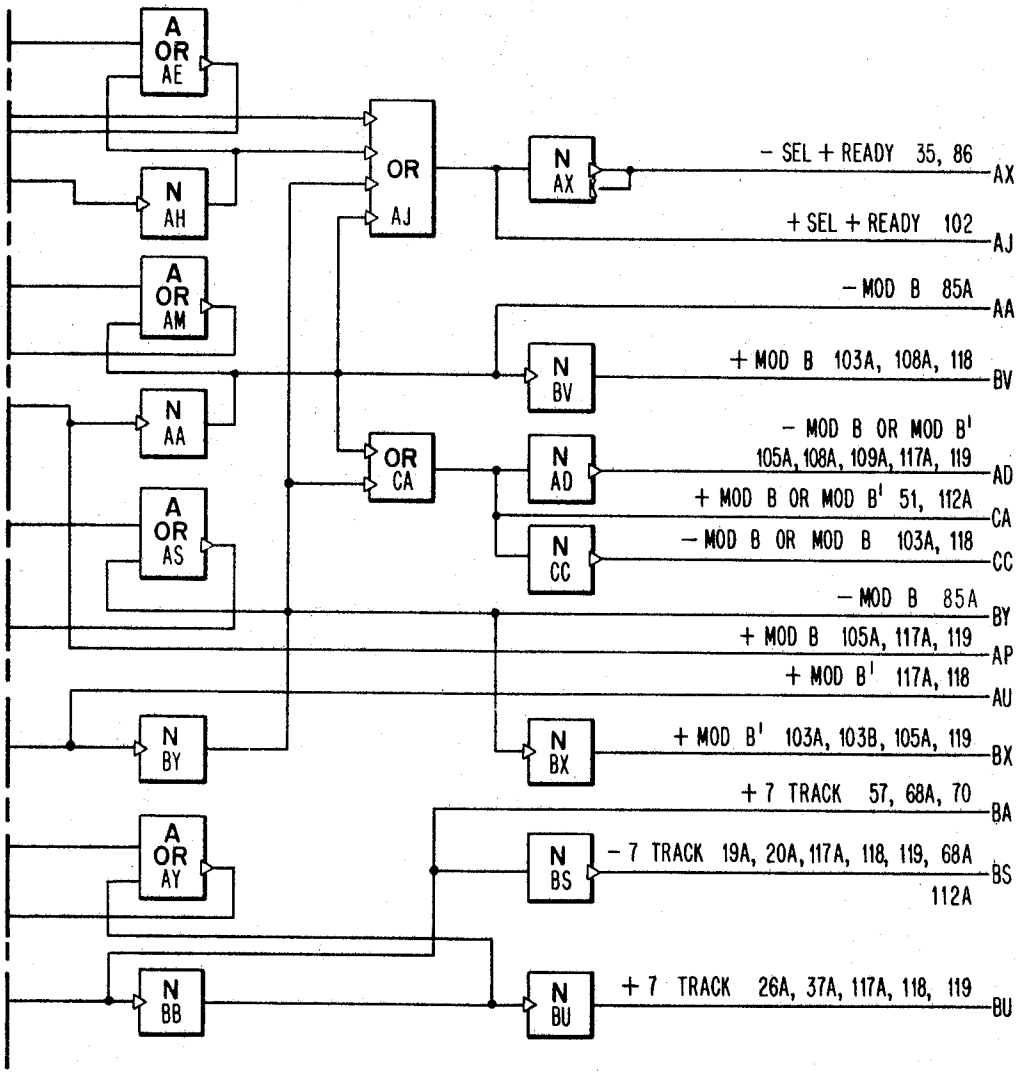

A OR AE
N AH
A OR AM
N AA
A OR AS
N BY
A OR AY
N BB
OR AJ
OR CA
N AX
N BV
N AD
N CC
N BX
N BS
N BU
- SEL + READY 35, 86
AX
+ SEL + READY 102
AJ
- MOD B 85A
AA
+ MOD B 103A, 108A, 118
BV
- MOD B OR MOD B'
105A, 108A, 109A, 117A, 119
AD
+ MOD B OR MOD B' 51, 112A
CA
- MOD B OR MOD B 103A, 118
CC
- MOD B 85A
BY
+ MOD B 105A, 117A, 119
AP
+ MOD B' 117A, 118
AU
+ MOD B' 103A, 103B, 105A, 119
BX
+ 7 TRACK 57, 68A, 70
BA
- 7 TRACK 19A, 20A, 117A, 118, 119, 68A
112A
BS
+ 7 TRACK 26A, 37A, 117A, 118, 119
BU

FIG. 101

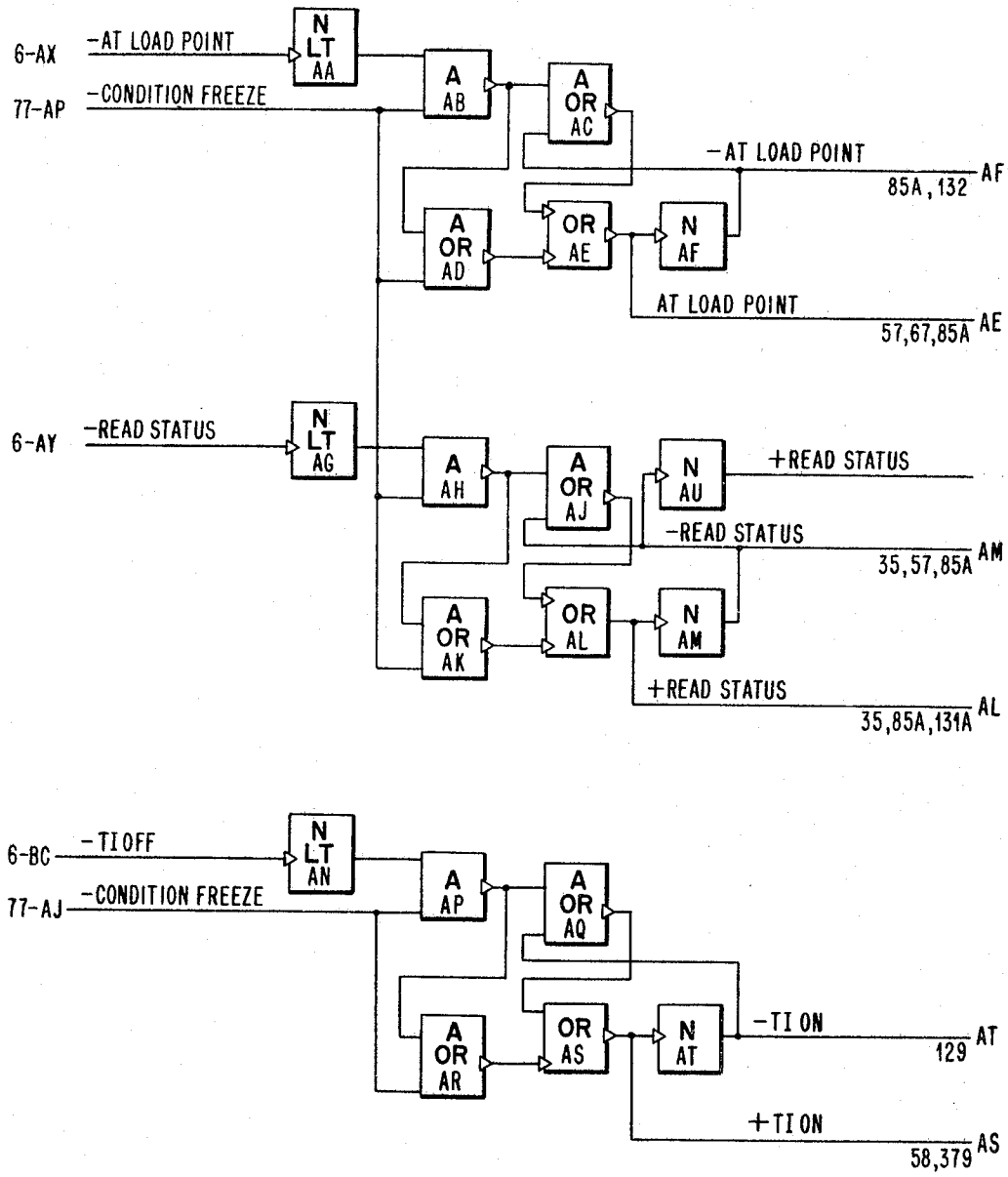

6-AX
-AT LOAD POINT
77-AP
-CONDITION FREEZE
N LT AA
A AB
A OR AC
A OR AD
OR AE
N AF
-AT LOAD POINT
85A,132
AF
AT LOAD POINT
57,67,85A
AE
6-AY
-READ STATUS
N LT AG
A AH
A OR AJ
A OR AK
OR AL
N AU
N AM
+READ STATUS
-READ STATUS
35,57,85A
AM
+READ STATUS
35,85A,131A
AL
6-BC
-TI OFF
77-AJ
-CONDITION FREEZE
N LT AN
A AP
A OR AQ
A OR AR
OR AS
N AT
-TI ON
129
AT
+TI ON
58,379
AS

FIG. 102

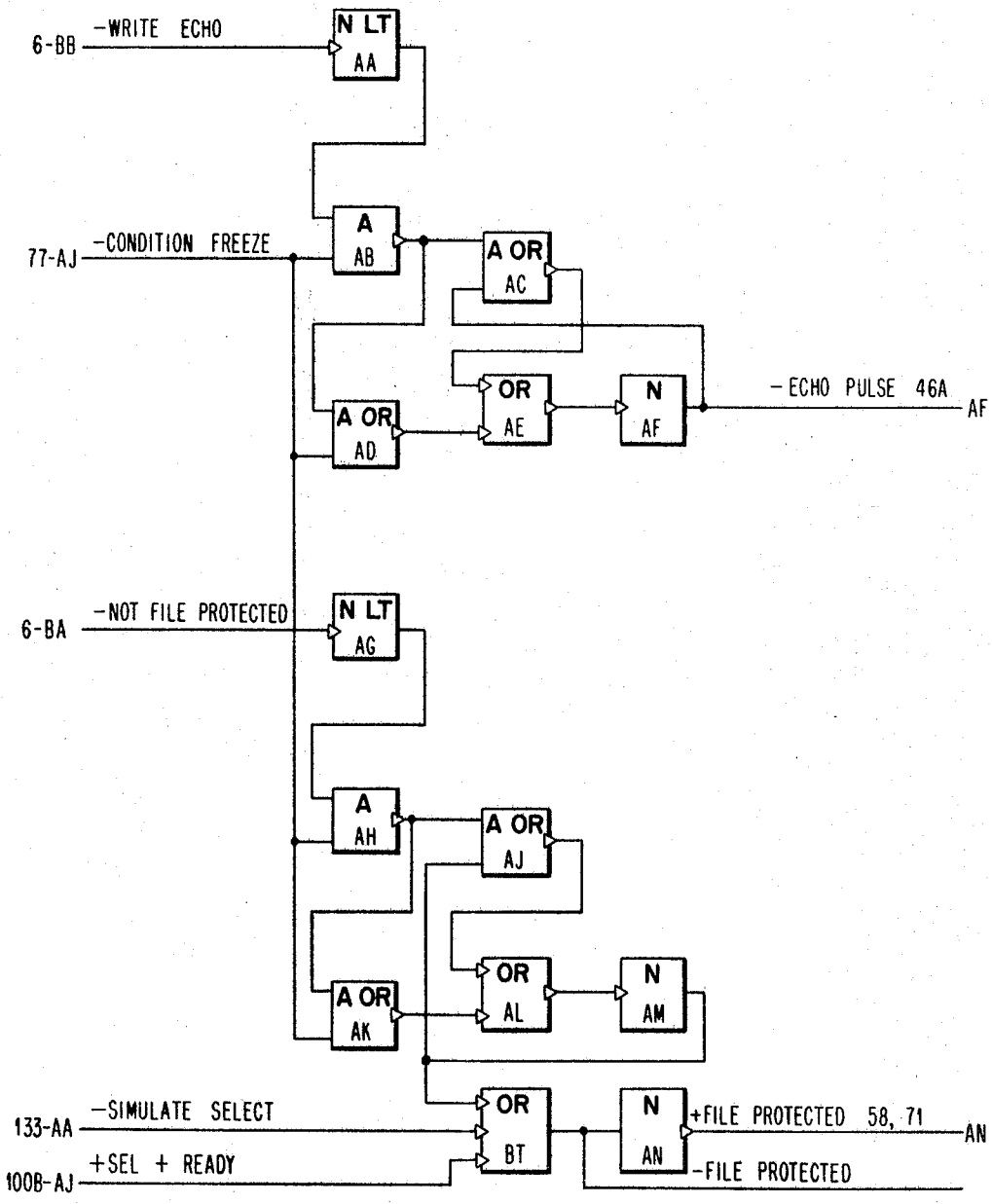

6-BB
-WRITE ECHO
N LT AA
77-AJ
-CONDITION FREEZE
A AB
A OR AC
A OR AD
OR AE
N AF
-ECHO PULSE 46A
AF
6-BA
-NOT FILE PROTECTED
N LT AG
A AH
A OR AJ
A OR AK
OR AL
N AM
133-AA
-SIMULATE SELECT
100B-AJ
+SEL + READY
OR BT
N AN
+FILE PROTECTED 58, 71
AN
-FILE PROTECTED

FIG. 103A

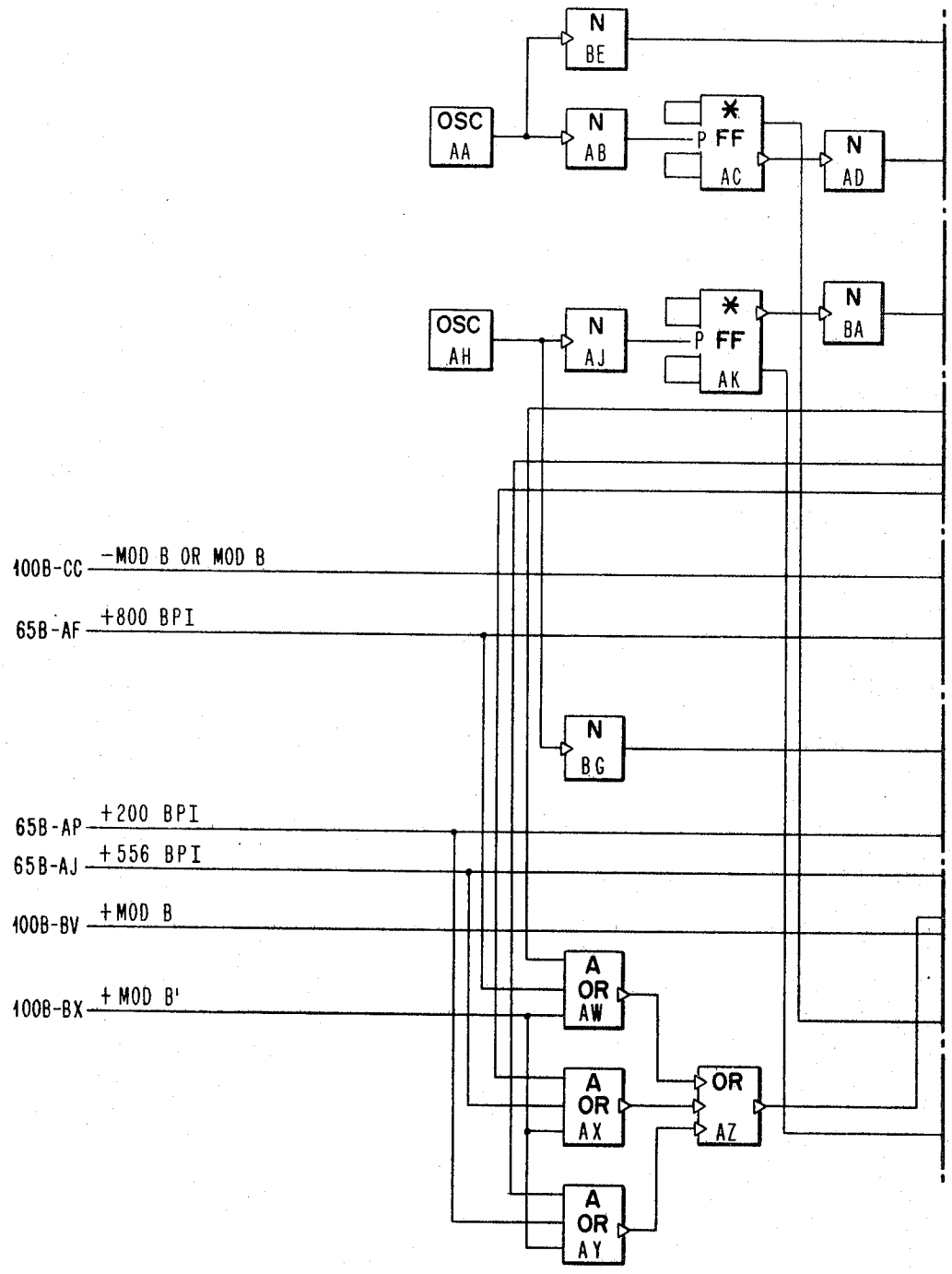

N BE
OSC AA
N AB
P FF AC
N AD
OSC AH
N AJ
P FF AK
N BA
100B-CC -MOD B OR MOD B
65B-AF +800 BPI
N BG
65B-AP +200 BPI
65B-AJ +556 BPI
100B-BV +MOD B
100B-BX +MOD B'
A OR AW
A OR AX
A OR AY
OR AZ

FIG. 103B

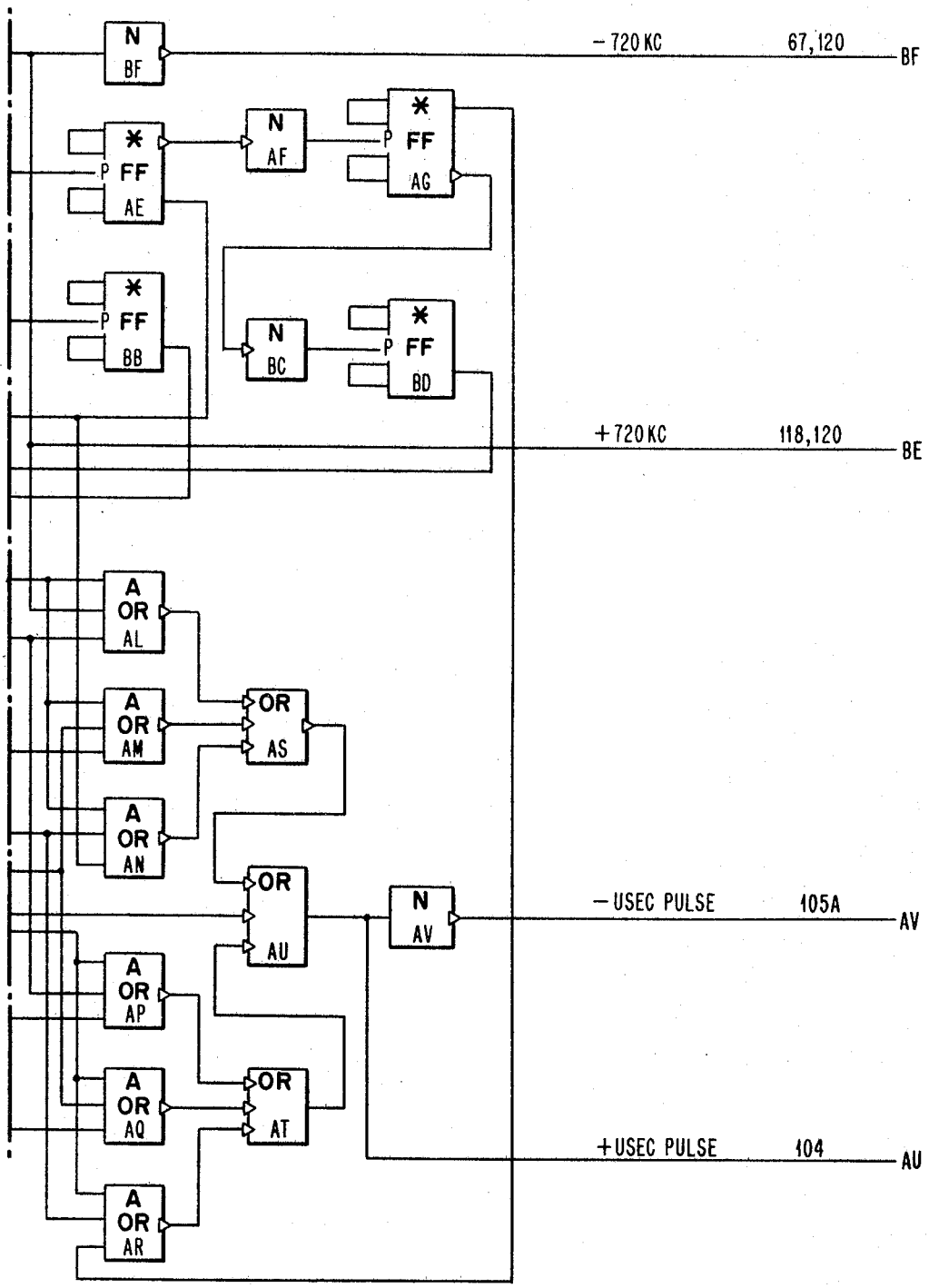

N BF
−720 KC
67,120
BF
* P FF AE
N AF
* P FF AG
* P FF BB
N BC
* P FF BD
+720 KC
118,120
BE
A OR AL
A OR AM
OR AS
A OR AN
OR AU
N AV
−USEC PULSE
105A
AV
A OR AP
A OR AQ
OR AT
+USEC PULSE
104
AU
A OR AR

FIG. 104

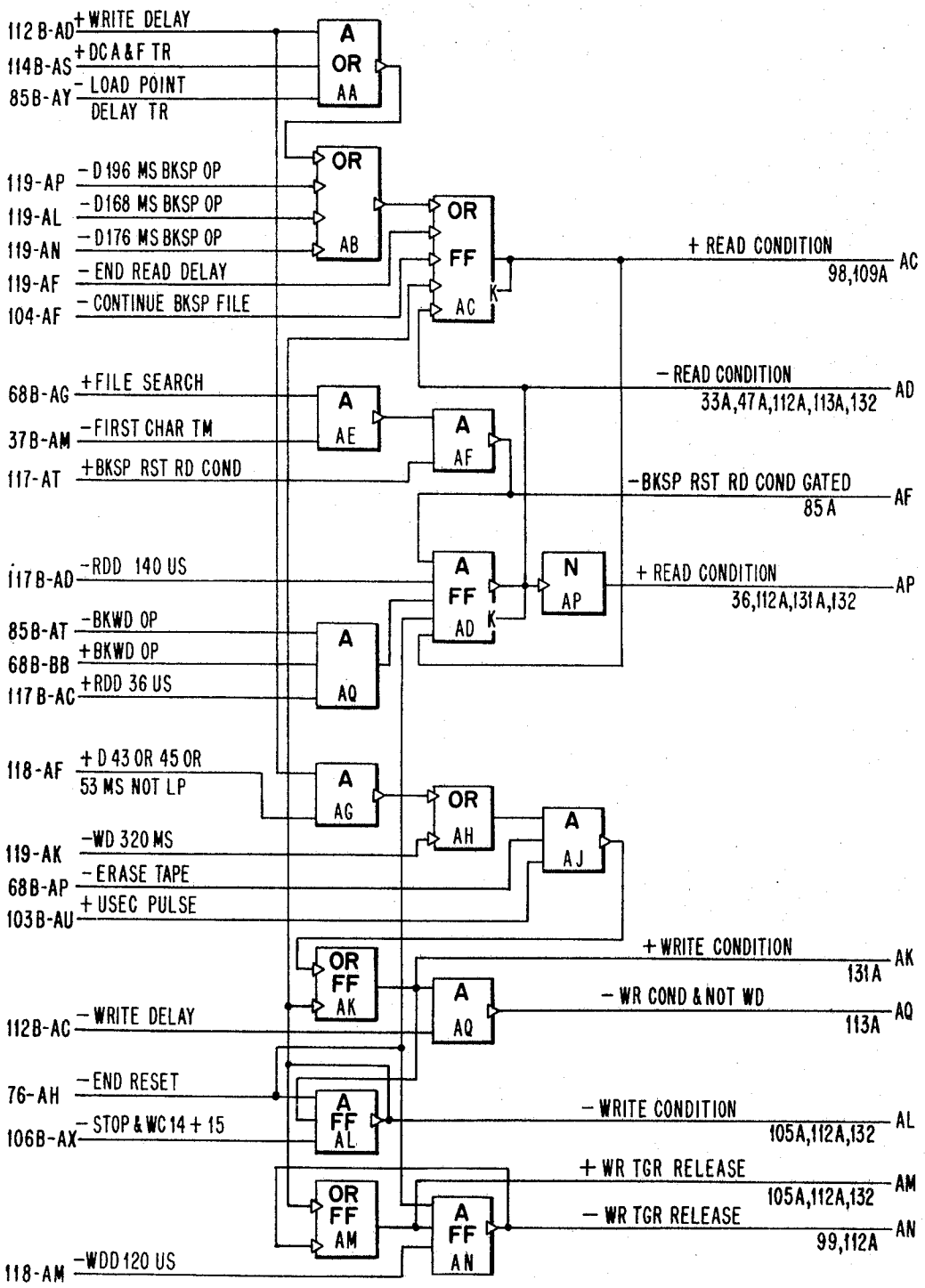

112B-AD +WRITE DELAY
114B-AS +DC A&F TR
85B-AY -LOAD POINT DELAY TR
A OR AA
119-AP -D196 MS BKSP OP
119-AL -D168 MS BKSP OP
119-AN -D176 MS BKSP OP
119-AF -END READ DELAY
104-AF -CONTINUE BKSP FILE
OR AB
OR FF AC K
+READ CONDITION 98,109A AC
68B-AG +FILE SEARCH
37B-AM -FIRST CHAR TM
117-AT +BKSP RST RD COND
A AE
A AF
-READ CONDITION 33A,47A,112A,113A,132 AD
-BKSP RST RD COND GATED 85A AF
117B-AD -RDD 140 US
A FF AD K
N AP
+READ CONDITION 36,112A,131A,132 AP
85B-AT -BKWD OP
68B-BB +BKWD OP
117B-AC +RDD 36 US
A AQ
118-AF +D43 OR 45 OR 53 MS NOT LP
A AG
OR AH
A AJ
119-AK -WD 320 MS
68B-AP -ERASE TAPE
103B-AU +USEC PULSE
OR FF AK
+WRITE CONDITION 131A AK
A AQ
-WR COND & NOT WD 113A AQ
112B-AC -WRITE DELAY
76-AH -END RESET
106B-AX -STOP & WC 14+15
A FF AL
-WRITE CONDITION 105A,112A,132 AL
+WR TGR RELEASE 105A,112A,132 AM
OR FF AM
A FF AN
-WR TGR RELEASE 99,112A AN
118-AM -WDD 120 US

FIG.105A

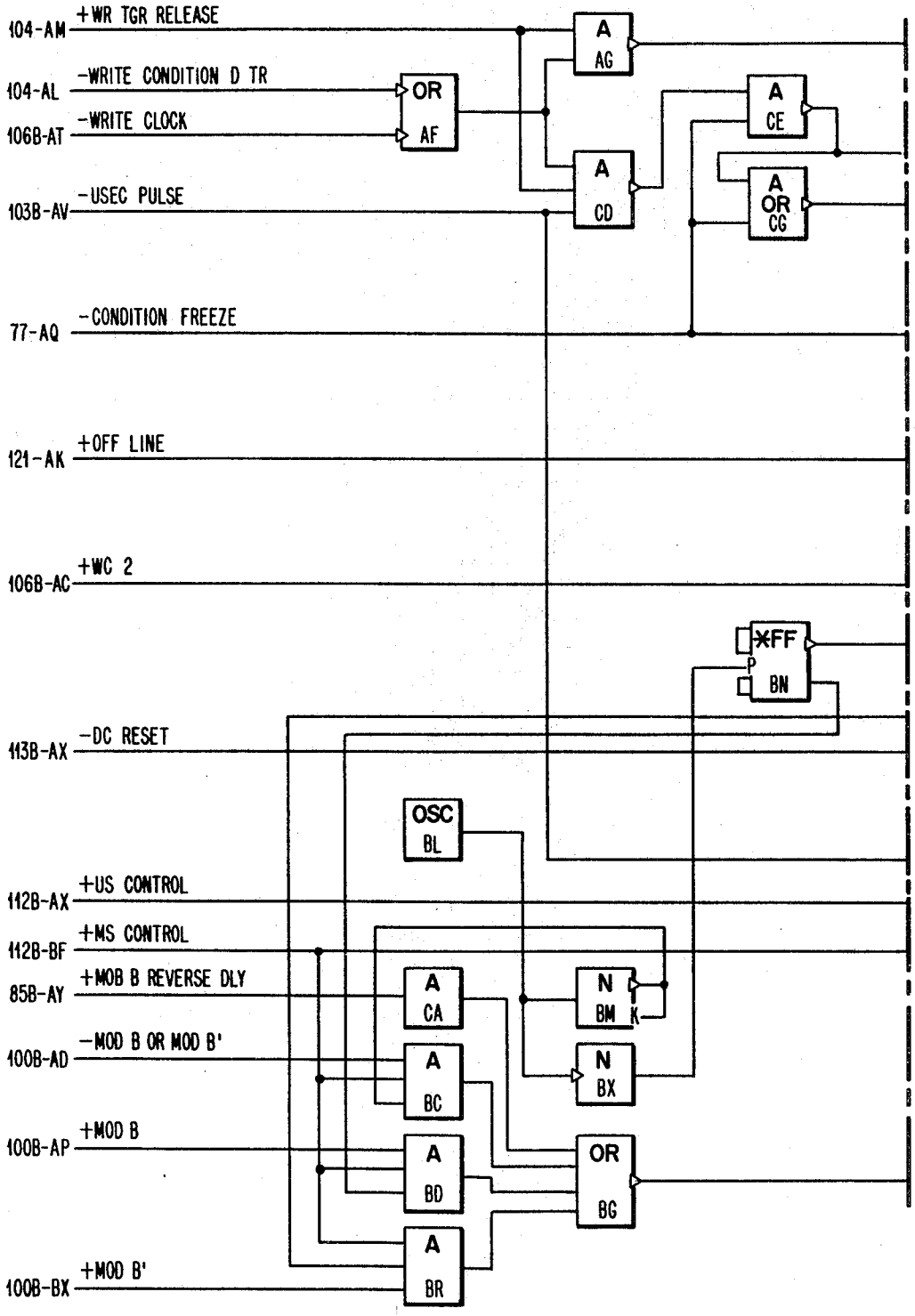

104-AM
+WR TGR RELEASE
104-AL
-WRITE CONDITION D TR
106B-AT
-WRITE CLOCK
OR
AF
A
AG
A
CE
A
CD
A
OR
CG
103B-AV
-USEC PULSE
77-AQ
-CONDITION FREEZE
121-AK
+OFF LINE
106B-AC
+WC 2
FF
P
BN
113B-AX
-DC RESET
OSC
BL
112B-AX
+US CONTROL
112B-BF
+MS CONTROL
85B-AY
+MOB B REVERSE DLY
A
CA
N
BM
K
100B-AD
-MOD B OR MOD B'
A
BC
N
BX
100B-AP
+MOD B
A
BD
OR
BG
100B-BX
+MOD B'
A
BR

FIG.105B

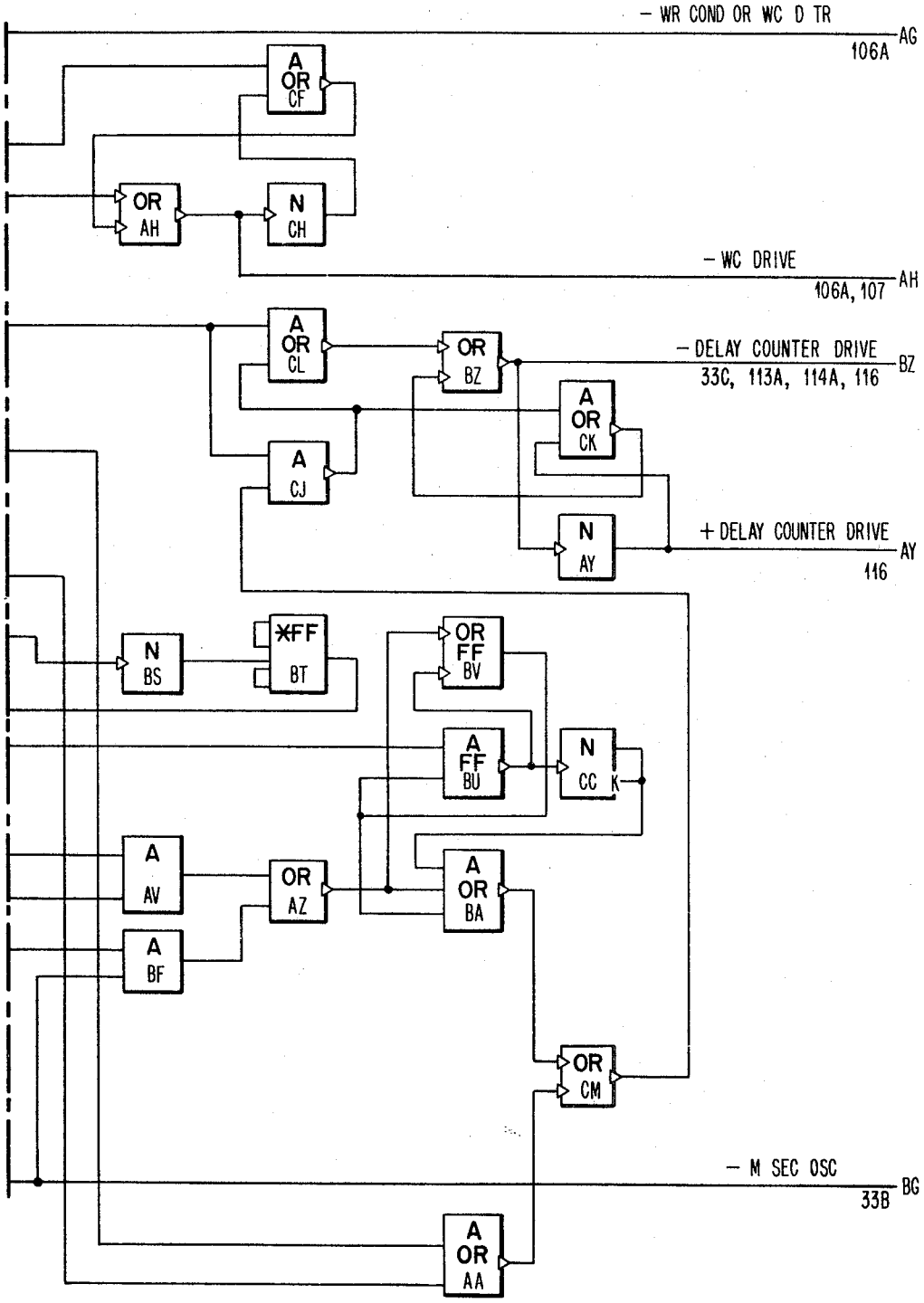

– WR COND OR WC D TR
106A
AG
A OR CF
OR AH
N CH
– WC DRIVE
106A, 107
AH
A OR CL
OR BZ
– DELAY COUNTER DRIVE
33C, 113A, 114A, 116
BZ
A OR CK
A CJ
N AY
+ DELAY COUNTER DRIVE
116
AY
N BS
XFF BT
OR FF BV
A FF BU
N CC K
A AV
OR AZ
A OR BA
A BF
OR CM
– M SEC OSC
33B
BG
A OR AA

FIG. 106A

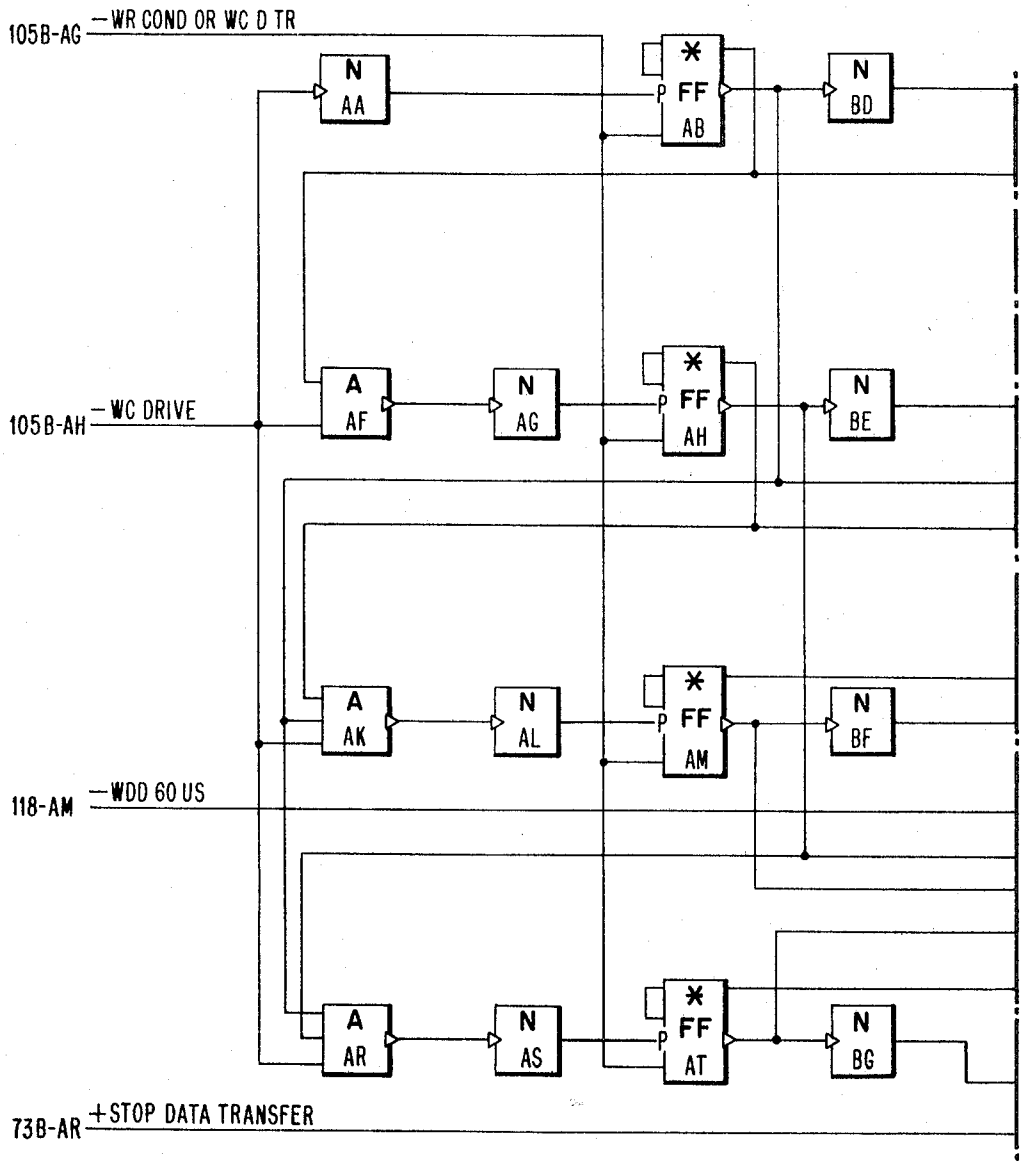

105B-AG
—WR COND OR WC D TR
N AA
* FF AB
N BD
105B-AH
—WC DRIVE
A AF
N AG
* FF AH
N BE
A AK
N AL
* FF AM
N BF
118-AM
—WDD 60 US
A AR
N AS
* FF AT
N BG
73B-AR
+STOP DATA TRANSFER

FIG.106B

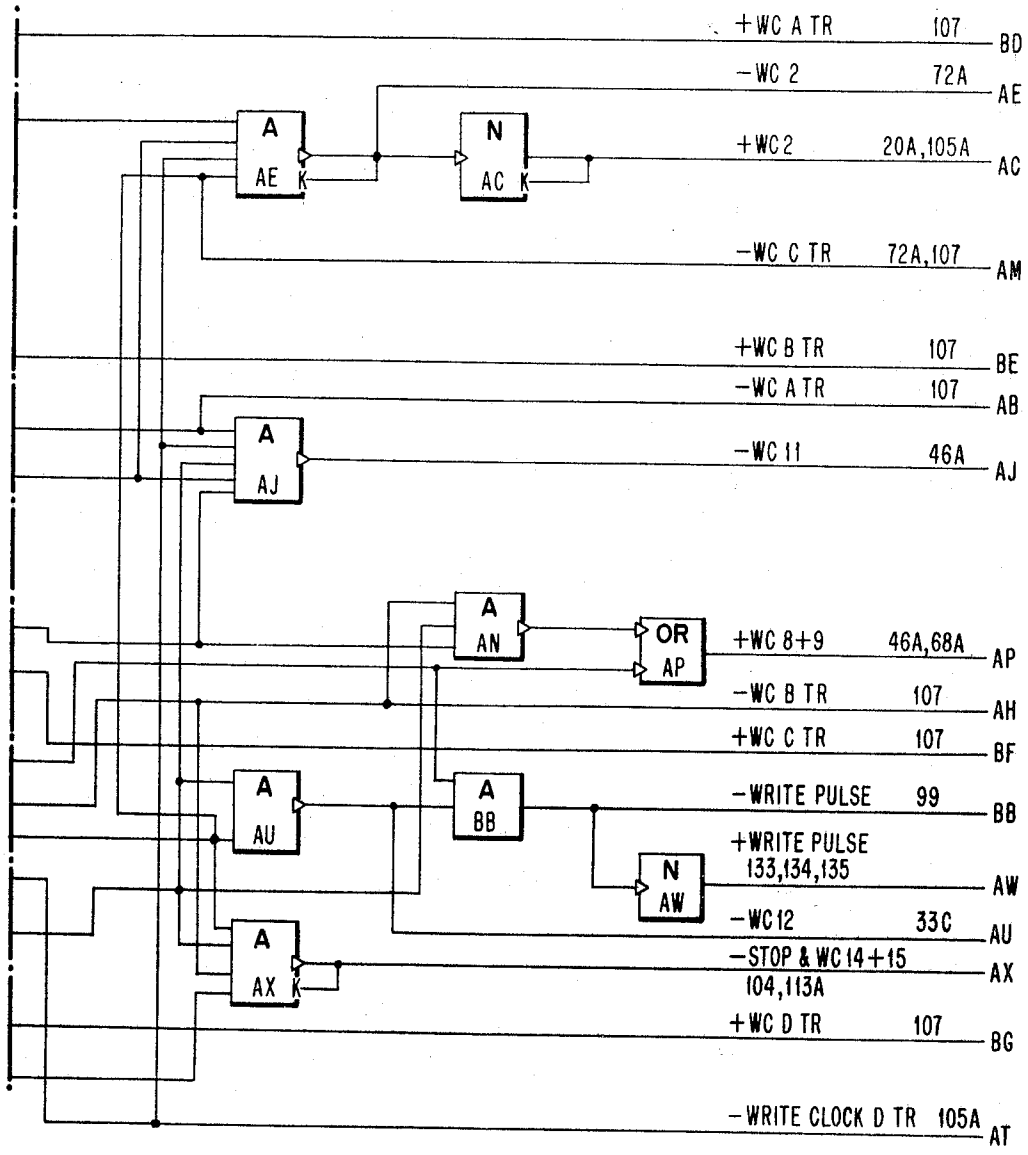

+WC A TR 107 BD
−WC 2 72A AE
A AE K
N AC K
+WC 2 20A,105A AC
−WC C TR 72A,107 AM
+WC B TR 107 BE
−WC A TR 107 AB
A AJ
−WC 11 46A AJ
A AN
OR AP
+WC 8+9 46A,68A AP
−WC B TR 107 AH
+WC C TR 107 BF
A AU
A BB
−WRITE PULSE 99 BB
+WRITE PULSE 133,134,135 AW
N AW
−WC 12 33C AU
A AX K
−STOP & WC 14+15 104,113A AX
+WC D TR 107 BG
−WRITE CLOCK D TR 105A AT

FIG. 107

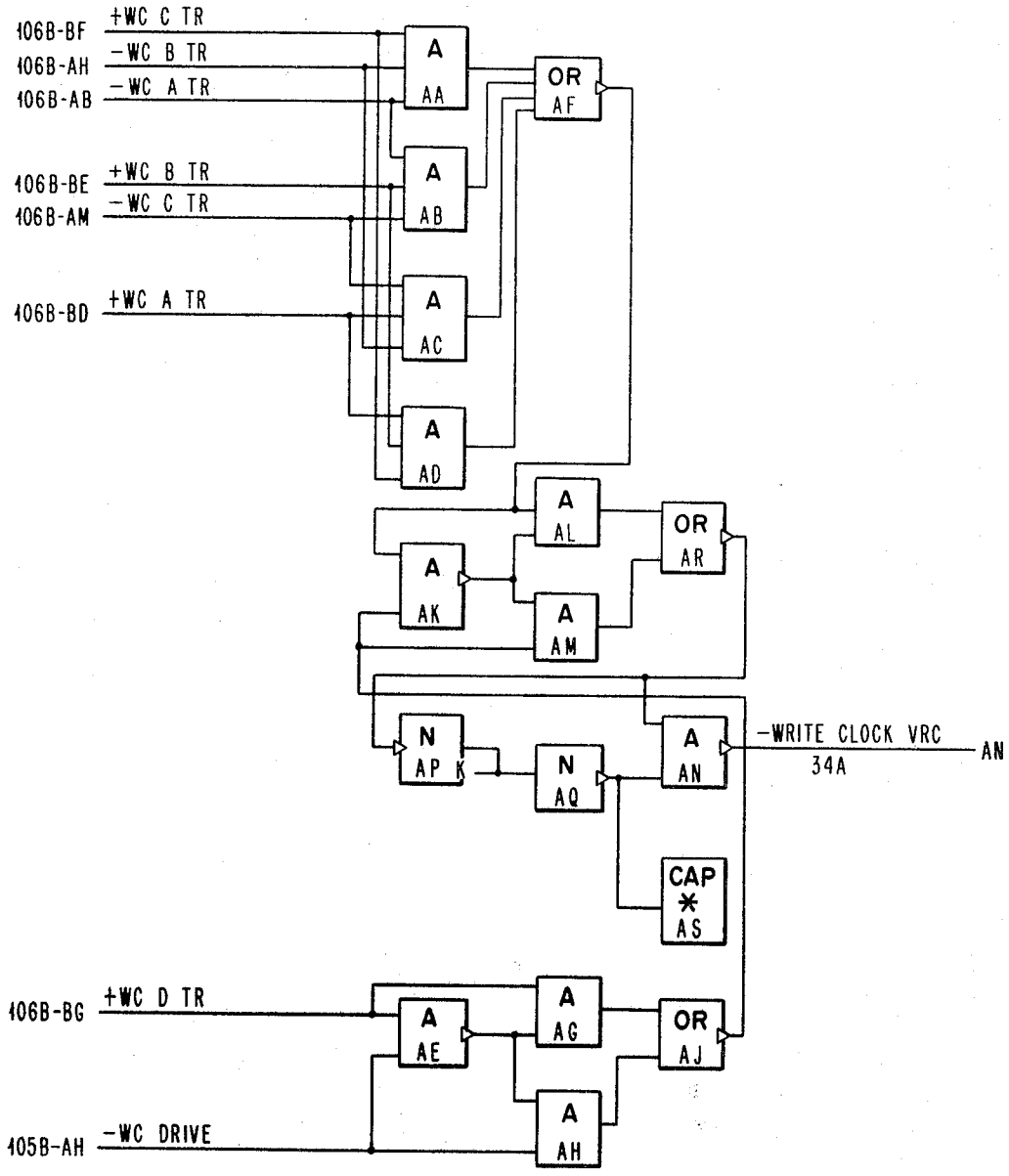

106B-BF
+WC C TR
106B-AH
-WC B TR
106B-AB
-WC A TR
106B-BE
+WC B TR
106B-AM
-WC C TR
106B-BD
+WC A TR
A AA
A AB
A AC
A AD
OR AF
A AL
A AK
A AM
OR AR
N AP K
N AQ
A AN
-WRITE CLOCK VRC 34A
AN
CAP * AS
106B-BG
+WC D TR
105B-AH
-WC DRIVE
A AE
A AG
A AH
OR AJ

FIG.108A

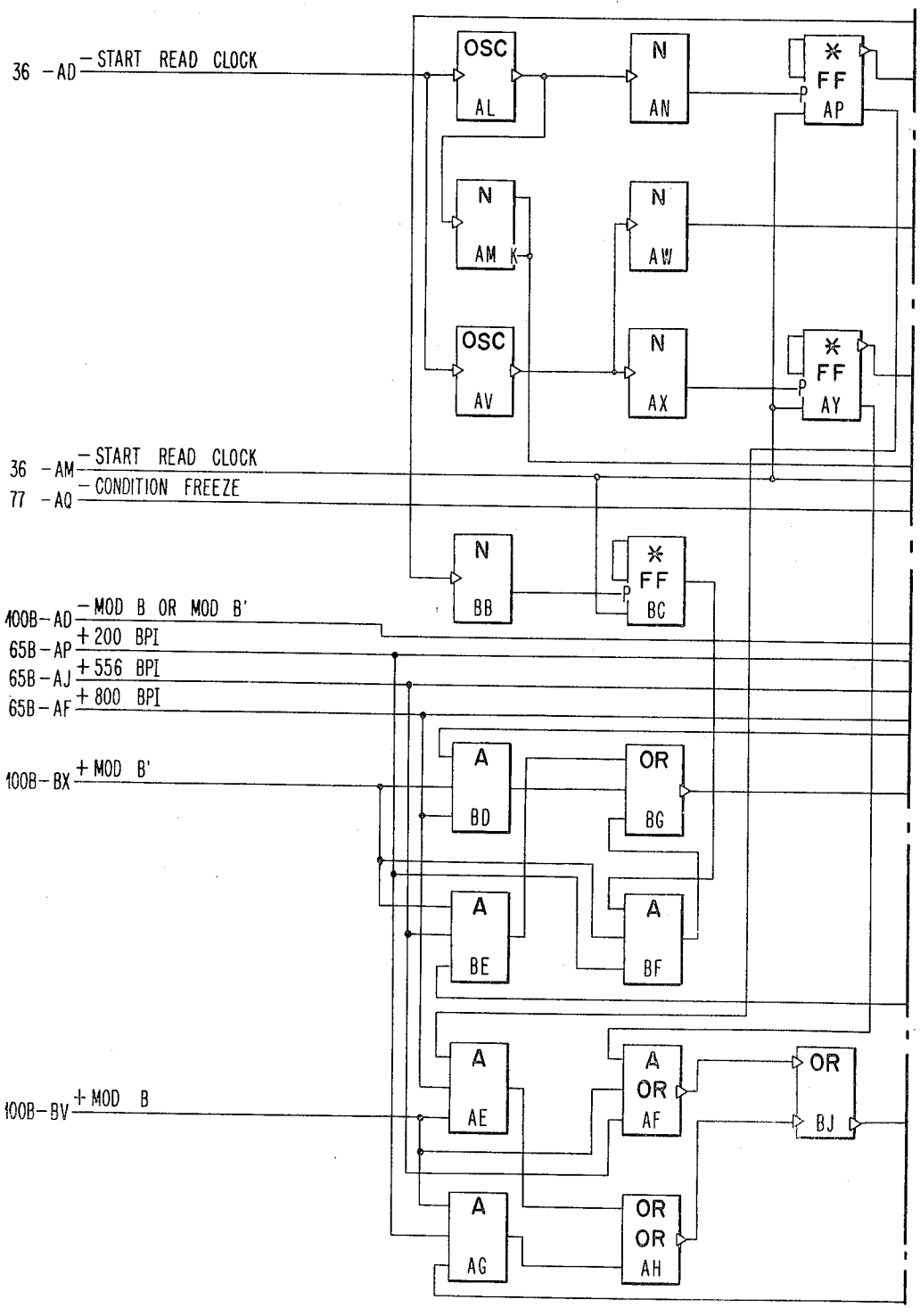

36 -AD
-START READ CLOCK
OSC AL
N AN
* FF AP
N AM
N AW
OSC AV
N AX
* FF AY
36 -AM
-START READ CLOCK
77 -AQ
-CONDITION FREEZE
N BB
* FF BC
100B-AD
-MOD B OR MOD B'
65B-AP
+200 BPI
65B-AJ
+556 BPI
65B-AF
+800 BPI
100B-BX
+MOD B'
A BD
OR BG
A BE
A BF
100B-BV
+MOD B
A AE
A OR AF
OR BJ
A AG
OR OR AH

FIG.108B

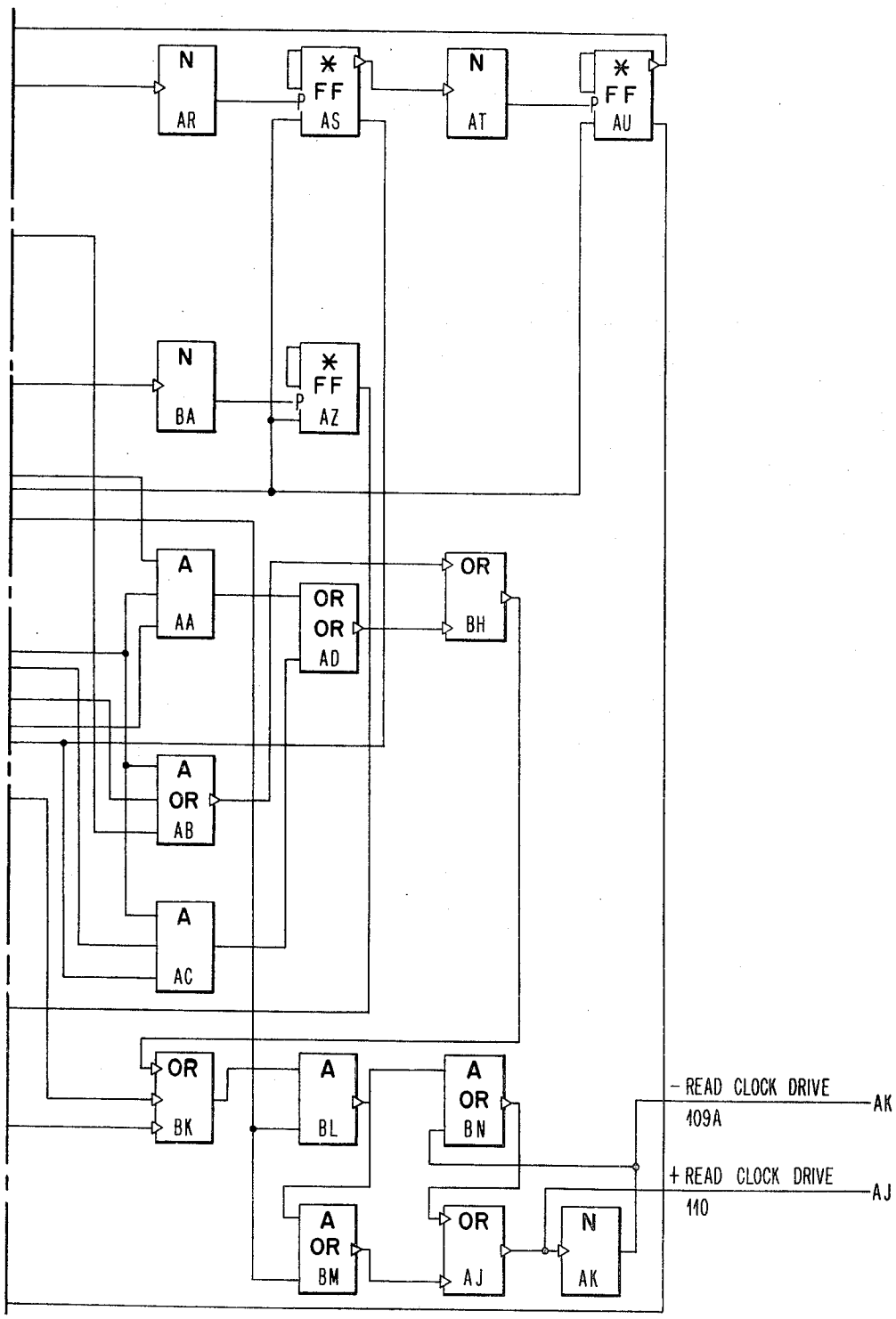

N
AR
* FF
AS
N
AT
* FF
AU
N
BA
* FF
AZ
A
AA
OR
OR
AD
OR
BH
A
OR
AB
A
AC
OR
BK
A
BL
A
OR
BN
- READ CLOCK DRIVE
109A
AK
+ READ CLOCK DRIVE
110
AJ
A
OR
BM
OR
AJ
N
AK

FIG. 109A

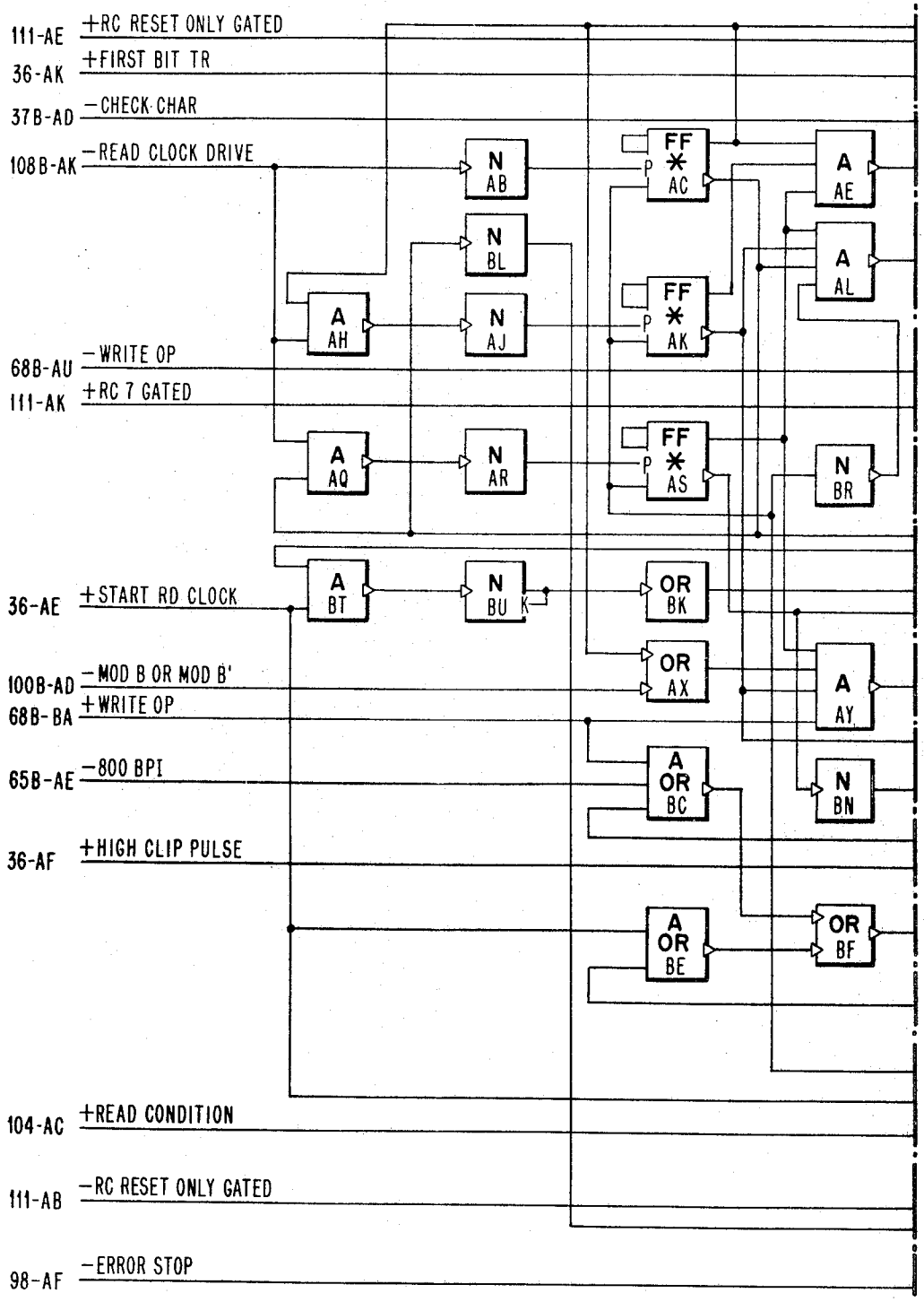

111-AE +RC RESET ONLY GATED
36-AK +FIRST BIT TR
37B-AD −CHECK CHAR
108B-AK −READ CLOCK DRIVE
68B-AU −WRITE OP
111-AK +RC 7 GATED
36-AE +START RD CLOCK
100B-AD −MOD B OR MOD B'
68B-BA +WRITE OP
65B-AE −800 BPI
36-AF +HIGH CLIP PULSE
104-AC +READ CONDITION
111-AB −RC RESET ONLY GATED
98-AF −ERROR STOP
N AB
N BL
A AH
N AJ
A AQ
N AR
A BT
N BU
FF AC
FF AK
FF AS
OR BK
OR AX
A OR BC
A OR BE
A AE
A AL
N BR
A AY
N BN
OR BF

FIG. 109B

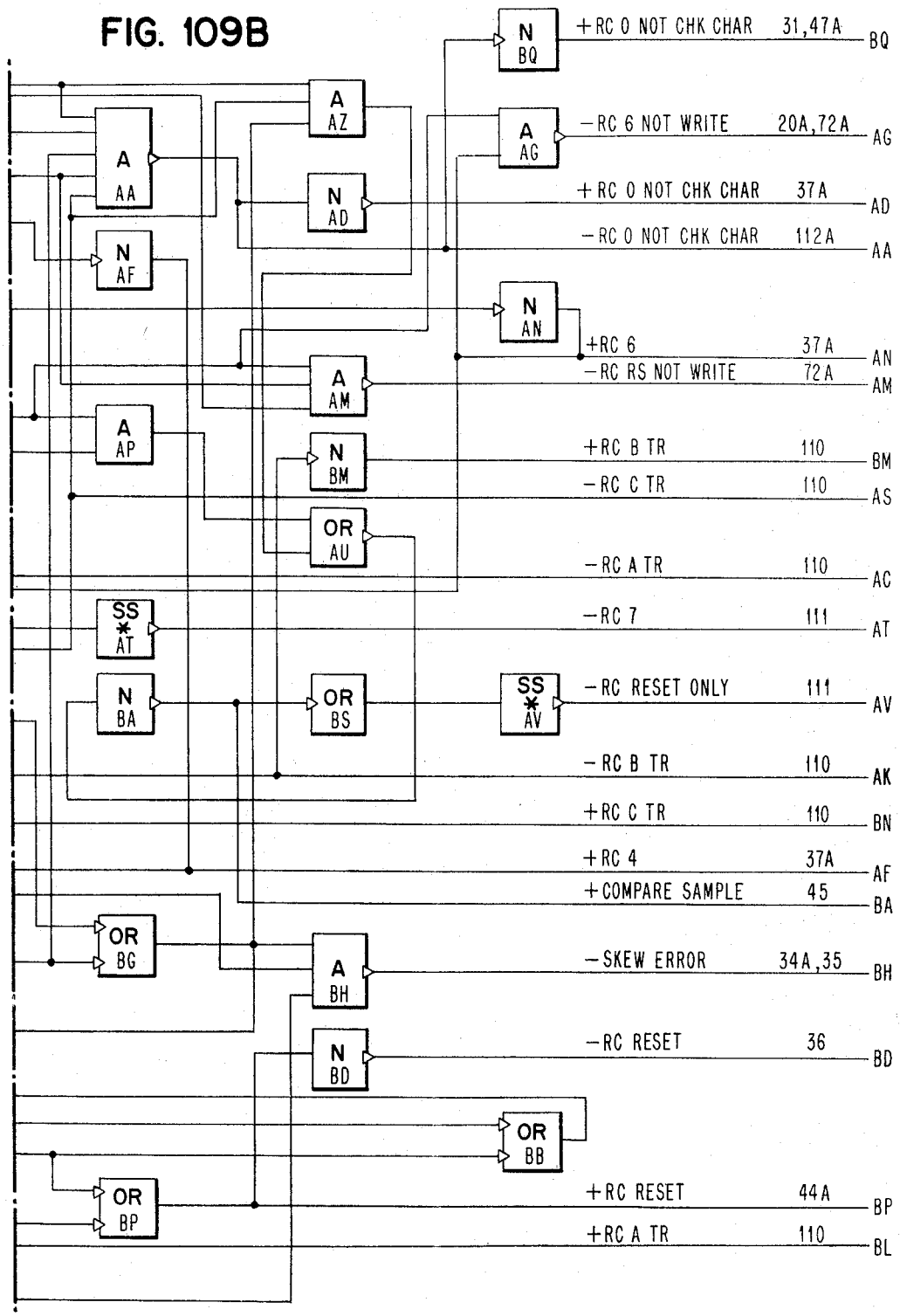

N BQ
+RC 0 NOT CHK CHAR 31,47A
BQ
A AZ
A AG
−RC 6 NOT WRITE 20A,72A
AG
A AA
N AD
+RC 0 NOT CHK CHAR 37A
AD
−RC 0 NOT CHK CHAR 112A
AA
N AF
N AN
+RC 6 37A
AN
−RC RS NOT WRITE 72A
AM
A AM
A AP
N BM
+RC B TR 110
BM
−RC C TR 110
AS
OR AU
−RC A TR 110
AC
SS * AT
−RC 7 111
AT
N BA
OR BS
SS * AV
−RC RESET ONLY 111
AV
−RC B TR 110
AK
+RC C TR 110
BN
+RC 4 37A
AF
+COMPARE SAMPLE 45
BA
OR BG
A BH
−SKEW ERROR 34A,35
BH
N BD
−RC RESET 36
BD
OR BB
OR BP
+RC RESET 44A
BP
+RC A TR 110
BL

FIG. 110

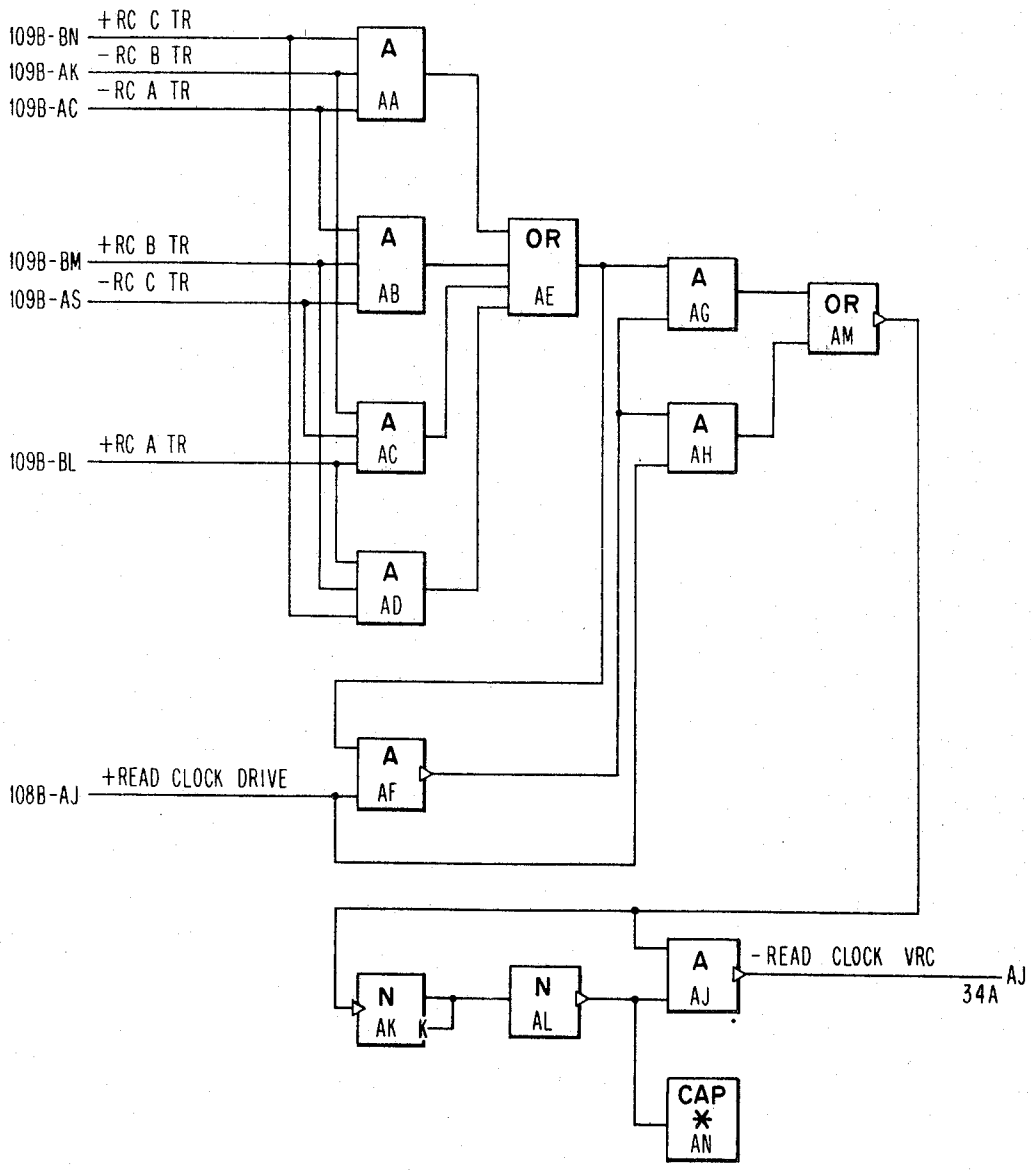

109B-BN
+RC C TR
109B-AK
-RC B TR
109B-AC
-RC A TR
109B-BM
+RC B TR
109B-AS
-RC C TR
109B-BL
+RC A TR
108B-AJ
+READ CLOCK DRIVE
A AA
A AB
A AC
A AD
OR AE
A AF
A AG
A AH
OR AM
N AK K
N AL
A AJ
-READ CLOCK VRC
AJ
34A
CAP * AN

FIG.111

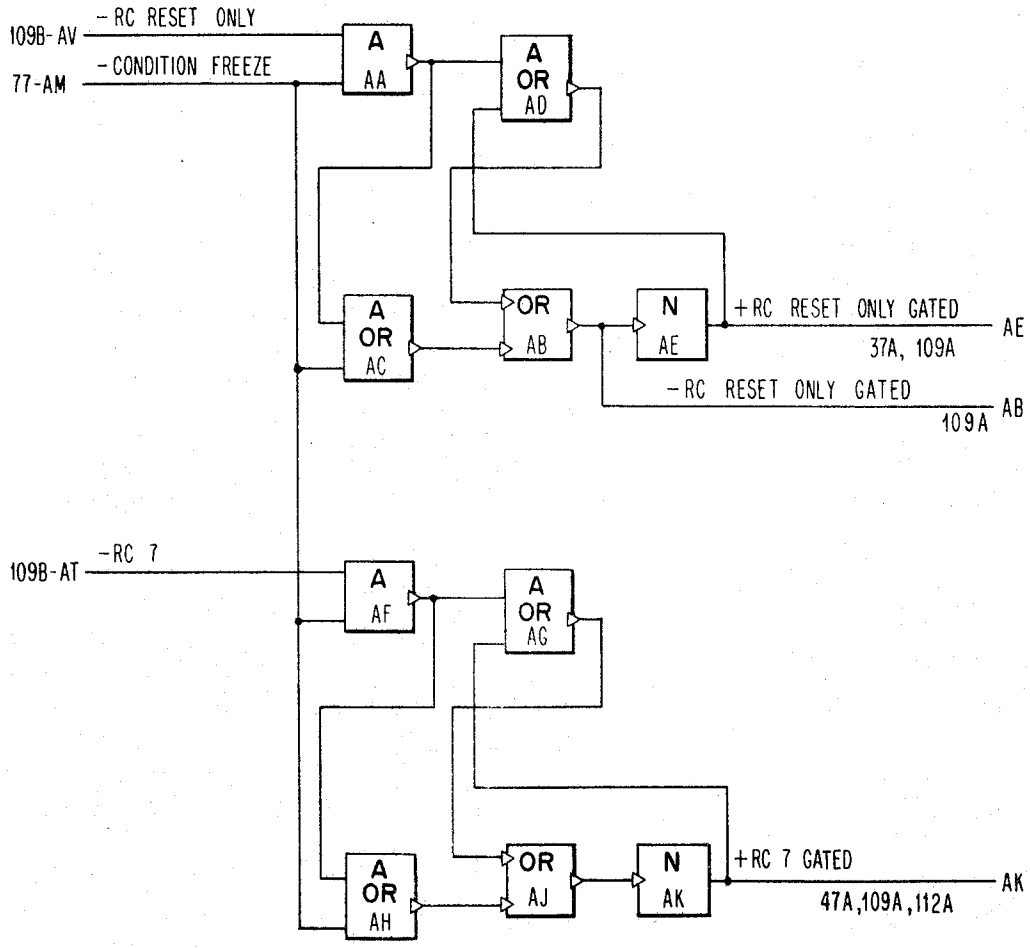

109B-AV
-RC RESET ONLY
77-AM
-CONDITION FREEZE
A AA
A OR AD
A OR AC
OR AB
N AE
+RC RESET ONLY GATED
37A, 109A
AE
-RC RESET ONLY GATED
109A
AB
109B-AT
-RC 7
A AF
A OR AG
A OR AH
OR AJ
N AK
+RC 7 GATED
47A,109A,112A
AK

FIG. 112A

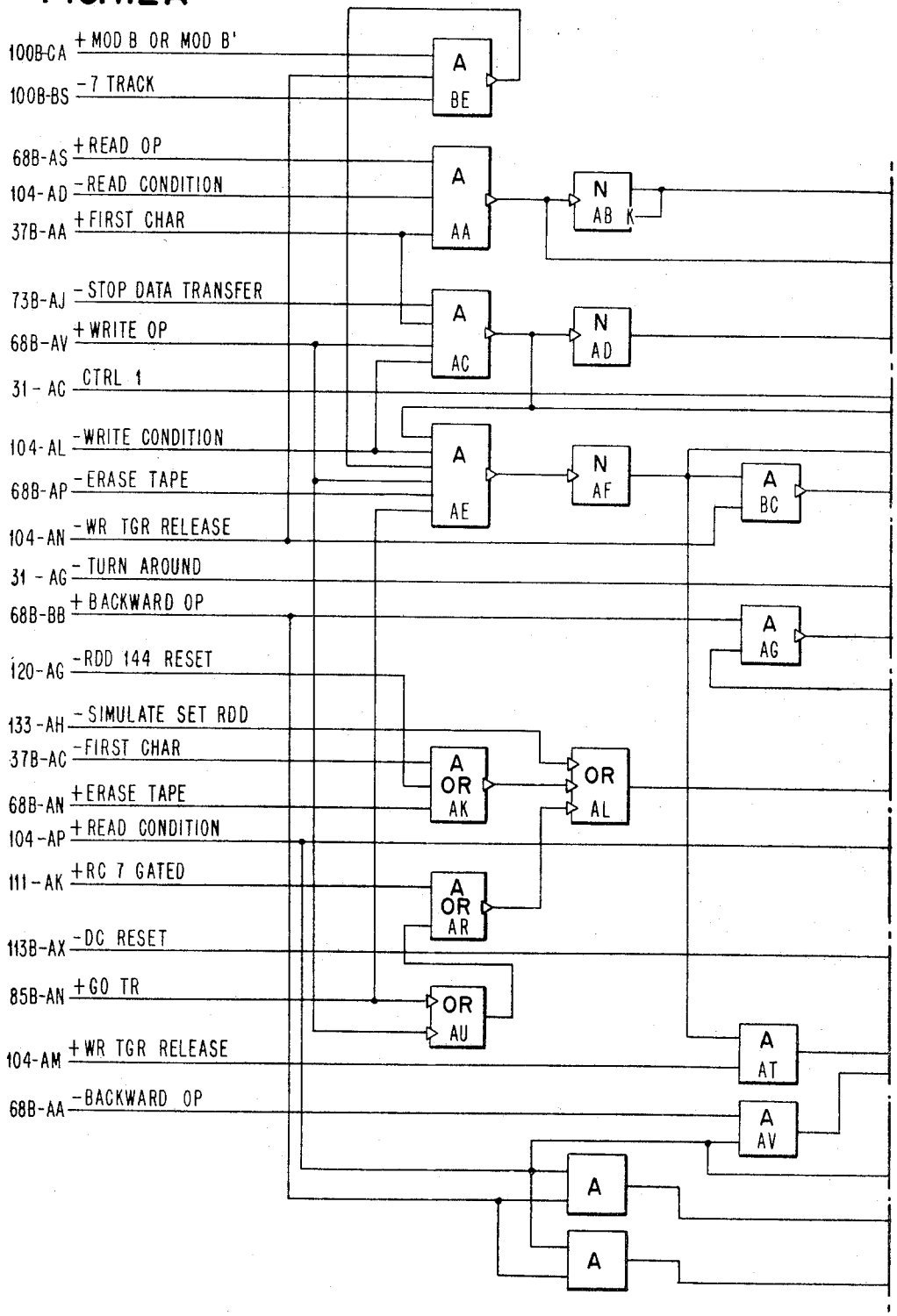

100B-CA +MOD B OR MOD B'
100B-BS -7 TRACK
68B-AS +READ OP
104-AD -READ CONDITION
37B-AA +FIRST CHAR
73B-AJ -STOP DATA TRANSFER
68B-AV +WRITE OP
31-AC CTRL 1
104-AL -WRITE CONDITION
68B-AP -ERASE TAPE
104-AN -WR TGR RELEASE
31-AG -TURN AROUND
68B-BB +BACKWARD OP
120-AG -RDD 144 RESET
133-AH -SIMULATE SET RDD
37B-AC -FIRST CHAR
68B-AN +ERASE TAPE
104-AP +READ CONDITION
111-AK +RC 7 GATED
113B-AX -DC RESET
85B-AN +GO TR
104-AM +WR TGR RELEASE
68B-AA -BACKWARD OP
A BE
A AA
N AB K
A AC
N AD
A AE
N AF
A BC
A AG
A OR AK
OR AL
A OR AR
OR AU
A AT
A AV
A
A

FIG. 112B

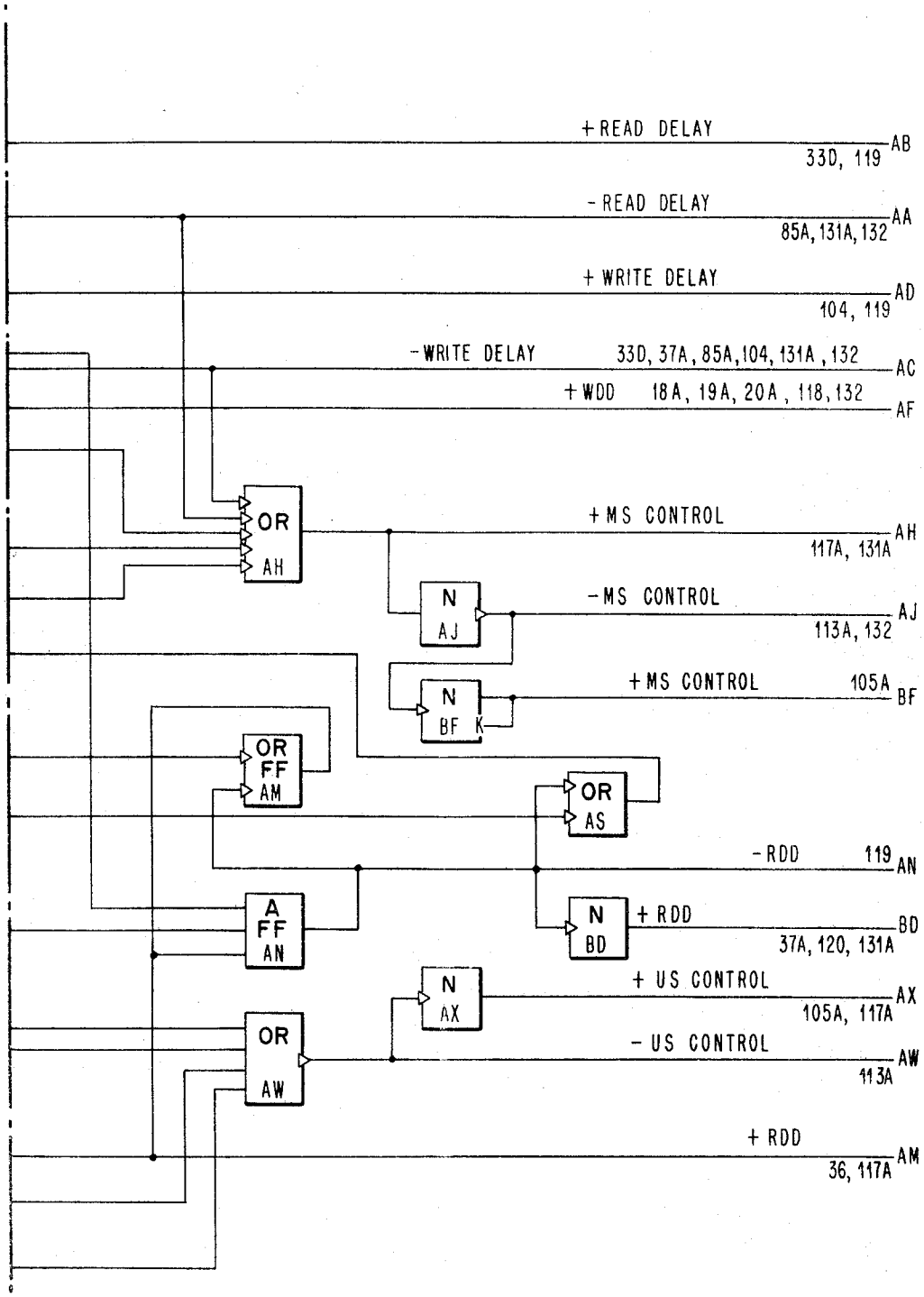

+READ DELAY
33D, 119
AB
-READ DELAY
85A, 131A, 132
AA
+WRITE DELAY
104, 119
AD
-WRITE DELAY
33D, 37A, 85A, 104, 131A, 132
AC
+WDD
18A, 19A, 20A, 118, 132
AF
OR
AH
+MS CONTROL
117A, 131A
AH
N
AJ
-MS CONTROL
113A, 132
AJ
N
BF K
+MS CONTROL
105A
BF
OR
FF
AM
OR
AS
-RDD
119
AN
A
FF
AN
N
BD
+RDD
37A, 120, 131A
BD
N
AX
+US CONTROL
105A, 117A
AX
OR
AW
-US CONTROL
113A
AW
+RDD
36, 117A
AM

FIG. 113A

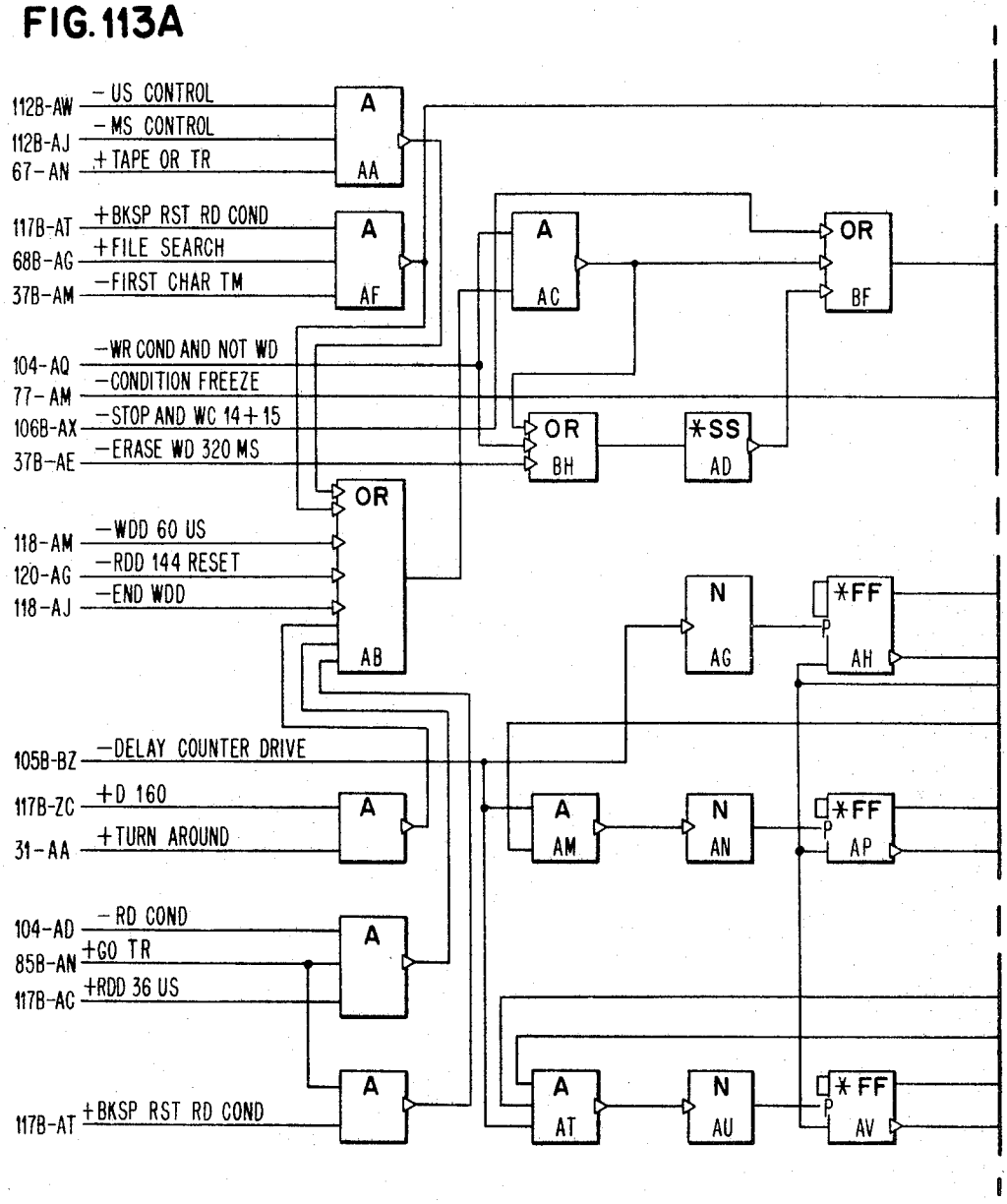

112B-AW
-US CONTROL
112B-AJ
-MS CONTROL
67-AN
+TAPE OR TR
A
AA
117B-AT
+BKSP RST RD COND
68B-AG
+FILE SEARCH
37B-AM
-FIRST CHAR TM
A
AF
A
AC
OR
BF
104-AQ
-WR COND AND NOT WD
77-AM
-CONDITION FREEZE
106B-AX
-STOP AND WC 14+15
37B-AE
-ERASE WD 320 MS
OR
BH
*SS
AD
OR
118-AM
-WDD 60 US
120-AG
-RDD 144 RESET
118-AJ
-END WDD
AB
N
AG
*FF
AH
105B-BZ
-DELAY COUNTER DRIVE
117B-ZC
+D 160
31-AA
+TURN AROUND
A
A
AM
N
AN
*FF
AP
104-AD
-RD COND
85B-AN
+GO TR
117B-AC
+RDD 36 US
A
A
AT
N
AU
*FF
AV
117B-AT
+BKSP RST RD COND

FIG.113B

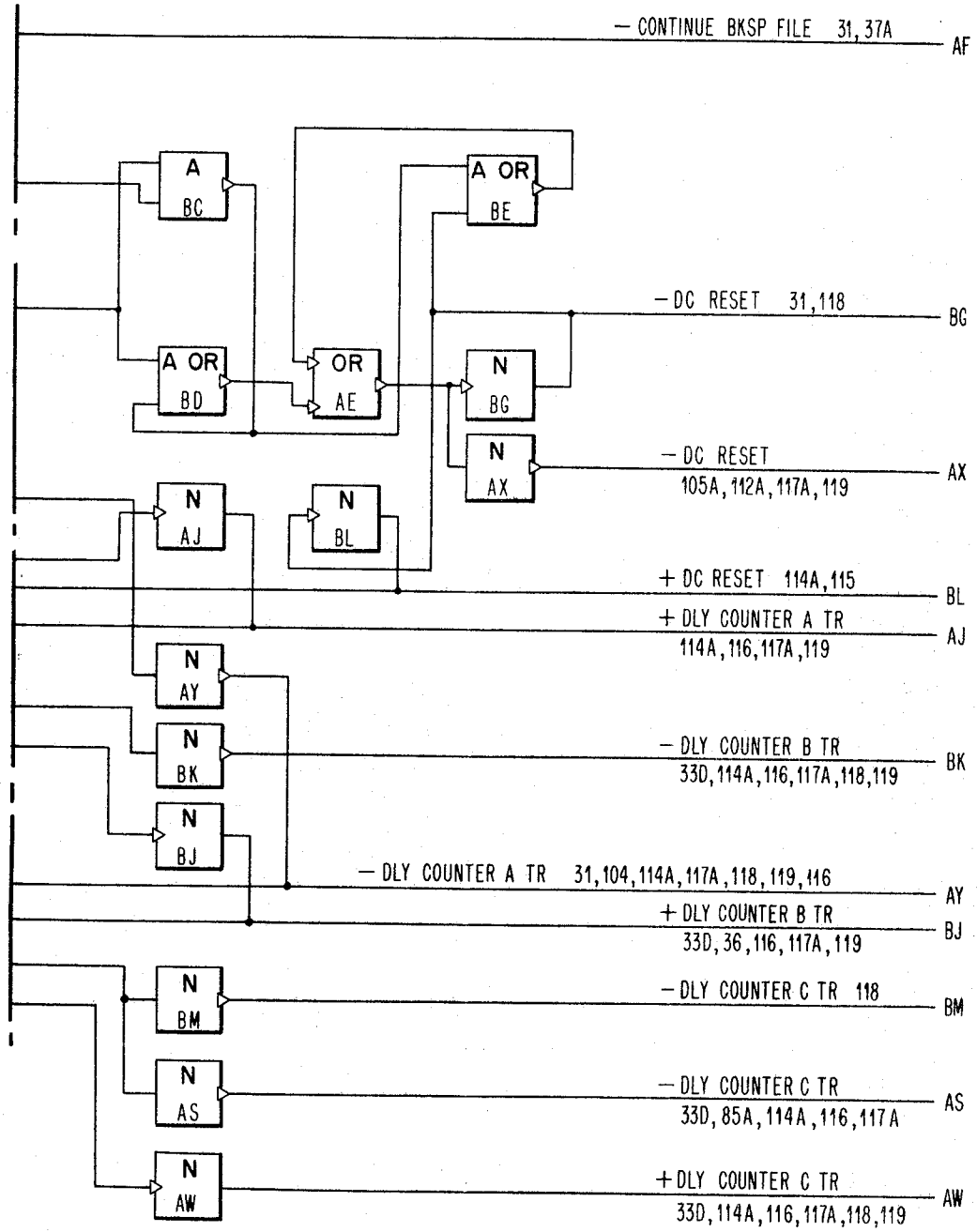

— CONTINUE BKSP FILE 31,37A
AF
A BC
A OR BE
— DC RESET 31,118
BG
A OR BD
OR AE
N BG
N AX
— DC RESET 105A,112A,117A,119
AX
N AJ
N BL
+ DC RESET 114A,115
BL
+ DLY COUNTER A TR 114A,116,117A,119
AJ
N AY
N BK
N BJ
— DLY COUNTER B TR 33D,114A,116,117A,118,119
BK
— DLY COUNTER A TR 31,104,114A,117A,118,119,116
AY
+ DLY COUNTER B TR 33D,36,116,117A,119
BJ
N BM
— DLY COUNTER C TR 118
BM
N AS
— DLY COUNTER C TR 33D,85A,114A,116,117A
AS
N AW
+ DLY COUNTER C TR 33D,114A,116,117A,118,119
AW

FIG. 114A

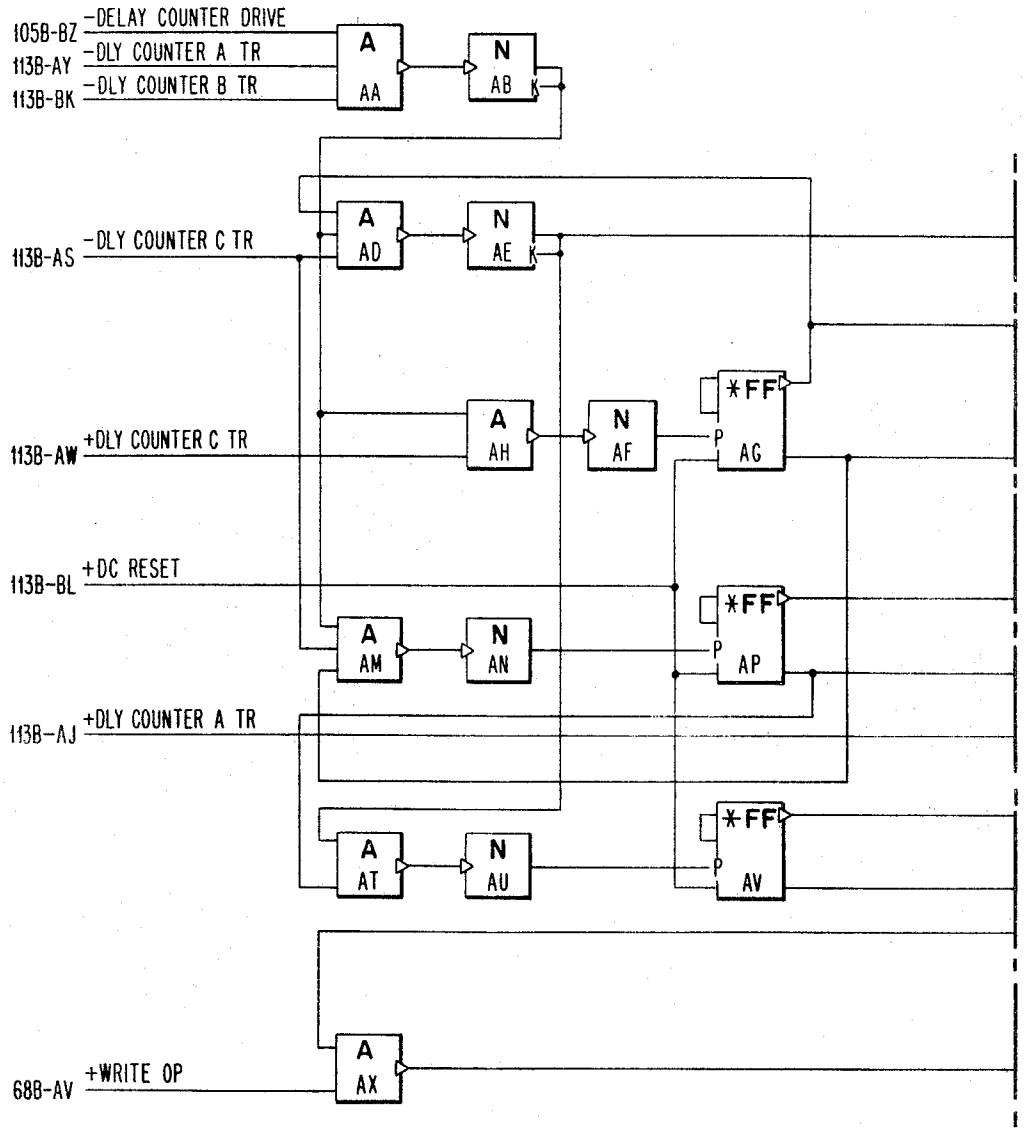

105B-BZ
-DELAY COUNTER DRIVE
113B-AY
-DLY COUNTER A TR
113B-BK
-DLY COUNTER B TR
A AA
N AB K
113B-AS
-DLY COUNTER C TR
A AD
N AE K
113B-AW
+DLY COUNTER C TR
A AH
N AF
*FF P AG
113B-BL
+DC RESET
A AM
N AN
*FF P AP
113B-AJ
+DLY COUNTER A TR
A AT
N AU
*FF P AV
68B-AV
+WRITE OP
A AX

FIG.114B

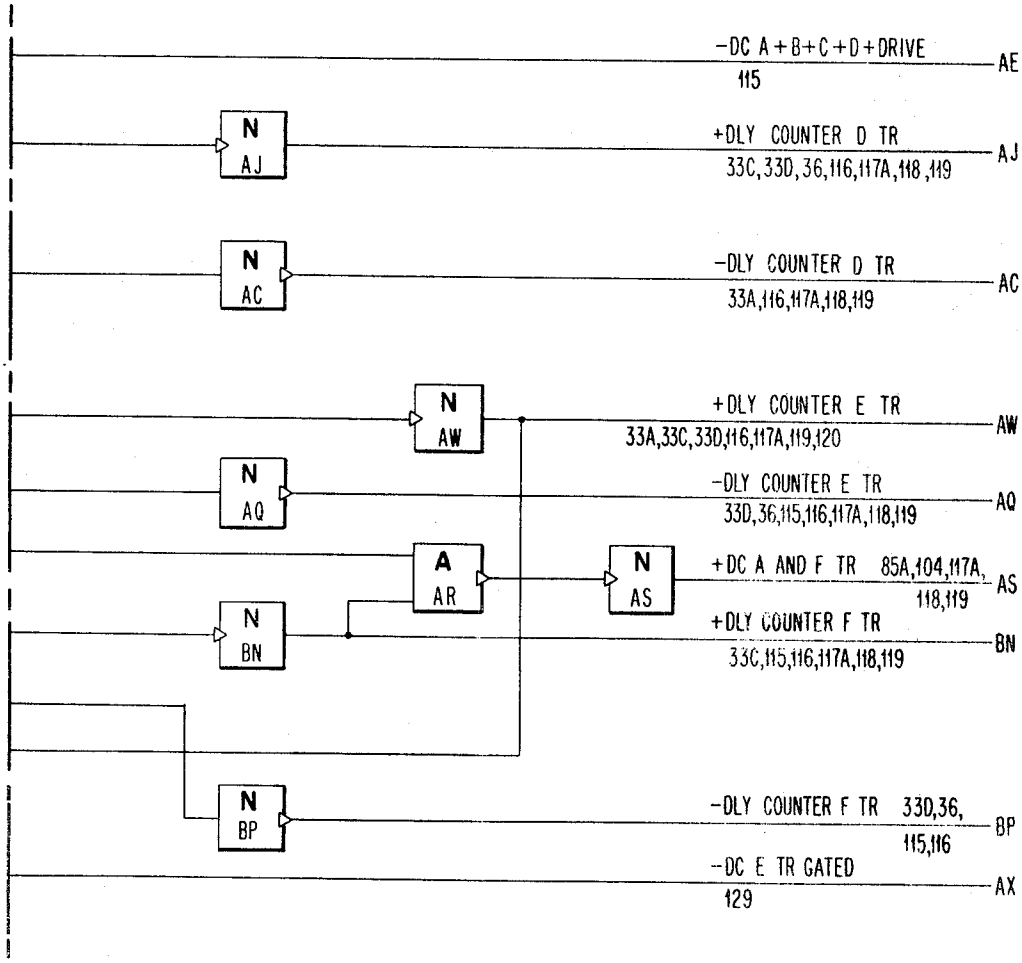

-DC A+B+C+D+DRIVE
115
AE
N
AJ
+DLY COUNTER D TR
33C,33D,36,116,117A,118,119
AJ
N
AC
-DLY COUNTER D TR
33A,116,117A,118,119
AC
N
AW
+DLY COUNTER E TR
33A,33C,33D,116,117A,119,120
AW
N
AQ
-DLY COUNTER E TR
33D,36,115,116,117A,118,119
AQ
A
AR
N
AS
+DC A AND F TR 85A,104,117A,
118,119
AS
N
BN
+DLY COUNTER F TR
33C,115,116,117A,118,119
BN
N
BP
-DLY COUNTER F TR 33D,36,
115,116
BP
-DC E TR GATED
129
AX

FIG. 115

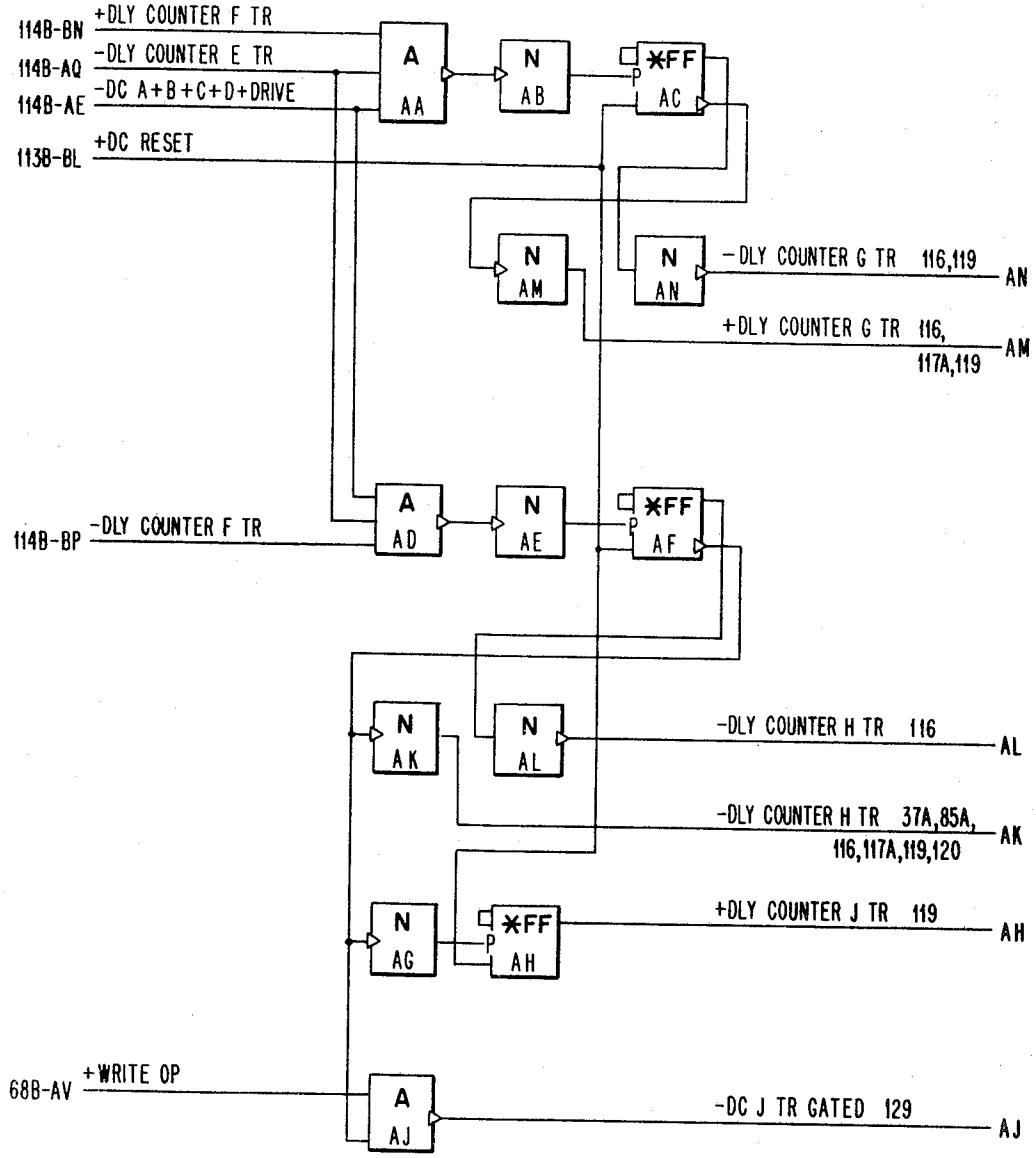

114B-BN +DLY COUNTER F TR
114B-AQ -DLY COUNTER E TR
114B-AE -DC A+B+C+D+DRIVE
113B-BL +DC RESET
A AA
N AB
*FF AC
P
N AM
N AN
-DLY COUNTER G TR 116,119 AN
+DLY COUNTER G TR 116, 117A,119 AM
114B-BP -DLY COUNTER F TR
A AD
N AE
*FF AF
N AK
N AL
-DLY COUNTER H TR 116 AL
-DLY COUNTER H TR 37A,85A, 116,117A,119,120 AK
N AG
*FF AH
+DLY COUNTER J TR 119 AH
68B-AV +WRITE OP
A AJ
-DC J TR GATED 129 AJ

FIG.116

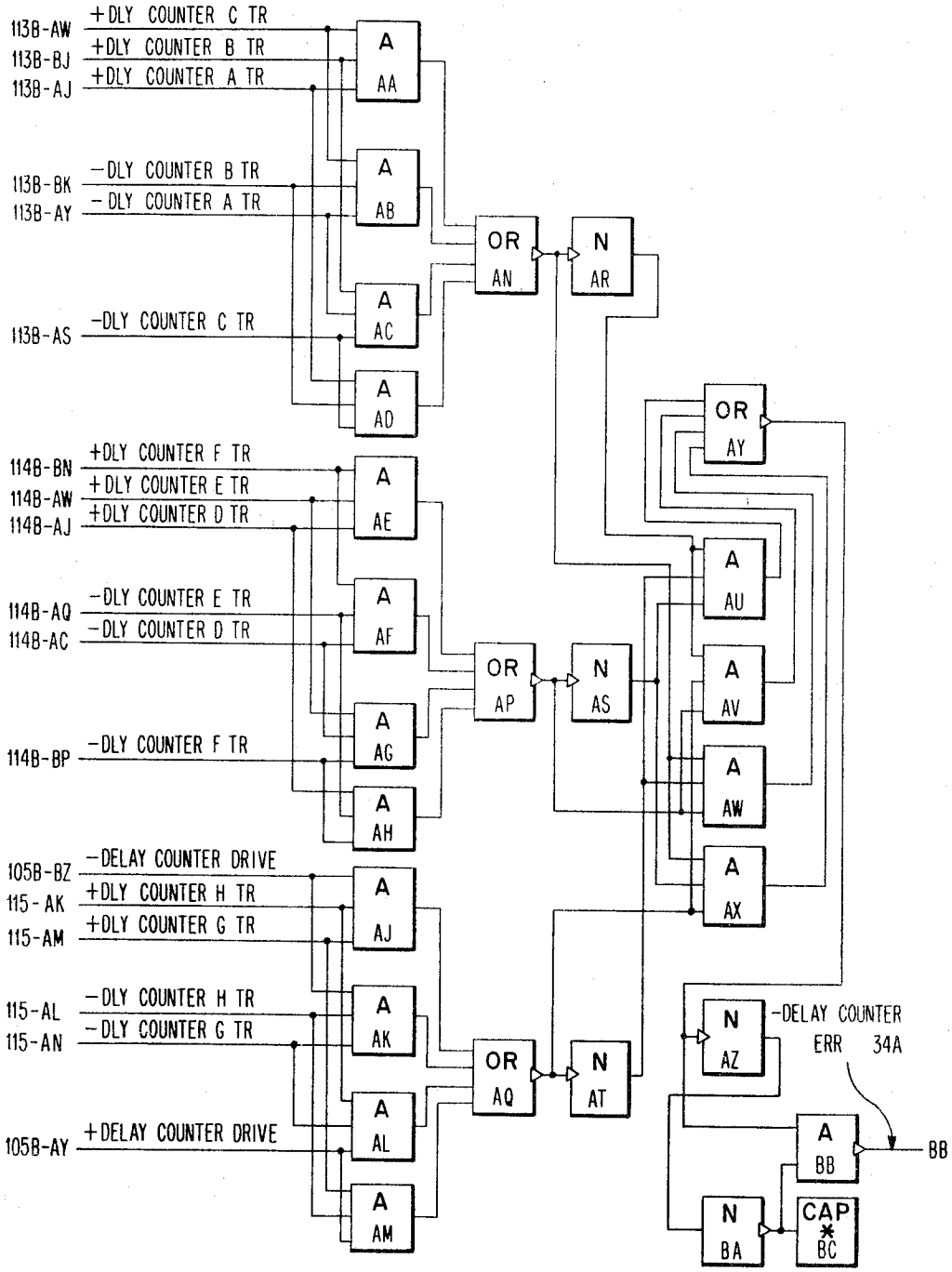

113B-AW
+DLY COUNTER C TR
113B-BJ
+DLY COUNTER B TR
113B-AJ
+DLY COUNTER A TR
113B-BK
-DLY COUNTER B TR
113B-AY
-DLY COUNTER A TR
113B-AS
-DLY COUNTER C TR
A AA
A AB
A AC
A AD
OR AN
N AR
114B-BN
+DLY COUNTER F TR
114B-AW
+DLY COUNTER E TR
114B-AJ
+DLY COUNTER D TR
114B-AQ
-DLY COUNTER E TR
114B-AC
-DLY COUNTER D TR
114B-BP
-DLY COUNTER F TR
A AE
A AF
A AG
A AH
OR AP
N AS
OR AY
A AU
A AV
A AW
A AX
105B-BZ
-DELAY COUNTER DRIVE
115-AK
+DLY COUNTER H TR
115-AM
+DLY COUNTER G TR
115-AL
-DLY COUNTER H TR
115-AN
-DLY COUNTER G TR
105B-AY
+DELAY COUNTER DRIVE
A AJ
A AK
A AL
A AM
OR AQ
N AT
N AZ
-DELAY COUNTER ERR 34A
A BB
BB
N BA
CAP * BC

FIG.117A

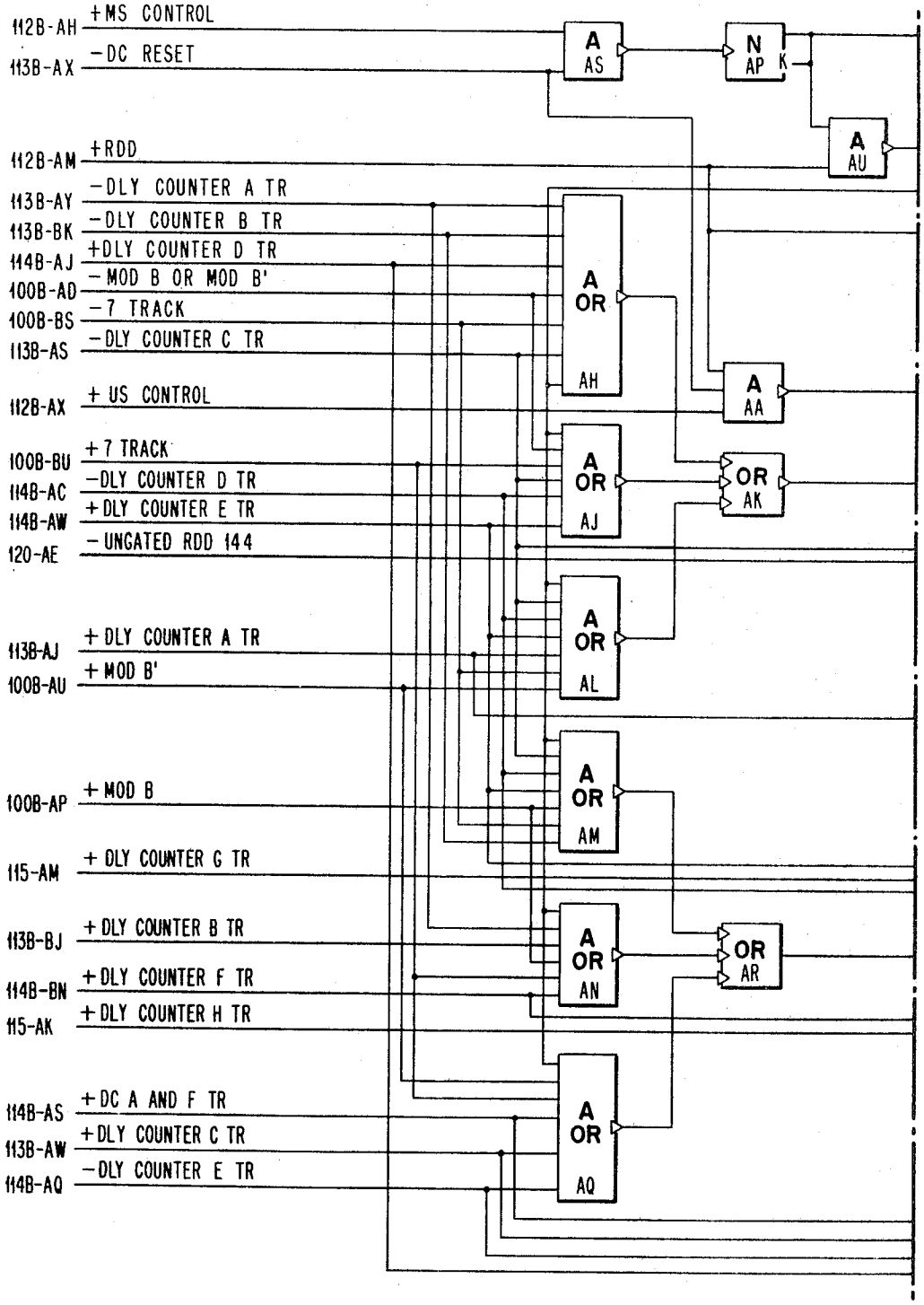

112B-AH +MS CONTROL
113B-AX −DC RESET
112B-AM +RDD
113B-AY −DLY COUNTER A TR
113B-BK −DLY COUNTER B TR
114B-AJ +DLY COUNTER D TR
100B-AD −MOD B OR MOD B'
100B-BS −7 TRACK
113B-AS −DLY COUNTER C TR
112B-AX + US CONTROL
100B-BU +7 TRACK
114B-AC −DLY COUNTER D TR
114B-AW +DLY COUNTER E TR
120-AE −UNGATED RDD 144
113B-AJ +DLY COUNTER A TR
100B-AU +MOD B'
100B-AP +MOD B
115-AM +DLY COUNTER G TR
113B-BJ +DLY COUNTER B TR
114B-BN +DLY COUNTER F TR
115-AK +DLY COUNTER H TR
114B-AS +DC A AND F TR
113B-AW +DLY COUNTER C TR
114B-AQ −DLY COUNTER E TR
A AS
N AP K
A AU
A OR AH
A AA
A OR AJ
OR AK
A OR AL
A OR AM
A OR AN
OR AR
A OR AQ

FIG.117B

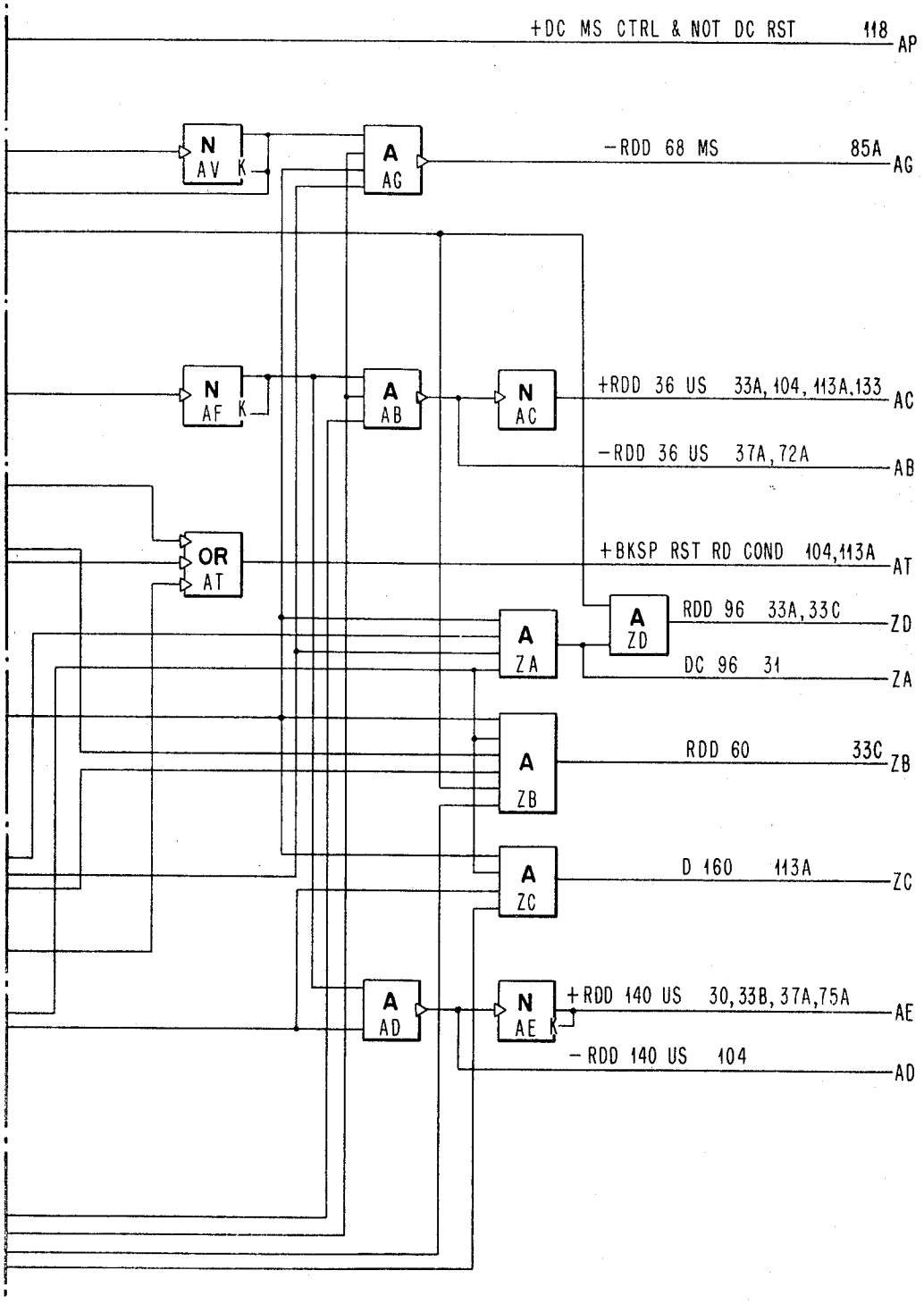

+DC MS CTRL & NOT DC RST 118 AP
N AV K
A AG
-RDD 68 MS 85A AG
N AF K
A AB
N AC
+RDD 36 US 33A,104,113A,133 AC
-RDD 36 US 37A,72A AB
OR AT
+BKSP RST RD COND 104,113A AT
A ZA
A ZD
RDD 96 33A,33C ZD
DC 96 31 ZA
A ZB
RDD 60 33C ZB
A ZC
D 160 113A ZC
A AD
N AE K
+RDD 140 US 30,33B,37A,75A AE
-RDD 140 US 104 AD

FIG.118

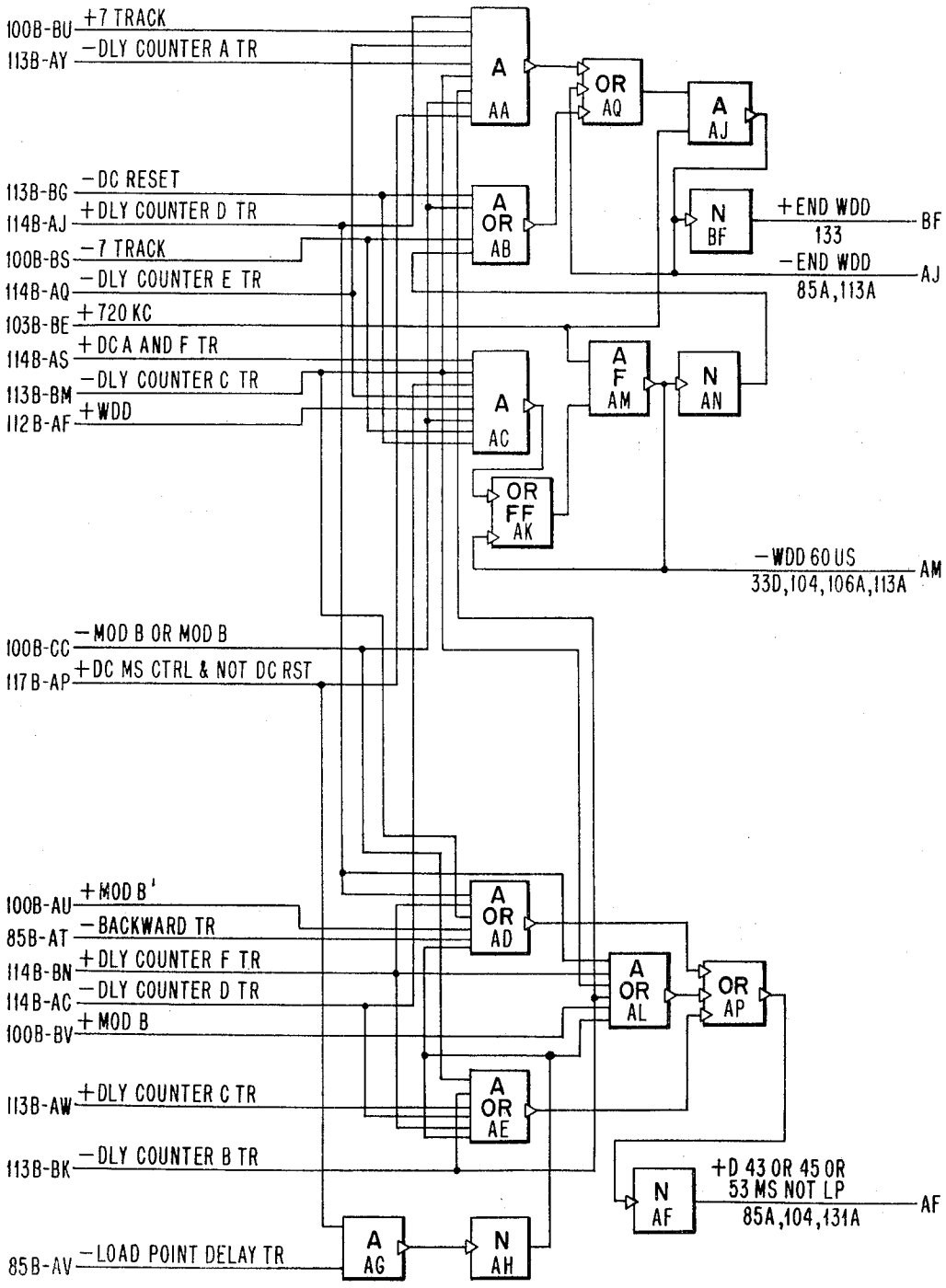

100B-BU +7 TRACK
113B-AY −DLY COUNTER A TR
A AA
OR AQ
A AJ
113B-BG −DC RESET
114B-AJ +DLY COUNTER D TR
100B-BS −7 TRACK
114B-AQ −DLY COUNTER E TR
A OR AB
N BF
+END WDD 133 BF
−END WDD 85A,113A AJ
103B-BE +720 KC
114B-AS +DC A AND F TR
113B-BM −DLY COUNTER C TR
112B-AF +WDD
A AC
A F AM
N AN
OR FF AK
−WDD 60 US 33D,104,106A,113A AM
100B-CC −MOD B OR MOD B
117B-AP +DC MS CTRL & NOT DC RST
100B-AU +MOD B'
85B-AT −BACKWARD TR
114B-BN +DLY COUNTER F TR
114B-AC −DLY COUNTER D TR
100B-BV +MOD B
A OR AD
A OR AL
OR AP
113B-AW +DLY COUNTER C TR
113B-BK −DLY COUNTER B TR
A OR AE
N AF
+D 43 OR 45 OR 53 MS NOT LP 85A,104,131A AF
85B-AV −LOAD POINT DELAY TR
A AG
N AH

FIG. 119

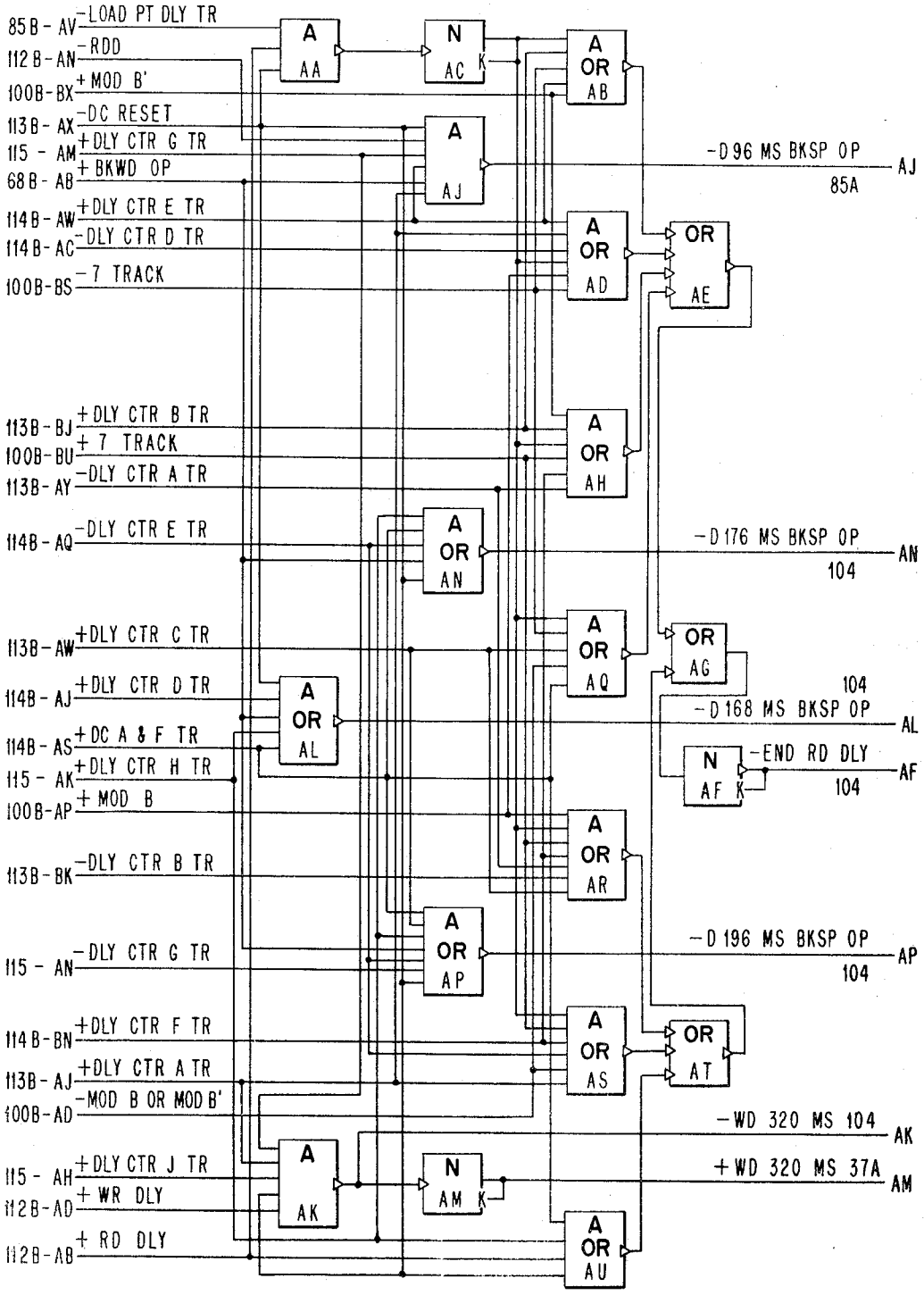

85B-AV -LOAD PT DLY TR
112B-AN -RDD
100B-BX +MOD B'
113B-AX -DC RESET
115 - AM +DLY CTR G TR
68B-AB +BKWD OP
114B-AW +DLY CTR E TR
114B-AC -DLY CTR D TR
100B-BS -7 TRACK
113B-BJ +DLY CTR B TR
100B-BU +7 TRACK
113B-AY -DLY CTR A TR
114B-AQ -DLY CTR E TR
113B-AW +DLY CTR C TR
114B-AJ +DLY CTR D TR
114B-AS +DC A & F TR
115 - AK +DLY CTR H TR
100B-AP +MOD B
113B-BK -DLY CTR B TR
115 - AN -DLY CTR G TR
114B-BN +DLY CTR F TR
113B-AJ +DLY CTR A TR
100B-AD -MOD B OR MOD B'
115 - AH +DLY CTR J TR
112B-AD +WR DLY
112B-AB +RD DLY
A AA
N AC K
A OR AB
A AJ
-D 96 MS BKSP OP
85A
AJ
A OR AD
OR AE
A OR AH
A OR AN
-D 176 MS BKSP OP
104
AN
A OR AQ
OR AG
104
-D 168 MS BKSP OP
AL
A OR AL
N AF K
-END RD DLY
104
AF
A OR AR
A OR AP
-D 196 MS BKSP OP
104
AP
A OR AS
OR AT
-WD 320 MS 104
AK
+WD 320 MS 37A
AM
A AK
N AM K
A OR AU

FIG.120

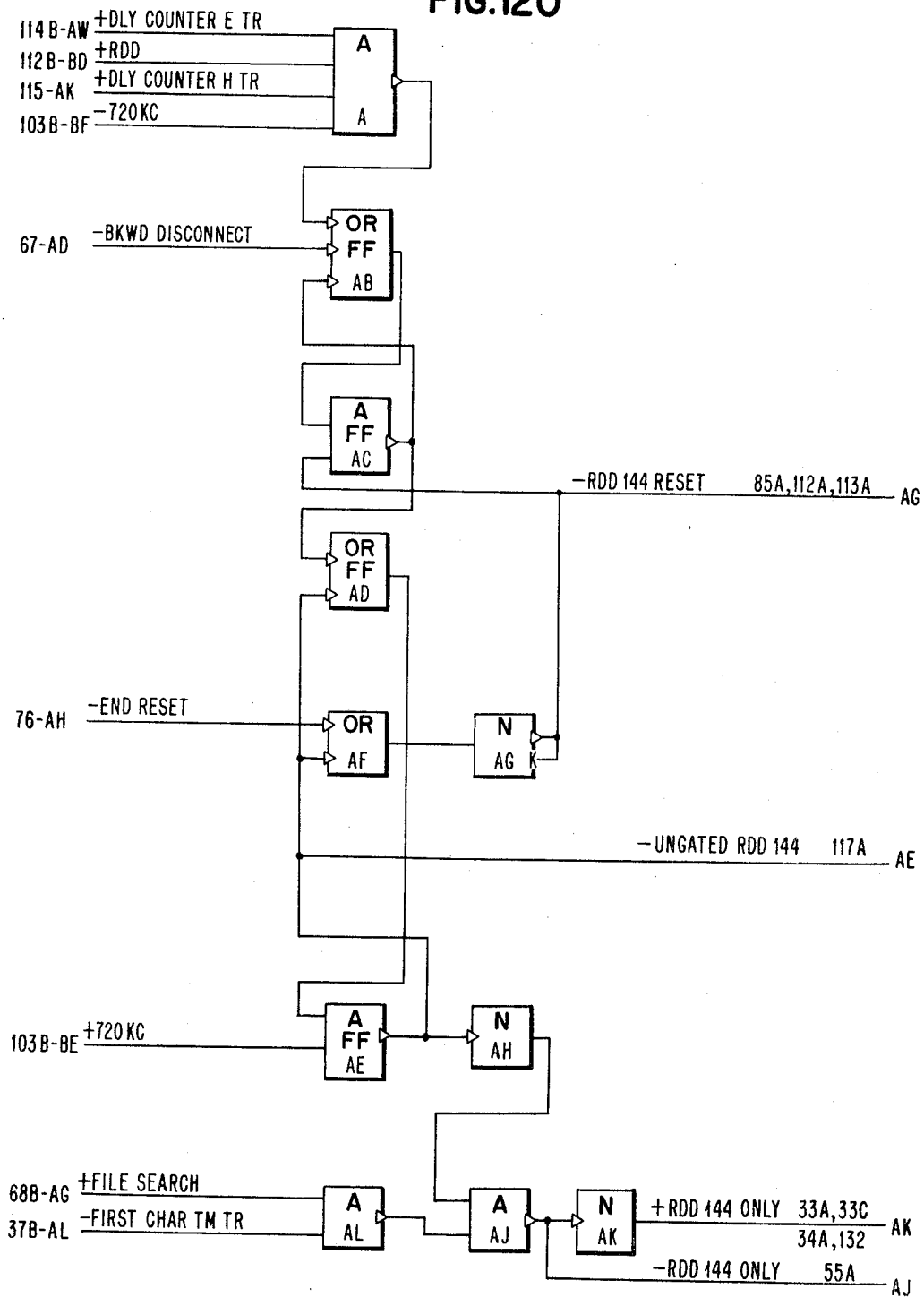

114B-AW +DLY COUNTER E TR
112B-BD +RDD
115-AK +DLY COUNTER H TR
103B-BF -720KC
A
A
67-AD -BKWD DISCONNECT
OR FF AB
A FF AC
-RDD 144 RESET 85A,112A,113A AG
OR FF AD
76-AH -END RESET
OR AF
N AG K
-UNGATED RDD 144 117A AE
103B-BE +720KC
A FF AE
N AH
68B-AG +FILE SEARCH
37B-AL -FIRST CHAR TM TR
A AL
A AJ
N AK
+RDD 144 ONLY 33A,33C 34A,132 AK
-RDD 144 ONLY 55A AJ

FIG.121

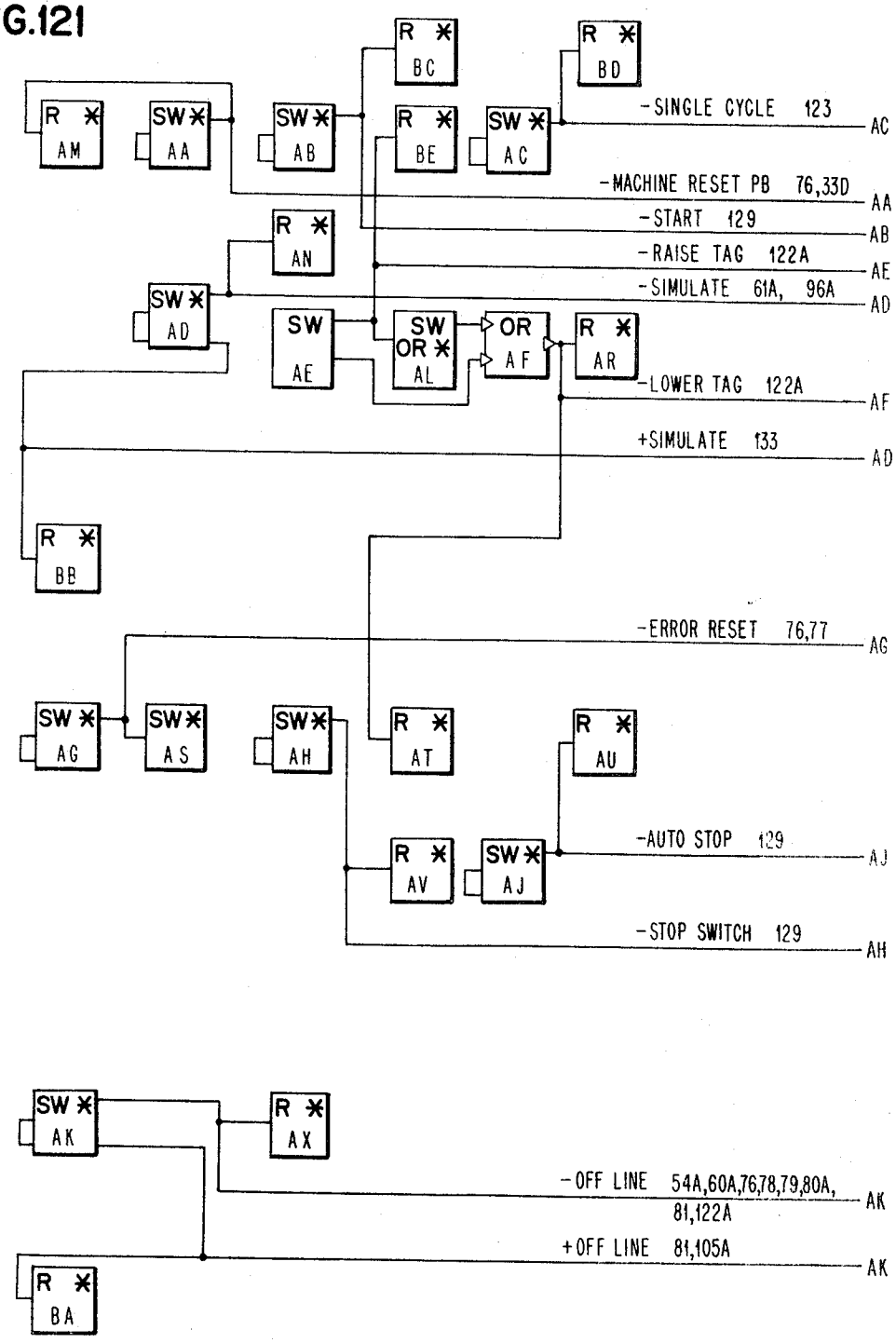

R * AM
SW * AA
SW * AB
R * BC
R * BE
SW * AC
R * BD
-SINGLE CYCLE 123
AC
-MACHINE RESET PB 76,33D
AA
-START 129
AB
-RAISE TAG 122A
AE
-SIMULATE 61A, 96A
AD
R * AN
SW * AD
SW AE
SW OR * AL
OR AF
R * AR
-LOWER TAG 122A
AF
+SIMULATE 133
AD
R * BB
-ERROR RESET 76,77
AG
SW * AG
SW * AS
SW * AH
R * AT
R * AU
-AUTO STOP 129
AJ
R * AV
SW * AJ
-STOP SWITCH 129
AH
SW * AK
R * AX
-OFF LINE 54A,60A,76,78,79,80A, 81,122A
AK
+OFF LINE 81,105A
AK
R * BA

FIG.122A

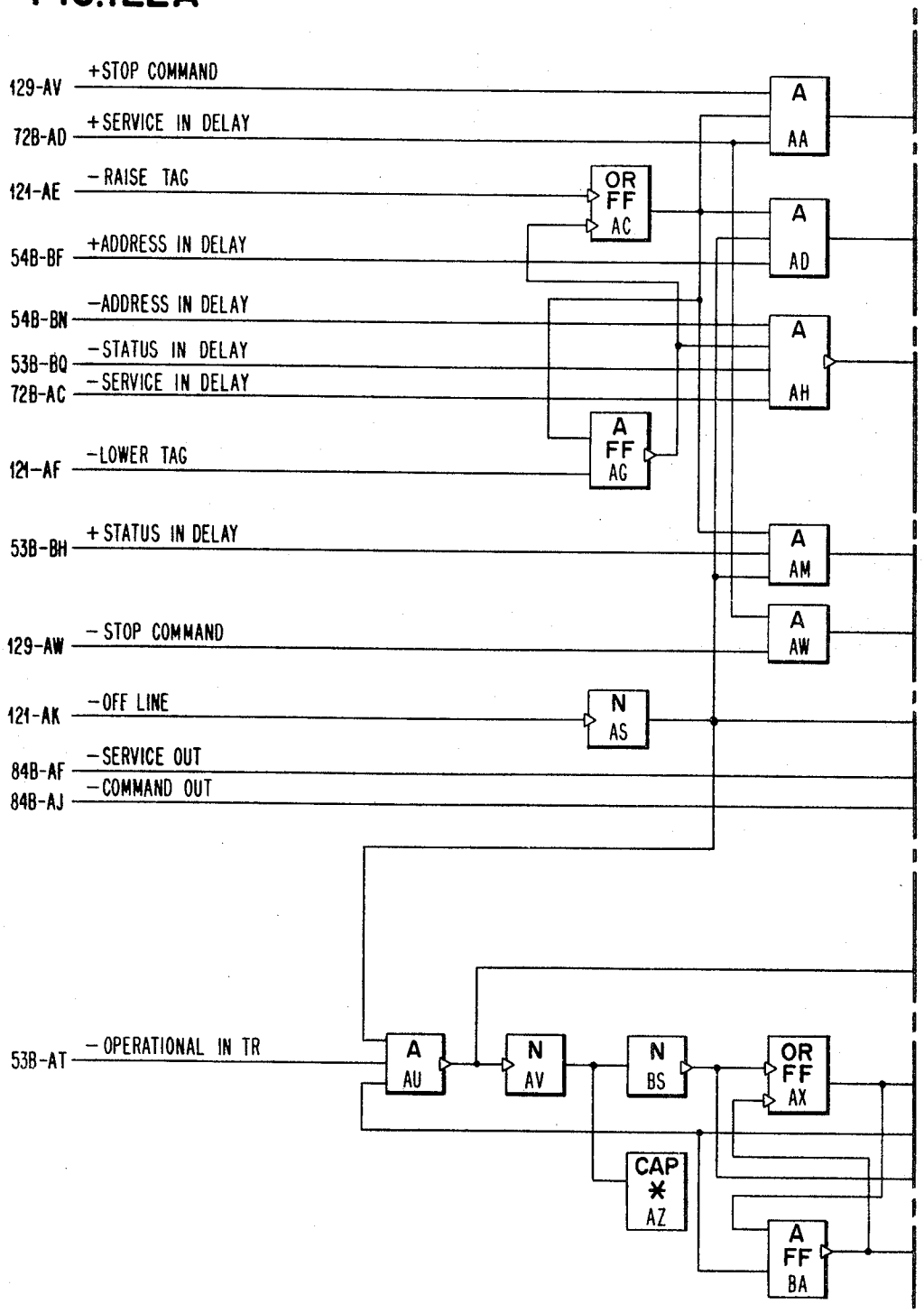

129-AV
+STOP COMMAND
72B-AD
+SERVICE IN DELAY
121-AE
-RAISE TAG
54B-BF
+ADDRESS IN DELAY
54B-BN
-ADDRESS IN DELAY
53B-BQ
-STATUS IN DELAY
72B-AC
-SERVICE IN DELAY
121-AF
-LOWER TAG
53B-BH
+STATUS IN DELAY
129-AW
-STOP COMMAND
121-AK
-OFF LINE
84B-AF
-SERVICE OUT
84B-AJ
-COMMAND OUT
53B-AT
-OPERATIONAL IN TR
A AA
A AD
A AH
A AM
A AW
OR FF AC
A FF AG
N AS
A AU
N AV
N BS
OR FF AX
CAP * AZ
A FF BA

FIG.122B

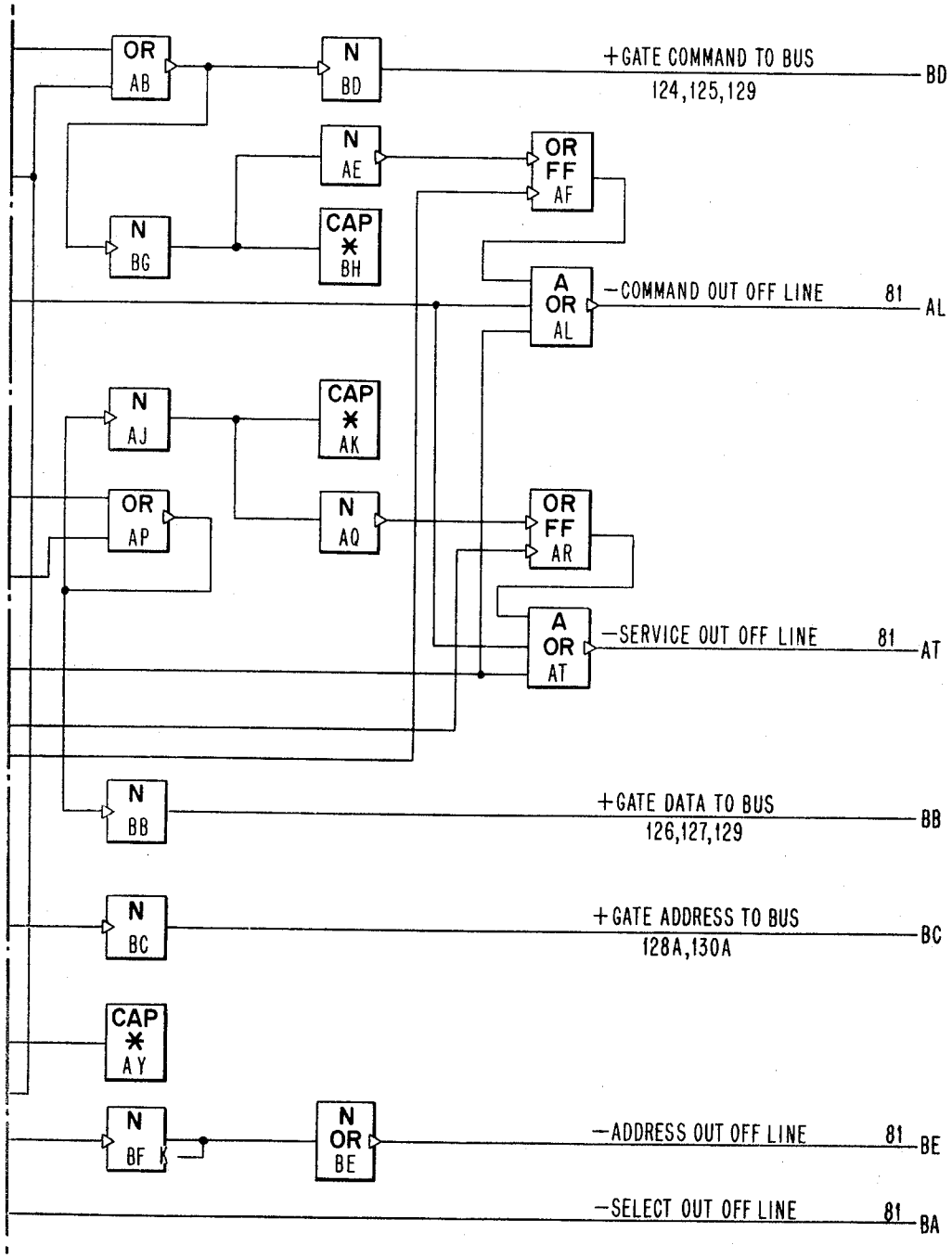

OR AB
N BD
+GATE COMMAND TO BUS 124,125,129
BD
N AE
OR FF AF
N BG
CAP * BH
A OR AL
-COMMAND OUT OFF LINE 81
AL
N AJ
CAP * AK
OR AP
N AQ
OR FF AR
A OR AT
-SERVICE OUT OFF LINE 81
AT
N BB
+GATE DATA TO BUS 126,127,129
BB
N BC
+GATE ADDRESS TO BUS 128A,130A
BC
CAP * AY
N BF K
N OR BE
-ADDRESS OUT OFF LINE 81
BE
-SELECT OUT OFF LINE 81
BA

FIG. 123

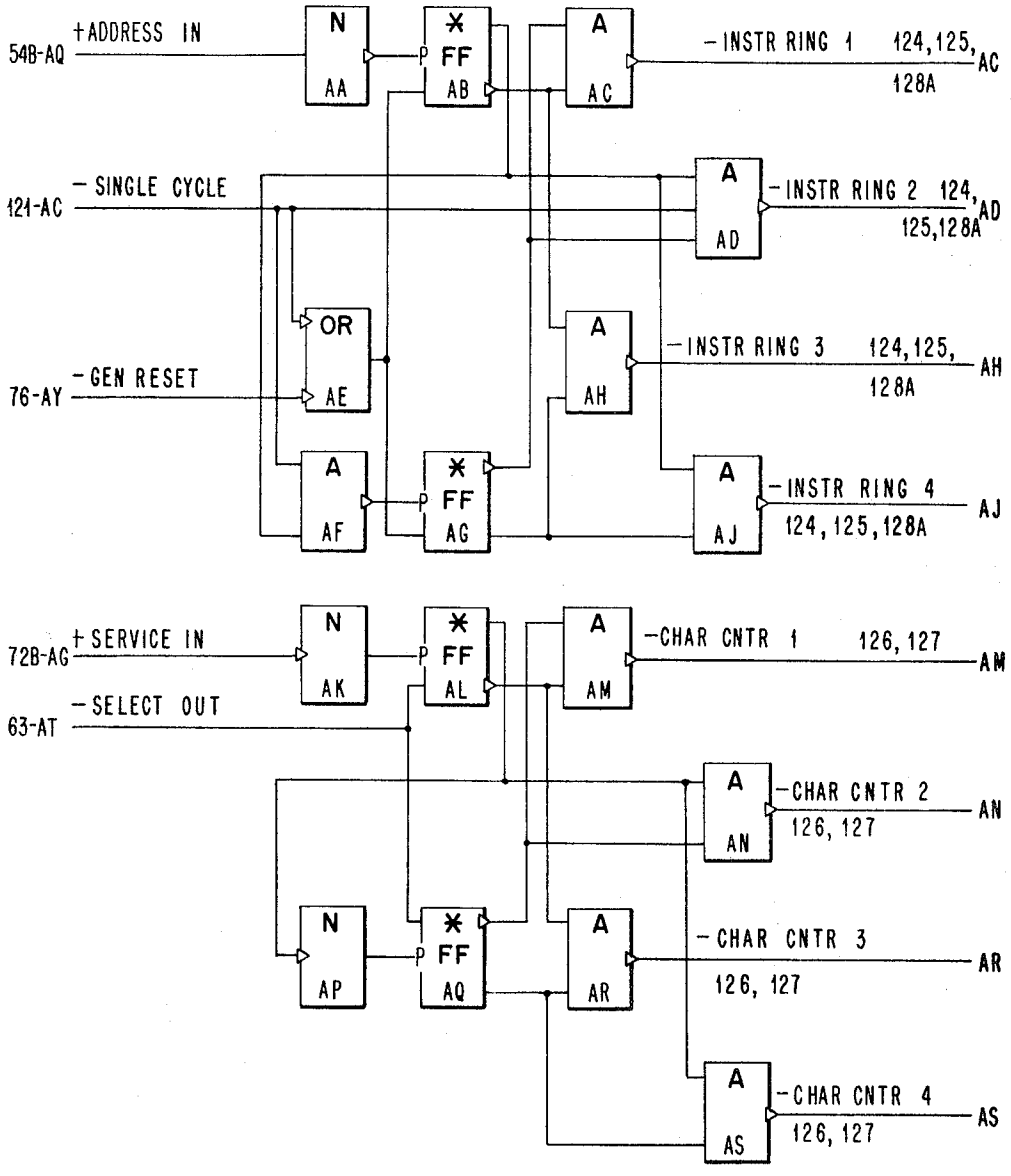

54B-AQ
+ADDRESS IN
N
AA
*
FF
AB
A
AC
-INSTR RING 1 124,125, 128A
AC
121-AC
-SINGLE CYCLE
A
AD
-INSTR RING 2 124, 125,128A
AD
OR
AE
76-AY
-GEN RESET
A
AH
-INSTR RING 3 124,125, 128A
AH
A
AF
*
FF
AG
A
AJ
-INSTR RING 4 124,125,128A
AJ
72B-AG
+SERVICE IN
N
AK
*
FF
AL
A
AM
-CHAR CNTR 1 126, 127
AM
63-AT
-SELECT OUT
A
AN
-CHAR CNTR 2 126, 127
AN
N
AP
*
FF
AQ
A
AR
-CHAR CNTR 3 126, 127
AR
A
AS
-CHAR CNTR 4 126, 127
AS

FIG. 124

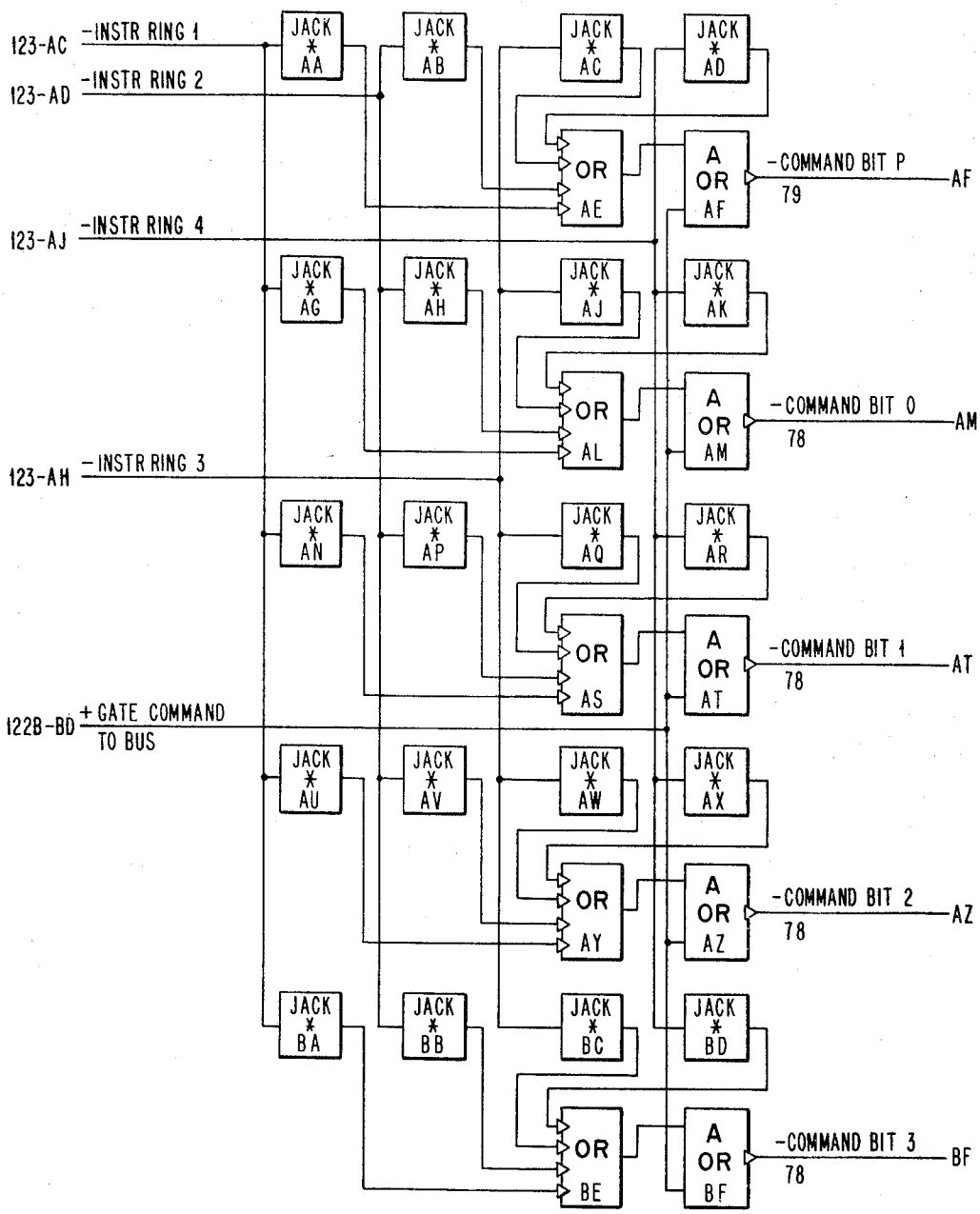

123-AC -INSTR RING 1
123-AD -INSTR RING 2
123-AJ -INSTR RING 4
123-AH -INSTR RING 3
122B-BD +GATE COMMAND TO BUS
JACK * AA
JACK * AB
JACK * AC
JACK * AD
OR AE
A OR AF
-COMMAND BIT P
79
AF
JACK * AG
JACK * AH
JACK * AJ
JACK * AK
OR AL
A OR AM
-COMMAND BIT 0
78
AM
JACK * AN
JACK * AP
JACK * AQ
JACK * AR
OR AS
A OR AT
-COMMAND BIT 1
78
AT
JACK * AU
JACK * AV
JACK * AW
JACK * AX
OR AY
A OR AZ
-COMMAND BIT 2
78
AZ
JACK * BA
JACK * BB
JACK * BC
JACK * BD
OR BE
A OR BF
-COMMAND BIT 3
78
BF

FIG. 125

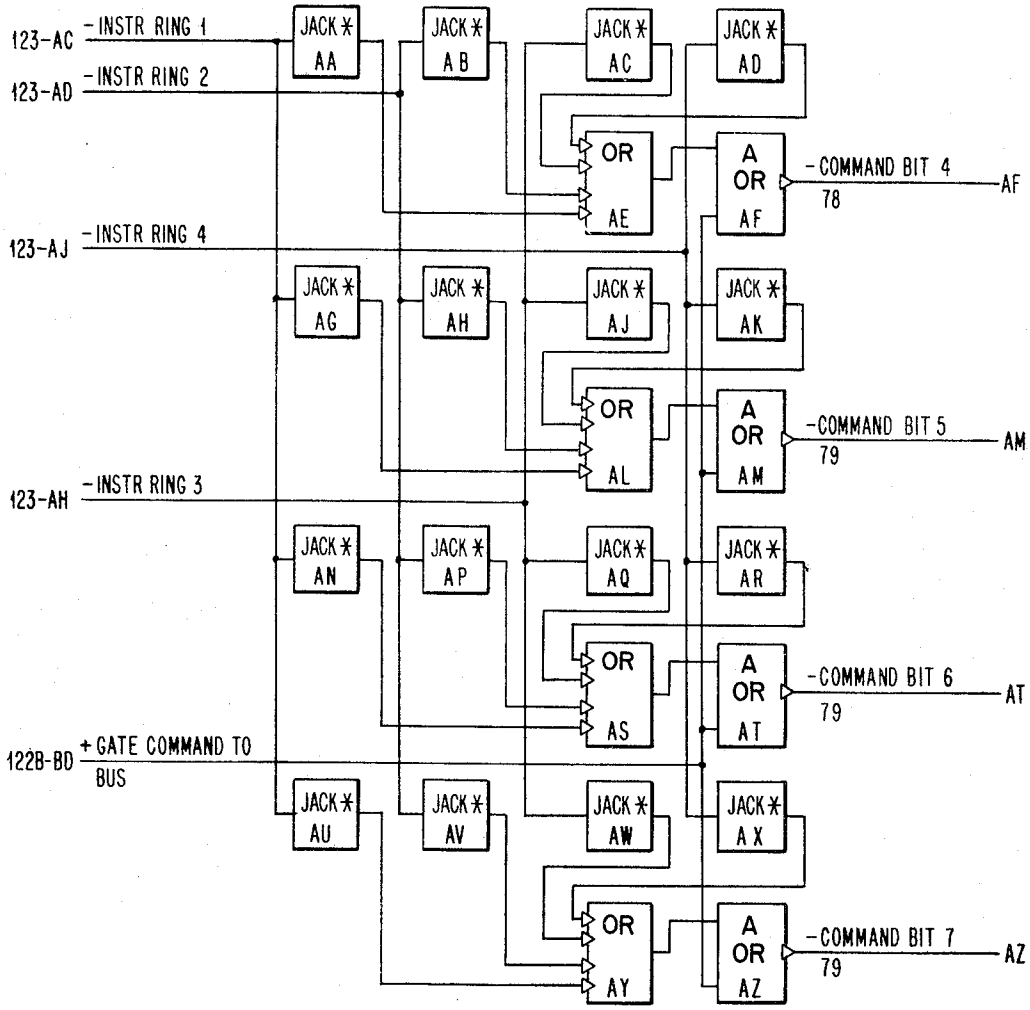

123-AC -INSTR RING 1
123-AD -INSTR RING 2
123-AJ -INSTR RING 4
123-AH -INSTR RING 3
122B-BD +GATE COMMAND TO BUS
JACK * AA
JACK * AB
JACK * AC
JACK * AD
OR AE
A OR AF
-COMMAND BIT 4
78
AF
JACK * AG
JACK * AH
JACK * AJ
JACK * AK
OR AL
A OR AM
-COMMAND BIT 5
79
AM
JACK * AN
JACK * AP
JACK * AQ
JACK * AR
OR AS
A OR AT
-COMMAND BIT 6
79
AT
JACK * AU
JACK * AV
JACK * AW
JACK * AX
OR AY
A OR AZ
-COMMAND BIT 7
79
AZ

FIG. 126

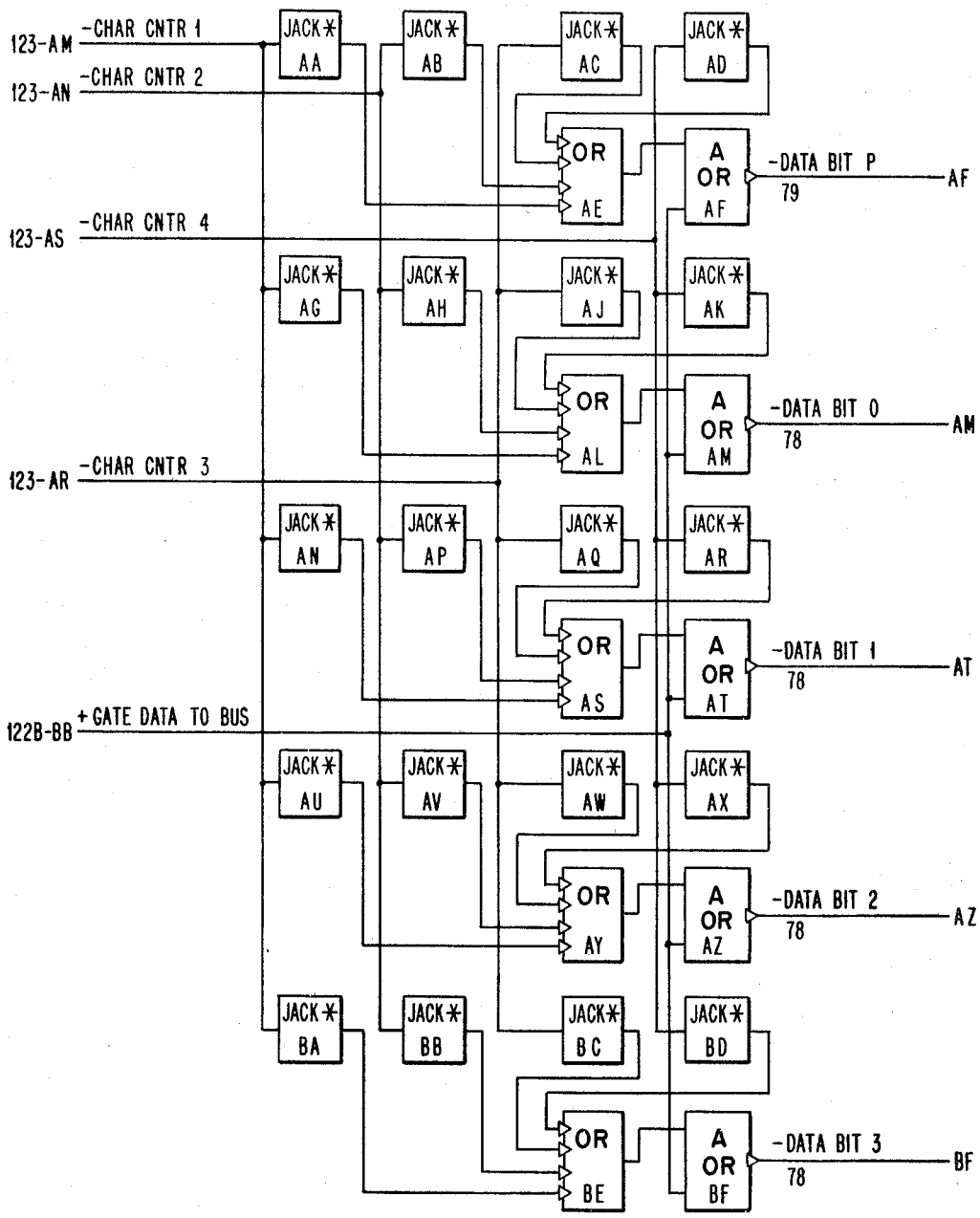

123-AM -CHAR CNTR 1
123-AN -CHAR CNTR 2
123-AS -CHAR CNTR 4
123-AR -CHAR CNTR 3
122B-BB +GATE DATA TO BUS
JACK AA
JACK AB
JACK AC
JACK AD
OR AE
A OR AF
-DATA BIT P
79
AF
JACK AG
JACK AH
JACK AJ
JACK AK
OR AL
A OR AM
-DATA BIT 0
78
AM
JACK AN
JACK AP
JACK AQ
JACK AR
OR AS
A OR AT
-DATA BIT 1
78
AT
JACK AU
JACK AV
JACK AW
JACK AX
OR AY
A OR AZ
-DATA BIT 2
78
AZ
JACK BA
JACK BB
JACK BC
JACK BD
OR BE
A OR BF
-DATA BIT 3
78
BF

FIG.127

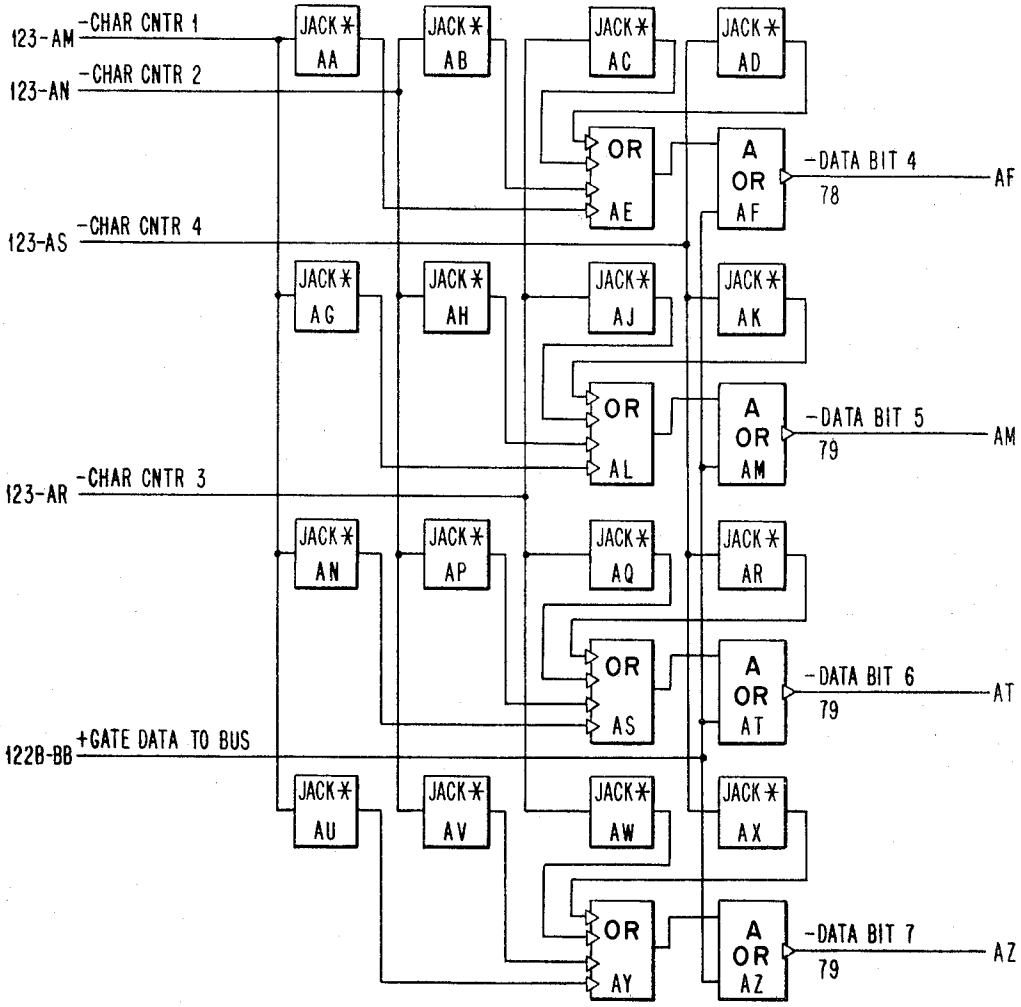

123-AM -CHAR CNTR 1
123-AN -CHAR CNTR 2
123-AS -CHAR CNTR 4
123-AR -CHAR CNTR 3
1228-BB +GATE DATA TO BUS
JACK AA
JACK AB
JACK AC
JACK AD
OR AE
A OR AF
-DATA BIT 4
78
AF
JACK AG
JACK AH
JACK AJ
JACK AK
OR AL
A OR AM
-DATA BIT 5
79
AM
JACK AN
JACK AP
JACK AQ
JACK AR
OR AS
A OR AT
-DATA BIT 6
79
AT
JACK AU
JACK AV
JACK AW
JACK AX
OR AY
A OR AZ
-DATA BIT 7
79
AZ

FIG.128A

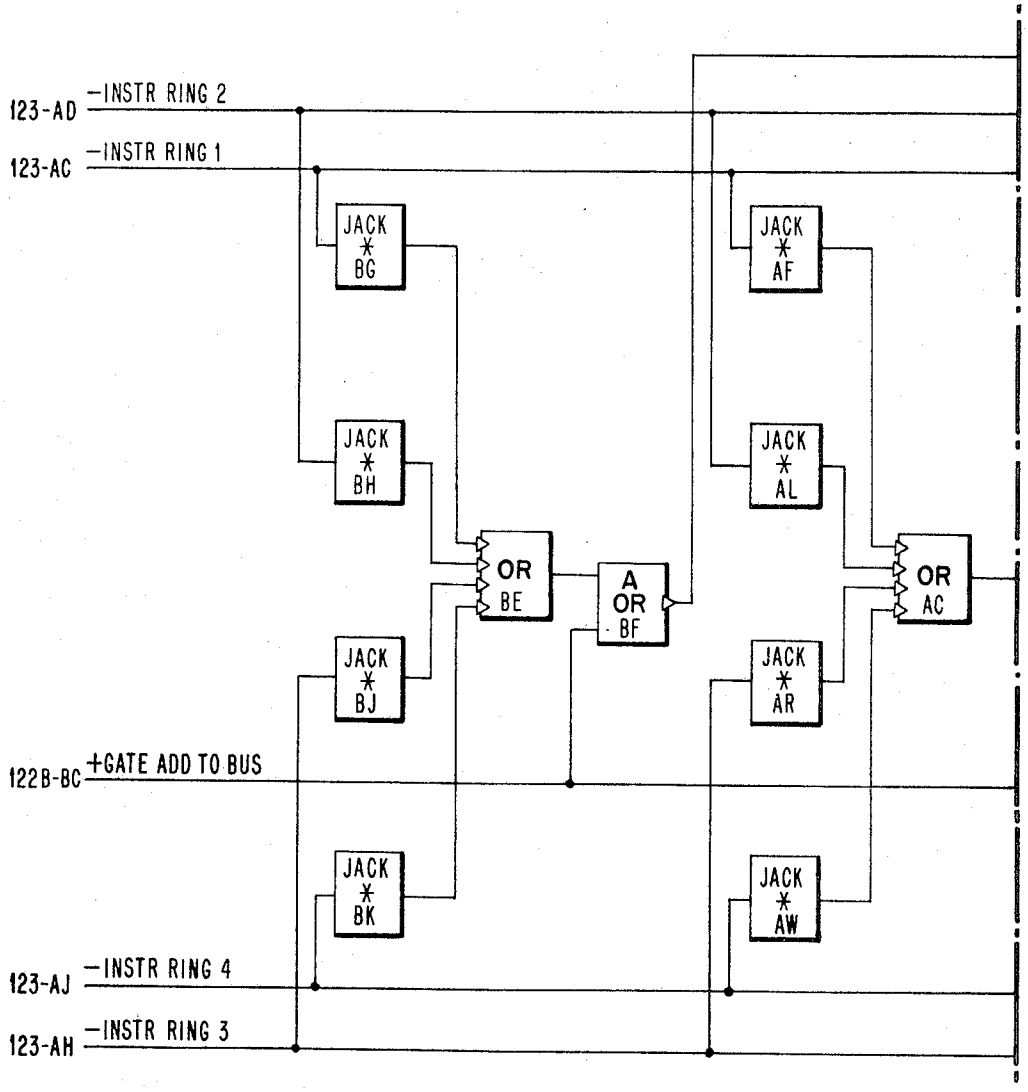

123-AD -INSTR RING 2
123-AC -INSTR RING 1
JACK * BG
JACK * BH
JACK * BJ
JACK * BK
OR BE
A OR BF
JACK * AF
JACK * AL
JACK * AR
JACK * AW
OR AC
122B-BC +GATE ADD TO BUS
123-AJ -INSTR RING 4
123-AH -INSTR RING 3

FIG. 128B

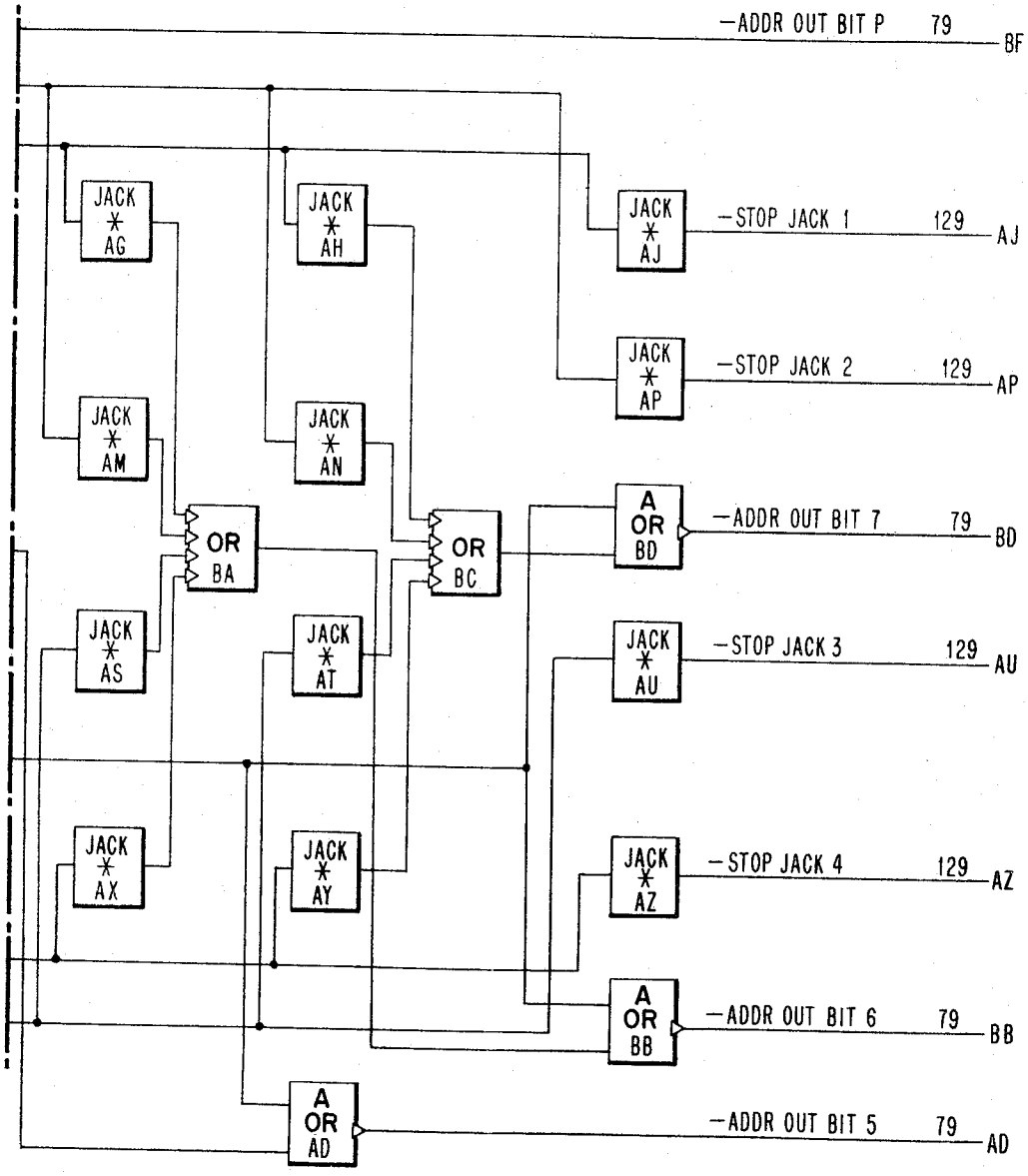

—ADDR OUT BIT P 79 BF
JACK * AG
JACK * AH
JACK * AJ
—STOP JACK 1 129 AJ
JACK * AP
—STOP JACK 2 129 AP
JACK * AM
JACK * AN
A OR BD
—ADDR OUT BIT 7 79 BD
OR BA
OR BC
JACK * AS
JACK * AT
JACK * AU
—STOP JACK 3 129 AU
JACK * AX
JACK * AY
JACK * AZ
—STOP JACK 4 129 AZ
A OR BB
—ADDR OUT BIT 6 79 BB
A OR AD
—ADDR OUT BIT 5 79 AD

FIG. 129

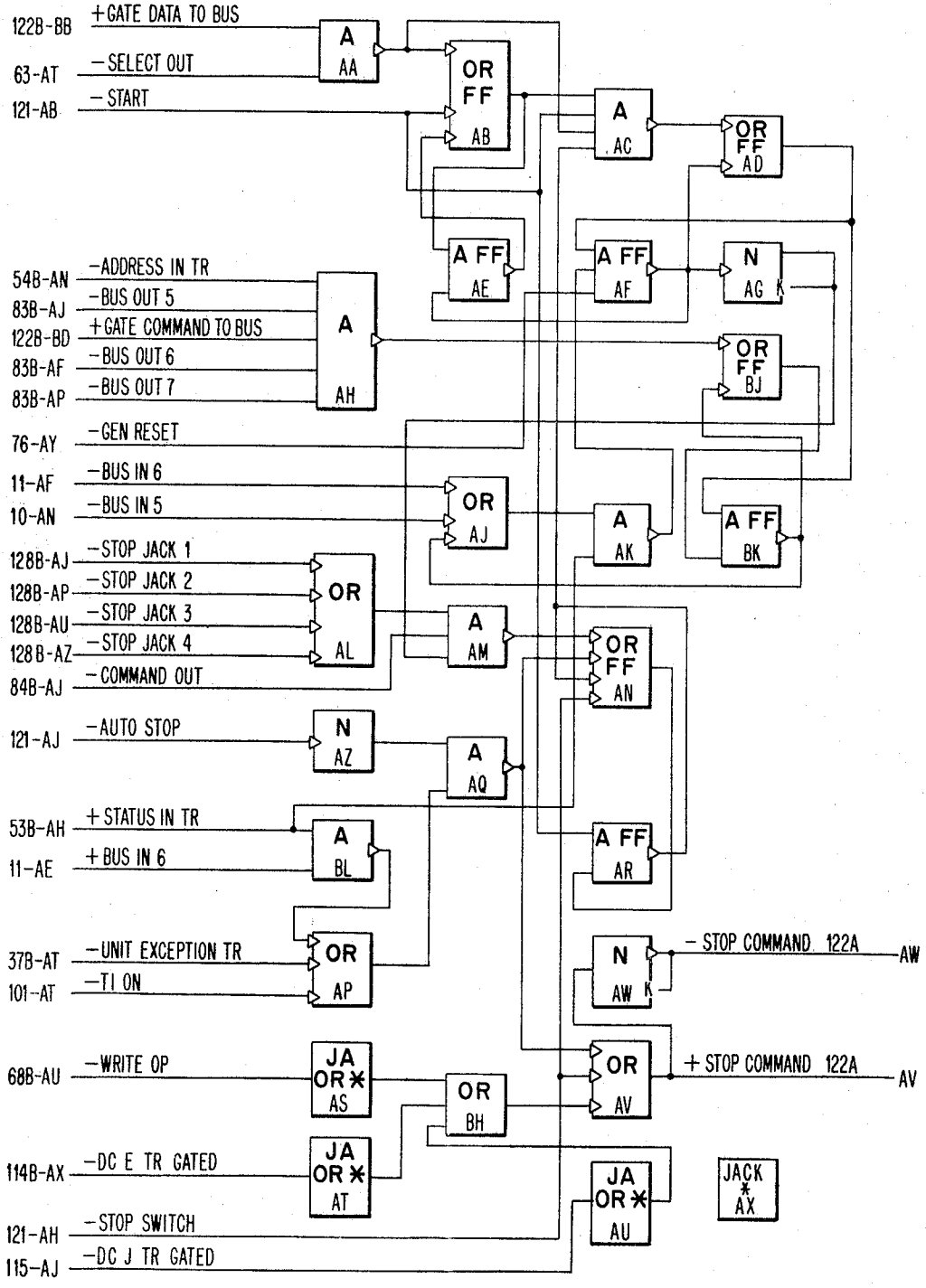

122B-BB
+GATE DATA TO BUS
63-AT
-SELECT OUT
121-AB
-START
A AA
OR FF AB
A AC
OR FF AD
A FF AE
A FF AF
N AG K
54B-AN
-ADDRESS IN TR
83B-AJ
-BUS OUT 5
122B-BD
+GATE COMMAND TO BUS
83B-AF
-BUS OUT 6
83B-AP
-BUS OUT 7
A AH
OR FF BJ
76-AY
-GEN RESET
11-AF
-BUS IN 6
10-AN
-BUS IN 5
OR AJ
A AK
A FF BK
128B-AJ
-STOP JACK 1
128B-AP
-STOP JACK 2
128B-AU
-STOP JACK 3
128B-AZ
-STOP JACK 4
84B-AJ
-COMMAND OUT
OR AL
A AM
OR FF AN
121-AJ
-AUTO STOP
N AZ
A AQ
53B-AH
+STATUS IN TR
11-AE
+BUS IN 6
A BL
A FF AR
37B-AT
-UNIT EXCEPTION TR
101-AT
-TI ON
OR AP
N AW K
-STOP COMMAND 122A
AW
68B-AU
-WRITE OP
JA OR* AS
OR BH
OR AV
+STOP COMMAND 122A
AV
114B-AX
-DC E TR GATED
JA OR* AT
121-AH
-STOP SWITCH
115-AJ
-DC J TR GATED
JA OR* AU
JACK * AX

FIG. 130A

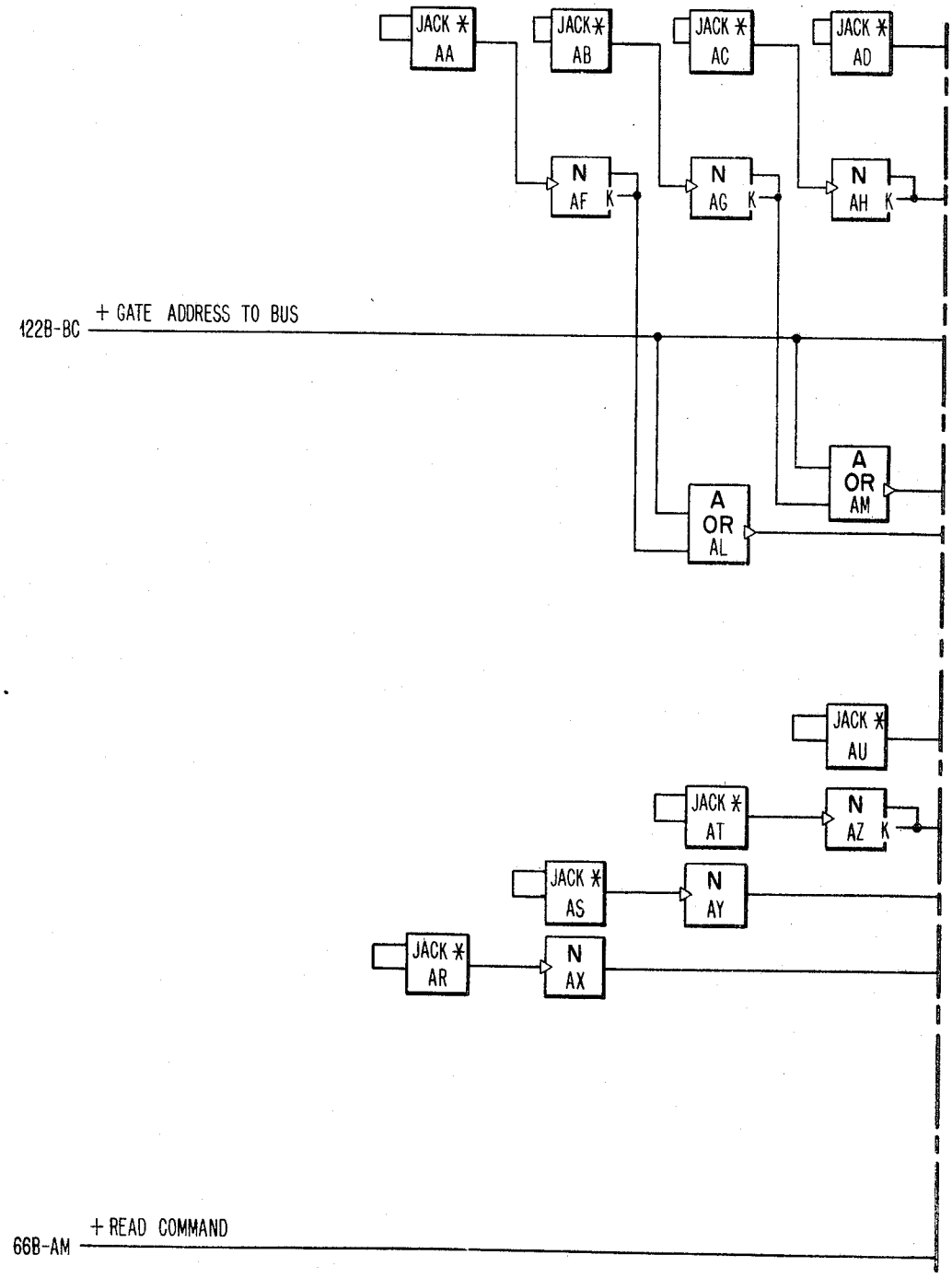

JACK * AA
JACK * AB
JACK * AC
JACK * AD
N AF K
N AG K
N AH K
122B-BC
+ GATE ADDRESS TO BUS
A OR AM
A OR AL
JACK * AU
JACK * AT
N AZ K
JACK * AS
N AY
JACK * AR
N AX
66B-AM
+ READ COMMAND

FIG. 130B

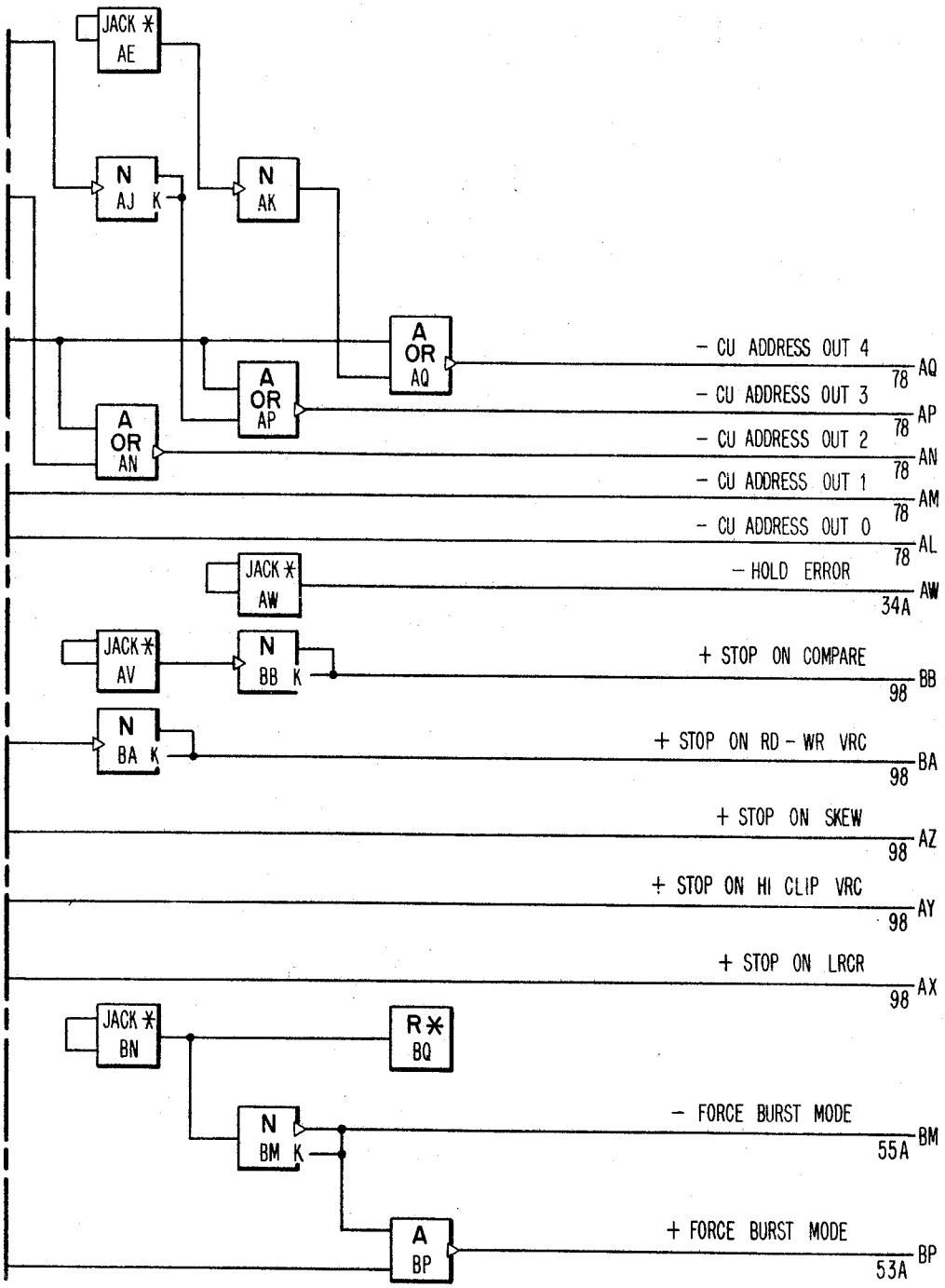

JACK * AE
N AJ K
N AK
A OR AQ
A OR AP
A OR AN
- CU ADDRESS OUT 4
78 AQ
- CU ADDRESS OUT 3
78 AP
- CU ADDRESS OUT 2
78 AN
- CU ADDRESS OUT 1
78 AM
- CU ADDRESS OUT 0
78 AL
JACK * AW
- HOLD ERROR
34A AW
JACK * AV
N BB K
+ STOP ON COMPARE
98 BB
N BA K
+ STOP ON RD - WR VRC
98 BA
+ STOP ON SKEW
98 AZ
+ STOP ON HI CLIP VRC
98 AY
+ STOP ON LRCR
98 AX
JACK * BN
R * BQ
N BM K
- FORCE BURST MODE
55A BM
A BP
+ FORCE BURST MODE
53A BP

FIG.131A

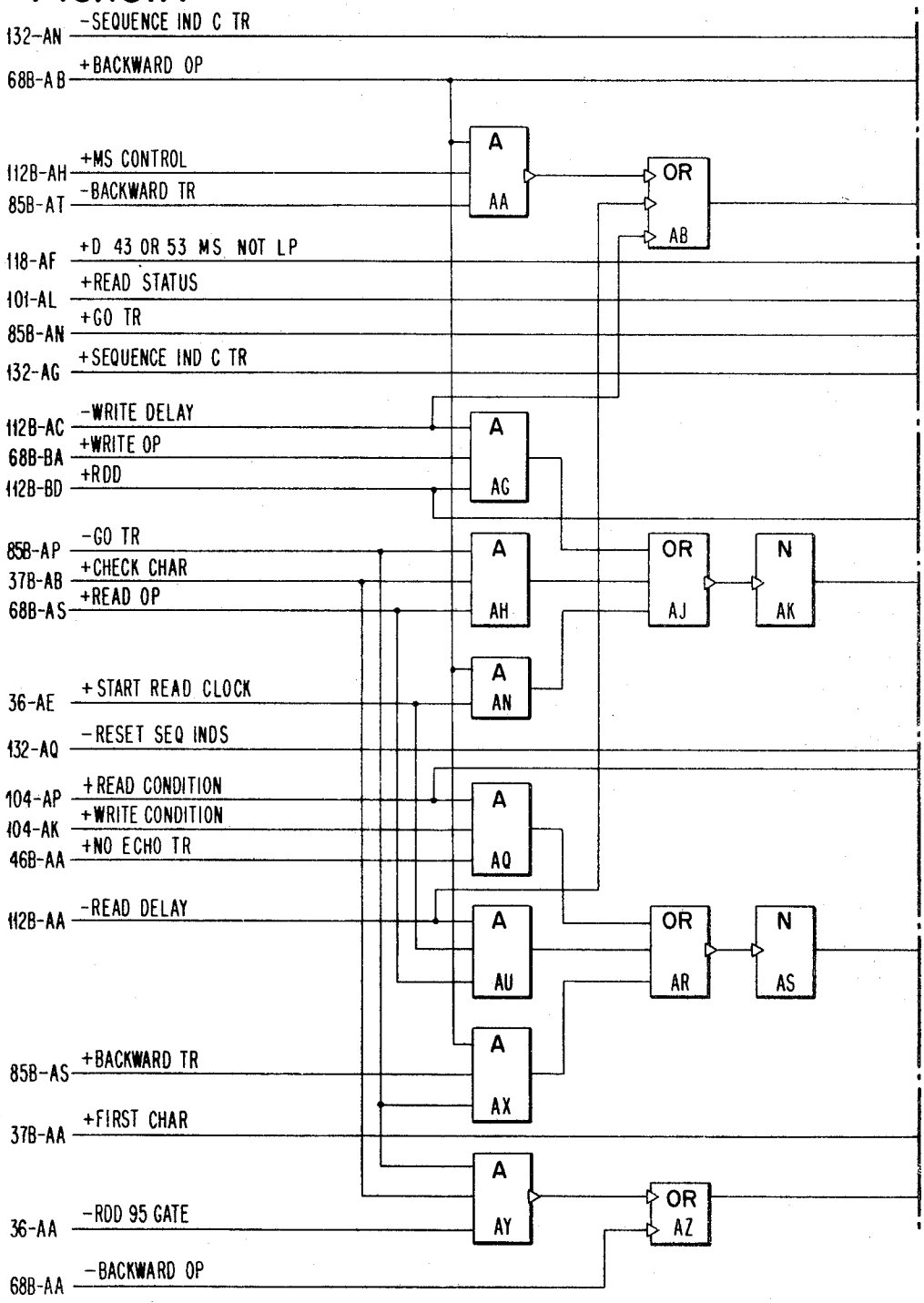

132-AN
-SEQUENCE IND C TR
68B-AB
+BACKWARD OP
112B-AH
+MS CONTROL
85B-AT
-BACKWARD TR
A
AA
OR
AB
118-AF
+D 43 OR 53 MS NOT LP
101-AL
+READ STATUS
85B-AN
+GO TR
132-AG
+SEQUENCE IND C TR
112B-AC
-WRITE DELAY
68B-BA
+WRITE OP
112B-BD
+RDD
A
AG
85B-AP
-GO TR
37B-AB
+CHECK CHAR
68B-AS
+READ OP
A
AH
OR
AJ
N
AK
A
AN
36-AE
+START READ CLOCK
132-AQ
-RESET SEQ INDS
104-AP
+READ CONDITION
104-AK
+WRITE CONDITION
46B-AA
+NO ECHO TR
A
AQ
112B-AA
-READ DELAY
A
AU
OR
AR
N
AS
85B-AS
+BACKWARD TR
A
AX
37B-AA
+FIRST CHAR
36-AA
-RDD 95 GATE
A
AY
OR
AZ
68B-AA
-BACKWARD OP

FIG.131B

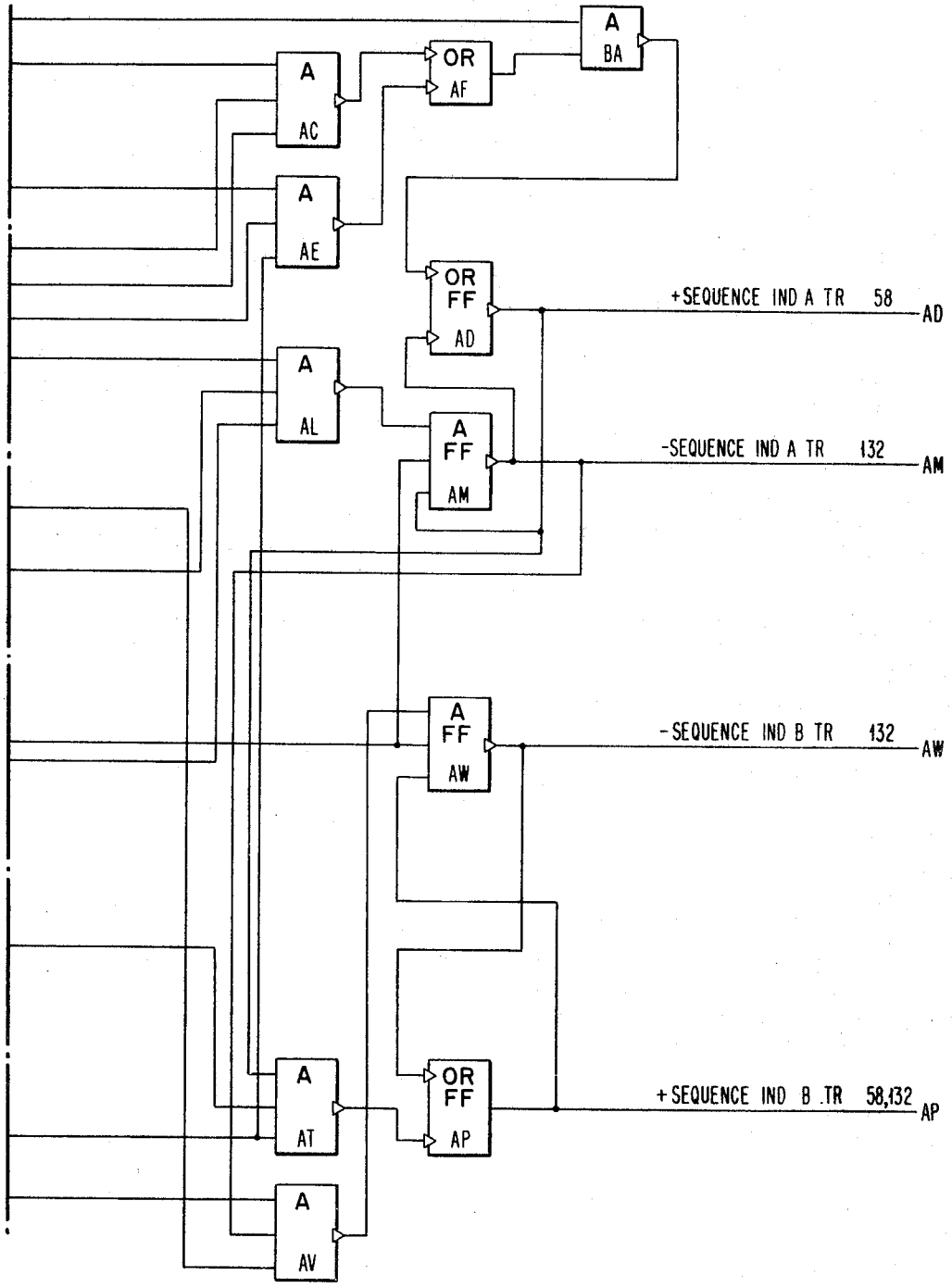

A
BA
OR
AF
A
AC
A
AE
OR
FF
AD
+SEQUENCE IND A TR 58
AD
A
AL
A
FF
AM
-SEQUENCE IND A TR 132
AM
A
FF
AW
-SEQUENCE IND B TR 132
AW
A
AT
OR
FF
AP
+SEQUENCE IND B TR 58,132
AP
A
AV

FIG. 132

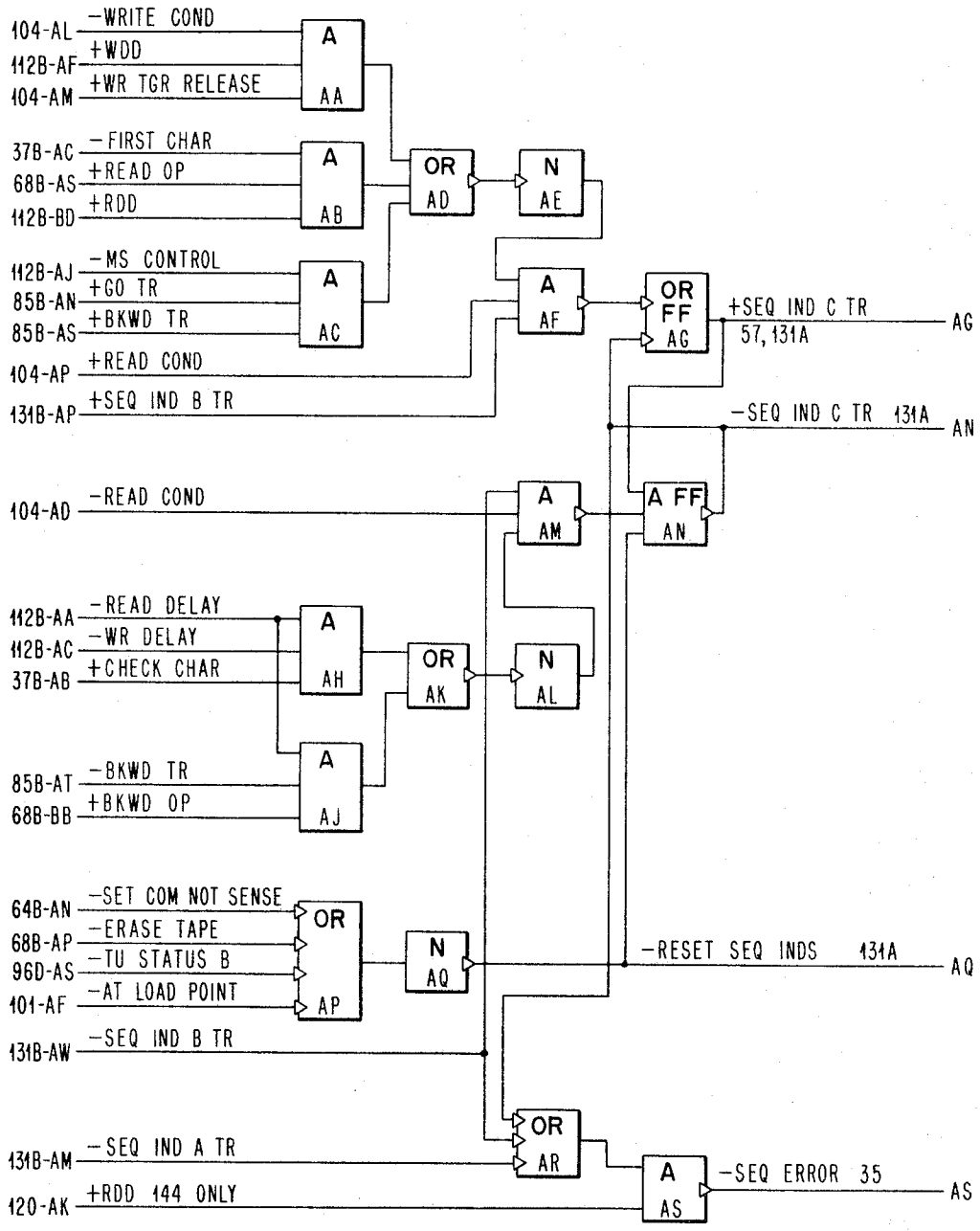

104-AL
-WRITE COND
112B-AF
+WDD
104-AM
+WR TGR RELEASE
A
AA
37B-AC
-FIRST CHAR
68B-AS
+READ OP
112B-BD
+RDD
A
AB
OR
AD
N
AE
112B-AJ
-MS CONTROL
85B-AN
+GO TR
85B-AS
+BKWD TR
A
AC
A
AF
OR
FF
AG
+SEQ IND C TR
57, 131A
AG
104-AP
+READ COND
131B-AP
+SEQ IND B TR
-SEQ IND C TR 131A
AN
104-AD
-READ COND
A
AM
A FF
AN
112B-AA
-READ DELAY
112B-AC
-WR DELAY
37B-AB
+CHECK CHAR
A
AH
OR
AK
N
AL
85B-AT
-BKWD TR
68B-BB
+BKWD OP
A
AJ
64B-AN
-SET COM NOT SENSE
68B-AP
-ERASE TAPE
96D-AS
-TU STATUS B
101-AF
-AT LOAD POINT
OR
AP
N
AQ
-RESET SEQ INDS 131A
AQ
131B-AW
-SEQ IND B TR
OR
AR
131B-AM
-SEQ IND A TR
120-AK
+RDD 144 ONLY
A
AS
-SEQ ERROR 35
AS

FIG.133

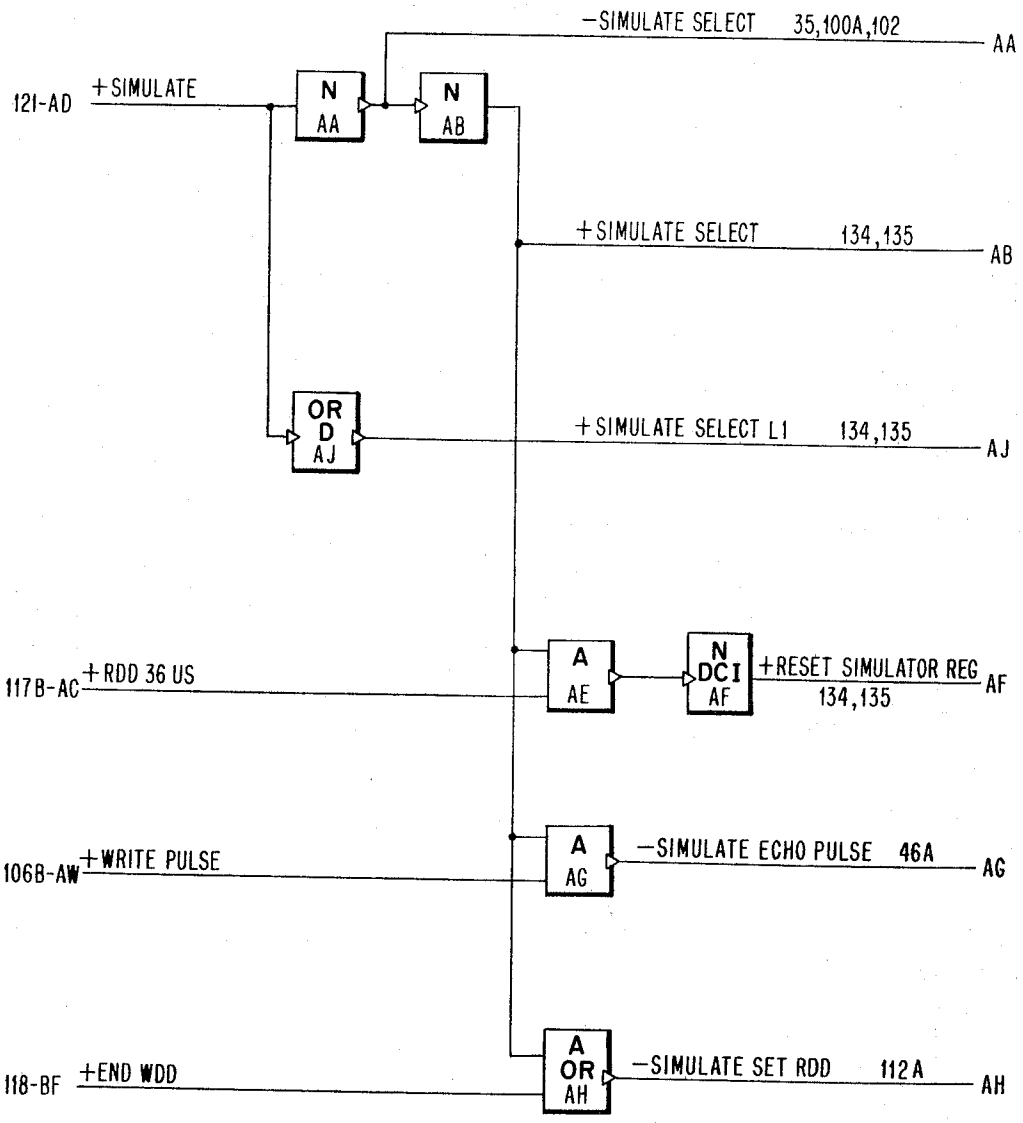

-SIMULATE SELECT 35,100A,102
AA
121-AD +SIMULATE
N AA
N AB
+SIMULATE SELECT 134,135
AB
OR D AJ
+SIMULATE SELECT L1 134,135
AJ
117B-AC +RDD 36 US
A AE
N DCI AF
+RESET SIMULATOR REG 134,135
AF
106B-AW +WRITE PULSE
A AG
-SIMULATE ECHO PULSE 46A
AG
118-BF +END WDD
A OR AH
-SIMULATE SET RDD 112A
AH

FIG. 134

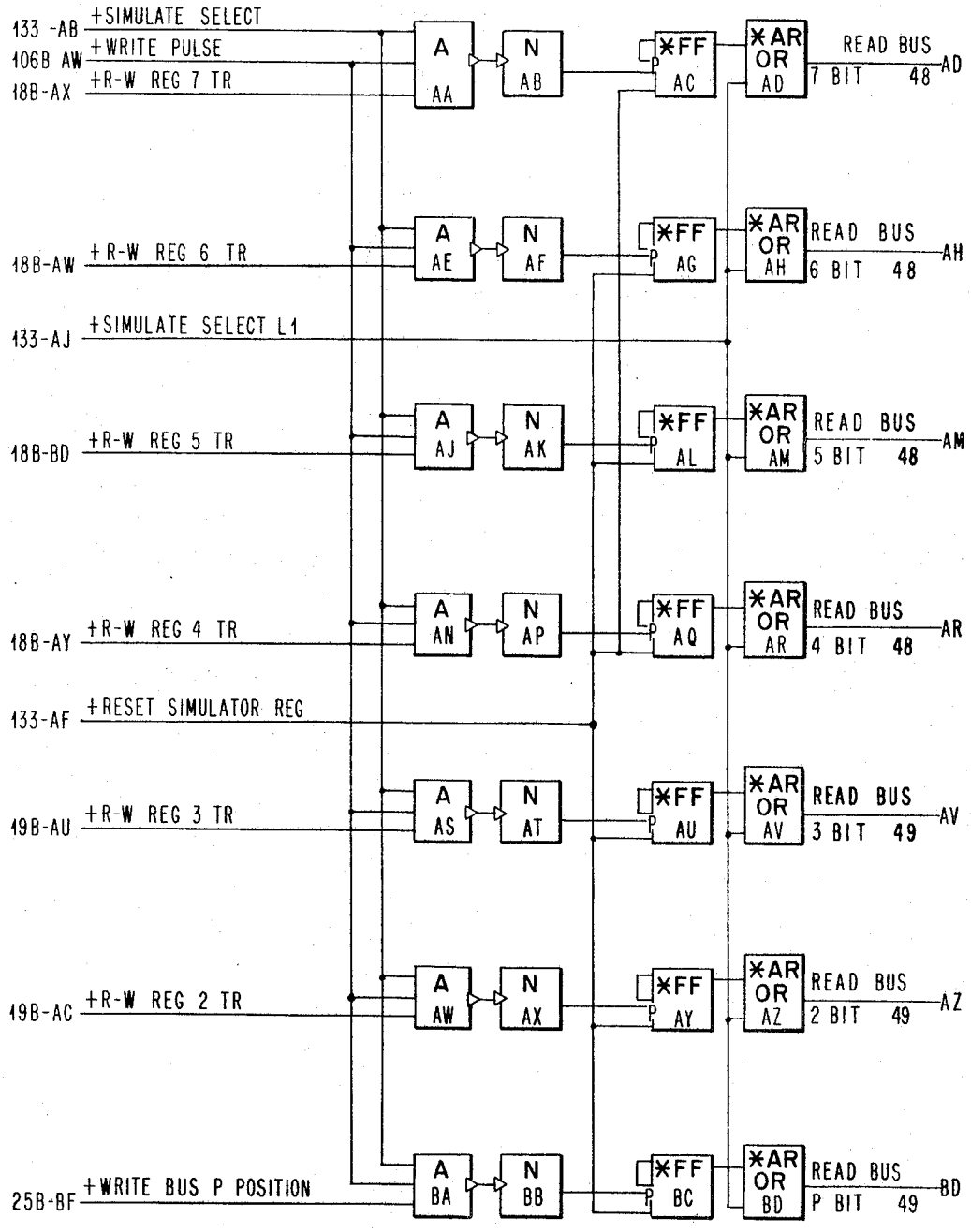
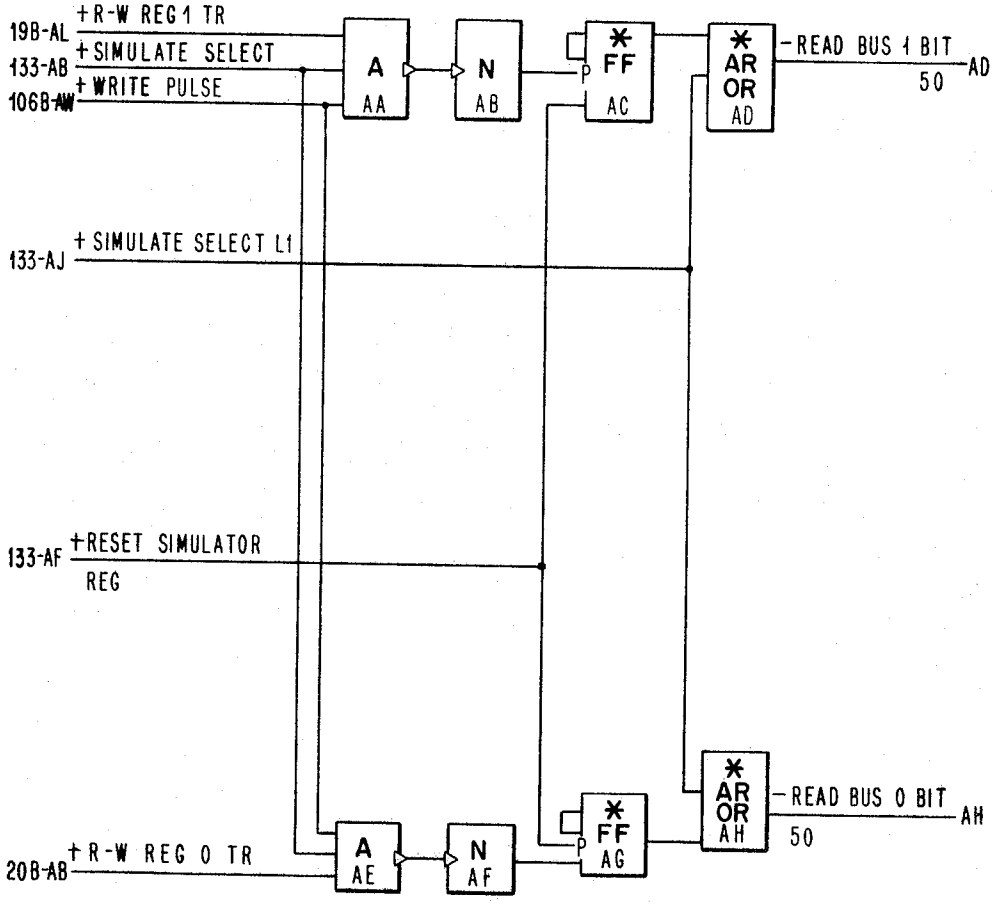
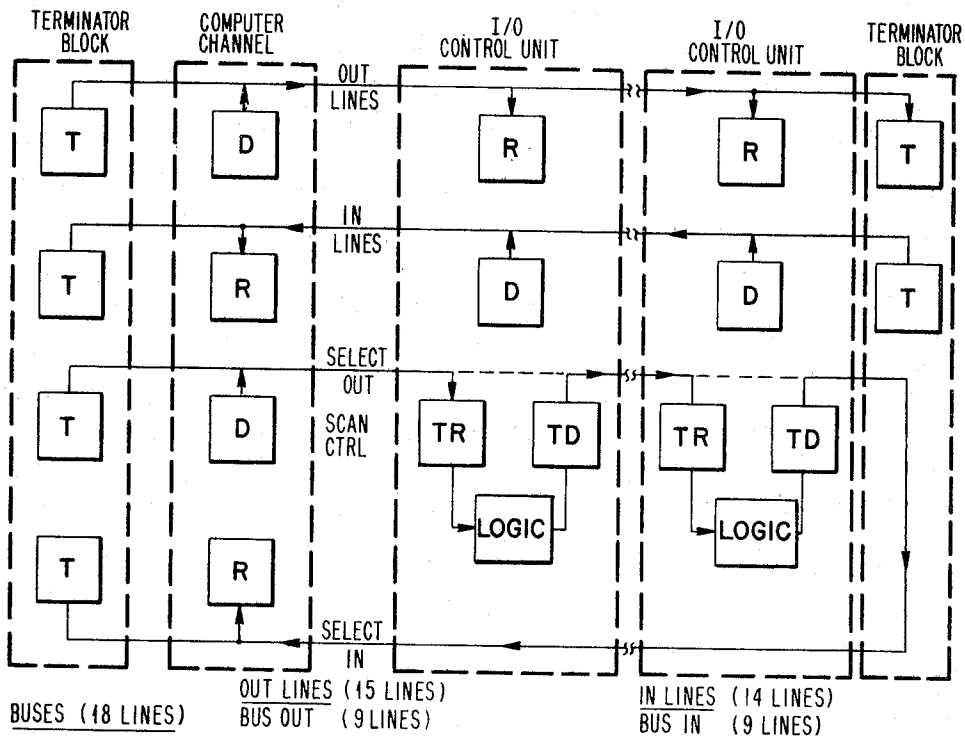

133 -AB +SIMULATE SELECT
106B AW +WRITE PULSE
18B-AX +R-W REG 7 TR
18B-AW +R-W REG 6 TR
133-AJ +SIMULATE SELECT L1
18B-BD +R-W REG 5 TR
18B-AY +R-W REG 4 TR
133-AF +RESET SIMULATOR REG
19B-AU +R-W REG 3 TR
19B-AC +R-W REG 2 TR
25B-BF +WRITE BUS P POSITION
A AA
N AB
*FF AC
*AR OR AD
READ BUS 7 BIT 48 AD
A AE
N AF
*FF AG
*AR OR AH
READ BUS 6 BIT 48 AH
A AJ
N AK
*FF AL
*AR OR AM
READ BUS 5 BIT 48 AM
A AN
N AP
*FF AQ
*AR OR AR
READ BUS 4 BIT 48 AR
A AS
N AT
*FF AU
*AR OR AV
READ BUS 3 BIT 49 AV
A AW
N AX
*FF AY
*AR OR AZ
READ BUS 2 BIT 49 AZ
A BA
N BB
*FF BC
*AR OR BD
READ BUS P BIT 49 BD

FIG. 135


19B-AL
+R-W REG 1 TR
133-AB
+SIMULATE SELECT
106B-AW
+WRITE PULSE
A
AA
N
AB
*
FF
AC
P
*
AR
OR
AD
-READ BUS 1 BIT
50
AD
133-AJ
+SIMULATE SELECT L1
133-AF
+RESET SIMULATOR
REG
20B-AB
+R-W REG 0 TR
A
AE
N
AF
*
FF
AG
P
*
AR
OR
AH
-READ BUS 0 BIT
50
AH

# FIG. 136




| | | | |
|---|---|---|---|
| BUSES (18 LINES) | OUT LINES (15 LINES)<br>BUS OUT (9 LINES) | | IN LINES (14 LINES)<br>BUS IN (9 LINES) |
| TAGS (6 LINES) | ADDRESS OUT<br>COMMAND OUT<br>SERVICE OUT | D.....DRIVER<br>R.....RECEIVER<br>T.....TERMINATOR | ADDRESS IN<br>STATUS IN<br>SERVICE IN |
| SCAN CTRLS (2 LINES) | SELECT OUT | | SELECT IN |
| INTERLOCKS (2 LINES) | OPERATIONAL OUT | | OPERATIONAL IN |
| SPECIAL CONTROL (5 LINE) | SUPPRESS OUT<br>METERING OUT<br>HOLD OUT | | METERING IN<br>REQUEST IN |

I/O CONTROL UNITS CONNECT IN SERIES TO SELECT OUT AND SELECT IN

I/O CONTROL UNITS CONNECT IN PARALLEL TO ALL OTHER INTERFACE LINES

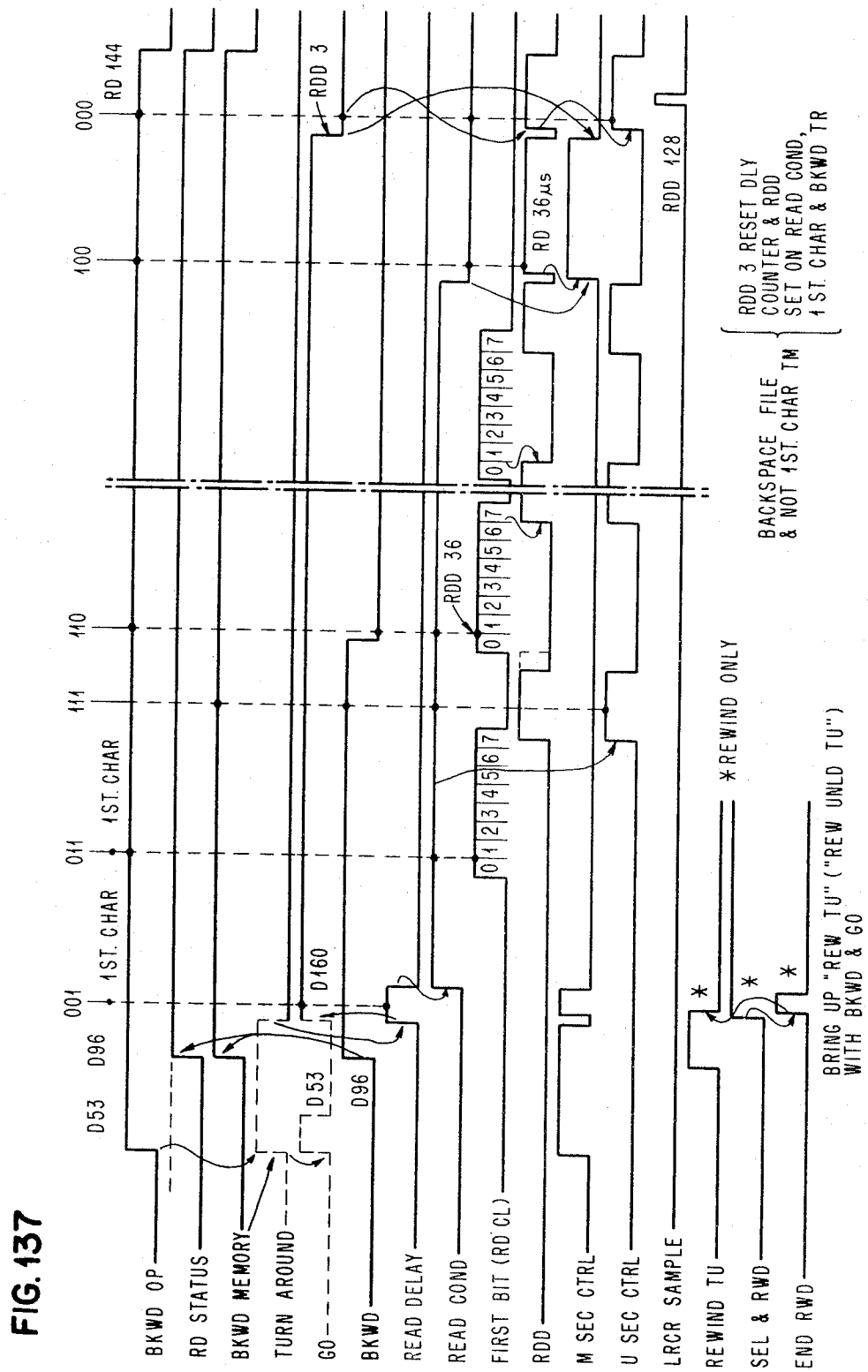

000
100
110
111
011
001
RD 144
RDD 3
RD 36 μs
RDD 128
RDD 36
1ST. CHAR
1ST. CHAR
D160
D53
D96
D53
D96
0 1 2 3 4 5 6 7
RDD 3 RESET DLY
COUNTER & RDD
SET ON READ COND,
1 ST. CHAR & BKWD TR
BACKSPACE FILE
& NOT 1ST. CHAR TM
*REWIND ONLY
BRING UP "REW TU" ("REW UNLD TU")
WITH BKWD & GO
BKWD OP
RD STATUS
BKWD MEMORY
TURN AROUND
GO
BKWD
READ DELAY
READ COND
FIRST BIT (RD CL)
RDD
M SEC CTRL
U SEC CTRL
LRCR SAMPLE
REWIND TU
SEL & RWD
END RWD

FIG. 137

FIG. 138A

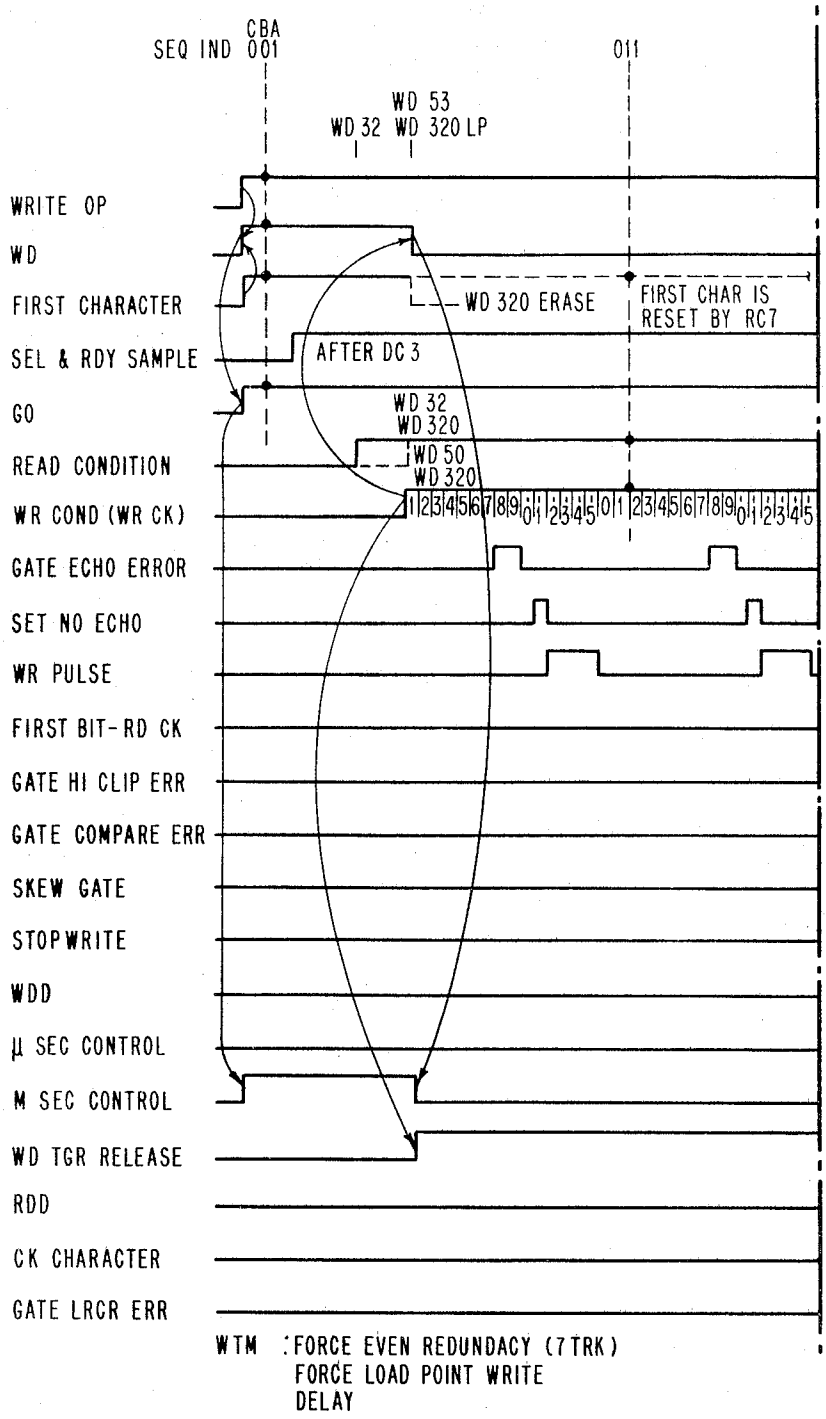

CBA
SEQ IND 001
011
WD 53
WD 32 WD 320 LP
WRITE OP
WD
FIRST CHARACTER
WD 320 ERASE
FIRST CHAR IS RESET BY RC7
SEL & RDY SAMPLE
AFTER DC 3
GO
WD 32
WD 320
READ CONDITION
WD 50
WD 320
WR COND (WR CK)
GATE ECHO ERROR
SET NO ECHO
WR PULSE
FIRST BIT-RD CK
GATE HI CLIP ERR
GATE COMPARE ERR
SKEW GATE
STOP WRITE
WDD
μ SEC CONTROL
M SEC CONTROL
WD TGR RELEASE
RDD
CK CHARACTER
GATE LRCR ERR
WTM : FORCE EVEN REDUNDACY (7 TRK)
FORCE LOAD POINT WRITE
DELAY

FIG.138B

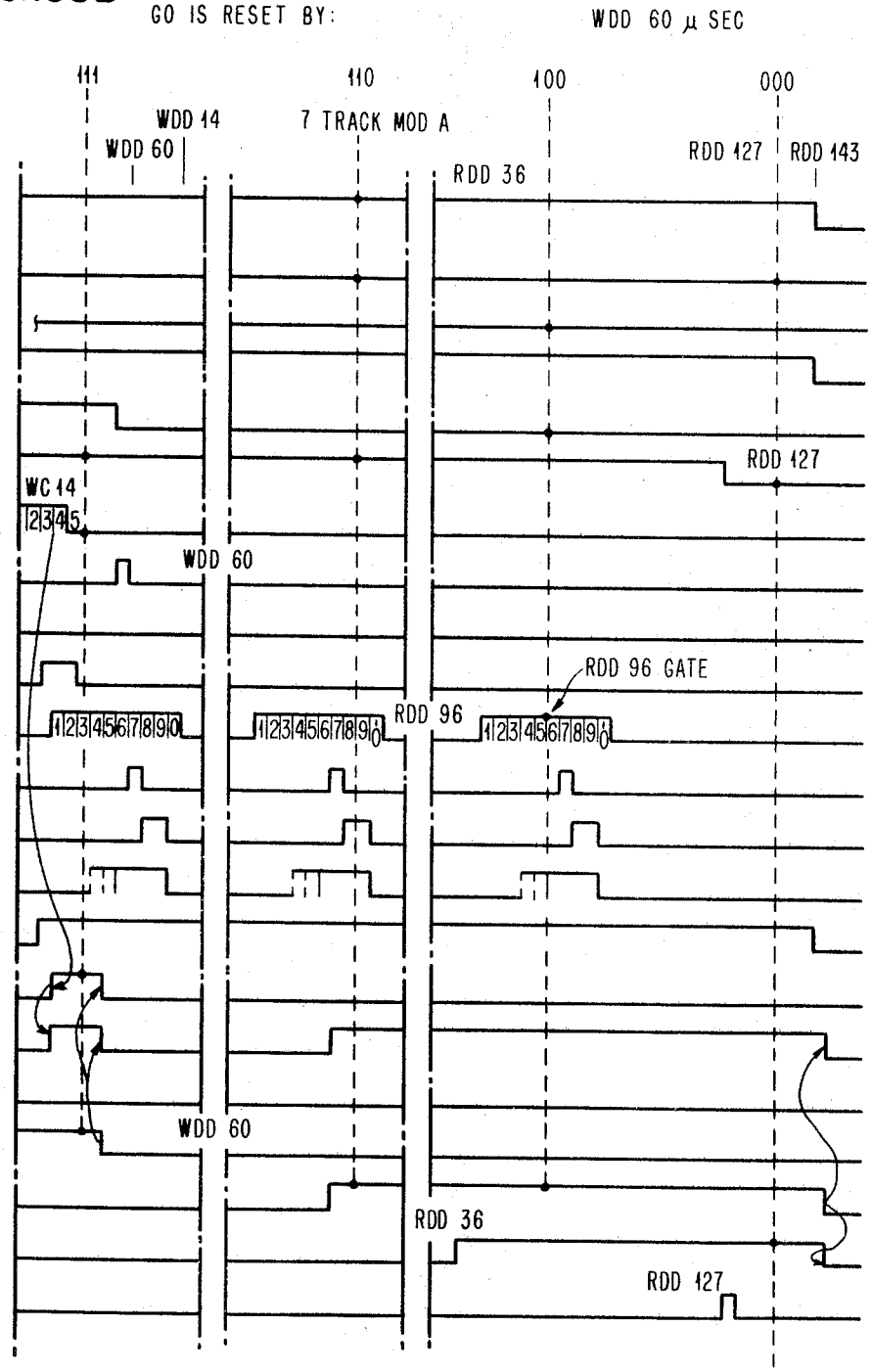

GO IS RESET BY:
WDD 60 μ SEC
111
110
100
000
WDD 14
WDD 60
7 TRACK MOD A
RDD 36
RDD 127
RDD 143
RDD 127
WC 14
WDD 60
RDD 96 GATE
RDD 96
WDD 60
RDD 36
RDD 127

FIG. 139A

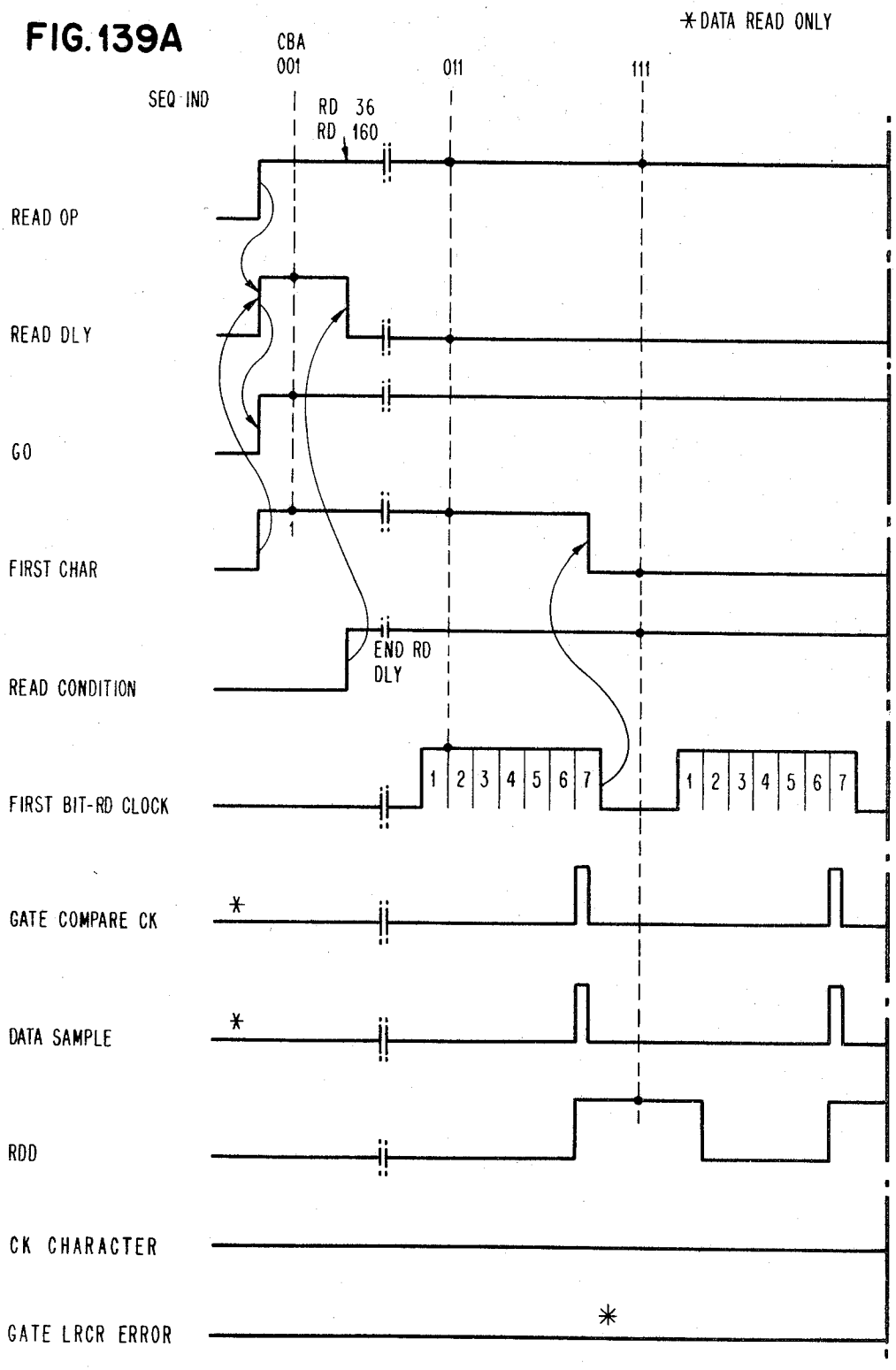

*DATA READ ONLY
CBA
001
011
111
SEQ IND
RD 36
RD 160
READ OP
READ DLY
GO
1
FIRST CHAR
END RD DLY
READ CONDITION
1 2 3 4 5 6 7
1 2 3 4 5 6 7
FIRST BIT-RD CLOCK
*
GATE COMPARE CK
*
DATA SAMPLE
RDD
CK CHARACTER
*
GATE LRCR ERROR

FIG. 139B

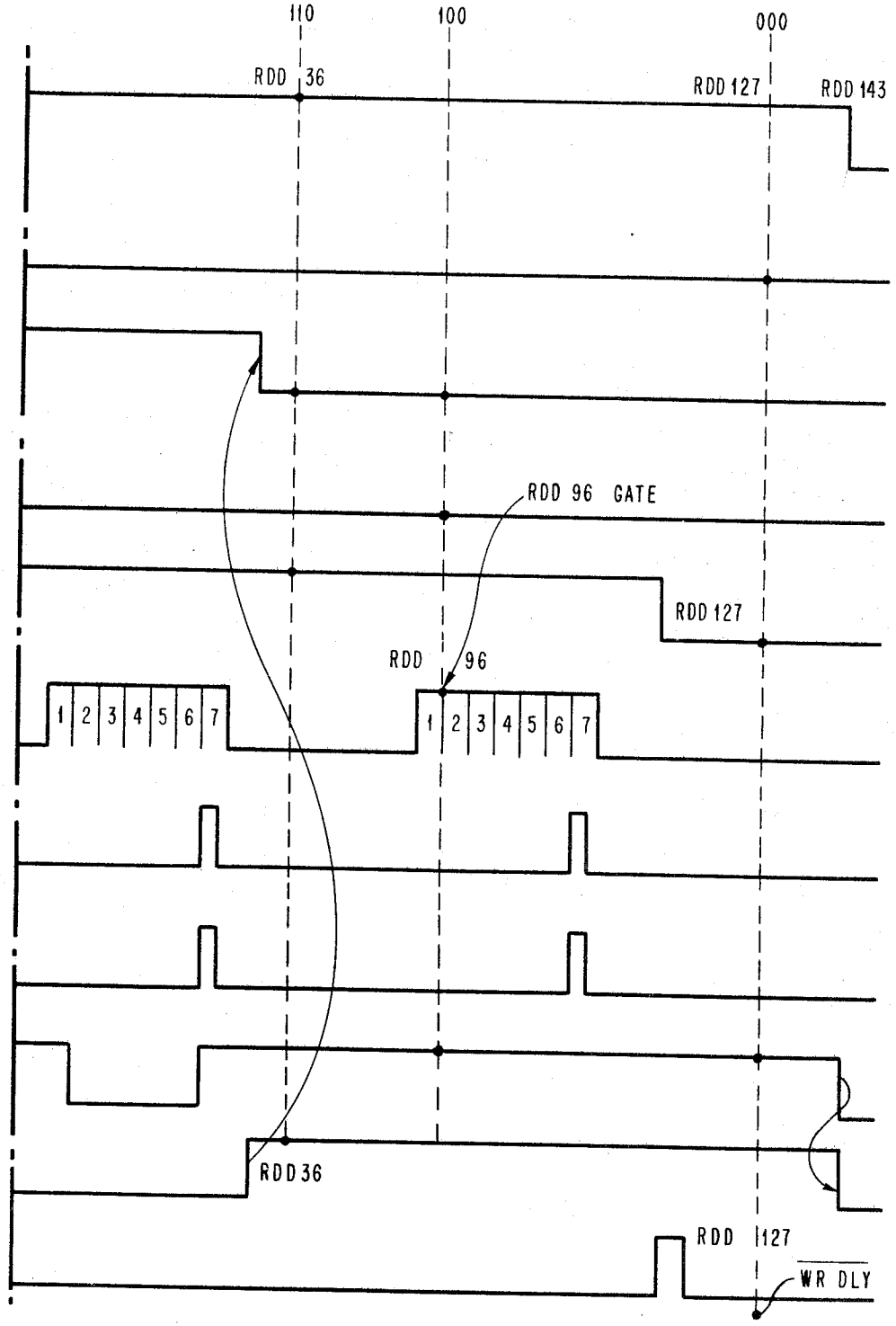

110
100
000
RDD 36
RDD 127
RDD 143
RDD 96 GATE
RDD 127
RDD 96
1 2 3 4 5 6 7
1 2 3 4 5 6 7
RDD 36
RDD 127
WR DLY

FIG. 140

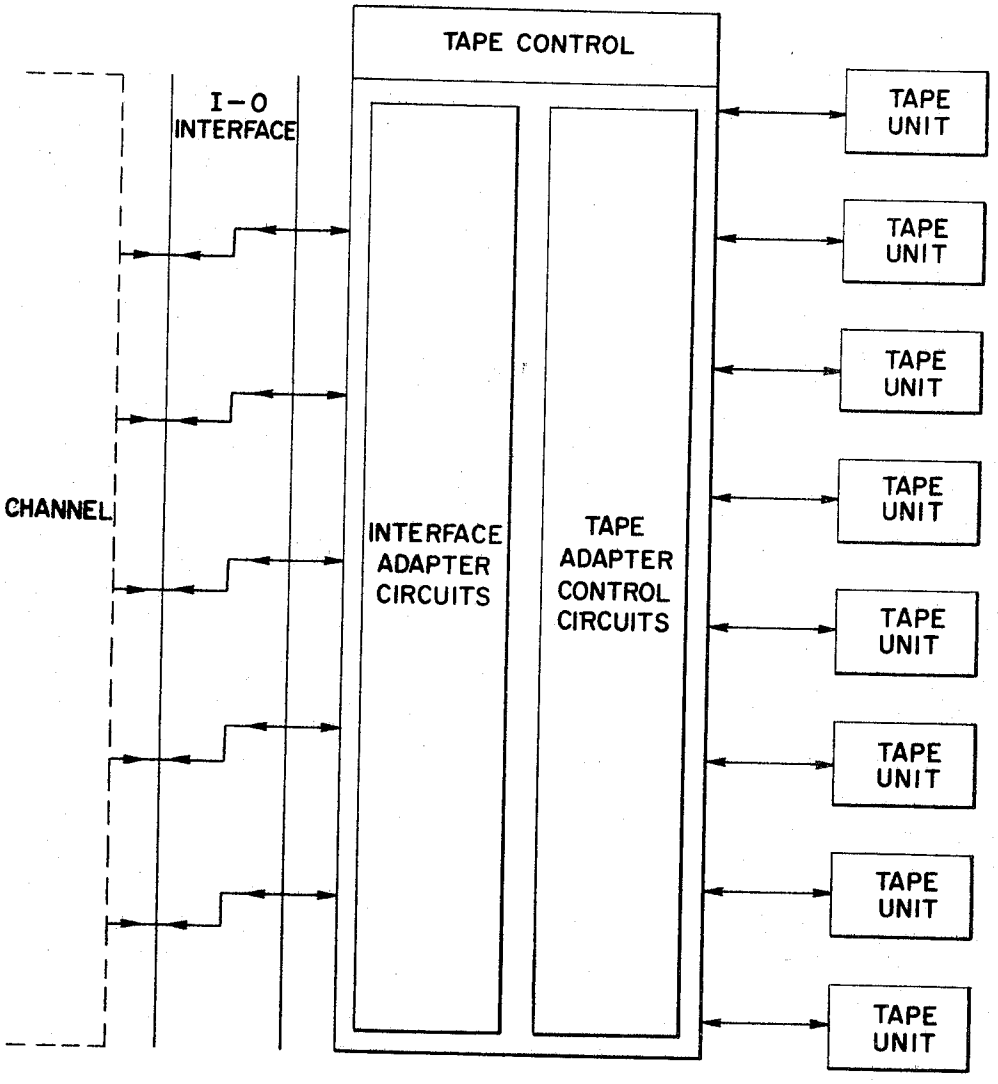
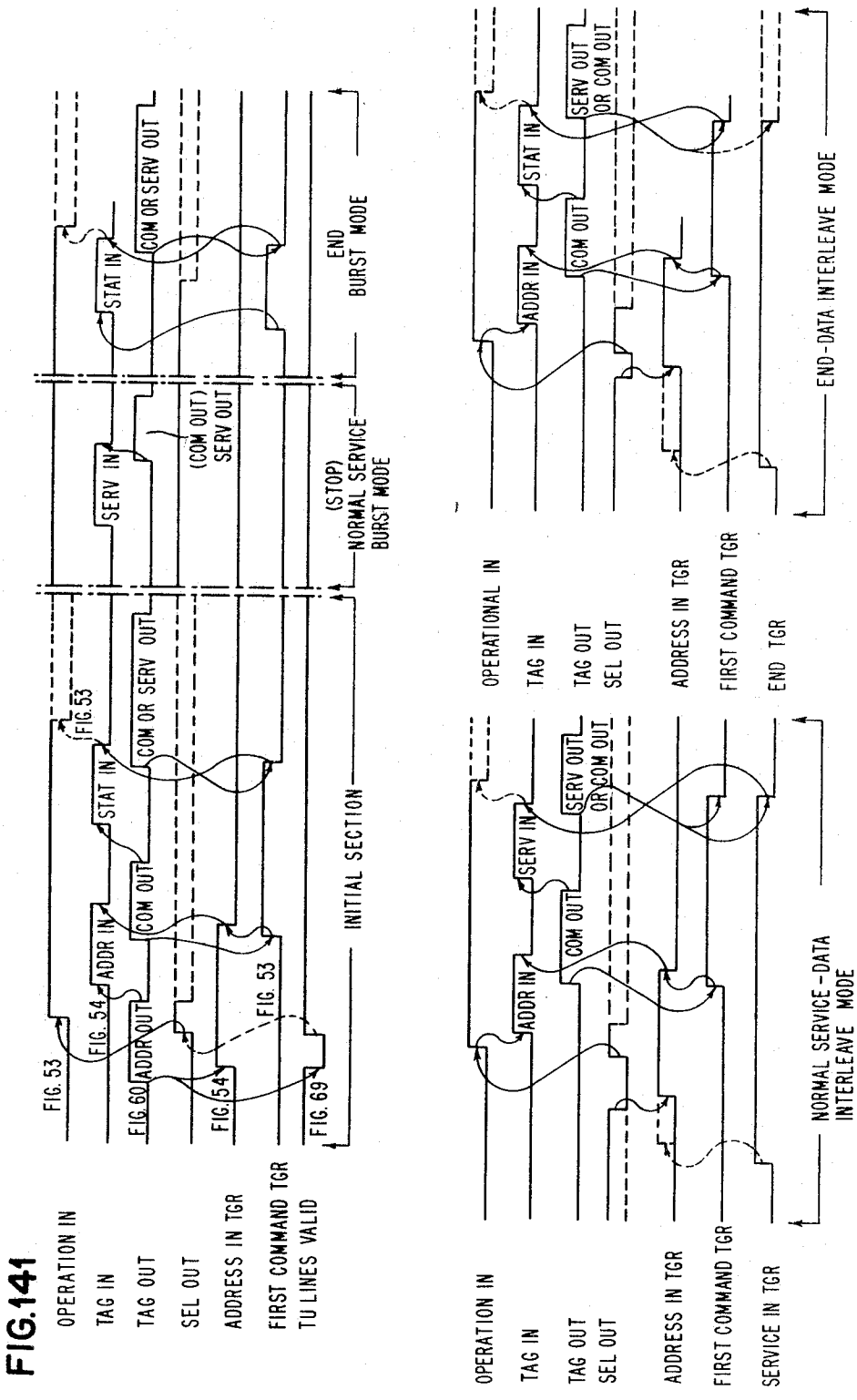

TAPE CONTROL
I-O INTERFACE
CHANNEL
INTERFACE ADAPTER CIRCUITS
TAPE ADAPTER CONTROL CIRCUITS
TAPE UNIT
TAPE UNIT
TAPE UNIT
TAPE UNIT
TAPE UNIT
TAPE UNIT
TAPE UNIT
TAPE UNIT

FIG.141


OPERATION IN
TAG IN
TAG OUT
SEL OUT
ADDRESS IN TGR
FIRST COMMAND TGR
TU LINES VALID
FIG. 53
FIG. 54
FIG. 60
FIG. 54
FIG. 53
FIG. 69
ADDR OUT
ADDR IN
STAT IN
COM OUT
COM OR SERV OUT
SERV IN
(COM OUT) SERV OUT
INITIAL SECTION
(STOP) NORMAL SERVICE BURST MODE
END BURST MODE
OPERATIONAL IN
TAG IN
TAG OUT
SEL OUT
ADDRESS IN TGR
FIRST COMMAND TGR
END TGR
ADDR IN
STAT IN
COM OUT
SERV OUT OR COM OUT
END-DATA INTERLEAVE MODE
SERVICE IN TGR
SERV IN
NORMAL SERVICE-DATA INTERLEAVE MODE

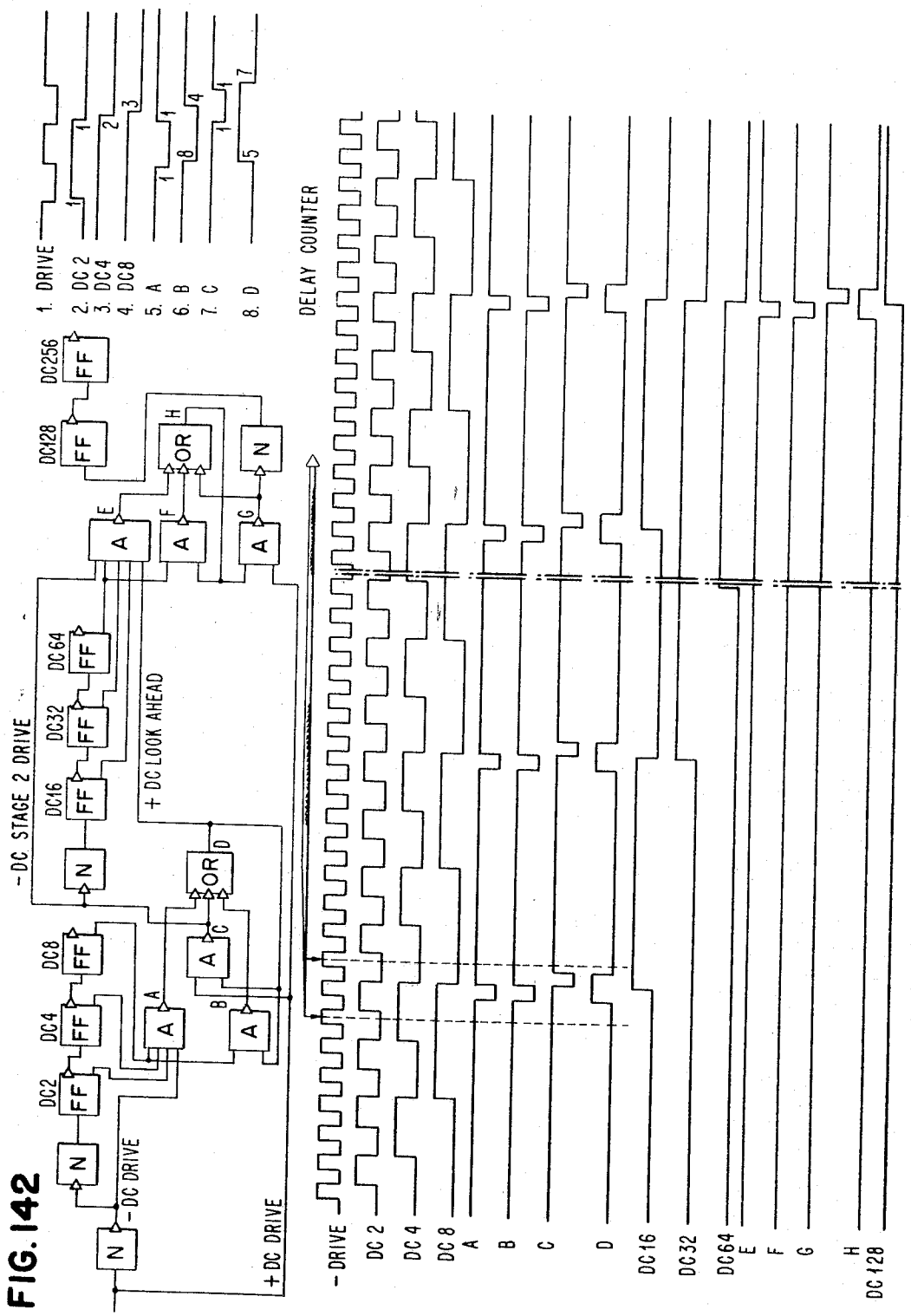
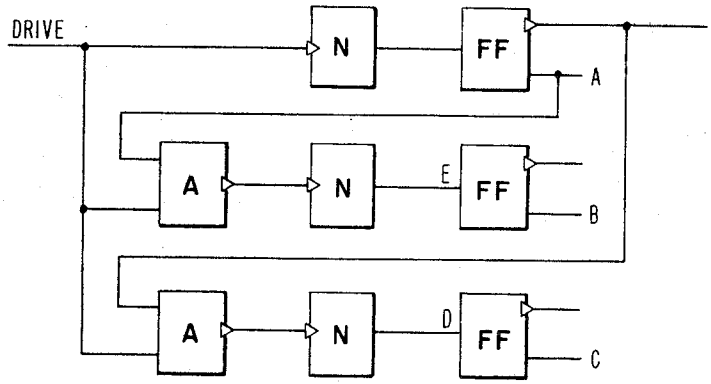

1. DRIVE
2. DC 2
3. DC4
4. DC8
5. A
6. B
7. C
8. D
DELAY COUNTER
DC256
DC128
DC64
DC32
DC16
DC8
DC4
DC2
FF
N
OR
A
H
E
F
G
D
C
B
A
-DC STAGE 2 DRIVE
+ DC LOOK AHEAD
-DC DRIVE
+DC DRIVE
- DRIVE
DC 2
DC 4
DC 8
A
B
C
D
DC 16
DC 32
DC 64
E
F
G
H
DC 128

FIG. 142

FIG.143




| C | B | A | |
|---|---|---|---|
| 0 | 0 | 1 | RC0 AND RC8 |
| 0 | 1 | 1 | RC1 AND RC9 |
| 0 | 1 | 0 | RC2 |
| 1 | 1 | 0 | RC3 |
| 1 | 1 | 1 | RC4 |
| 1 | 0 | 1 | RC5 |
| 1 | 0 | 0 | RC6 |
| 0 | 0 | 0 | RC7 |

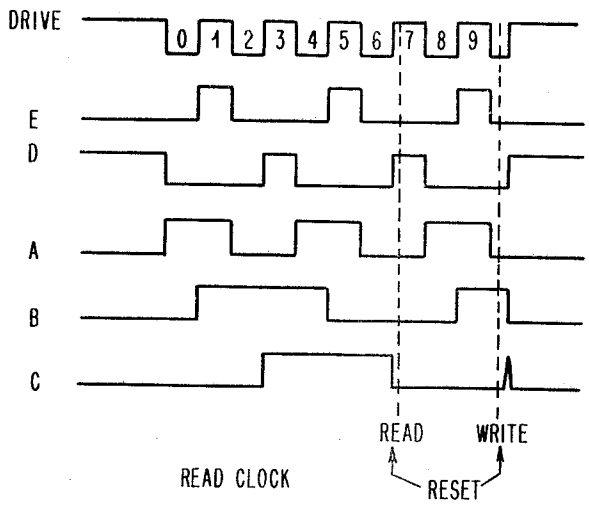
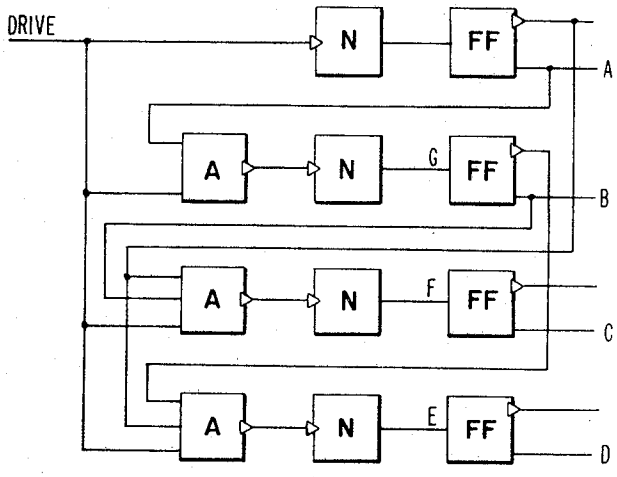


FIG. 144




| D | C | B | A | |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | WC1 |
| 0 | 0 | 1 | 1 | WC2 |
| 0 | 0 | 1 | 0 | WC3 |
| 0 | 1 | 1 | 0 | WC4 |
| 0 | 1 | 1 | 1 | WC5 |
| 0 | 1 | 0 | 1 | WC6 |
| 0 | 1 | 0 | 0 | WC7 |
| 1 | 1 | 0 | 0 | WC8 |
| 1 | 1 | 0 | 1 | WC9 |
| 1 | 1 | 1 | 1 | WC10 |
| 1 | 1 | 1 | 0 | WC11 |
| 1 | 0 | 1 | 0 | WC12 |
| 1 | 0 | 1 | 1 | WC13 |
| 1 | 0 | 0 | 1 | WC14 |
| 1 | 0 | 0 | 0 | WC15 |
| 0 | 0 | 0 | 0 | WC0 |

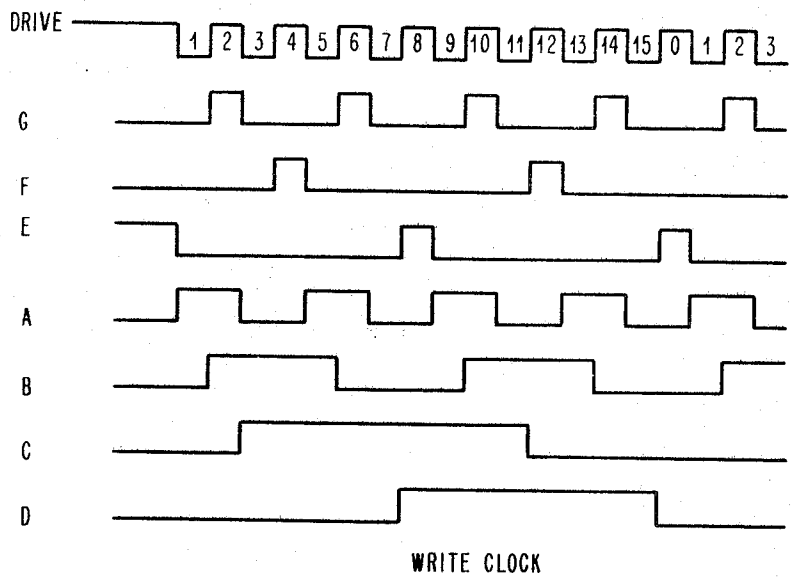


FIG.145

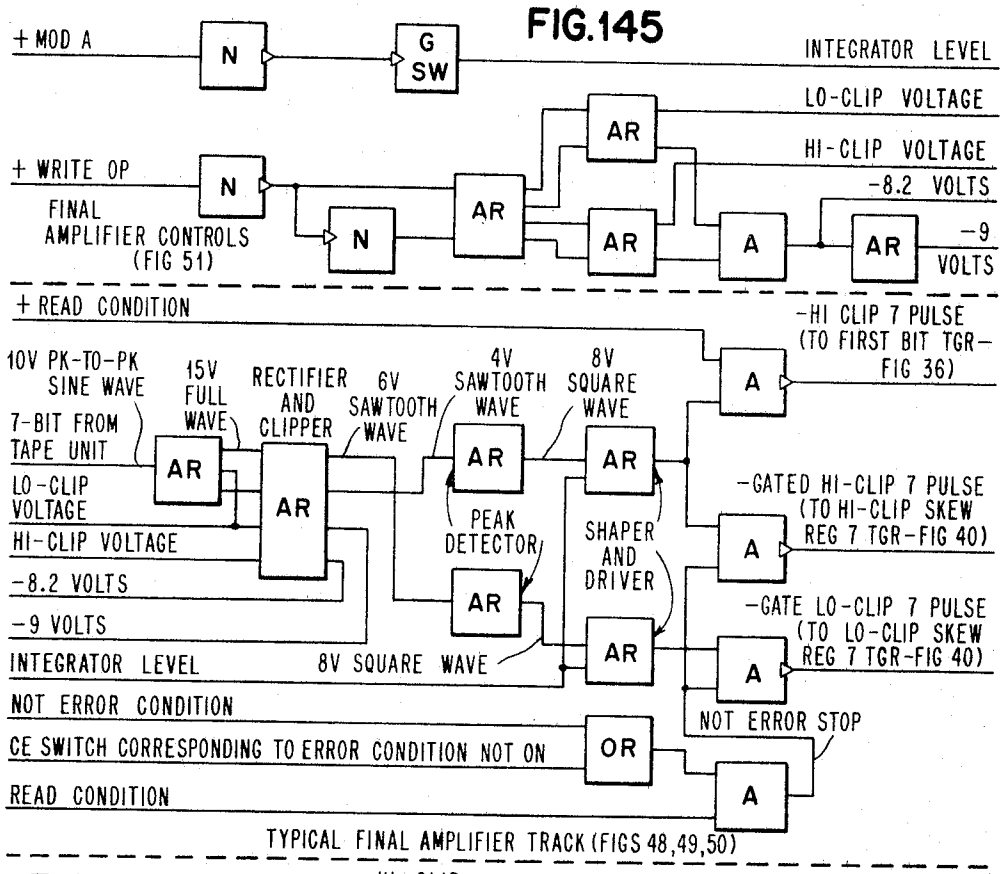
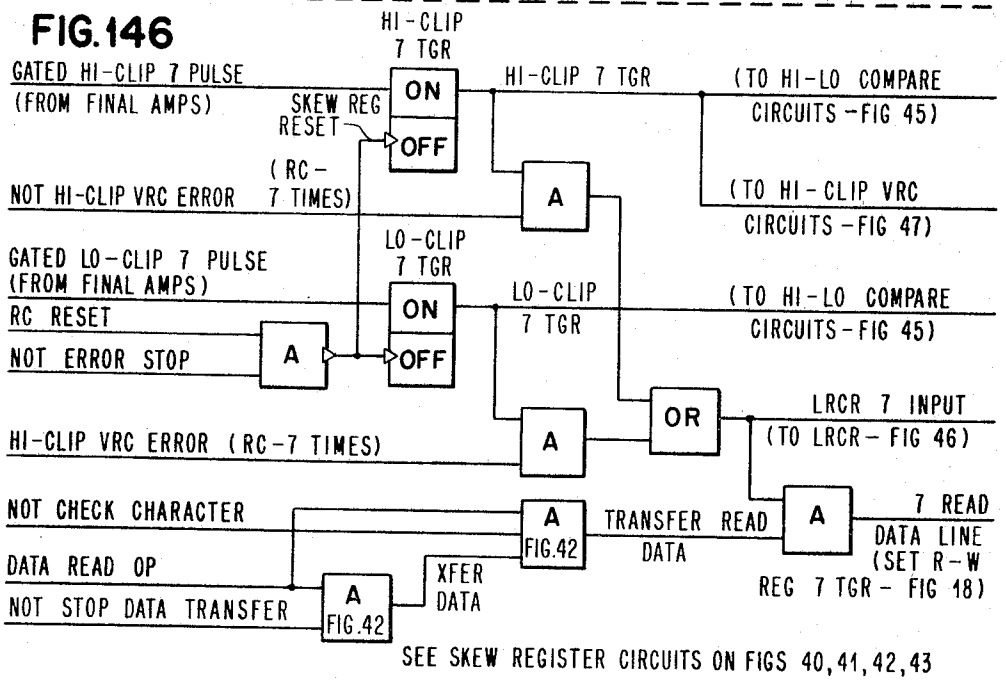



FIG.146



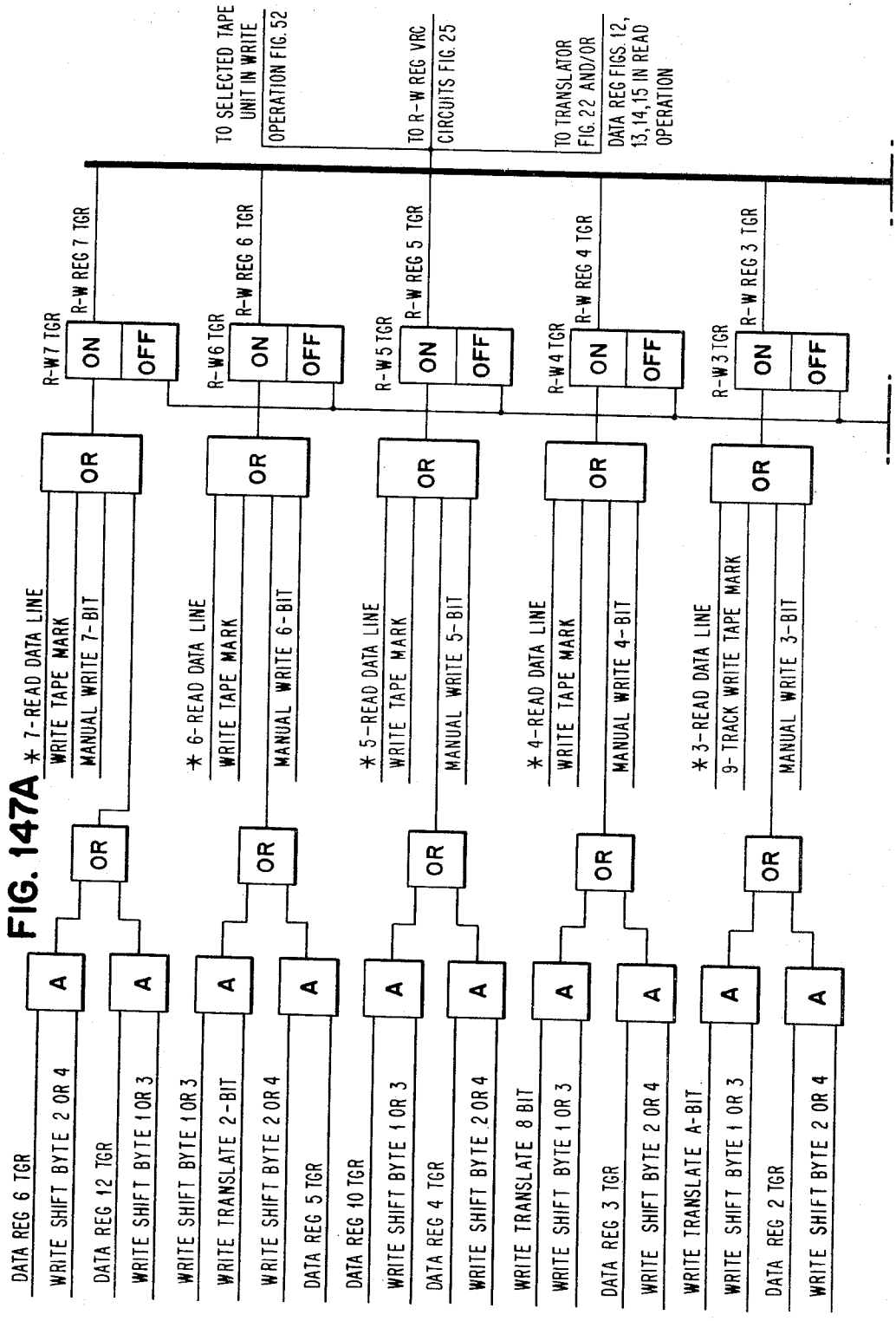

TO SELECTED TAPE UNIT IN WRITE OPERATION FIG. 52
TO R-W REG VRC CIRCUITS FIG. 25
TO TRANSLATOR FIG. 22 AND/OR DATA REG FIGS. 12, 13, 14, 15 IN READ OPERATION
R-W7 TGR
R-W REG 7 TGR
R-W6 TGR
R-W REG 6 TGR
R-W5 TGR
R-W REG 5 TGR
R-W4 TGR
R-W REG 4 TGR
R-W3 TGR
R-W REG 3 TGR
ON
OFF
OR
A
* 7-READ DATA LINE
WRITE TAPE MARK
MANUAL WRITE 7-BIT
* 6-READ DATA LINE
WRITE TAPE MARK
MANUAL WRITE 6-BIT
* 5-READ DATA LINE
WRITE TAPE MARK
MANUAL WRITE 5-BIT
* 4-READ DATA LINE
WRITE TAPE MARK
MANUAL WRITE 4-BIT
* 3-READ DATA LINE
9-TRACK WRITE TAPE MARK
MANUAL WRITE 3-BIT
DATA REG 6 TGR
WRITE SHIFT BYTE 2 OR 4
DATA REG 12 TGR
WRITE SHIFT BYTE 1 OR 3
WRITE SHIFT BYTE 1 OR 3
WRITE TRANSLATE 2-BIT
WRITE SHIFT BYTE 2 OR 4
DATA REG 5 TGR
DATA REG 10 TGR
WRITE SHIFT BYTE 1 OR 3
DATA REG 4 TGR
WRITE SHIFT BYTE 2 OR 4
WRITE TRANSLATE 8 BIT
WRITE SHIFT BYTE 1 OR 3
DATA REG 3 TGR
WRITE SHIFT BYTE 2 OR 4
WRITE TRANSLATE A-BIT
WRITE SHIFT BYTE 1 OR 3
DATA REG 2 TGR
WRITE SHIFT BYTE 2 OR 4

FIG. 147A

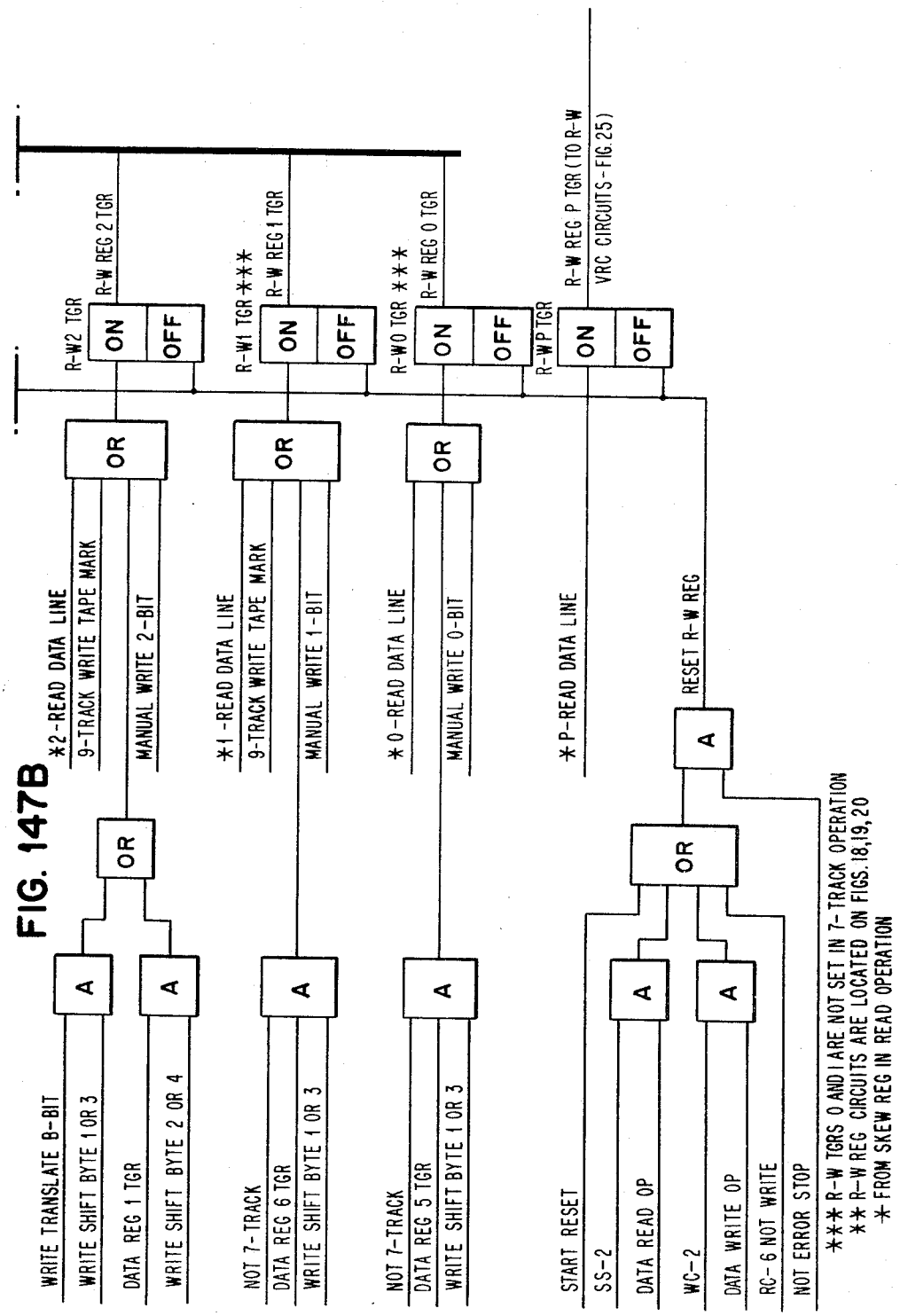
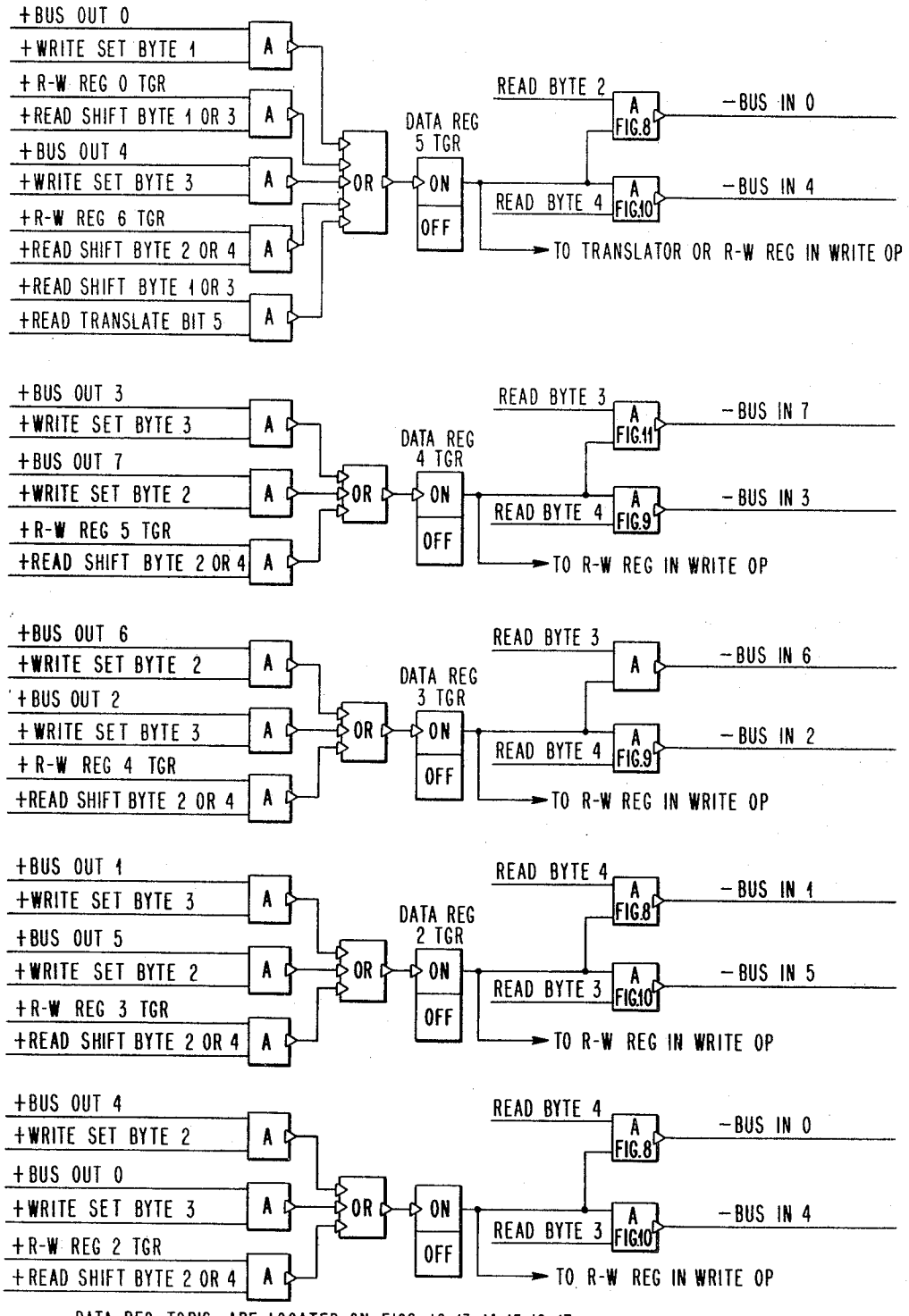

R-W2 TGR
R-W REG 2 TGR
ON
OFF
R-W1 TGR ***
R-W REG 1 TGR
ON
OFF
R-W0 TGR ***
R-W REG 0 TGR
ON
OFF
R-WP TGR
R-W REG P TGR (TO R-W VRC CIRCUITS - FIG. 25)
ON
OFF
OR
OR
OR
*2-READ DATA LINE
9-TRACK WRITE TAPE MARK
MANUAL WRITE 2-BIT
*1-READ DATA LINE
9-TRACK WRITE TAPE MARK
MANUAL WRITE 1-BIT
*0-READ DATA LINE
MANUAL WRITE 0-BIT
*P-READ DATA LINE
RESET R-W REG
A
A
A
A
A
A
OR
WRITE TRANSLATE B-BIT
WRITE SHIFT BYTE 1 OR 3
DATA REG 1 TGR
WRITE SHIFT BYTE 2 OR 4
NOT 7-TRACK
DATA REG 6 TGR
WRITE SHIFT BYTE 1 OR 3
NOT 7-TRACK
DATA REG 5 TGR
WRITE SHIFT BYTE 1 OR 3
START RESET
SS-2
DATA READ OP
WC-2
DATA WRITE OP
RC-6 NOT WRITE
NOT ERROR STOP
*** R-W TGRS 0 AND 1 ARE NOT SET IN 7-TRACK OPERATION
** R-W REG CIRCUITS ARE LOCATED ON FIGS. 18, 19, 20
* FROM SKEW REG IN READ OPERATION

FIG. 147B

FIG. 148




DATA REG TGR'S ARE LOCATED ON FIGS. 12,13,14,15,16,17

FIG.149

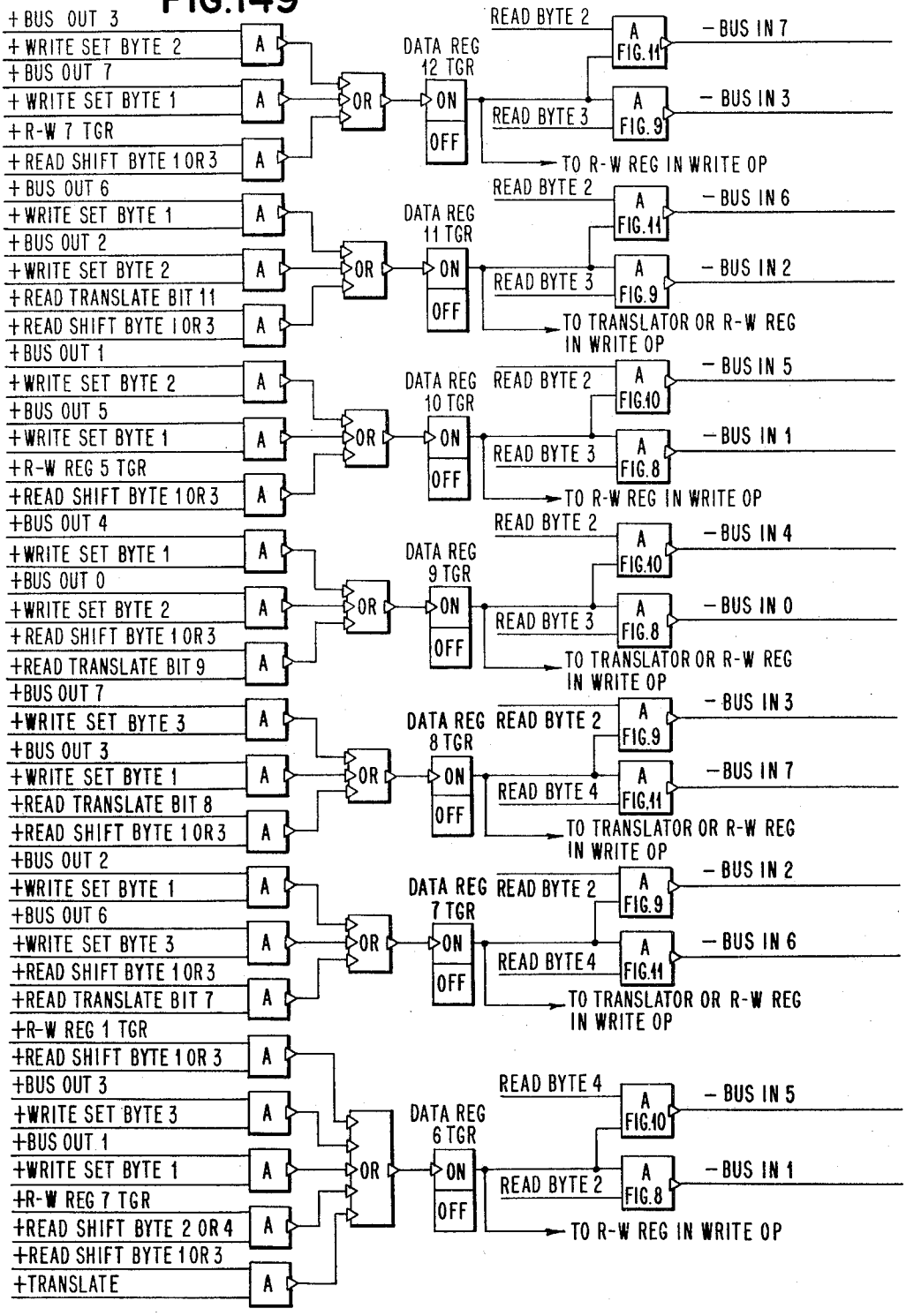
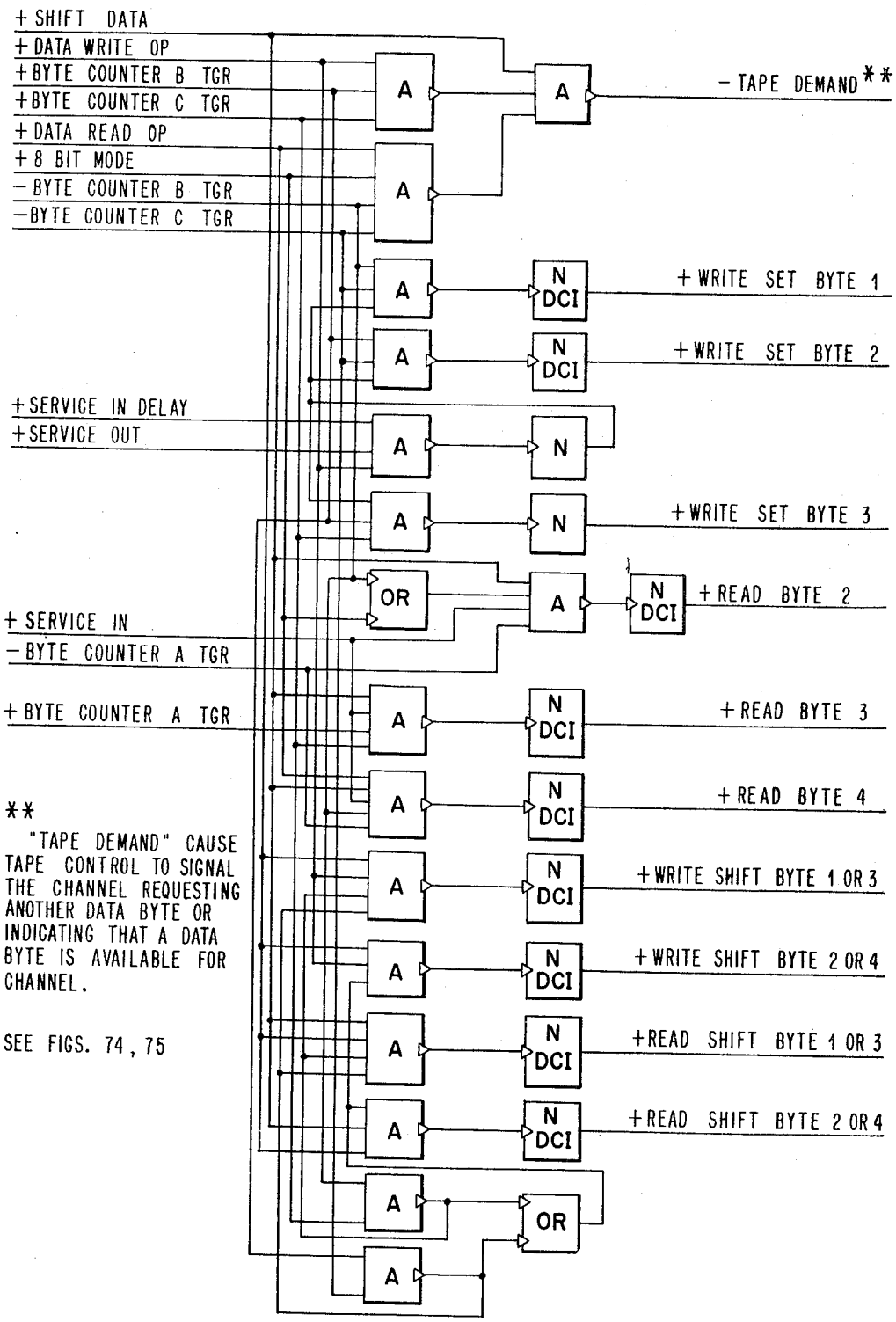

+BUS OUT 3
+WRITE SET BYTE 2
+BUS OUT 7
+WRITE SET BYTE 1
+R-W 7 TGR
+READ SHIFT BYTE 1 OR 3
A
OR
DATA REG 12 TGR
ON
OFF
READ BYTE 2
READ BYTE 3
A FIG.11
A FIG.9
-BUS IN 7
-BUS IN 3
TO R-W REG IN WRITE OP
+BUS OUT 6
+WRITE SET BYTE 1
+BUS OUT 2
+WRITE SET BYTE 2
+READ TRANSLATE BIT 11
+READ SHIFT BYTE 1 OR 3
DATA REG 11 TGR
-BUS IN 6
-BUS IN 2
TO TRANSLATOR OR R-W REG IN WRITE OP
+BUS OUT 1
+WRITE SET BYTE 2
+BUS OUT 5
+WRITE SET BYTE 1
+R-W REG 5 TGR
+READ SHIFT BYTE 1 OR 3
DATA REG 10 TGR
A FIG.10
A FIG.8
-BUS IN 5
-BUS IN 1
TO R-W REG IN WRITE OP
+BUS OUT 4
+WRITE SET BYTE 1
+BUS OUT 0
+WRITE SET BYTE 2
+READ SHIFT BYTE 1 OR 3
+READ TRANSLATE BIT 9
DATA REG 9 TGR
-BUS IN 4
-BUS IN 0
TO TRANSLATOR OR R-W REG IN WRITE OP
+BUS OUT 7
+WRITE SET BYTE 3
+BUS OUT 3
+WRITE SET BYTE 1
+READ TRANSLATE BIT 8
+READ SHIFT BYTE 1 OR 3
DATA REG 8 TGR
-BUS IN 3
-BUS IN 7
TO TRANSLATOR OR R-W REG IN WRITE OP
+BUS OUT 2
+WRITE SET BYTE 1
+BUS OUT 6
+WRITE SET BYTE 3
+READ SHIFT BYTE 1 OR 3
+READ TRANSLATE BIT 7
+R-W REG 1 TGR
DATA REG 7 TGR
READ BYTE 4
-BUS IN 2
-BUS IN 6
TO TRANSLATOR OR R-W REG IN WRITE OP
+READ SHIFT BYTE 1 OR 3
+BUS OUT 3
+WRITE SET BYTE 3
+BUS OUT 1
+WRITE SET BYTE 1
+R-W REG 7 TGR
+READ SHIFT BYTE 2 OR 4
+READ SHIFT BYTE 1 OR 3
+TRANSLATE
DATA REG 6 TGR
-BUS IN 5
-BUS IN 1
TO R-W REG IN WRITE OP

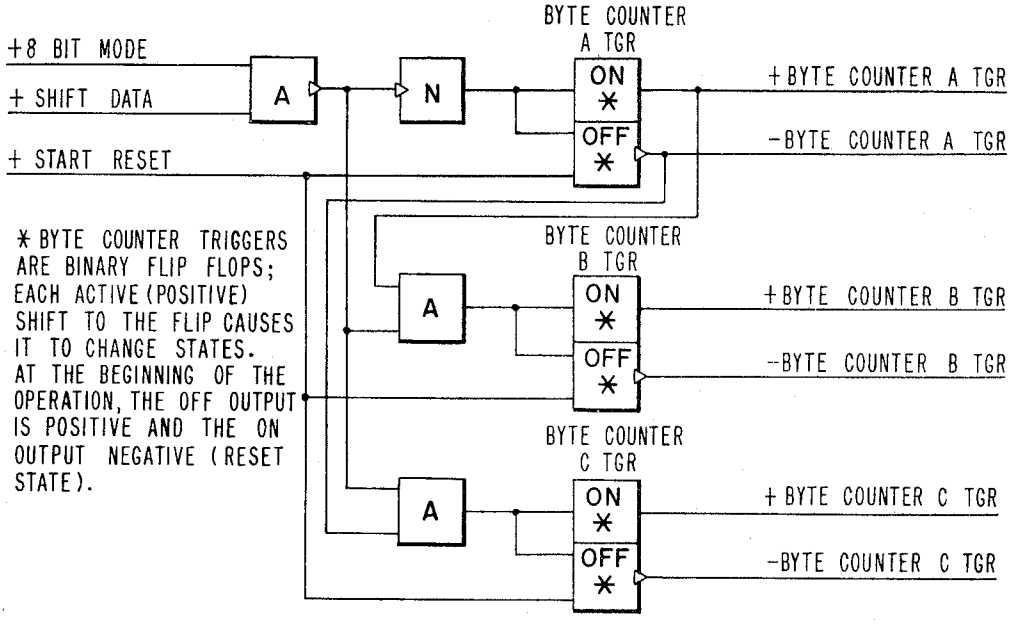
FIG. 150A




** "TAPE DEMAND" CAUSE TAPE CONTROL TO SIGNAL THE CHANNEL REQUESTING ANOTHER DATA BYTE OR INDICATING THAT A DATA BYTE IS AVAILABLE FOR CHANNEL.

SEE FIGS. 74, 75

FIG.150B


+8 BIT MODE
+ SHIFT DATA
+ START RESET
A
N
BYTE COUNTER A TGR
ON *
OFF *
+BYTE COUNTER A TGR
-BYTE COUNTER A TGR
BYTE COUNTER B TGR
+BYTE COUNTER B TGR
-BYTE COUNTER B TGR
BYTE COUNTER C TGR
+BYTE COUNTER C TGR
-BYTE COUNTER C TGR
* BYTE COUNTER TRIGGERS ARE BINARY FLIP FLOPS; EACH ACTIVE (POSITIVE) SHIFT TO THE FLIP CAUSES IT TO CHANGE STATES. AT THE BEGINNING OF THE OPERATION, THE OFF OUTPUT IS POSITIVE AND THE ON OUTPUT NEGATIVE (RESET STATE).

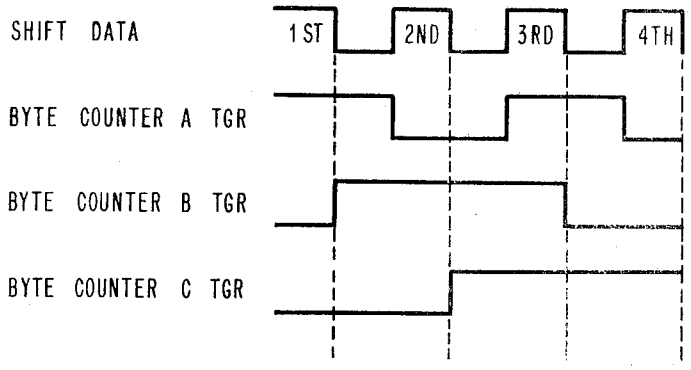

SHIFT DATA
1ST
2ND
3RD
4TH
BYTE COUNTER A TGR
BYTE COUNTER B TGR
BYTE COUNTER C TGR

FIG.151

WRITE

| DATA REGISTER POSITIONS | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BYTE 1 FROM INTERFACE (8-BIT CODE) | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | "WRITE SET BYTE 1" ACTIVE |
| BYTE 1 TO R-W REG (BCD) | | | | | | | B | A | 8 | 4 | 2 | 1 | "WRITE SHIFT BYTE 1 OR 3" ACTIVE |
| BYTE 2 FROM INTERFACE (8-BIT CODE) | 4 | 3 | 2 | 1 | | | | | 0 | 7 | 6 | 5 | "WRITE SET BYTE 2" ACTIVE |
| BYTE 2 TO R-W REG (BCD) | B | A | 8 | 4 | 2 | 1 | | | | | | | "WRITE SHIFT BYTE 2 OR 4" ACTIVE |
| BYTE 3 FROM INTERFACE (8-BIT CODE) | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | | | | | "WRITE SET BYTE 3" ACTIVE |
| BYTE 3 TO R-W REG (BCD) | | | | | | | B | A | 8 | 4 | 2 | 1 | "WRITE SHIFT BYTE 1 OR 3" ACTIVE |
| BYTE 4 TO R-W REG (BCD)** | B | A | 8 | 4 | 2 | 1 | | | | | | | "WRITE SHIFT BYTE 2 OR 4" ACTIVE |

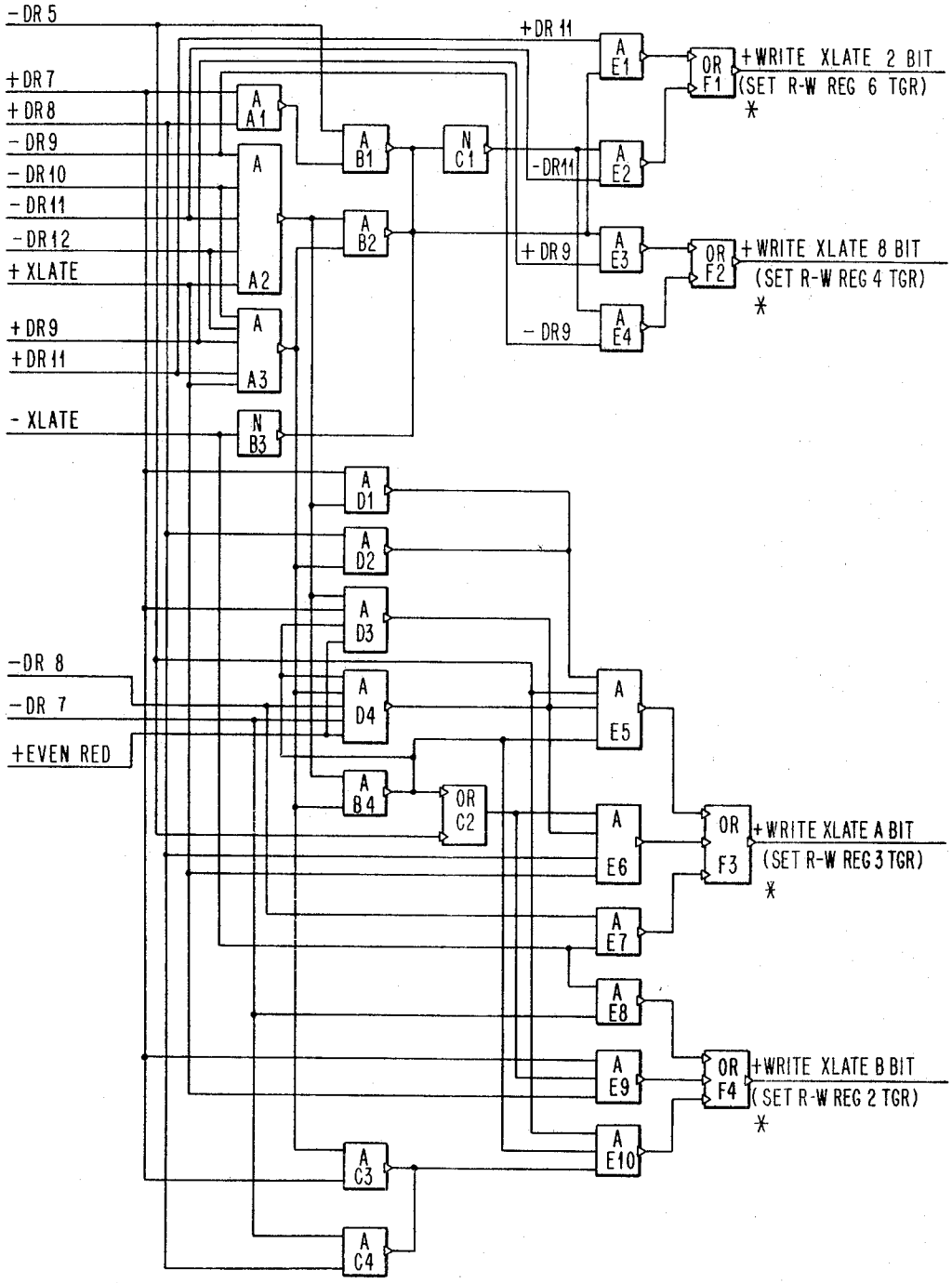
* DATA REG IS FULL

** DATA REG IS EMPTY

READ

| DATA REGISTER POSITIONS | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BYTE 1 FROM R-W REG (BCD) | | | | | | | B | A | 8 | 4 | 2 | 1 | "READ SHIFT BYTE 1 OR 3" ACTIVE |
| BYTE 2 FROM R-W REG (BCD)* | B | A | 8 | 4 | 2 | 1 | | | | | | | "READ SHIFT BYTE 2 OR 4" ACTIVE |
| BYTE 1 TO INTERFACE (8-BIT CODE) | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | "READ BYTE 2" ACTIVE |
| BYTE 3 FROM R-W REG (BCD) | | | | | | | B | A | 8 | 4 | 2 | 1 | "READ SHIFT BYTE 1 OR 3" ACTIVE |
| BYTE 2 TO INTERFACE (8-BIT CODE) | 4 | 5 | 6 | 7 | | | | | 0 | 1 | 2 | 3 | "READ BYTE 3" ACTIVE |
| BYTE 4 FROM R-W REG (BCD) | B | A | 8 | 4 | 2 | 1 | | | | | | | "READ SHIFT BYTE 2 OR 4" ACTIVE |
| BYTE 3 TO INTERFACE (8-BIT CODE)** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | | "READ BYTE 4" ACTIVE |

FIG. 152




* DATE REG 12 TGR SETS R-W REG 7 TGR; DATA REG 10 TGR SETS R-W REG 5 TGR

** SEE FIG. 21

FIG. 153

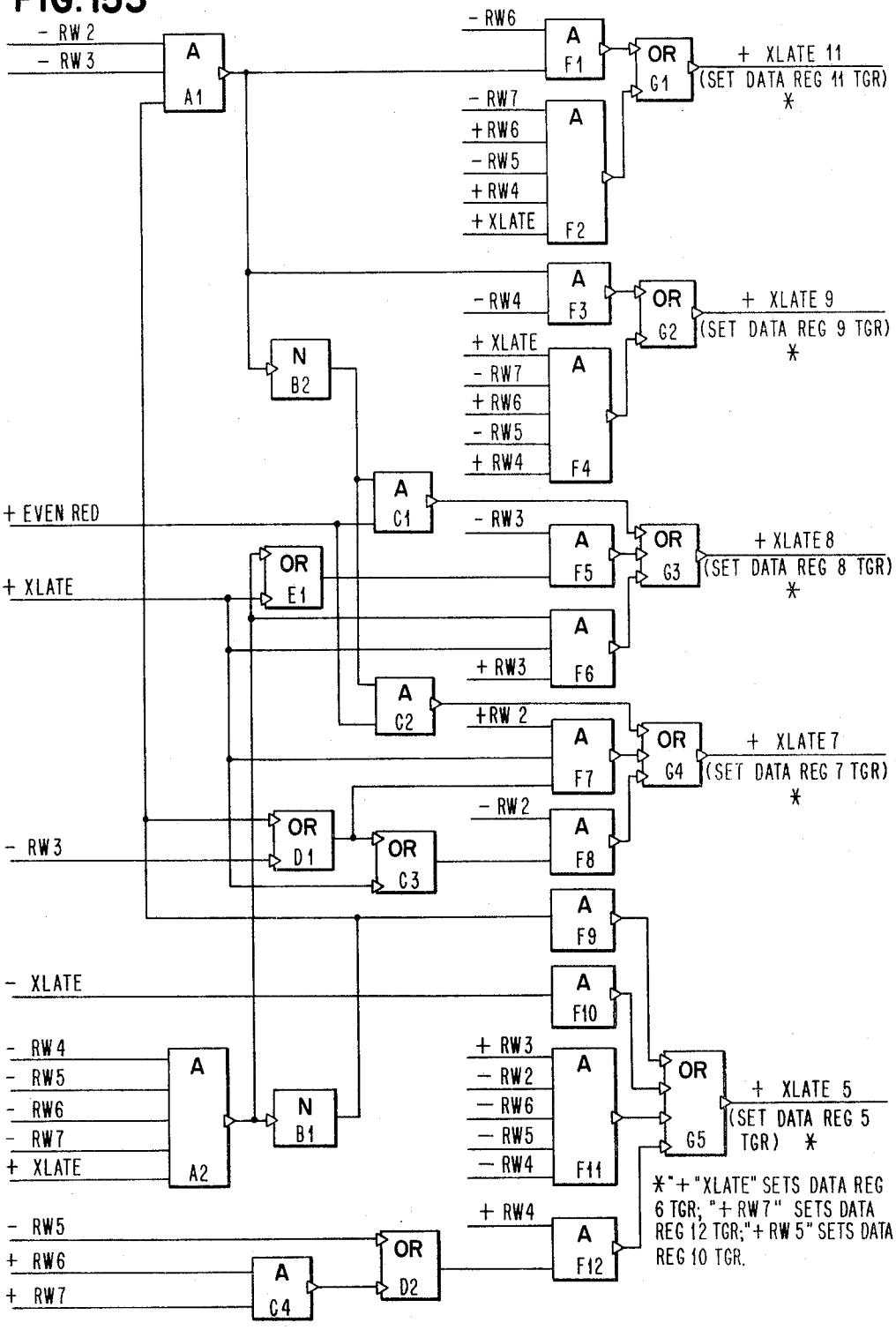

- RW 2
- RW 3
A A1
- RW6
A F1
OR G1
+ XLATE 11
(SET DATA REG 11 TGR)
*
- RW7
+ RW6
- RW5
+ RW4
+ XLATE
A F2
- RW4
A F3
OR G2
+ XLATE 9
(SET DATA REG 9 TGR)
*
N B2
+ XLATE
- RW7
+ RW6
- RW5
+ RW4
A F4
A C1
+ EVEN RED
- RW3
A F5
OR G3
+ XLATE 8
(SET DATA REG 8 TGR)
*
OR E1
+ XLATE
A F6
+ RW3
A C2
+RW 2
A F7
OR G4
+ XLATE 7
(SET DATA REG 7 TGR)
*
- RW 2
A F8
OR D1
OR C3
- RW3
A F9
- XLATE
A F10
- RW 4
- RW5
- RW6
- RW7
+ XLATE
A A2
N B1
+ RW3
- RW2
- RW6
- RW5
- RW4
A F11
OR G5
+ XLATE 5
(SET DATA REG 5 TGR) *
*"+"XLATE" SETS DATA REG 6 TGR; "+ RW7" SETS DATA REG 12 TGR;"+ RW 5" SETS DATA REG 10 TGR.
- RW5
+ RW6
+ RW7
A C4
OR D2
+ RW4
A F12

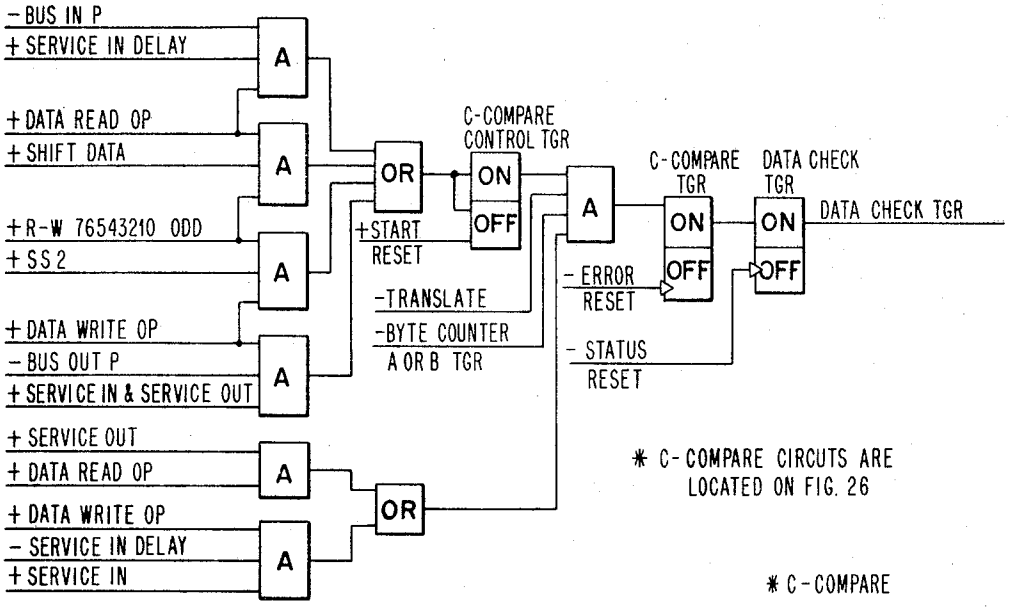
FIG.154A

| COLLATING SEQUENCE | GRAPHICS 8 BIT | GRAPHICS BCD | 8 BIT CODE 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | BCD B | A | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | BLANK | BLANK | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 01 | | | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 02 | ← | ¤ ) | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 03 | ( | [ | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 04 | + | ≺ | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| 05 | GM | GM | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 06 | & | &+ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 07 | $ | $ | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 08 | * | * | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 09 | ) | ] | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 10 | ; | ; | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 11 | MC | MC | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 12 | - | - | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 13 | / | / | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 14 | , | , | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 15 | % | %( | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 16 | WS | WS | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 17 | ↓ | \ | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| 18 | SM | SM | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 19 | ↄ | ↄ | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 20 | # | #= | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 21 | @ | @' | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 22 | ∇ | : | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 23 | = | > | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 24 | TM | TM | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 25 | ♂ | ♂ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 26 | A | A | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 27 | B | B | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 28 | C | C | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 29 | D | D | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 30 | E | E | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 31 | F | F | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 32 | G | G | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 33 | H | H | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 34 | I | I | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 35 | ō | ō | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 36 | J | J | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

FIG. 154B

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 37 | K | K | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 38 | L | L | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 39 | M | M | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 40 | N | N | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 41 | O | O | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 42 | P | P | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 43 | Q | Q | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 44 | R | R | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 45 | RM | RM | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 46 | S | S | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 47 | T | T | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 48 | U | U | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 49 | V | V | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 50 | W | W | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 51 | X | X | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 52 | Y | Y | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 53 | Z | Z | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 54 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 55 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 56 | 2 | 2 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 57 | 3 | 3 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 58 | 4 | 4 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 59 | 5 | 5 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 60 | 6 | 6 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 61 | 7 | 7 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 62 | 8 | 8 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 63 | 9 | 9 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

FIG. 155


-BUS IN P
+SERVICE IN DELAY
+DATA READ OP
+SHIFT DATA
+R-W 76543210 ODD
+SS2
+DATA WRITE OP
-BUS OUT P
+SERVICE IN & SERVICE OUT
+SERVICE OUT
+DATA READ OP
+DATA WRITE OP
-SERVICE IN DELAY
+SERVICE IN
A
OR
C-COMPARE CONTROL TGR
ON
OFF
+START RESET
-TRANSLATE
-BYTE COUNTER A OR B TGR
C-COMPARE TGR
DATA CHECK TGR
DATA CHECK TGR
-ERROR RESET
-STATUS RESET
* C-COMPARE CIRCUTS ARE LOCATED ON FIG. 26

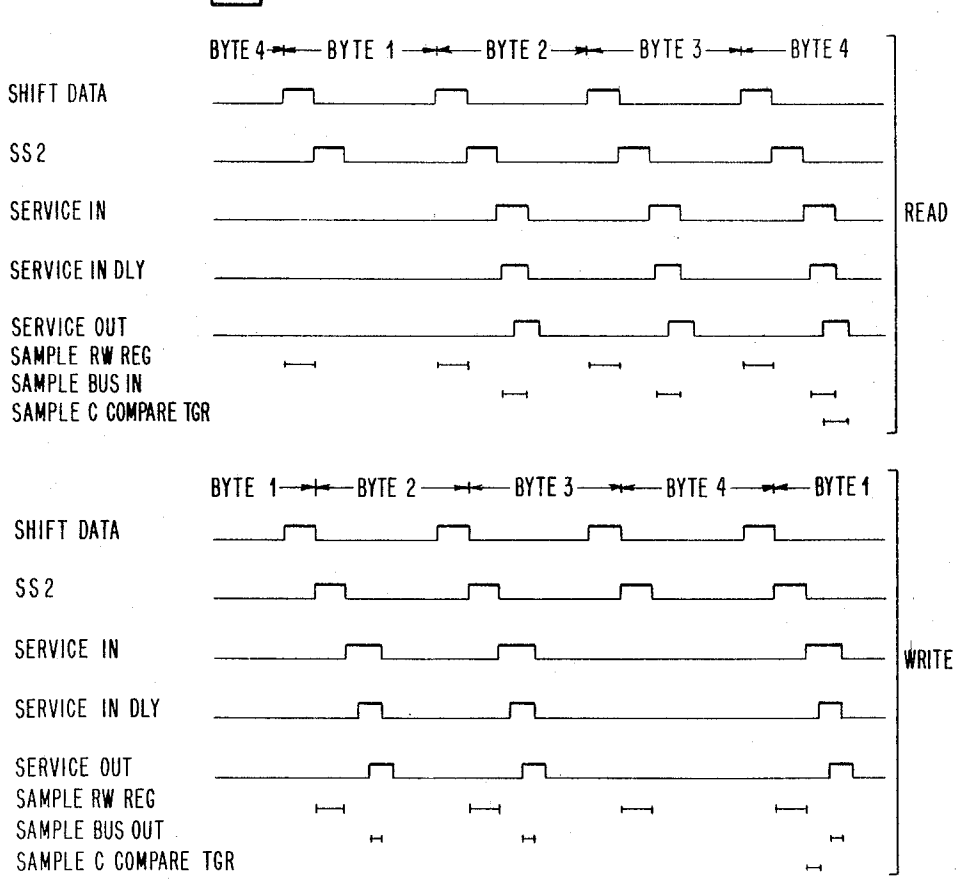
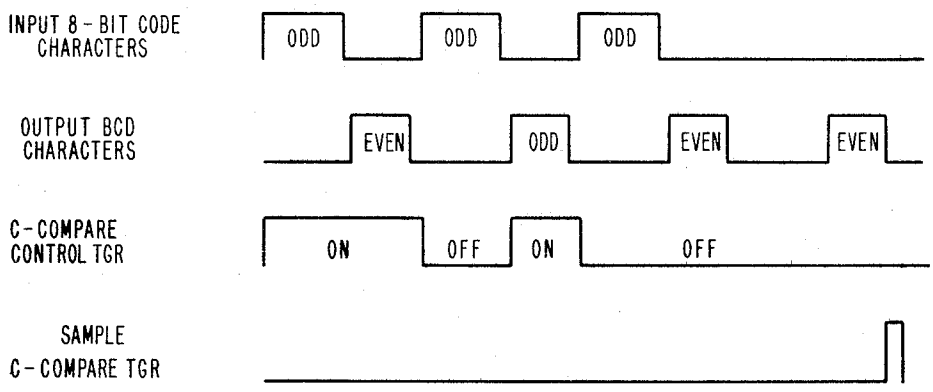

* C-COMPARE
BYTE 4
BYTE 1
BYTE 2
BYTE 3
BYTE 4
SHIFT DATA
SS2
SERVICE IN
SERVICE IN DLY
SERVICE OUT
SAMPLE RW REG
SAMPLE BUS IN
SAMPLE C COMPARE TGR
READ
BYTE 1
BYTE 2
BYTE 3
BYTE 4
BYTE 1
SHIFT DATA
SS2
SERVICE IN
SERVICE IN DLY
SERVICE OUT
SAMPLE RW REG
SAMPLE BUS OUT
SAMPLE C COMPARE TGR
WRITE

FIG. 156

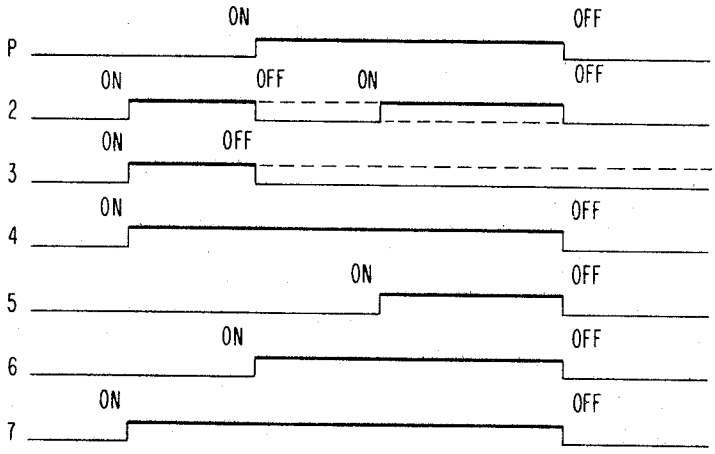


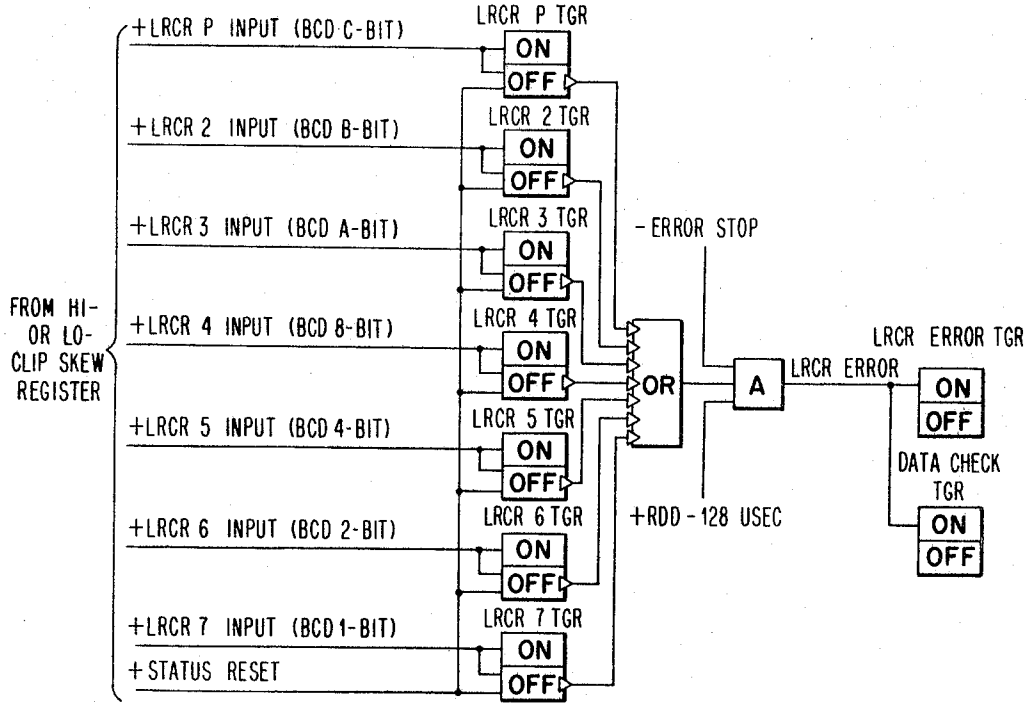

| | |
|---|---|
| 8-BIT CODE INPUT CHARACTERS (8-BIT BYTES) | 1 1 1 1 0 0 0 1 |
| | 1 1 0 0 1 1 0 1 |
| | 0 1 0 1 0 1 0 0 |
| BCD OUTPUT CHARACTERS (6-BIT BYTES) | 1 1 1 1 0 0 |
| | 0 1 1 1 0 0 |
| | 1 1 0 1 0 1 |
| | 0 1 0 1 0 0 |

C-COMPARE CIRCUTS IN A SPECIFIC WRITE OPERATION

FIG. 157

| | 1 | B | M | CHECK CHARACTER |
|---|---|---|---|---|
| P | 0 | 1 | 0 | 1 |
| 2 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 0 | 0 | 1 | 1 |
| 6 | 0 | 1 | 0 | 1 |
| 7 | 1 | 0 | 0 | 1 |



ERROR DETECTION AND CORRECTION SYSTEM

Filed April 6, 1964 316 Sheets-Sheet 237

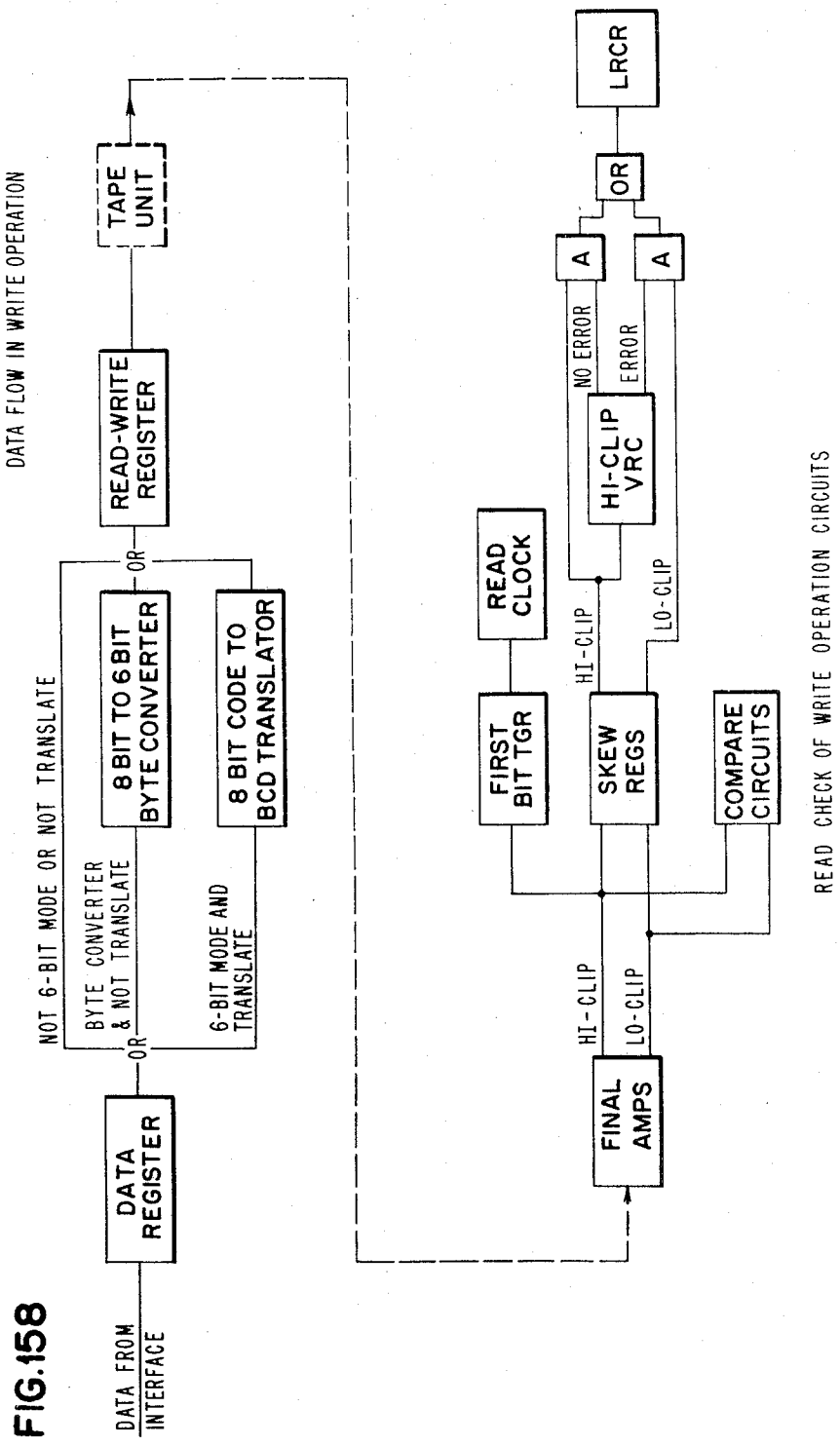



FIG. 158

FIG. 159A

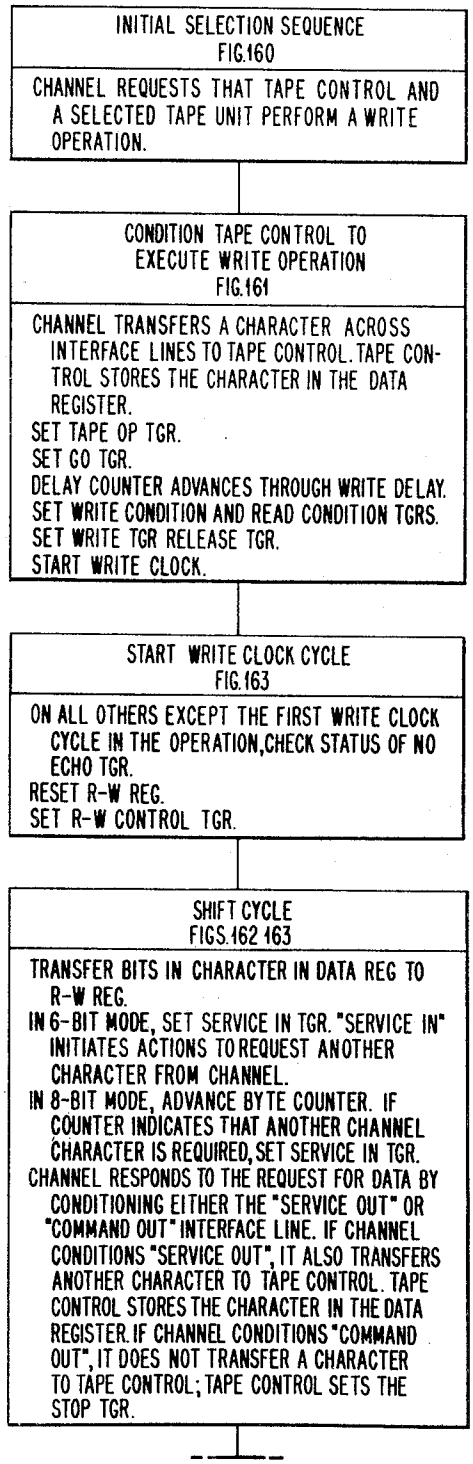

INITIAL SELECTION SEQUENCE
FIG.160
CHANNEL REQUESTS THAT TAPE CONTROL AND A SELECTED TAPE UNIT PERFORM A WRITE OPERATION.
CONDITION TAPE CONTROL TO EXECUTE WRITE OPERATION
FIG.161
CHANNEL TRANSFERS A CHARACTER ACROSS INTERFACE LINES TO TAPE CONTROL. TAPE CONTROL STORES THE CHARACTER IN THE DATA REGISTER.
SET TAPE OP TGR.
SET GO TGR.
DELAY COUNTER ADVANCES THROUGH WRITE DELAY.
SET WRITE CONDITION AND READ CONDITION TGRS.
SET WRITE TGR RELEASE TGR.
START WRITE CLOCK.
START WRITE CLOCK CYCLE
FIG.163
ON ALL OTHERS EXCEPT THE FIRST WRITE CLOCK CYCLE IN THE OPERATION, CHECK STATUS OF NO ECHO TGR.
RESET R-W REG.
SET R-W CONTROL TGR.
SHIFT CYCLE
FIGS.162 163
TRANSFER BITS IN CHARACTER IN DATA REG TO R-W REG.
IN 6-BIT MODE, SET SERVICE IN TGR. "SERVICE IN" INITIATES ACTIONS TO REQUEST ANOTHER CHARACTER FROM CHANNEL.
IN 8-BIT MODE, ADVANCE BYTE COUNTER. IF COUNTER INDICATES THAT ANOTHER CHANNEL CHARACTER IS REQUIRED, SET SERVICE IN TGR.
CHANNEL RESPONDS TO THE REQUEST FOR DATA BY CONDITIONING EITHER THE "SERVICE OUT" OR "COMMAND OUT" INTERFACE LINE. IF CHANNEL CONDITIONS "SERVICE OUT", IT ALSO TRANSFERS ANOTHER CHARACTER TO TAPE CONTROL. TAPE CONTROL STORES THE CHARACTER IN THE DATA REGISTER. IF CHANNEL CONDITIONS "COMMAND OUT", IT DOES NOT TRANSFER A CHARACTER TO TAPE CONTROL; TAPE CONTROL SETS THE STOP TGR.

FIG. 159B

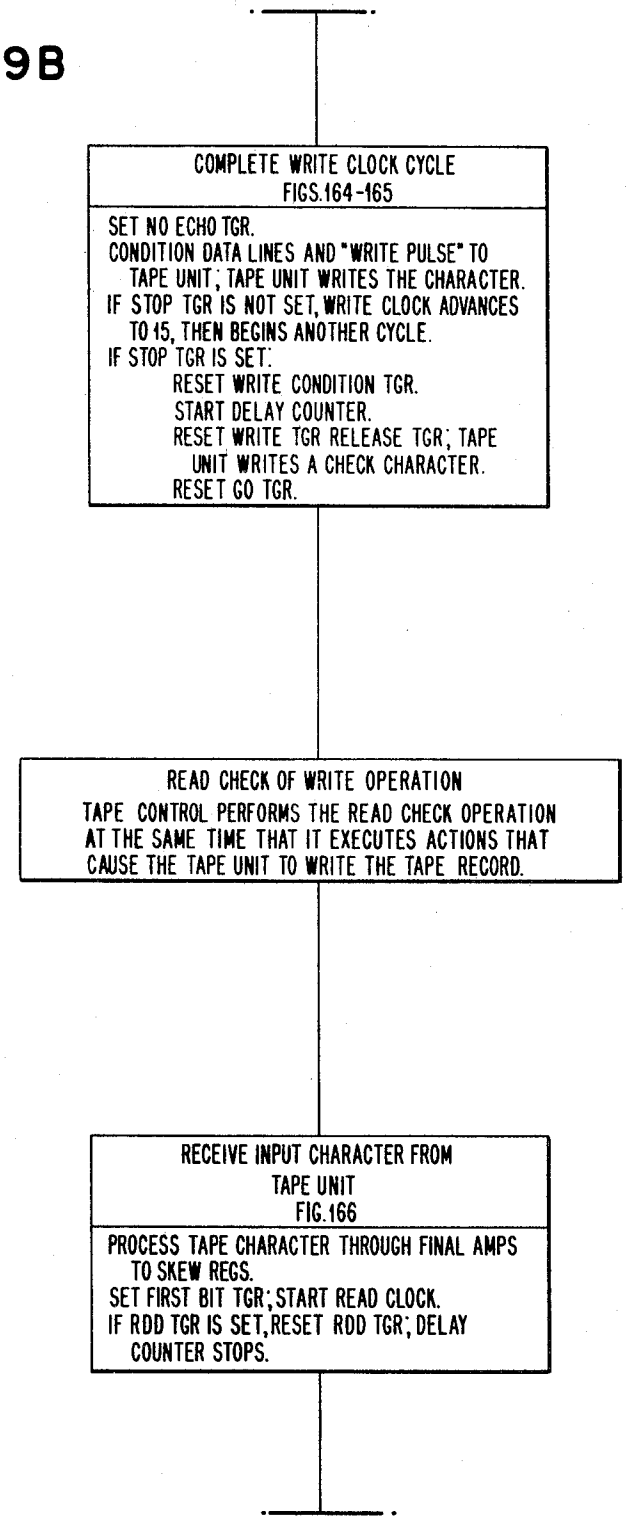

COMPLETE WRITE CLOCK CYCLE
FIGS. 164-165
SET NO ECHO TGR.
CONDITION DATA LINES AND "WRITE PULSE" TO TAPE UNIT; TAPE UNIT WRITES THE CHARACTER.
IF STOP TGR IS NOT SET, WRITE CLOCK ADVANCES TO 15, THEN BEGINS ANOTHER CYCLE.
IF STOP TGR IS SET:
RESET WRITE CONDITION TGR.
START DELAY COUNTER.
RESET WRITE TGR RELEASE TGR; TAPE UNIT WRITES A CHECK CHARACTER.
RESET GO TGR.
READ CHECK OF WRITE OPERATION
TAPE CONTROL PERFORMS THE READ CHECK OPERATION AT THE SAME TIME THAT IT EXECUTES ACTIONS THAT CAUSE THE TAPE UNIT TO WRITE THE TAPE RECORD.
RECEIVE INPUT CHARACTER FROM TAPE UNIT
FIG. 166
PROCESS TAPE CHARACTER THROUGH FINAL AMPS TO SKEW REGS.
SET FIRST BIT TGR; START READ CLOCK.
IF RDD TGR IS SET, RESET RDD TGR; DELAY COUNTER STOPS.

FIG. 159C

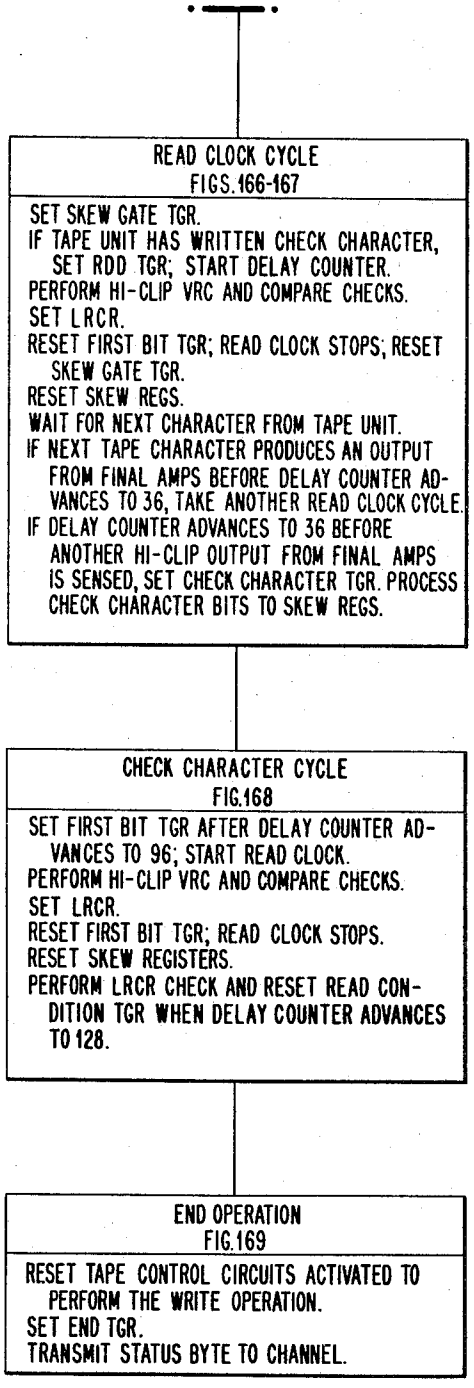

READ CLOCK CYCLE
FIGS. 166-167
SET SKEW GATE TGR.
IF TAPE UNIT HAS WRITTEN CHECK CHARACTER, SET RDD TGR; START DELAY COUNTER.
PERFORM HI-CLIP VRC AND COMPARE CHECKS.
SET LRCR.
RESET FIRST BIT TGR; READ CLOCK STOPS; RESET SKEW GATE TGR.
RESET SKEW REGS.
WAIT FOR NEXT CHARACTER FROM TAPE UNIT.
IF NEXT TAPE CHARACTER PRODUCES AN OUTPUT FROM FINAL AMPS BEFORE DELAY COUNTER ADVANCES TO 36, TAKE ANOTHER READ CLOCK CYCLE.
IF DELAY COUNTER ADVANCES TO 36 BEFORE ANOTHER HI-CLIP OUTPUT FROM FINAL AMPS IS SENSED, SET CHECK CHARACTER TGR. PROCESS CHECK CHARACTER BITS TO SKEW REGS.
CHECK CHARACTER CYCLE
FIG. 168
SET FIRST BIT TGR AFTER DELAY COUNTER ADVANCES TO 96; START READ CLOCK.
PERFORM HI-CLIP VRC AND COMPARE CHECKS.
SET LRCR.
RESET FIRST BIT TGR; READ CLOCK STOPS.
RESET SKEW REGISTERS.
PERFORM LRCR CHECK AND RESET READ CONDITION TGR WHEN DELAY COUNTER ADVANCES TO 128.
END OPERATION
FIG. 169
RESET TAPE CONTROL CIRCUITS ACTIVATED TO PERFORM THE WRITE OPERATION.
SET END TGR.
TRANSMIT STATUS BYTE TO CHANNEL.

FIG. 160A

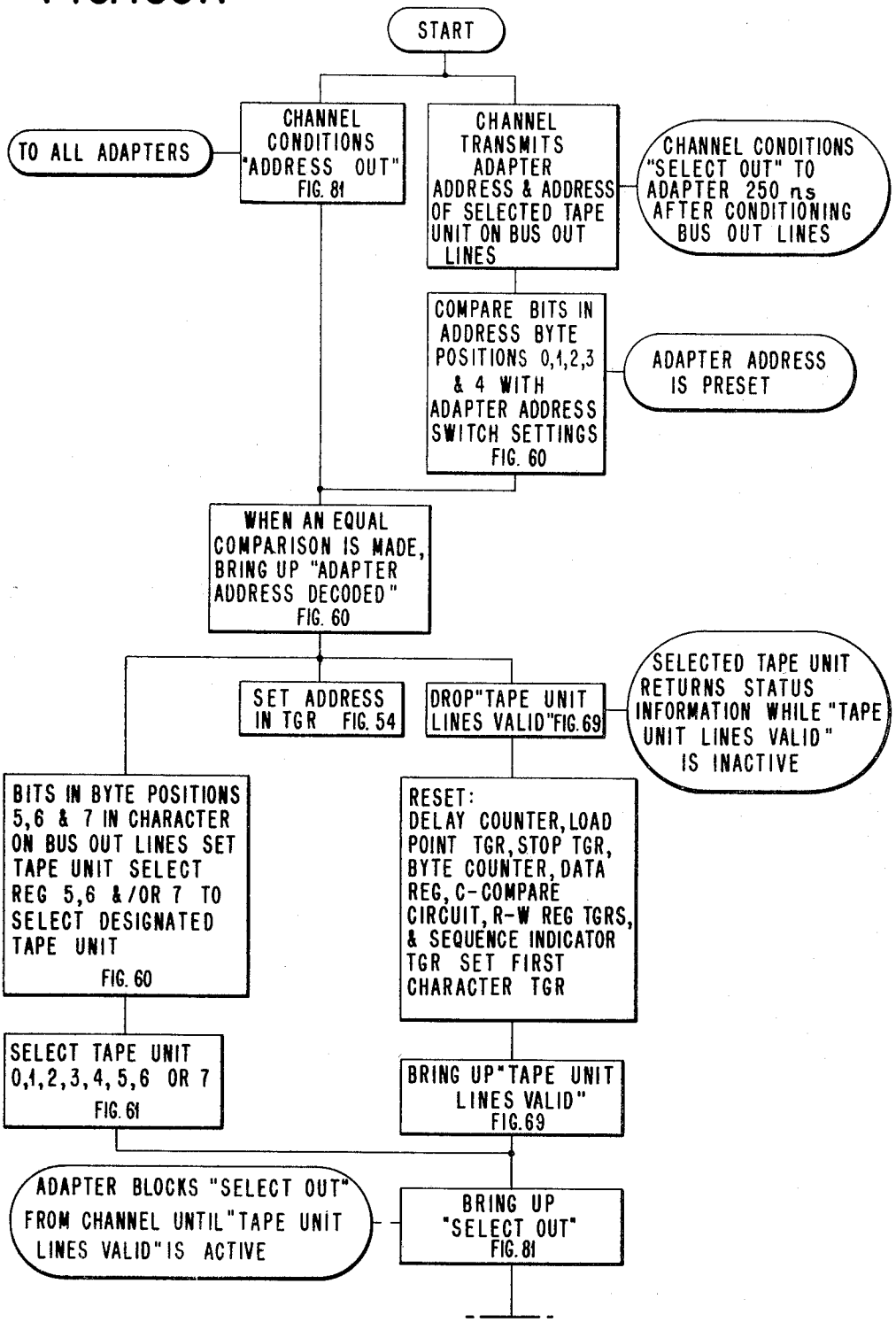

START
TO ALL ADAPTERS
CHANNEL CONDITIONS "ADDRESS OUT" FIG. 81
CHANNEL TRANSMITS ADAPTER ADDRESS & ADDRESS OF SELECTED TAPE UNIT ON BUS OUT LINES
CHANNEL CONDITIONS "SELECT OUT" TO ADAPTER 250 ns AFTER CONDITIONING BUS OUT LINES
COMPARE BITS IN ADDRESS BYTE POSITIONS 0,1,2,3 & 4 WITH ADAPTER ADDRESS SWITCH SETTINGS FIG. 60
ADAPTER ADDRESS IS PRESET
WHEN AN EQUAL COMPARISON IS MADE, BRING UP "ADAPTER ADDRESS DECODED" FIG. 60
SET ADDRESS IN TGR FIG. 54
DROP "TAPE UNIT LINES VALID" FIG. 69
SELECTED TAPE UNIT RETURNS STATUS INFORMATION WHILE "TAPE UNIT LINES VALID" IS INACTIVE
BITS IN BYTE POSITIONS 5,6 & 7 IN CHARACTER ON BUS OUT LINES SET TAPE UNIT SELECT REG 5,6 &/OR 7 TO SELECT DESIGNATED TAPE UNIT FIG. 60
RESET: DELAY COUNTER, LOAD POINT TGR, STOP TGR, BYTE COUNTER, DATA REG, C-COMPARE CIRCUIT, R-W REG TGRS, & SEQUENCE INDICATOR TGR SET FIRST CHARACTER TGR
SELECT TAPE UNIT 0,1,2,3,4,5,6 OR 7 FIG. 61
BRING UP "TAPE UNIT LINES VALID" FIG. 69
ADAPTER BLOCKS "SELECT OUT" FROM CHANNEL UNTIL "TAPE UNIT LINES VALID" IS ACTIVE
BRING UP "SELECT OUT" FIG. 81

FIG. 160B

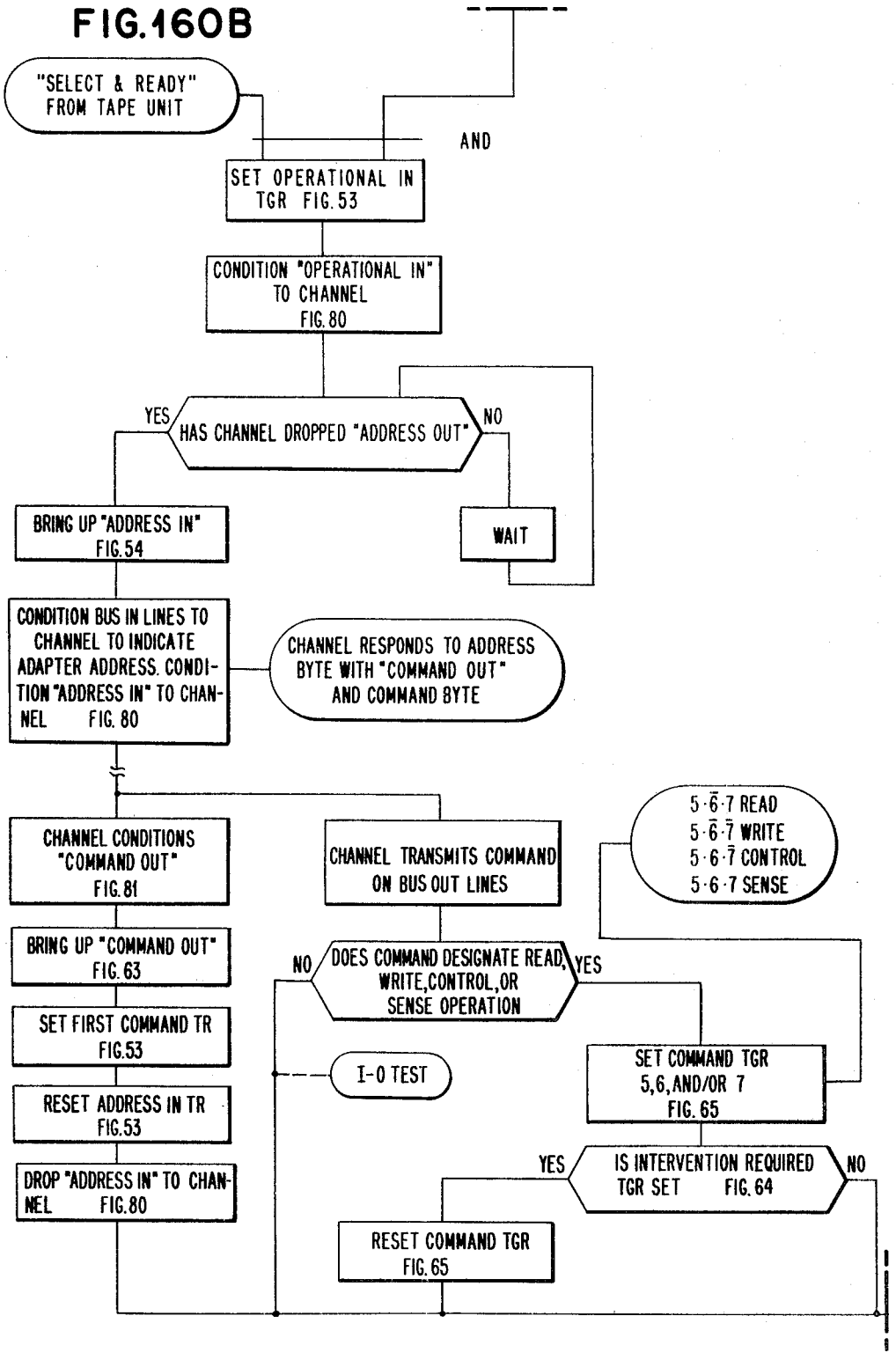

"SELECT & READY" FROM TAPE UNIT
AND
SET OPERATIONAL IN TGR FIG. 53
CONDITION "OPERATIONAL IN" TO CHANNEL FIG. 80
YES
HAS CHANNEL DROPPED "ADDRESS OUT"
NO
BRING UP "ADDRESS IN" FIG. 54
WAIT
CONDITION BUS IN LINES TO CHANNEL TO INDICATE ADAPTER ADDRESS. CONDITION "ADDRESS IN" TO CHANNEL FIG. 80
CHANNEL RESPONDS TO ADDRESS BYTE WITH "COMMAND OUT" AND COMMAND BYTE
CHANNEL CONDITIONS "COMMAND OUT" FIG. 81
CHANNEL TRANSMITS COMMAND ON BUS OUT LINES
5·6̄·7 READ
5·6̄·7̄ WRITE
5·6·7̄ CONTROL
5·6·7 SENSE
BRING UP "COMMAND OUT" FIG. 63
NO
DOES COMMAND DESIGNATE READ, WRITE, CONTROL, OR SENSE OPERATION
YES
SET FIRST COMMAND TR FIG. 53
I-O TEST
SET COMMAND TGR 5,6, AND/OR 7 FIG. 65
RESET ADDRESS IN TR FIG. 53
YES
IS INTERVENTION REQUIRED TGR SET FIG. 64
NO
DROP "ADDRESS IN" TO CHANNEL FIG. 80
RESET COMMAND TGR FIG. 65

FIG.160C

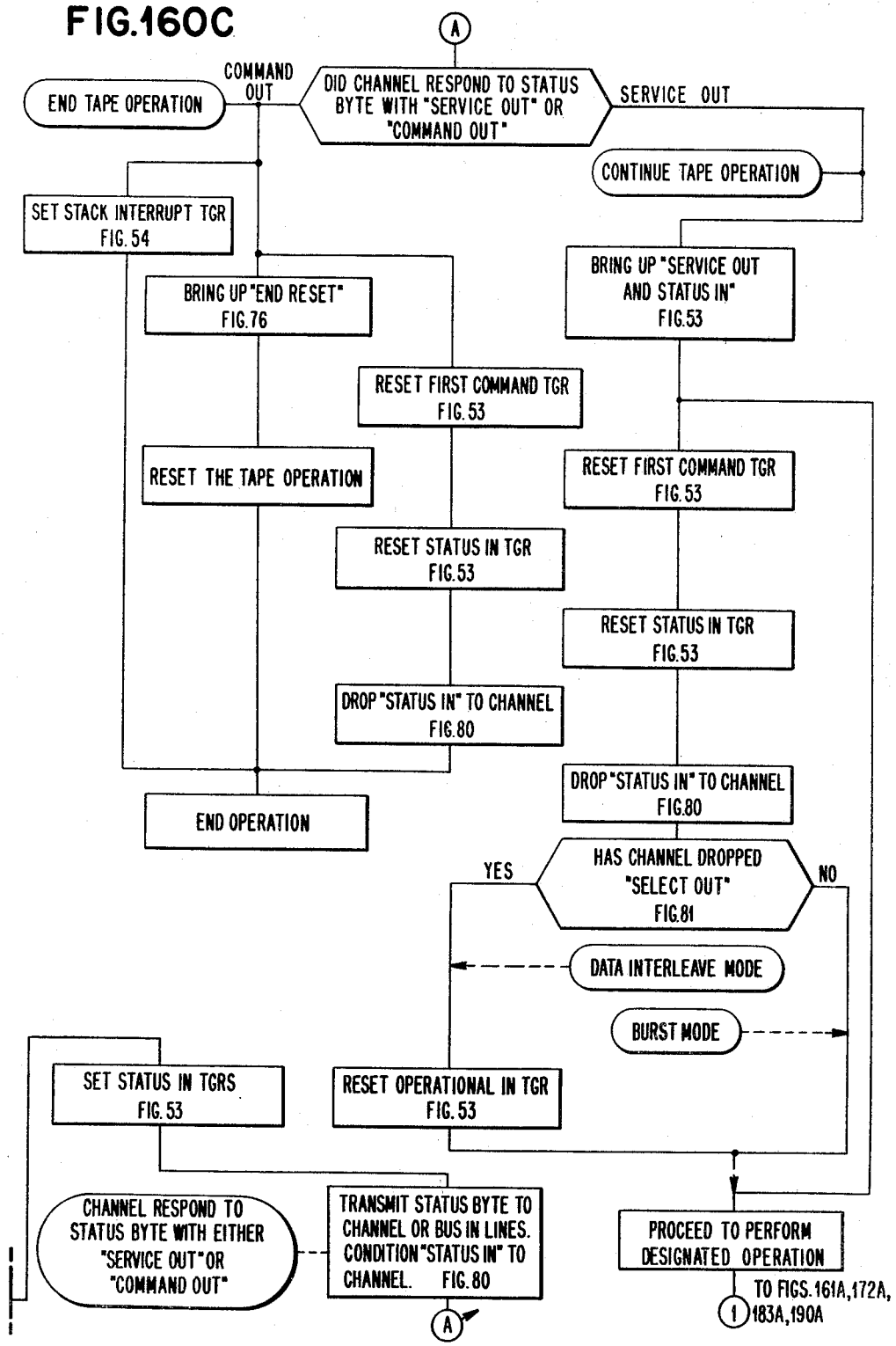

A
END TAPE OPERATION
COMMAND OUT
DID CHANNEL RESPOND TO STATUS BYTE WITH "SERVICE OUT" OR "COMMAND OUT"
SERVICE OUT
CONTINUE TAPE OPERATION
SET STACK INTERRUPT TGR FIG. 54
BRING UP "END RESET" FIG. 76
BRING UP "SERVICE OUT AND STATUS IN" FIG. 53
RESET FIRST COMMAND TGR FIG. 53
RESET THE TAPE OPERATION
RESET FIRST COMMAND TGR FIG. 53
RESET STATUS IN TGR FIG. 53
RESET STATUS IN TGR FIG. 53
DROP "STATUS IN" TO CHANNEL FIG. 80
DROP "STATUS IN" TO CHANNEL FIG. 80
END OPERATION
YES
HAS CHANNEL DROPPED "SELECT OUT" FIG. 81
NO
DATA INTERLEAVE MODE
BURST MODE
SET STATUS IN TGRS FIG. 53
RESET OPERATIONAL IN TGR FIG. 53
CHANNEL RESPOND TO STATUS BYTE WITH EITHER "SERVICE OUT" OR "COMMAND OUT"
TRANSMIT STATUS BYTE TO CHANNEL OR BUS IN LINES. CONDITION "STATUS IN" TO CHANNEL. FIG. 80
PROCEED TO PERFORM DESIGNATED OPERATION
TO FIGS. 161A, 172A, 183A, 190A
1
A

FIG. 161A

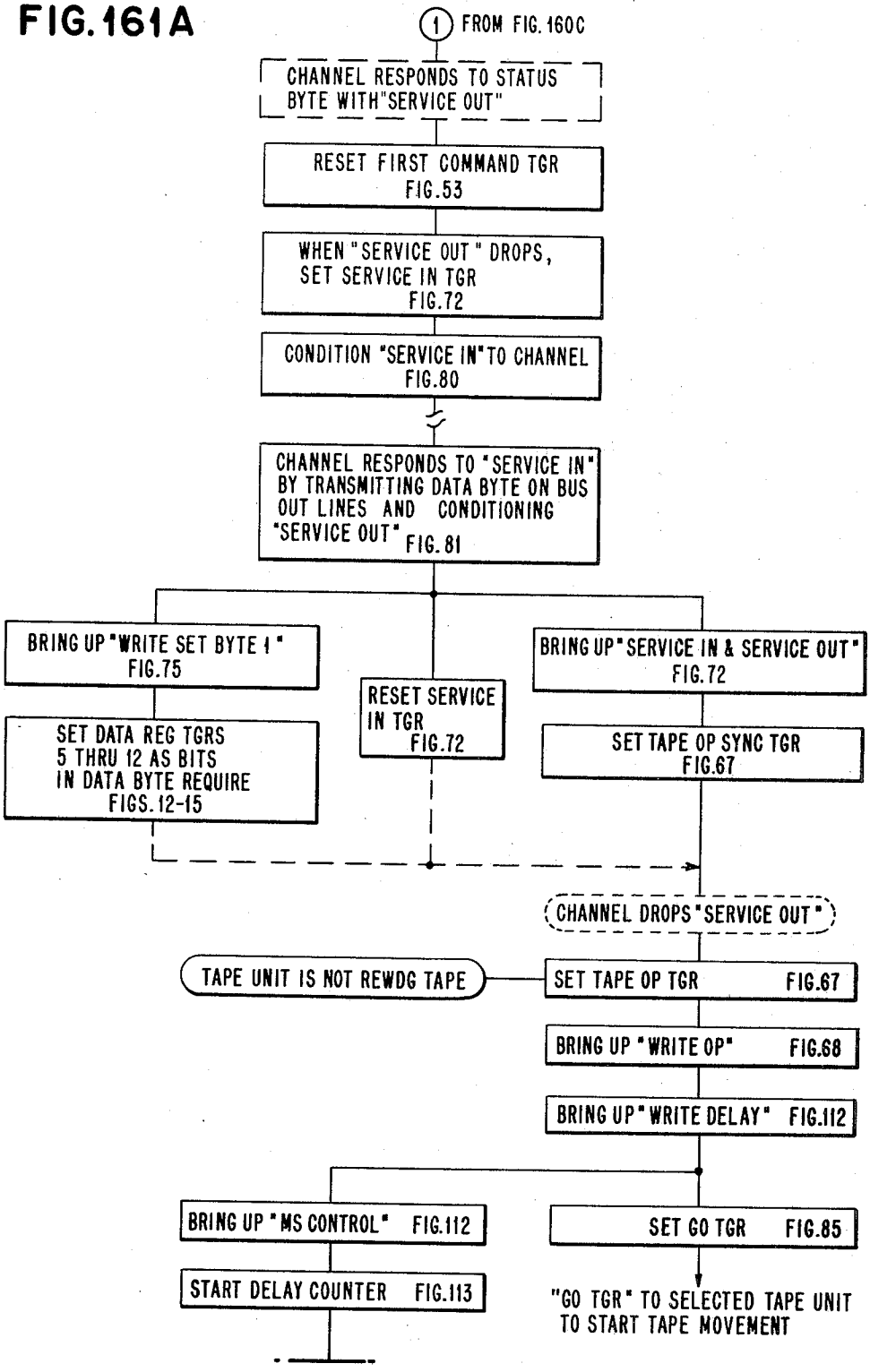

1 FROM FIG. 160C
CHANNEL RESPONDS TO STATUS BYTE WITH "SERVICE OUT"
RESET FIRST COMMAND TGR FIG. 53
WHEN "SERVICE OUT" DROPS, SET SERVICE IN TGR FIG. 72
CONDITION "SERVICE IN" TO CHANNEL FIG. 80
CHANNEL RESPONDS TO "SERVICE IN" BY TRANSMITTING DATA BYTE ON BUS OUT LINES AND CONDITIONING "SERVICE OUT" FIG. 81
BRING UP "WRITE SET BYTE 1" FIG. 75
SET DATA REG TGRS 5 THRU 12 AS BITS IN DATA BYTE REQUIRE FIGS. 12-15
RESET SERVICE IN TGR FIG. 72
BRING UP "SERVICE IN & SERVICE OUT" FIG. 72
SET TAPE OP SYNC TGR FIG. 67
CHANNEL DROPS "SERVICE OUT"
TAPE UNIT IS NOT REWDG TAPE
SET TAPE OP TGR FIG. 67
BRING UP "WRITE OP" FIG. 68
BRING UP "WRITE DELAY" FIG. 112
BRING UP "MS CONTROL" FIG. 112
START DELAY COUNTER FIG. 113
SET GO TGR FIG. 85
"GO TGR" TO SELECTED TAPE UNIT TO START TAPE MOVEMENT

FIG.161B

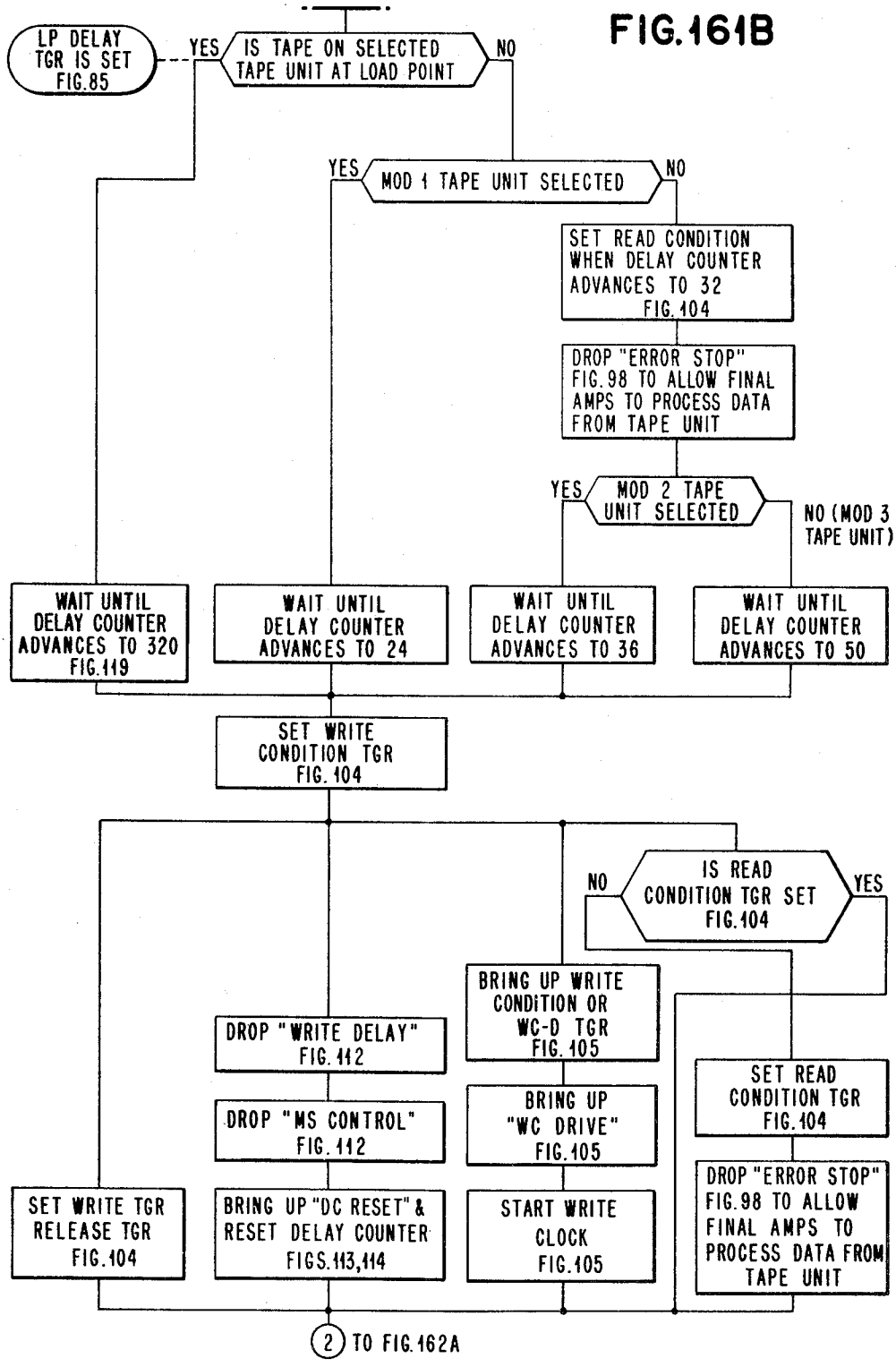

LP DELAY TGR IS SET FIG.85
IS TAPE ON SELECTED TAPE UNIT AT LOAD POINT
YES
NO
MOD 1 TAPE UNIT SELECTED
SET READ CONDITION WHEN DELAY COUNTER ADVANCES TO 32 FIG.104
DROP "ERROR STOP" FIG.98 TO ALLOW FINAL AMPS TO PROCESS DATA FROM TAPE UNIT
MOD 2 TAPE UNIT SELECTED
NO (MOD 3 TAPE UNIT)
WAIT UNTIL DELAY COUNTER ADVANCES TO 320 FIG.119
WAIT UNTIL DELAY COUNTER ADVANCES TO 24
WAIT UNTIL DELAY COUNTER ADVANCES TO 36
WAIT UNTIL DELAY COUNTER ADVANCES TO 50
SET WRITE CONDITION TGR FIG.104
IS READ CONDITION TGR SET FIG.104
BRING UP WRITE CONDITION OR WC-D TGR FIG.105
DROP "WRITE DELAY" FIG.112
DROP "MS CONTROL" FIG.112
BRING UP "WC DRIVE" FIG.105
SET READ CONDITION TGR FIG.104
DROP "ERROR STOP" FIG.98 TO ALLOW FINAL AMPS TO PROCESS DATA FROM TAPE UNIT
SET WRITE TGR RELEASE TGR FIG.104
BRING UP "DC RESET" & RESET DELAY COUNTER FIGS.113,114
START WRITE CLOCK FIG.105
2 TO FIG.162A

FIG.162A

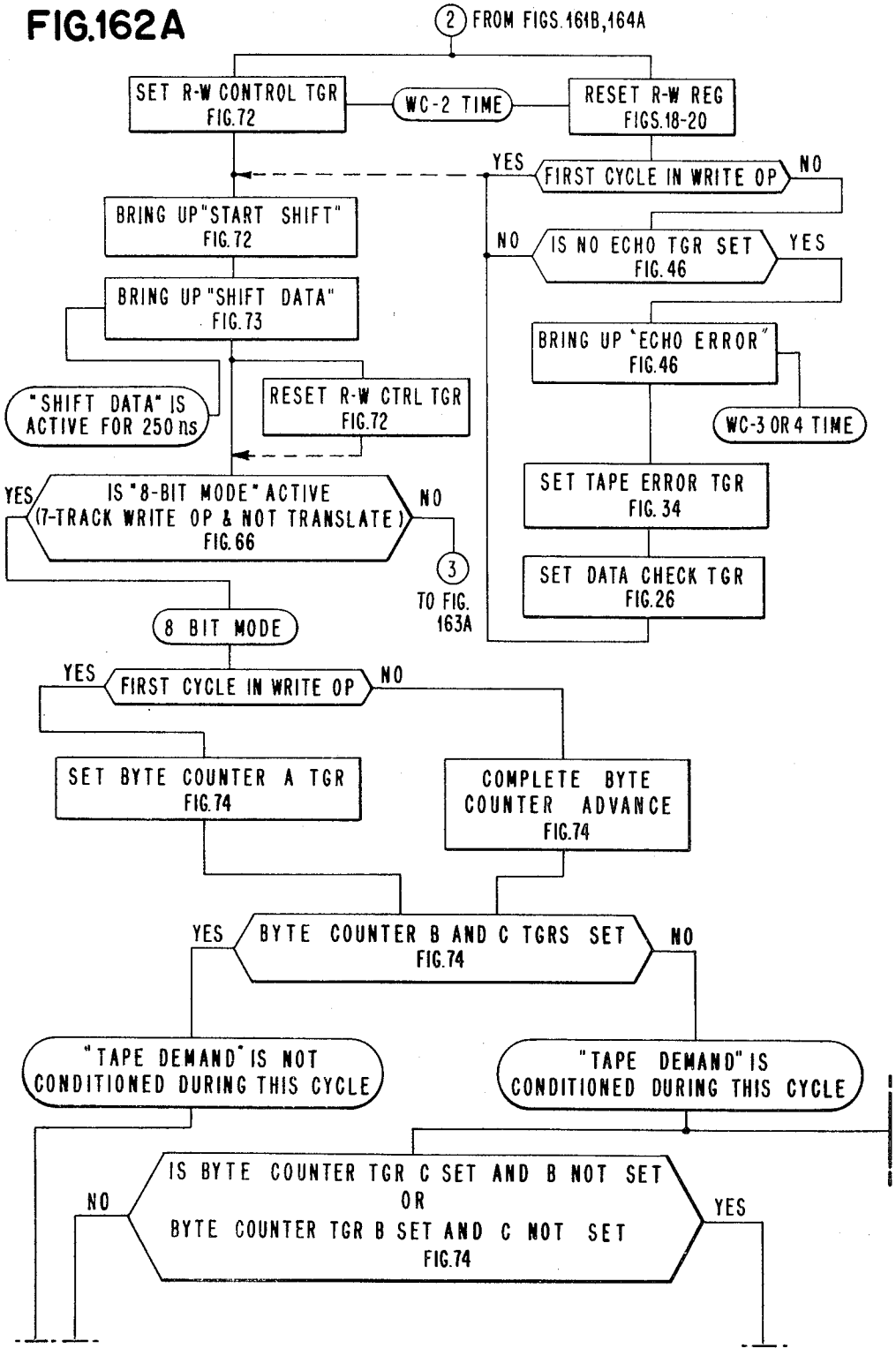

2 FROM FIGS.161B,164A
SET R-W CONTROL TGR FIG.72
WC-2 TIME
RESET R-W REG FIGS.18-20
YES
FIRST CYCLE IN WRITE OP
NO
BRING UP "START SHIFT" FIG.72
NO
IS NO ECHO TGR SET FIG.46
YES
BRING UP "SHIFT DATA" FIG.73
BRING UP "ECHO ERROR" FIG.46
"SHIFT DATA" IS ACTIVE FOR 250 ns
RESET R-W CTRL TGR FIG.72
WC-3 OR 4 TIME
YES
IS "8-BIT MODE" ACTIVE (7-TRACK WRITE OP & NOT TRANSLATE) FIG.66
NO
SET TAPE ERROR TGR FIG.34
3
TO FIG. 163A
SET DATA CHECK TGR FIG.26
8 BIT MODE
YES
FIRST CYCLE IN WRITE OP
NO
SET BYTE COUNTER A TGR FIG.74
COMPLETE BYTE COUNTER ADVANCE FIG.74
YES
BYTE COUNTER B AND C TGRS SET FIG.74
NO
"TAPE DEMAND" IS NOT CONDITIONED DURING THIS CYCLE
"TAPE DEMAND" IS CONDITIONED DURING THIS CYCLE
NO
IS BYTE COUNTER TGR C SET AND B NOT SET OR BYTE COUNTER TGR B SET AND C NOT SET FIG.74
YES

FIG.162B

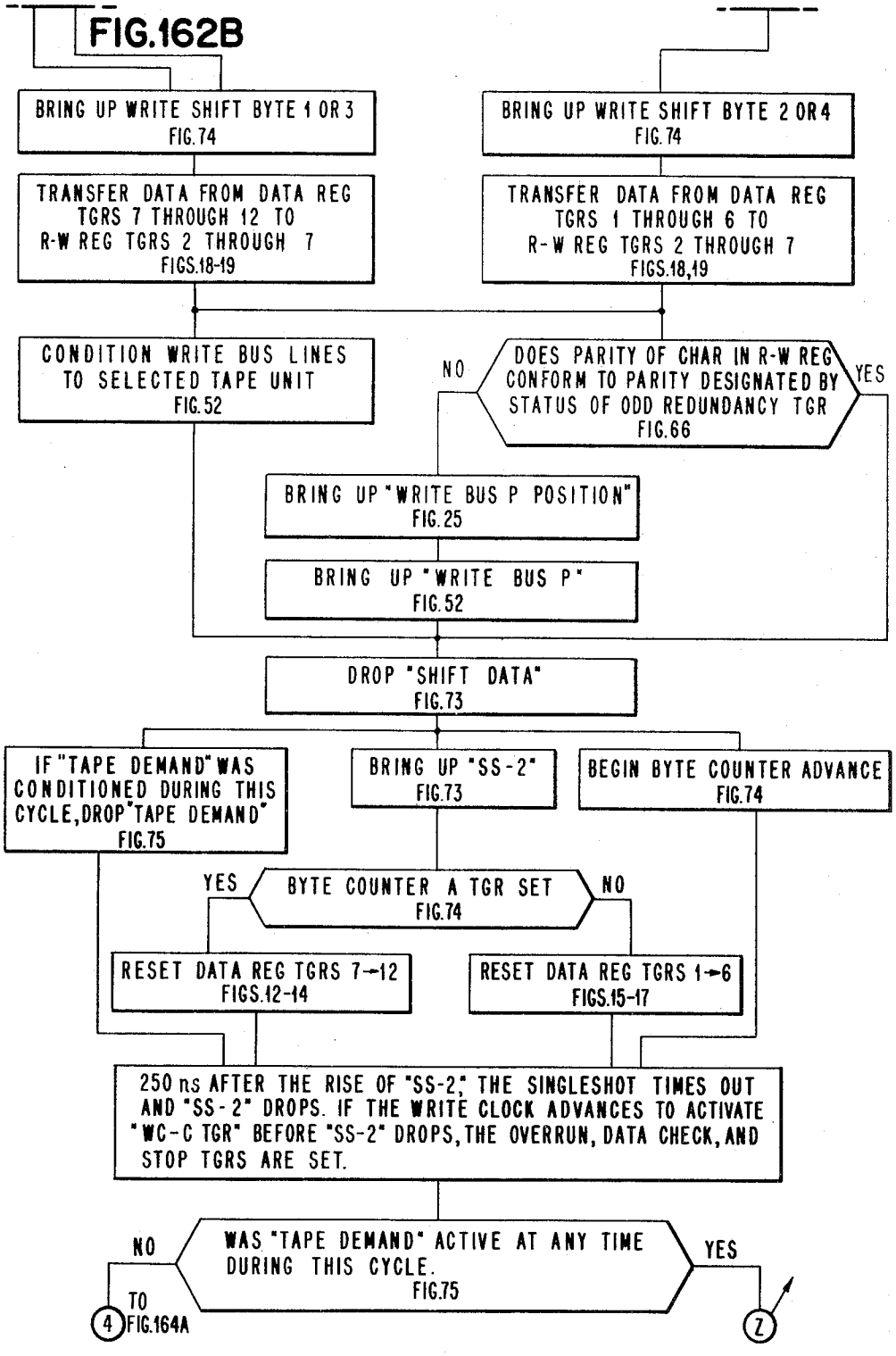

BRING UP WRITE SHIFT BYTE 1 OR 3
FIG.74
BRING UP WRITE SHIFT BYTE 2 OR 4
FIG.74
TRANSFER DATA FROM DATA REG TGRS 7 THROUGH 12 TO R-W REG TGRS 2 THROUGH 7
FIGS.18-19
TRANSFER DATA FROM DATA REG TGRS 1 THROUGH 6 TO R-W REG TGRS 2 THROUGH 7
FIGS.18,19
CONDITION WRITE BUS LINES TO SELECTED TAPE UNIT
FIG.52
NO
DOES PARITY OF CHAR IN R-W REG CONFORM TO PARITY DESIGNATED BY STATUS OF ODD REDUNDANCY TGR
FIG.66
YES
BRING UP "WRITE BUS P POSITION"
FIG.25
BRING UP "WRITE BUS P"
FIG.52
DROP "SHIFT DATA"
FIG.73
IF "TAPE DEMAND" WAS CONDITIONED DURING THIS CYCLE, DROP "TAPE DEMAND"
FIG.75
BRING UP "SS-2"
FIG.73
BEGIN BYTE COUNTER ADVANCE
FIG.74
YES
BYTE COUNTER A TGR SET
FIG.74
NO
RESET DATA REG TGRS 7→12
FIGS.12-14
RESET DATA REG TGRS 1→6
FIGS.15-17
250 ns AFTER THE RISE OF "SS-2," THE SINGLESHOT TIMES OUT AND "SS-2" DROPS. IF THE WRITE CLOCK ADVANCES TO ACTIVATE "WC-C TGR" BEFORE "SS-2" DROPS, THE OVERRUN, DATA CHECK, AND STOP TGRS ARE SET.
NO
WAS "TAPE DEMAND" ACTIVE AT ANY TIME DURING THIS CYCLE.
FIG.75
YES
TO
4
FIG.164A
Z

FIG.162C

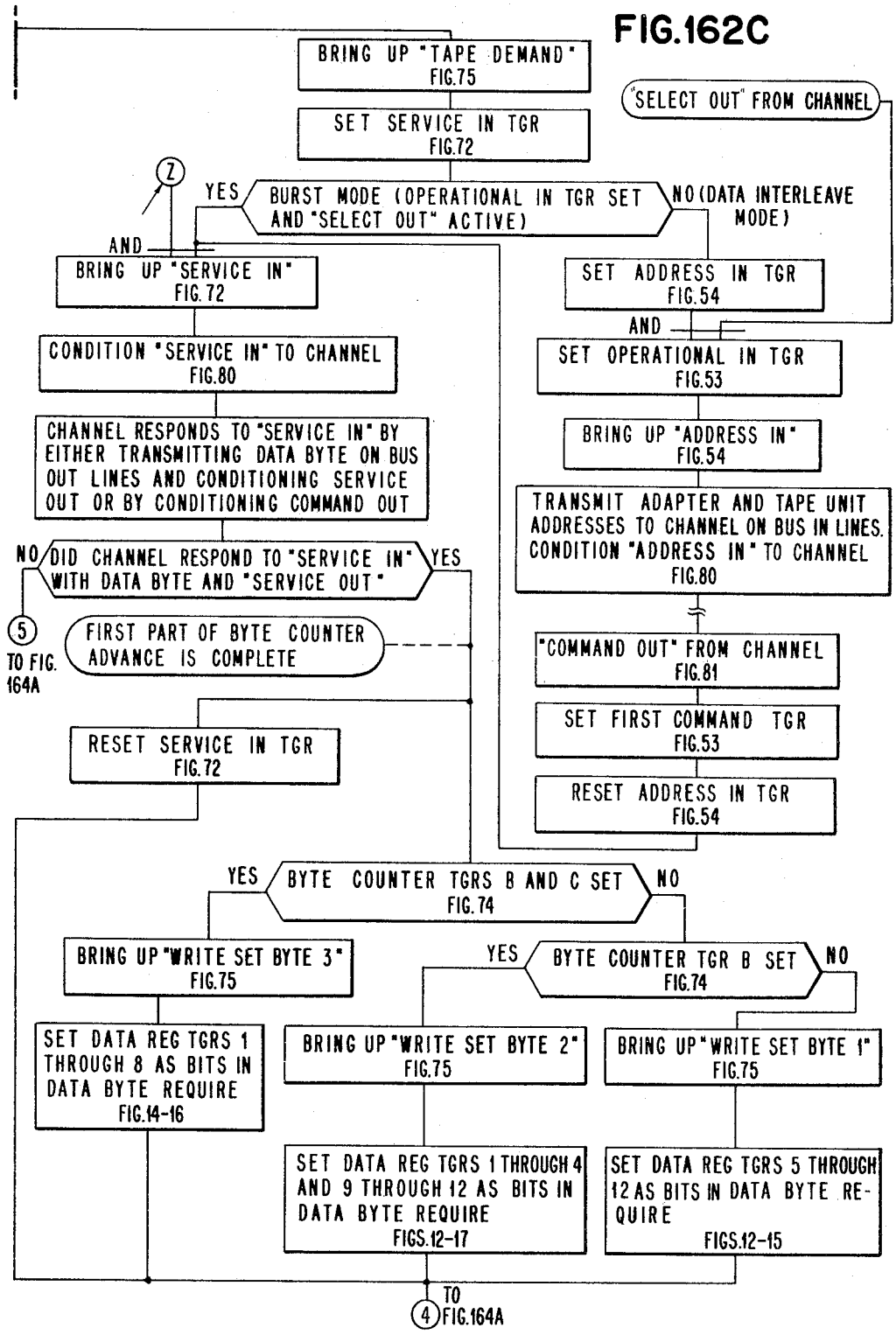

BRING UP "TAPE DEMAND" FIG.75
SET SERVICE IN TGR FIG.72
"SELECT OUT" FROM CHANNEL
Z
YES
BURST MODE (OPERATIONAL IN TGR SET AND "SELECT OUT" ACTIVE)
NO (DATA INTERLEAVE MODE)
AND
BRING UP "SERVICE IN" FIG.72
SET ADDRESS IN TGR FIG.54
AND
CONDITION "SERVICE IN" TO CHANNEL FIG.80
SET OPERATIONAL IN TGR FIG.53
CHANNEL RESPONDS TO "SERVICE IN" BY EITHER TRANSMITTING DATA BYTE ON BUS OUT LINES AND CONDITIONING SERVICE OUT OR BY CONDITIONING COMMAND OUT
BRING UP "ADDRESS IN" FIG.54
TRANSMIT ADAPTER AND TAPE UNIT ADDRESSES TO CHANNEL ON BUS IN LINES. CONDITION "ADDRESS IN" TO CHANNEL FIG.80
NO
DID CHANNEL RESPOND TO "SERVICE IN" WITH DATA BYTE AND "SERVICE OUT"
YES
5
TO FIG. 164A
FIRST PART OF BYTE COUNTER ADVANCE IS COMPLETE
"COMMAND OUT" FROM CHANNEL FIG.81
SET FIRST COMMAND TGR FIG.53
RESET SERVICE IN TGR FIG.72
RESET ADDRESS IN TGR FIG.54
YES
BYTE COUNTER TGRS B AND C SET FIG.74
NO
BRING UP "WRITE SET BYTE 3" FIG.75
YES
BYTE COUNTER TGR B SET FIG.74
NO
SET DATA REG TGRS 1 THROUGH 8 AS BITS IN DATA BYTE REQUIRE FIG.14-16
BRING UP "WRITE SET BYTE 2" FIG.75
BRING UP "WRITE SET BYTE 1" FIG.75
SET DATA REG TGRS 1 THROUGH 4 AND 9 THROUGH 12 AS BITS IN DATA BYTE REQUIRE FIGS.12-17
SET DATA REG TGRS 5 THROUGH 12 AS BITS IN DATA BYTE REQUIRE FIGS.12-15
TO
4
FIG.164A

FIG.163A

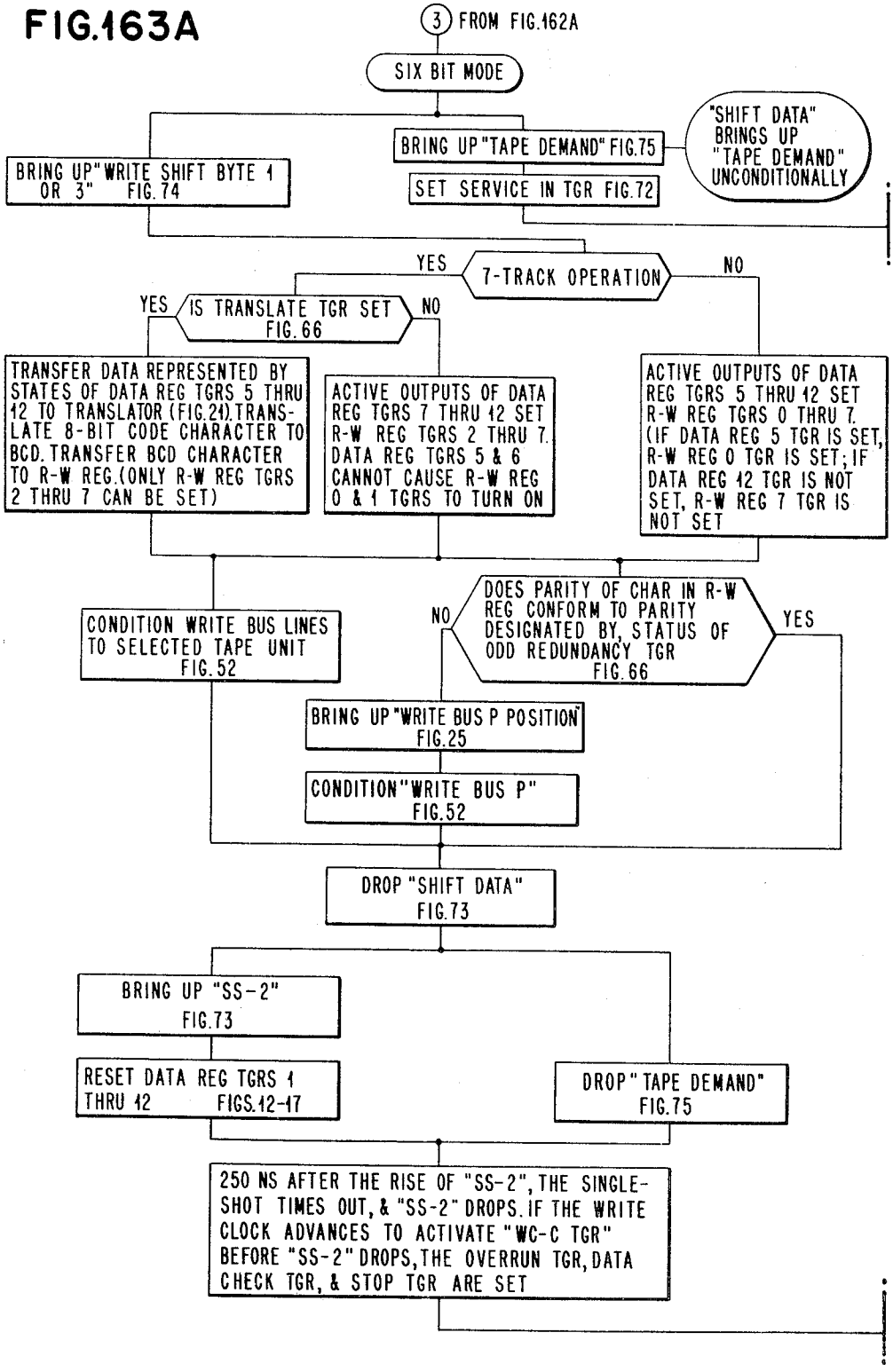

3 FROM FIG.162A
SIX BIT MODE
BRING UP "WRITE SHIFT BYTE 1 OR 3" FIG.74
BRING UP "TAPE DEMAND" FIG.75
SET SERVICE IN TGR FIG.72
"SHIFT DATA" BRINGS UP "TAPE DEMAND" UNCONDITIONALLY
YES
7-TRACK OPERATION
NO
YES
IS TRANSLATE TGR SET FIG.66
NO
TRANSFER DATA REPRESENTED BY STATES OF DATA REG TGRS 5 THRU 12 TO TRANSLATOR (FIG.21). TRANSLATE 8-BIT CODE CHARACTER TO BCD. TRANSFER BCD CHARACTER TO R-W REG. (ONLY R-W REG TGRS 2 THRU 7 CAN BE SET)
ACTIVE OUTPUTS OF DATA REG TGRS 7 THRU 12 SET R-W REG TGRS 2 THRU 7. DATA REG TGRS 5 & 6 CANNOT CAUSE R-W REG 0 & 1 TGRS TO TURN ON
ACTIVE OUTPUTS OF DATA REG TGRS 5 THRU 12 SET R-W REG TGRS 0 THRU 7. (IF DATA REG 5 TGR IS SET, R-W REG 0 TGR IS SET; IF DATA REG 12 TGR IS NOT SET, R-W REG 7 TGR IS NOT SET
CONDITION WRITE BUS LINES TO SELECTED TAPE UNIT FIG.52
NO
DOES PARITY OF CHAR IN R-W REG CONFORM TO PARITY DESIGNATED BY, STATUS OF ODD REDUNDANCY TGR FIG.66
YES
BRING UP "WRITE BUS P POSITION" FIG.25
CONDITION "WRITE BUS P" FIG.52
DROP "SHIFT DATA" FIG.73
BRING UP "SS-2" FIG.73
RESET DATA REG TGRS 1 THRU 12 FIGS.12-17
DROP "TAPE DEMAND" FIG.75
250 NS AFTER THE RISE OF "SS-2", THE SINGLE-SHOT TIMES OUT, & "SS-2" DROPS. IF THE WRITE CLOCK ADVANCES TO ACTIVATE "WC-C TGR" BEFORE "SS-2" DROPS, THE OVERRUN TGR, DATA CHECK TGR, & STOP TGR ARE SET

FIG. 163B

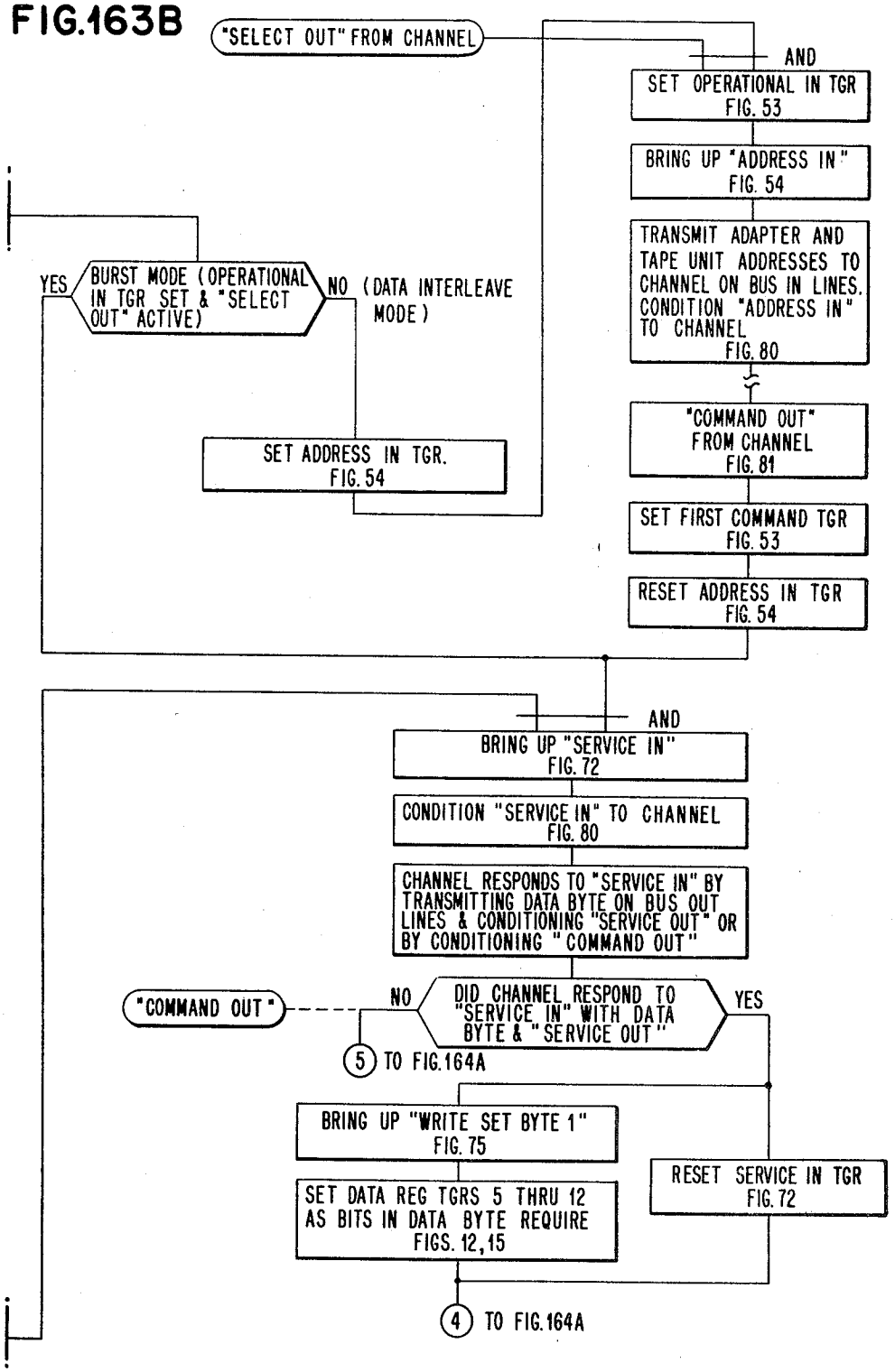

"SELECT OUT" FROM CHANNEL
AND
SET OPERATIONAL IN TGR FIG. 53
BRING UP "ADDRESS IN" FIG. 54
TRANSMIT ADAPTER AND TAPE UNIT ADDRESSES TO CHANNEL ON BUS IN LINES. CONDITION "ADDRESS IN" TO CHANNEL FIG. 80
"COMMAND OUT" FROM CHANNEL FIG. 81
SET FIRST COMMAND TGR FIG. 53
RESET ADDRESS IN TGR FIG. 54
YES
BURST MODE (OPERATIONAL IN TGR SET & "SELECT OUT" ACTIVE)
NO (DATA INTERLEAVE MODE)
SET ADDRESS IN TGR. FIG. 54
AND
BRING UP "SERVICE IN" FIG. 72
CONDITION "SERVICE IN" TO CHANNEL FIG. 80
CHANNEL RESPONDS TO "SERVICE IN" BY TRANSMITTING DATA BYTE ON BUS OUT LINES & CONDITIONING "SERVICE OUT" OR BY CONDITIONING "COMMAND OUT"
"COMMAND OUT"
NO
DID CHANNEL RESPOND TO "SERVICE IN" WITH DATA BYTE & "SERVICE OUT"
YES
5 TO FIG. 164A
BRING UP "WRITE SET BYTE 1" FIG. 75
SET DATA REG TGRS 5 THRU 12 AS BITS IN DATA BYTE REQUIRE FIGS. 12,15
RESET SERVICE IN TGR FIG. 72
4 TO FIG. 164A

FIG. 164A

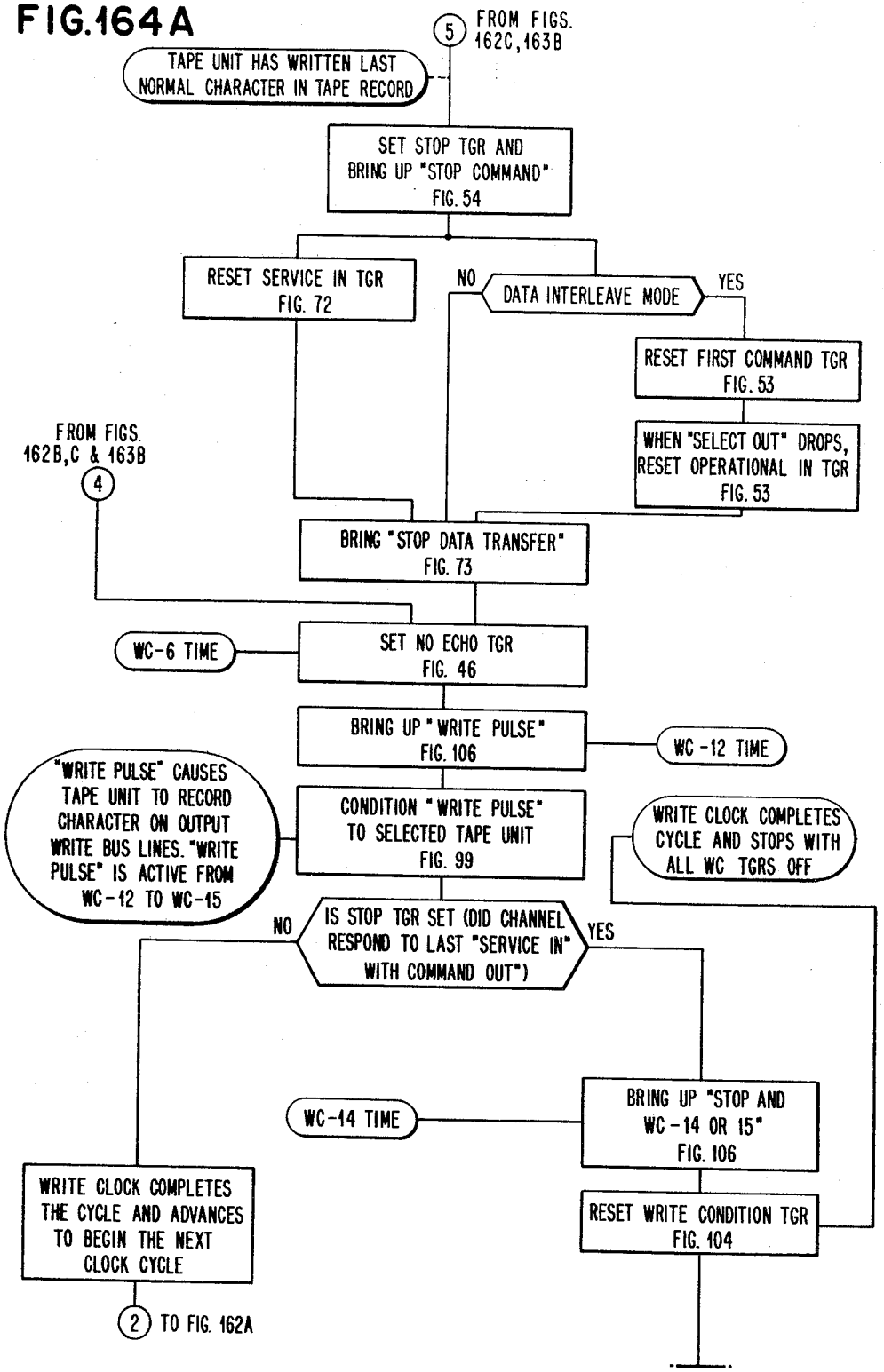
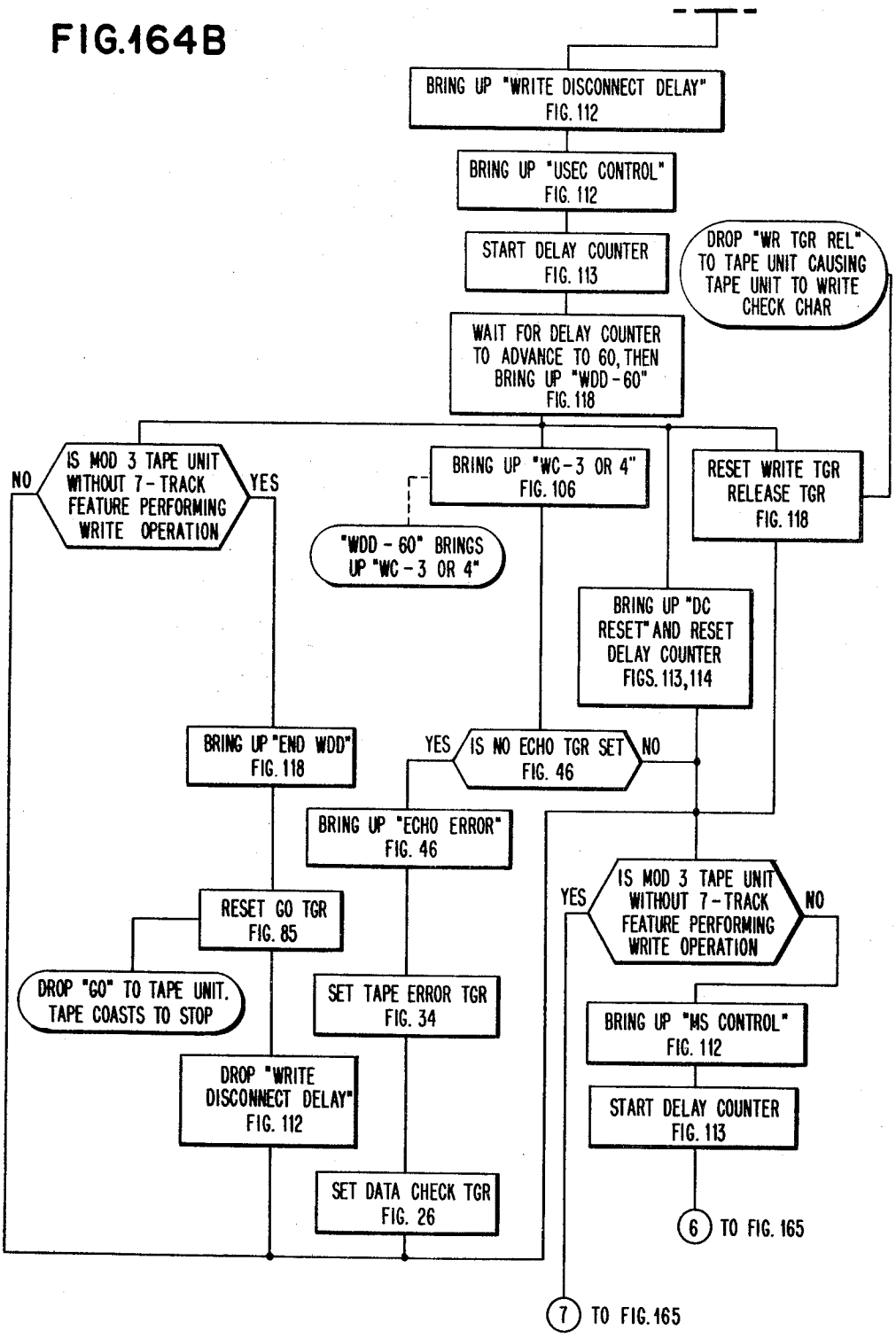

5
FROM FIGS. 162C, 163B
TAPE UNIT HAS WRITTEN LAST NORMAL CHARACTER IN TAPE RECORD
SET STOP TGR AND BRING UP "STOP COMMAND" FIG. 54
RESET SERVICE IN TGR FIG. 72
NO
DATA INTERLEAVE MODE
YES
RESET FIRST COMMAND TGR FIG. 53
WHEN "SELECT OUT" DROPS, RESET OPERATIONAL IN TGR FIG. 53
FROM FIGS. 162B, C & 163B
4
BRING "STOP DATA TRANSFER" FIG. 73
WC-6 TIME
SET NO ECHO TGR FIG. 46
BRING UP "WRITE PULSE" FIG. 106
WC-12 TIME
"WRITE PULSE" CAUSES TAPE UNIT TO RECORD CHARACTER ON OUTPUT WRITE BUS LINES. "WRITE PULSE" IS ACTIVE FROM WC-12 TO WC-15
CONDITION "WRITE PULSE" TO SELECTED TAPE UNIT FIG. 99
WRITE CLOCK COMPLETES CYCLE AND STOPS WITH ALL WC TGRS OFF
NO
IS STOP TGR SET (DID CHANNEL RESPOND TO LAST "SERVICE IN" WITH COMMAND OUT")
YES
WC-14 TIME
BRING UP "STOP AND WC-14 OR 15" FIG. 106
RESET WRITE CONDITION TGR FIG. 104
WRITE CLOCK COMPLETES THE CYCLE AND ADVANCES TO BEGIN THE NEXT CLOCK CYCLE
2
TO FIG. 162A

FIG.164B


BRING UP "WRITE DISCONNECT DELAY" FIG. 112
BRING UP "USEC CONTROL" FIG. 112
START DELAY COUNTER FIG. 113
DROP "WR TGR REL" TO TAPE UNIT CAUSING TAPE UNIT TO WRITE CHECK CHAR
WAIT FOR DELAY COUNTER TO ADVANCE TO 60, THEN BRING UP "WDD-60" FIG. 118
IS MOD 3 TAPE UNIT WITHOUT 7-TRACK FEATURE PERFORMING WRITE OPERATION
NO
YES
BRING UP "WC-3 OR 4" FIG. 106
"WDD-60" BRINGS UP "WC-3 OR 4"
RESET WRITE TGR RELEASE TGR FIG. 118
BRING UP "DC RESET" AND RESET DELAY COUNTER FIGS. 113,114
BRING UP "END WDD" FIG. 118
YES
IS NO ECHO TGR SET FIG. 46
NO
BRING UP "ECHO ERROR" FIG. 46
RESET GO TGR FIG. 85
DROP "GO" TO TAPE UNIT. TAPE COASTS TO STOP
SET TAPE ERROR TGR FIG. 34
DROP "WRITE DISCONNECT DELAY" FIG. 112
SET DATA CHECK TGR FIG. 26
YES
IS MOD 3 TAPE UNIT WITHOUT 7-TRACK FEATURE PERFORMING WRITE OPERATION
NO
BRING UP "MS CONTROL" FIG. 112
START DELAY COUNTER FIG. 113
6 TO FIG. 165
7 TO FIG. 165

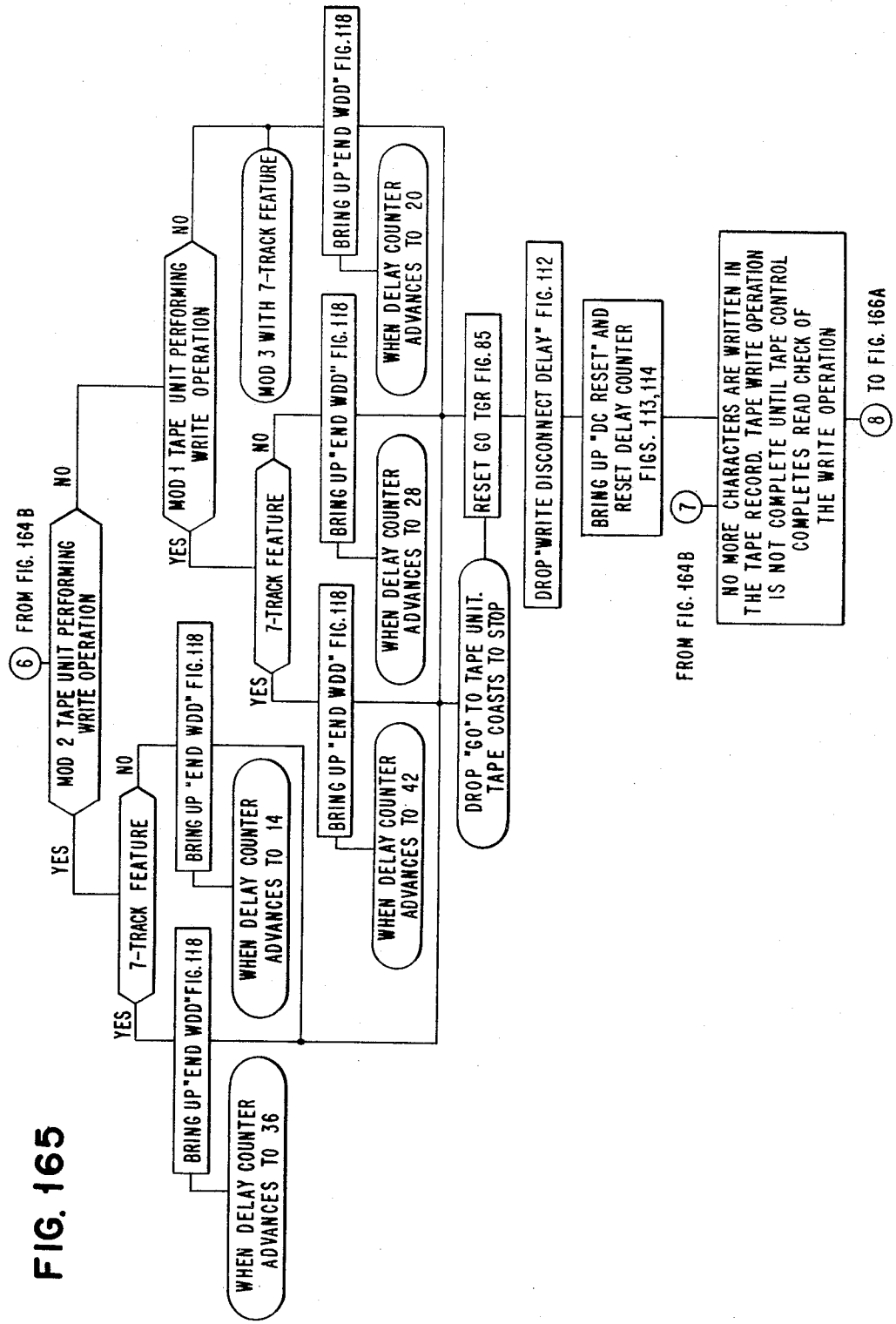

6 FROM FIG. 164 B
MOD 2 TAPE UNIT PERFORMING WRITE OPERATION
YES
NO
7-TRACK FEATURE
YES
NO
BRING UP "END WDD" FIG. 118
WHEN DELAY COUNTER ADVANCES TO 14
BRING UP "END WDD" FIG. 118
WHEN DELAY COUNTER ADVANCES TO 36
MOD 1 TAPE UNIT PERFORMING WRITE OPERATION
YES
NO
7-TRACK FEATURE
YES
NO
BRING UP "END WDD" FIG. 118
WHEN DELAY COUNTER ADVANCES TO 28
BRING UP "END WDD" FIG. 118
WHEN DELAY COUNTER ADVANCES TO 42
MOD 3 WITH 7-TRACK FEATURE
BRING UP "END WDD" FIG. 118
WHEN DELAY COUNTER ADVANCES TO 20
DROP "GO" TO TAPE UNIT. TAPE COASTS TO STOP
RESET GO TGR FIG. 85
DROP "WRITE DISCONNECT DELAY" FIG. 112
BRING UP "DC RESET" AND RESET DELAY COUNTER FIGS. 113, 114
FROM FIG. 164B 7
NO MORE CHARACTERS ARE WRITTEN IN THE TAPE RECORD. TAPE WRITE OPERATION IS NOT COMPLETE UNTIL TAPE CONTROL COMPLETES READ CHECK OF THE WRITE OPERATION
8 TO FIG. 166A

FIG. 165

FIG. 166A

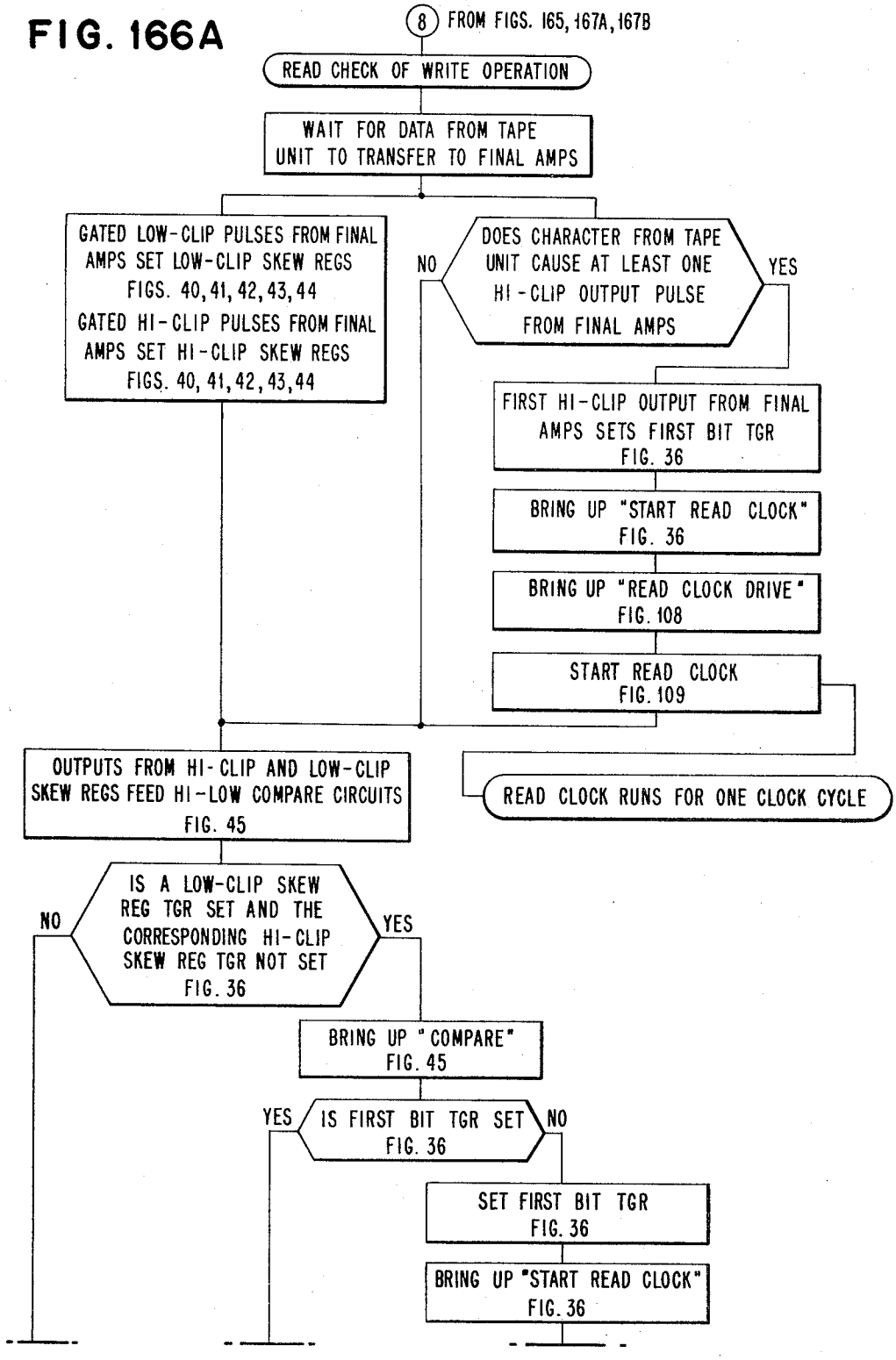
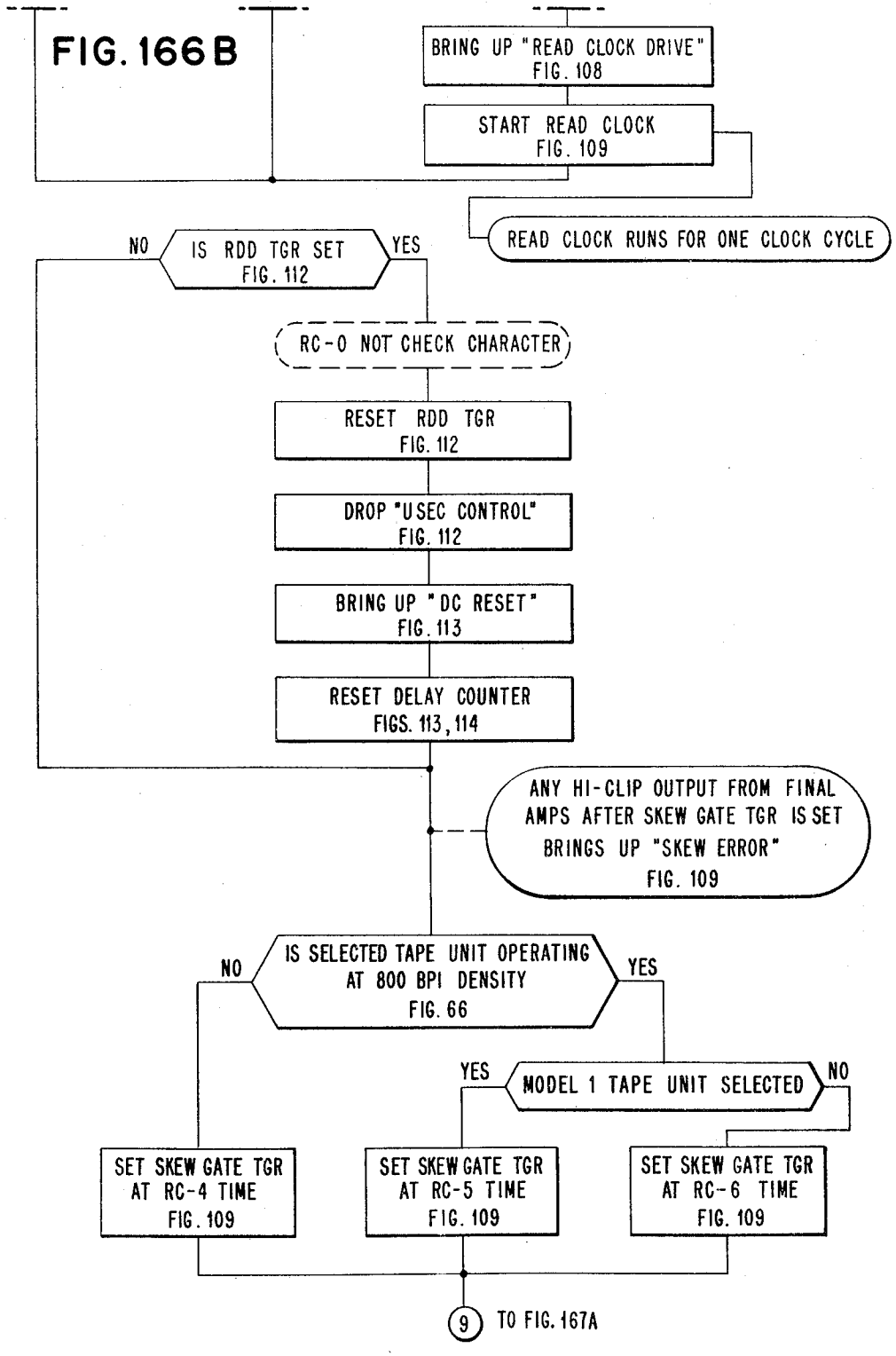

8 FROM FIGS. 165, 167A, 167B
READ CHECK OF WRITE OPERATION
WAIT FOR DATA FROM TAPE UNIT TO TRANSFER TO FINAL AMPS
GATED LOW-CLIP PULSES FROM FINAL AMPS SET LOW-CLIP SKEW REGS FIGS. 40, 41, 42, 43, 44
GATED HI-CLIP PULSES FROM FINAL AMPS SET HI-CLIP SKEW REGS FIGS. 40, 41, 42, 43, 44
DOES CHARACTER FROM TAPE UNIT CAUSE AT LEAST ONE HI-CLIP OUTPUT PULSE FROM FINAL AMPS
NO
YES
FIRST HI-CLIP OUTPUT FROM FINAL AMPS SETS FIRST BIT TGR FIG. 36
BRING UP "START READ CLOCK" FIG. 36
BRING UP "READ CLOCK DRIVE" FIG. 108
START READ CLOCK FIG. 109
READ CLOCK RUNS FOR ONE CLOCK CYCLE
OUTPUTS FROM HI-CLIP AND LOW-CLIP SKEW REGS FEED HI-LOW COMPARE CIRCUITS FIG. 45
IS A LOW-CLIP SKEW REG TGR SET AND THE CORRESPONDING HI-CLIP SKEW REG TGR NOT SET FIG. 36
NO
YES
BRING UP "COMPARE" FIG. 45
IS FIRST BIT TGR SET FIG. 36
YES
NO
SET FIRST BIT TGR FIG. 36
BRING UP "START READ CLOCK" FIG. 36

FIG. 166B


BRING UP "READ CLOCK DRIVE" FIG. 108
START READ CLOCK FIG. 109
READ CLOCK RUNS FOR ONE CLOCK CYCLE
NO
IS RDD TGR SET FIG. 112
YES
RC-0 NOT CHECK CHARACTER
RESET RDD TGR FIG. 112
DROP "USEC CONTROL" FIG. 112
BRING UP "DC RESET" FIG. 113
RESET DELAY COUNTER FIGS. 113, 114
ANY HI-CLIP OUTPUT FROM FINAL AMPS AFTER SKEW GATE TGR IS SET BRINGS UP "SKEW ERROR" FIG. 109
NO
IS SELECTED TAPE UNIT OPERATING AT 800 BPI DENSITY FIG. 66
YES
YES
MODEL 1 TAPE UNIT SELECTED
NO
SET SKEW GATE TGR AT RC-4 TIME FIG. 109
SET SKEW GATE TGR AT RC-5 TIME FIG. 109
SET SKEW GATE TGR AT RC-6 TIME FIG. 109
9
TO FIG. 167A

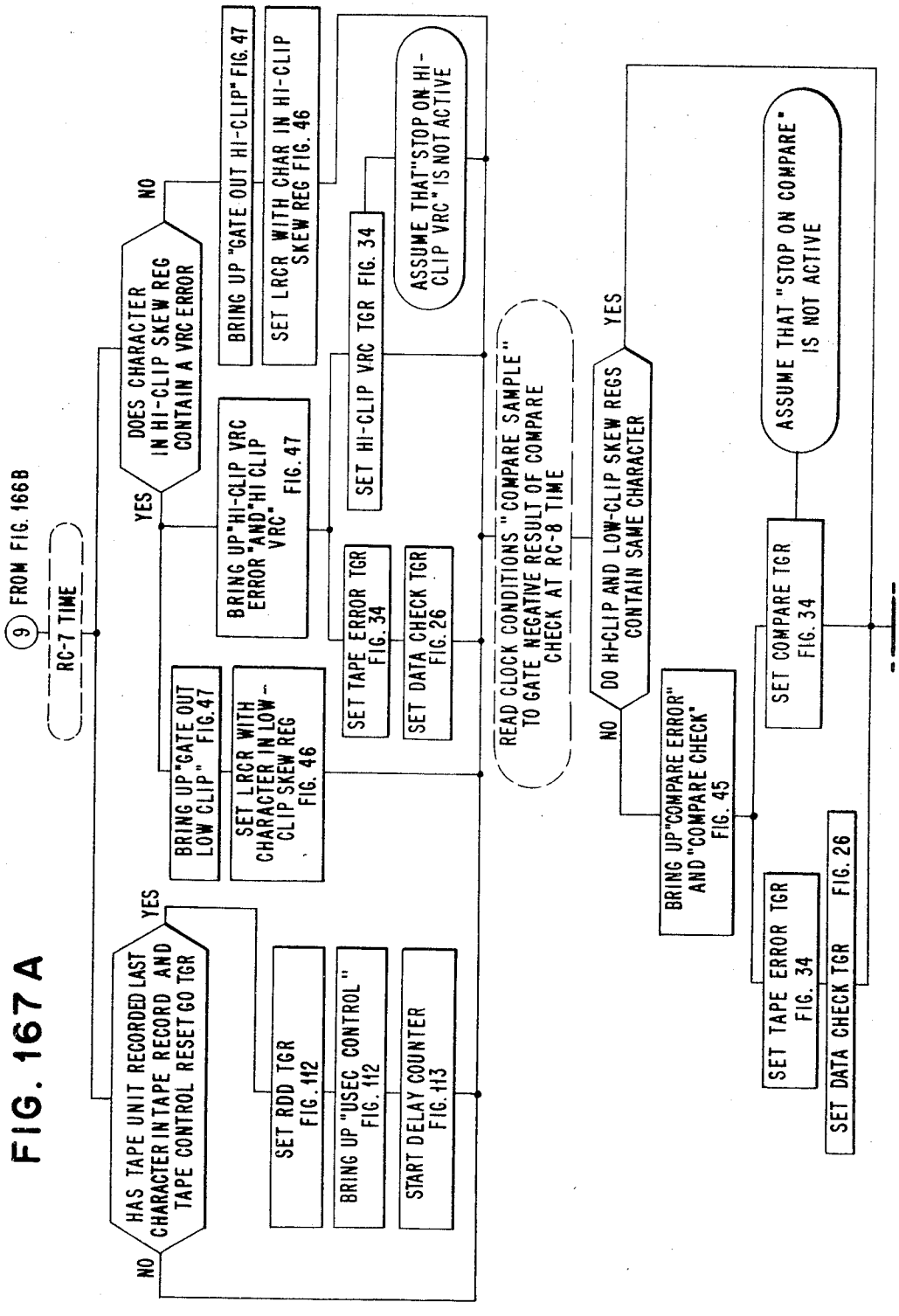

9 FROM FIG. 166B
RC-7 TIME
DOES CHARACTER IN HI-CLIP SKEW REG CONTAIN A VRC ERROR
YES
NO
BRING UP "GATE OUT HI-CLIP" FIG. 47
SET LRCR WITH CHAR IN HI-CLIP SKEW REG FIG. 46
BRING UP "HI-CLIP VRC ERROR "AND" HI CLIP VRC" FIG. 47
SET HI-CLIP VRC TGR FIG. 34
ASSUME THAT "STOP ON HI-CLIP VRC" IS NOT ACTIVE
BRING UP "GATE OUT LOW CLIP" FIG. 47
SET LRCR WITH CHARACTER IN LOW-CLIP SKEW REG FIG. 46
SET TAPE ERROR TGR FIG. 34
SET DATA CHECK TGR FIG. 26
READ CLOCK CONDITIONS "COMPARE SAMPLE" TO GATE NEGATIVE RESULT OF COMPARE CHECK AT RC-8 TIME
DO HI-CLIP AND LOW-CLIP SKEW REGS CONTAIN SAME CHARACTER
YES
NO
ASSUME THAT "STOP ON COMPARE" IS NOT ACTIVE
BRING UP "COMPARE ERROR" AND "COMPARE CHECK" FIG. 45
SET COMPARE TGR FIG. 34
SET TAPE ERROR TGR FIG. 34
SET DATA CHECK TGR FIG. 26
HAS TAPE UNIT RECORDED LAST CHARACTER IN TAPE RECORD AND TAPE CONTROL RESET GO TGR
YES
NO
SET RDD TGR FIG. 112
BRING UP "USEC CONTROL" FIG. 112
START DELAY COUNTER FIG. 113

FIG. 167A

FIG. 167B

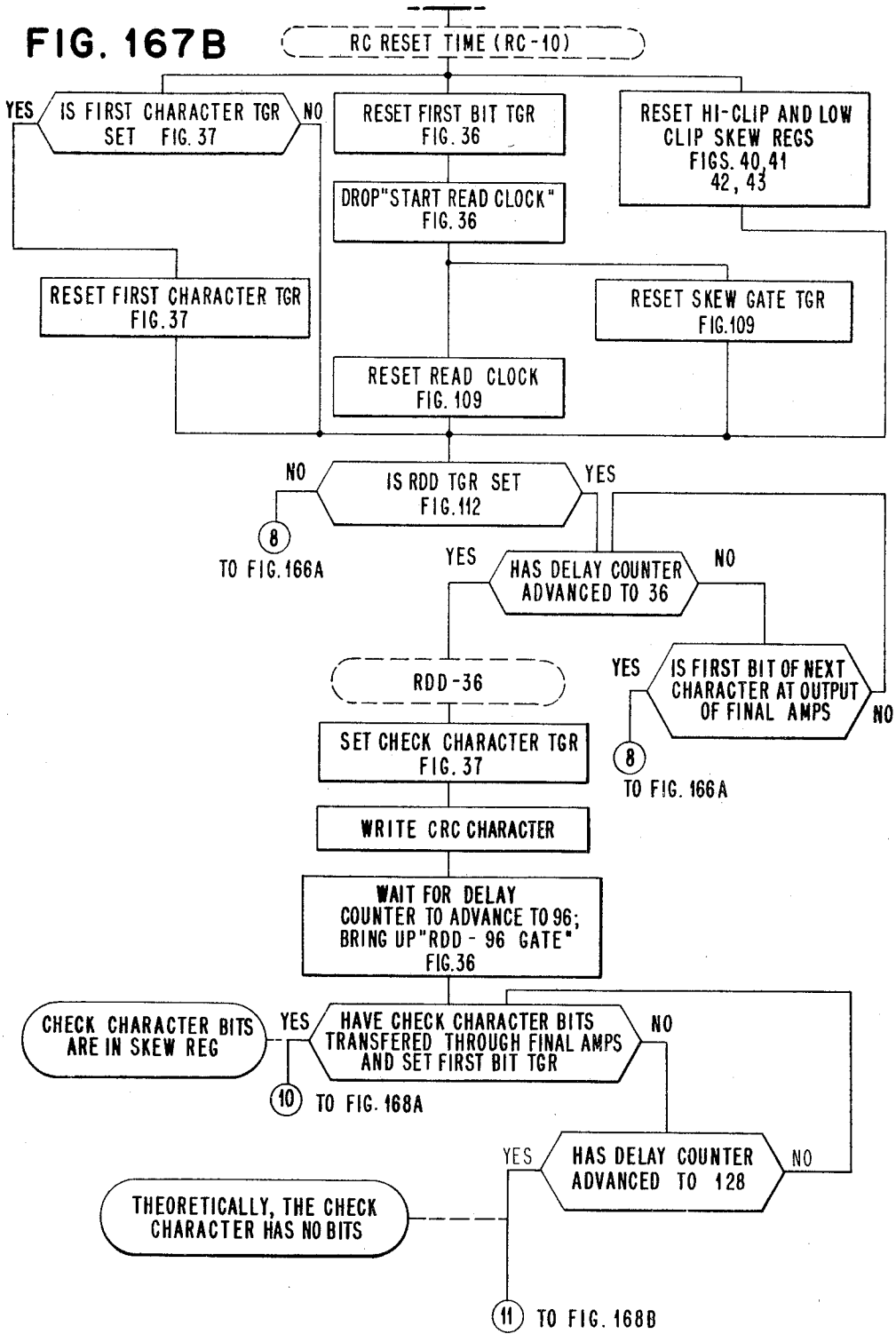

RC RESET TIME (RC-10)
IS FIRST CHARACTER TGR SET FIG. 37
YES
NO
RESET FIRST BIT TGR FIG. 36
RESET HI-CLIP AND LOW CLIP SKEW REGS FIGS. 40, 41 42, 43
DROP "START READ CLOCK" FIG. 36
RESET FIRST CHARACTER TGR FIG. 37
RESET SKEW GATE TGR FIG. 109
RESET READ CLOCK FIG. 109
IS RDD TGR SET FIG. 112
NO
YES
8
TO FIG. 166A
HAS DELAY COUNTER ADVANCED TO 36
YES
NO
RDD-36
IS FIRST BIT OF NEXT CHARACTER AT OUTPUT OF FINAL AMPS
YES
NO
8
TO FIG. 166A
SET CHECK CHARACTER TGR FIG. 37
WRITE CRC CHARACTER
WAIT FOR DELAY COUNTER TO ADVANCE TO 96; BRING UP "RDD - 96 GATE" FIG. 36
CHECK CHARACTER BITS ARE IN SKEW REG
YES
HAVE CHECK CHARACTER BITS TRANSFERED THROUGH FINAL AMPS AND SET FIRST BIT TGR
NO
10
TO FIG. 168A
YES
HAS DELAY COUNTER ADVANCED TO 128
NO
THEORETICALLY, THE CHECK CHARACTER HAS NO BITS
11
TO FIG. 168B

FIG. 168A

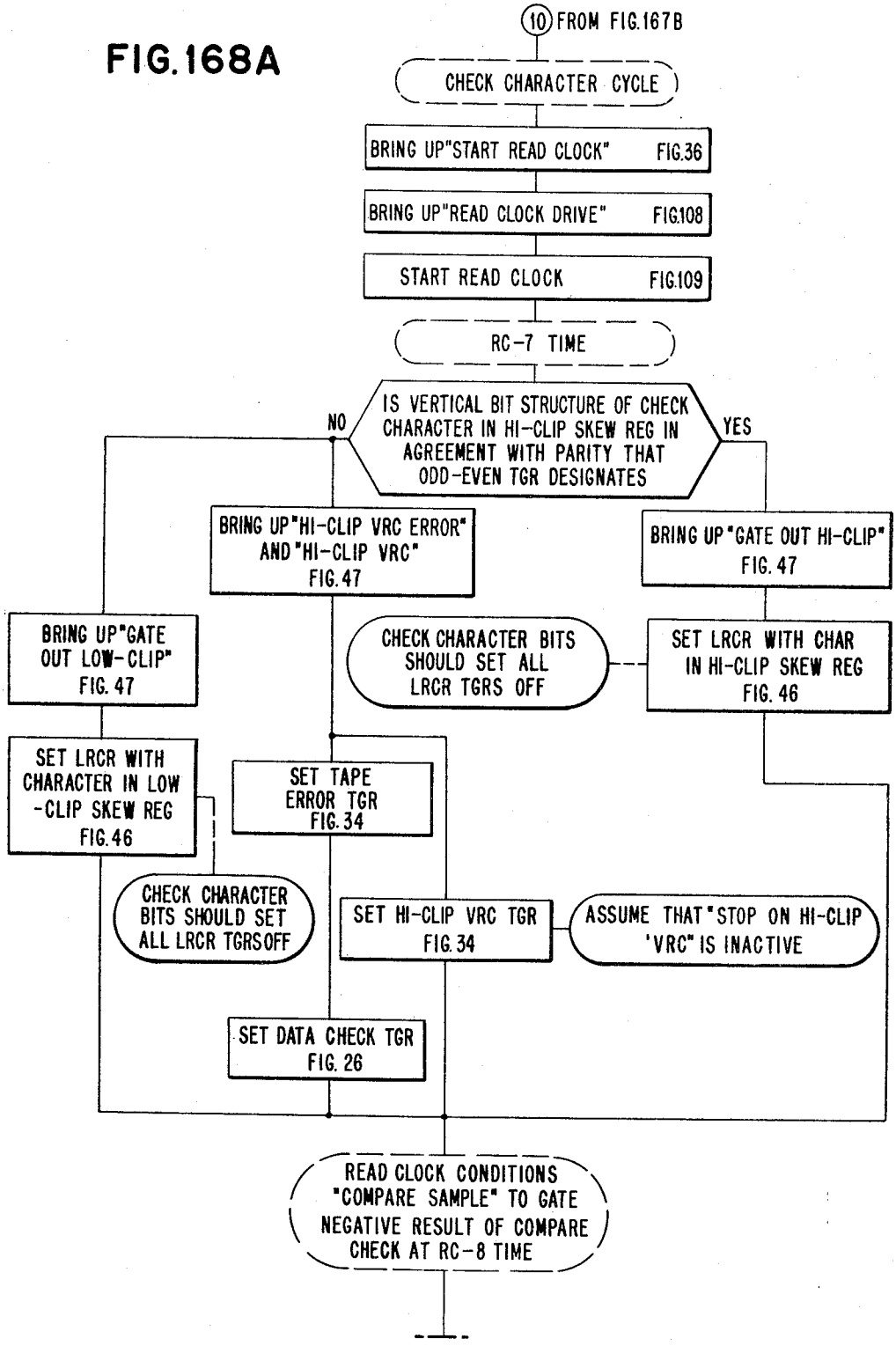

(10) FROM FIG. 167B
CHECK CHARACTER CYCLE
BRING UP "START READ CLOCK" FIG. 36
BRING UP "READ CLOCK DRIVE" FIG. 108
START READ CLOCK FIG. 109
RC-7 TIME
IS VERTICAL BIT STRUCTURE OF CHECK CHARACTER IN HI-CLIP SKEW REG IN AGREEMENT WITH PARITY THAT ODD-EVEN TGR DESIGNATES
NO
YES
BRING UP "HI-CLIP VRC ERROR" AND "HI-CLIP VRC" FIG. 47
BRING UP "GATE OUT HI-CLIP" FIG. 47
BRING UP "GATE OUT LOW-CLIP" FIG. 47
CHECK CHARACTER BITS SHOULD SET ALL LRCR TGRS OFF
SET LRCR WITH CHAR IN HI-CLIP SKEW REG FIG. 46
SET LRCR WITH CHARACTER IN LOW-CLIP SKEW REG FIG. 46
SET TAPE ERROR TGR FIG. 34
CHECK CHARACTER BITS SHOULD SET ALL LRCR TGRS OFF
SET HI-CLIP VRC TGR FIG. 34
ASSUME THAT "STOP ON HI-CLIP 'VRC" IS INACTIVE
SET DATA CHECK TGR FIG. 26
READ CLOCK CONDITIONS "COMPARE SAMPLE" TO GATE NEGATIVE RESULT OF COMPARE CHECK AT RC-8 TIME

FIG. 168B

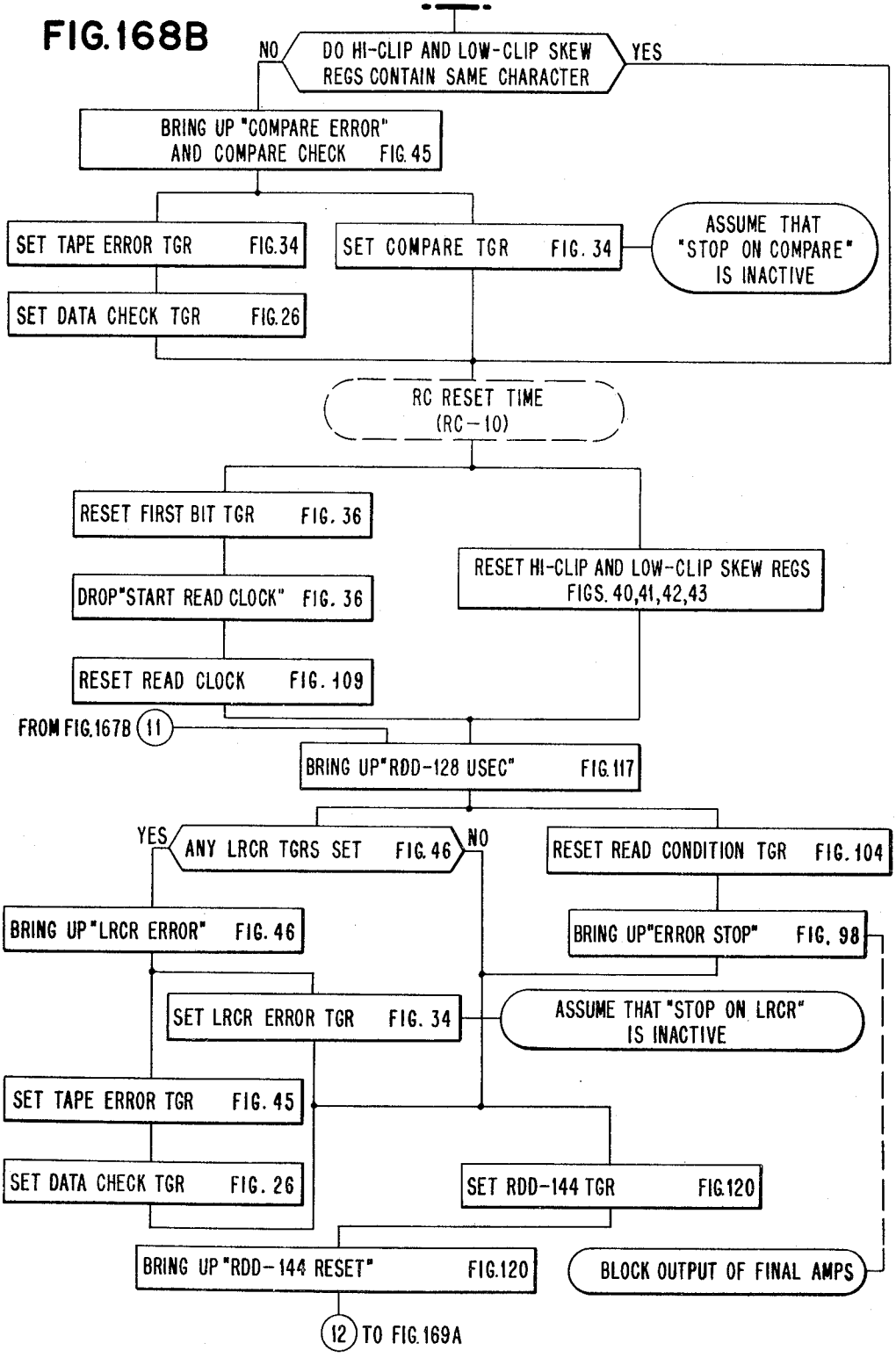

DO HI-CLIP AND LOW-CLIP SKEW REGS CONTAIN SAME CHARACTER
NO
YES
BRING UP "COMPARE ERROR" AND COMPARE CHECK FIG. 45
SET TAPE ERROR TGR FIG. 34
SET DATA CHECK TGR FIG. 26
SET COMPARE TGR FIG. 34
ASSUME THAT "STOP ON COMPARE" IS INACTIVE
RC RESET TIME (RC-10)
RESET FIRST BIT TGR FIG. 36
DROP "START READ CLOCK" FIG. 36
RESET READ CLOCK FIG. 109
RESET HI-CLIP AND LOW-CLIP SKEW REGS FIGS. 40,41,42,43
FROM FIG. 167B (11)
BRING UP "RDD-128 USEC" FIG. 117
YES
ANY LRCR TGRS SET FIG. 46
NO
RESET READ CONDITION TGR FIG. 104
BRING UP "LRCR ERROR" FIG. 46
BRING UP "ERROR STOP" FIG. 98
SET LRCR ERROR TGR FIG. 34
ASSUME THAT "STOP ON LRCR" IS INACTIVE
SET TAPE ERROR TGR FIG. 45
SET DATA CHECK TGR FIG. 26
SET RDD-144 TGR FIG. 120
BRING UP "RDD-144 RESET" FIG. 120
BLOCK OUTPUT OF FINAL AMPS
(12) TO FIG. 169A

FIG. 169A

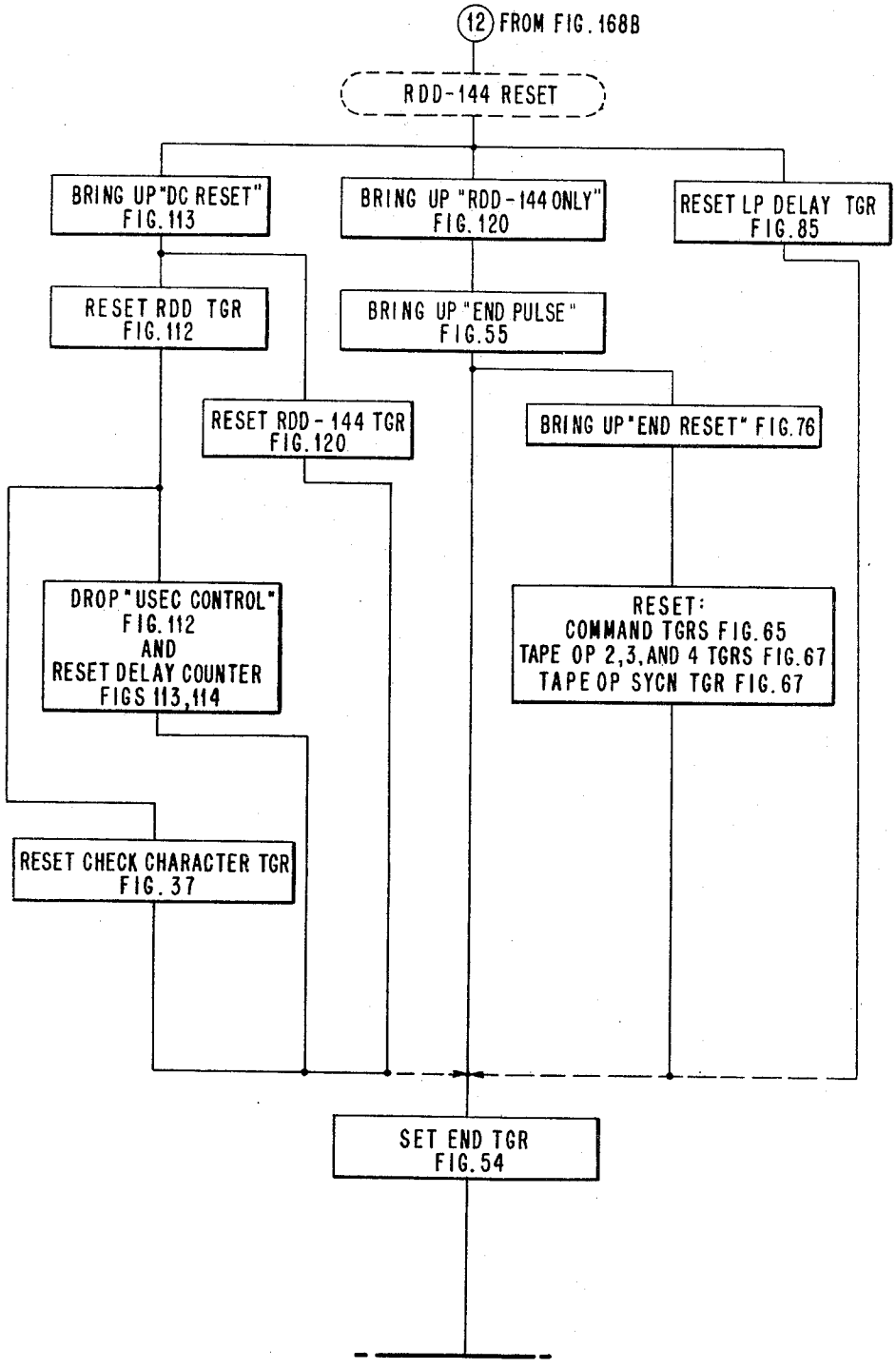

12 FROM FIG. 168B
RDD-144 RESET
BRING UP "DC RESET" FIG. 113
BRING UP "RDD-144 ONLY" FIG. 120
RESET LP DELAY TGR FIG. 85
RESET RDD TGR FIG. 112
BRING UP "END PULSE" FIG. 55
RESET RDD-144 TGR FIG. 120
BRING UP "END RESET" FIG. 76
DROP "USEC CONTROL" FIG. 112 AND RESET DELAY COUNTER FIGS 113, 114
RESET: COMMAND TGRS FIG. 65 TAPE OP 2,3, AND 4 TGRS FIG. 67 TAPE OP SYCN TGR FIG. 67
RESET CHECK CHARACTER TGR FIG. 37
SET END TGR FIG. 54

FIG. 169B

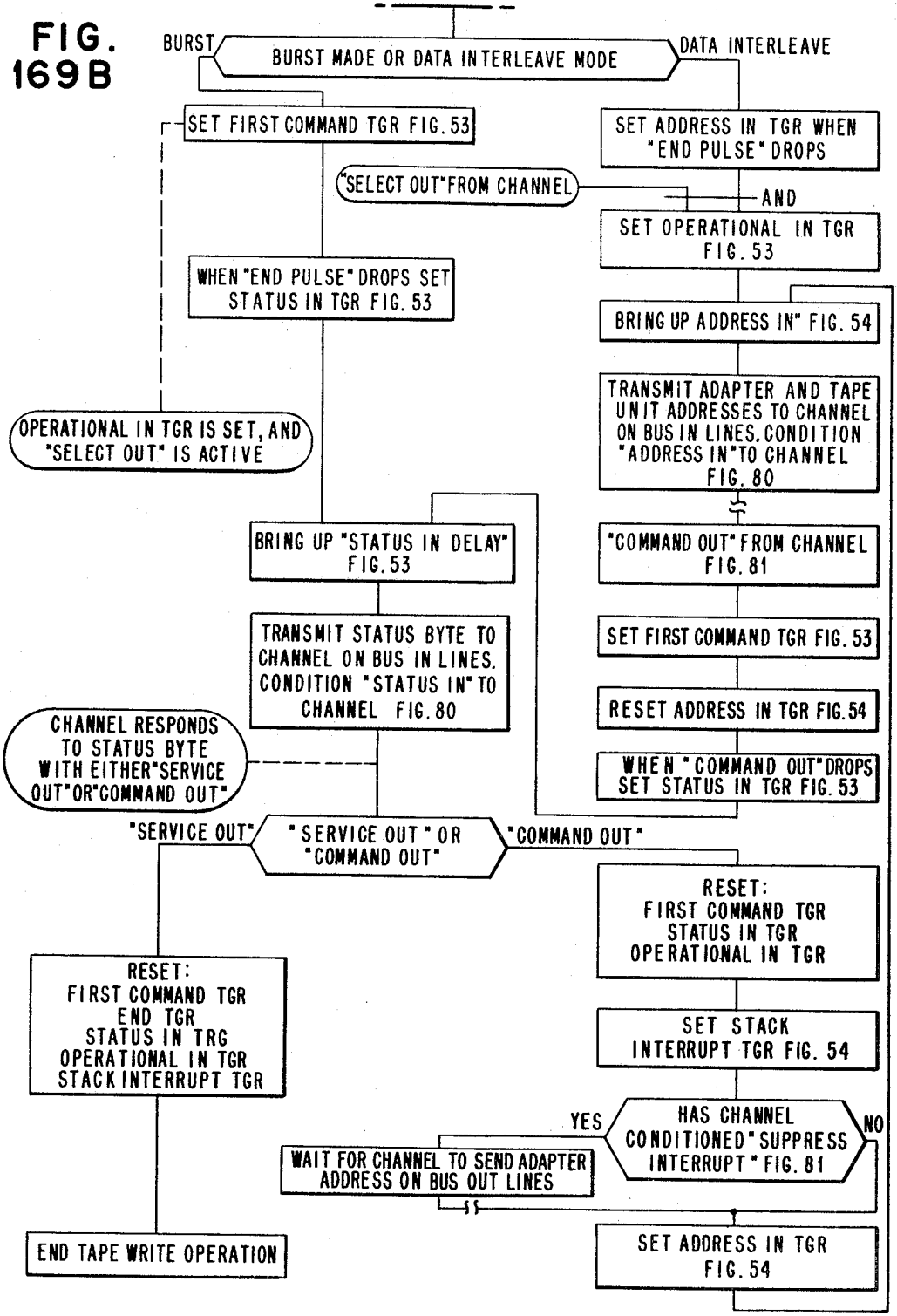

BURST
BURST MADE OR DATA INTERLEAVE MODE
DATA INTERLEAVE
SET FIRST COMMAND TGR FIG. 53
SET ADDRESS IN TGR WHEN "END PULSE" DROPS
"SELECT OUT" FROM CHANNEL
AND
SET OPERATIONAL IN TGR FIG. 53
WHEN "END PULSE" DROPS SET STATUS IN TGR FIG. 53
BRING UP ADDRESS IN" FIG. 54
TRANSMIT ADAPTER AND TAPE UNIT ADDRESSES TO CHANNEL ON BUS IN LINES. CONDITION "ADDRESS IN" TO CHANNEL FIG. 80
OPERATIONAL IN TGR IS SET, AND "SELECT OUT" IS ACTIVE
BRING UP "STATUS IN DELAY" FIG. 53
"COMMAND OUT" FROM CHANNEL FIG. 81
TRANSMIT STATUS BYTE TO CHANNEL ON BUS IN LINES. CONDITION "STATUS IN" TO CHANNEL FIG. 80
SET FIRST COMMAND TGR FIG. 53
RESET ADDRESS IN TGR FIG. 54
CHANNEL RESPONDS TO STATUS BYTE WITH EITHER "SERVICE OUT" OR "COMMAND OUT"
WHEN "COMMAND OUT" DROPS SET STATUS IN TGR FIG. 53
"SERVICE OUT"
"SERVICE OUT" OR "COMMAND OUT"
"COMMAND OUT"
RESET: FIRST COMMAND TGR STATUS IN TGR OPERATIONAL IN TGR
RESET: FIRST COMMAND TGR END TGR STATUS IN TRG OPERATIONAL IN TGR STACK INTERRUPT TGR
SET STACK INTERRUPT TGR FIG. 54
YES
HAS CHANNEL CONDITIONED "SUPPRESS INTERRUPT" FIG. 81
NO
WAIT FOR CHANNEL TO SEND ADAPTER ADDRESS ON BUS OUT LINES
END TAPE WRITE OPERATION
SET ADDRESS IN TGR FIG. 54

FIG.170

DATA FLOW IN READ OPERATION

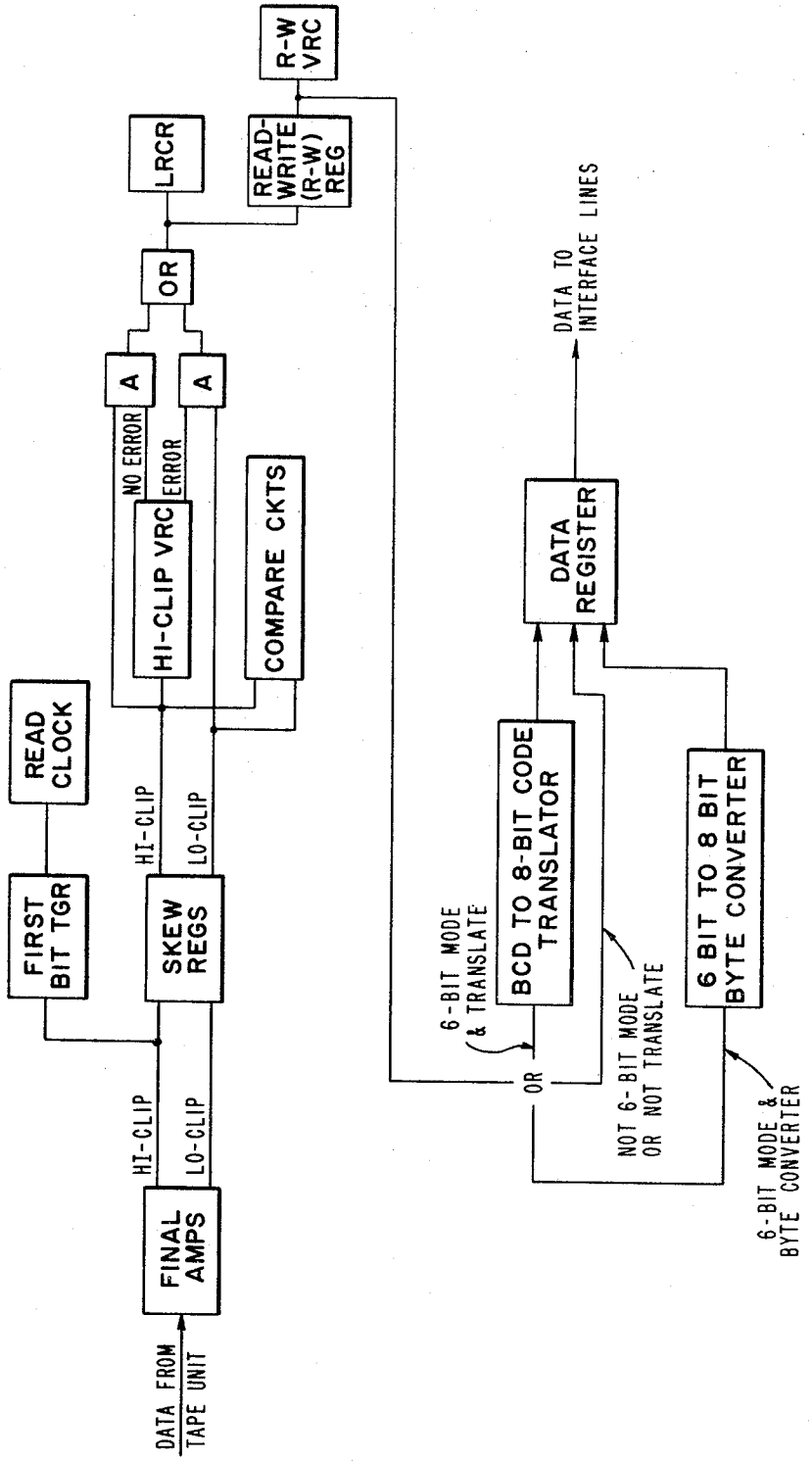


FIG. 171A

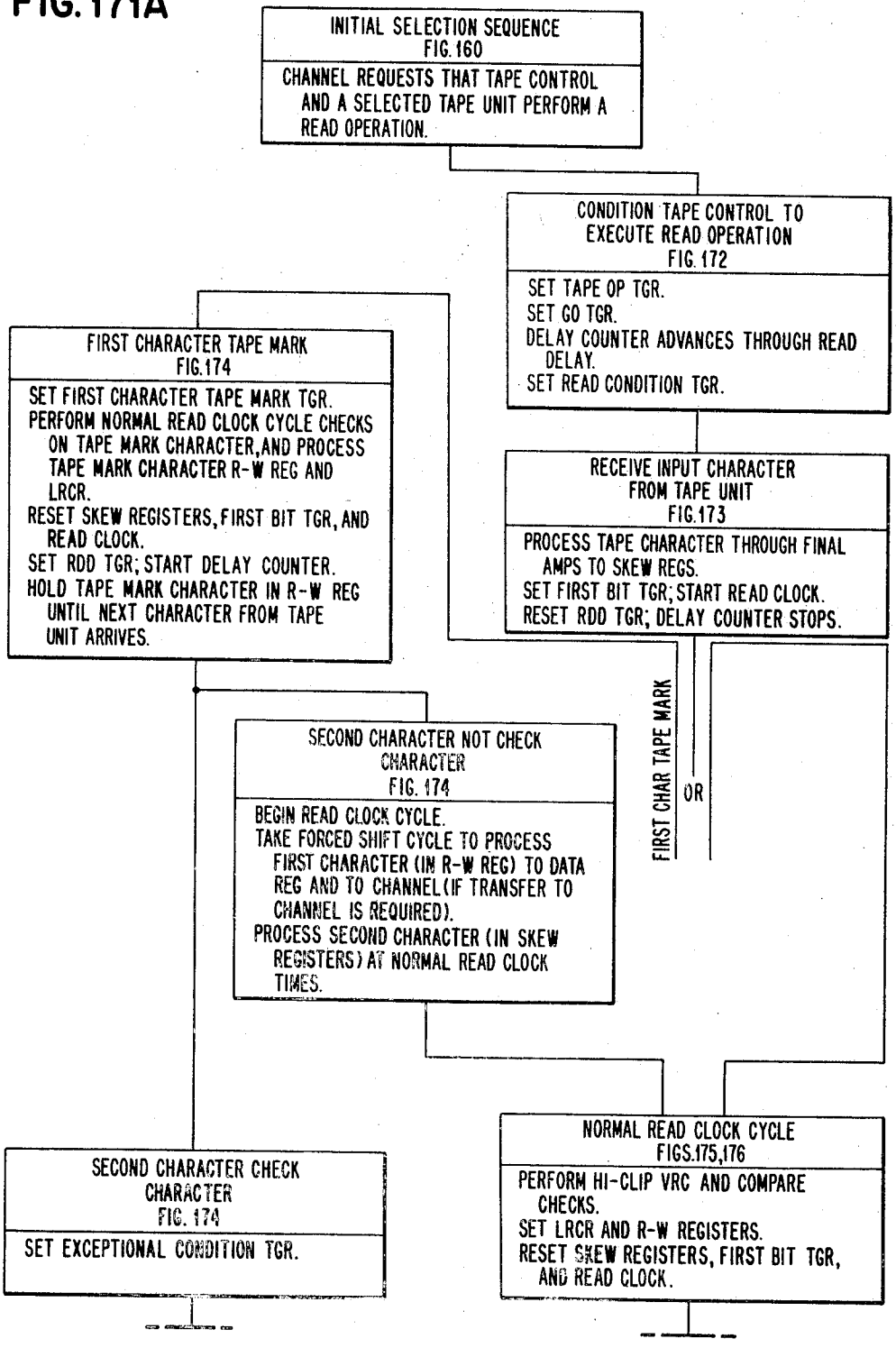

INITIAL SELECTION SEQUENCE
FIG. 160
CHANNEL REQUESTS THAT TAPE CONTROL AND A SELECTED TAPE UNIT PERFORM A READ OPERATION.
CONDITION TAPE CONTROL TO EXECUTE READ OPERATION
FIG. 172
SET TAPE OP TGR.
SET GO TGR.
DELAY COUNTER ADVANCES THROUGH READ DELAY.
SET READ CONDITION TGR.
FIRST CHARACTER TAPE MARK
FIG. 174
SET FIRST CHARACTER TAPE MARK TGR.
PERFORM NORMAL READ CLOCK CYCLE CHECKS ON TAPE MARK CHARACTER, AND PROCESS TAPE MARK CHARACTER R-W REG AND LRCR.
RESET SKEW REGISTERS, FIRST BIT TGR, AND READ CLOCK.
SET RDD TGR; START DELAY COUNTER.
HOLD TAPE MARK CHARACTER IN R-W REG UNTIL NEXT CHARACTER FROM TAPE UNIT ARRIVES.
RECEIVE INPUT CHARACTER FROM TAPE UNIT
FIG. 173
PROCESS TAPE CHARACTER THROUGH FINAL AMPS TO SKEW REGS.
SET FIRST BIT TGR; START READ CLOCK.
RESET RDD TGR; DELAY COUNTER STOPS.
FIRST CHAR TAPE MARK
OR
SECOND CHARACTER NOT CHECK CHARACTER
FIG. 174
BEGIN READ CLOCK CYCLE.
TAKE FORCED SHIFT CYCLE TO PROCESS FIRST CHARACTER (IN R-W REG) TO DATA REG AND TO CHANNEL (IF TRANSFER TO CHANNEL IS REQUIRED).
PROCESS SECOND CHARACTER (IN SKEW REGISTERS) AT NORMAL READ CLOCK TIMES.
SECOND CHARACTER CHECK CHARACTER
FIG. 174
SET EXCEPTIONAL CONDITION TGR.
NORMAL READ CLOCK CYCLE
FIGS. 175, 176
PERFORM HI-CLIP VRC AND COMPARE CHECKS.
SET LRCR AND R-W REGISTERS.
RESET SKEW REGISTERS, FIRST BIT TGR, AND READ CLOCK.

FIG.171B

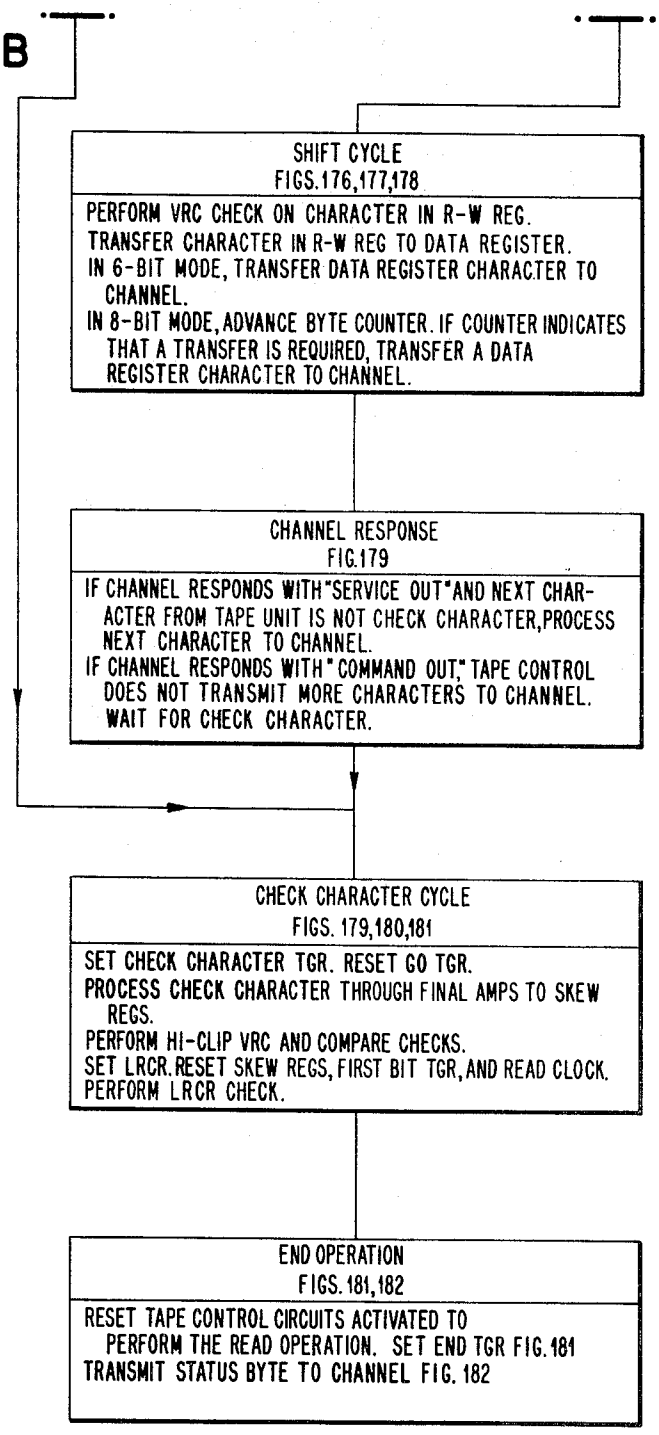

SHIFT CYCLE
FIGS.176,177,178
PERFORM VRC CHECK ON CHARACTER IN R-W REG.
TRANSFER CHARACTER IN R-W REG TO DATA REGISTER.
IN 6-BIT MODE, TRANSFER DATA REGISTER CHARACTER TO CHANNEL.
IN 8-BIT MODE, ADVANCE BYTE COUNTER. IF COUNTER INDICATES THAT A TRANSFER IS REQUIRED, TRANSFER A DATA REGISTER CHARACTER TO CHANNEL.
CHANNEL RESPONSE
FIG.179
IF CHANNEL RESPONDS WITH "SERVICE OUT" AND NEXT CHARACTER FROM TAPE UNIT IS NOT CHECK CHARACTER, PROCESS NEXT CHARACTER TO CHANNEL.
IF CHANNEL RESPONDS WITH "COMMAND OUT," TAPE CONTROL DOES NOT TRANSMIT MORE CHARACTERS TO CHANNEL. WAIT FOR CHECK CHARACTER.
CHECK CHARACTER CYCLE
FIGS. 179,180,181
SET CHECK CHARACTER TGR. RESET GO TGR.
PROCESS CHECK CHARACTER THROUGH FINAL AMPS TO SKEW REGS.
PERFORM HI-CLIP VRC AND COMPARE CHECKS.
SET LRCR. RESET SKEW REGS, FIRST BIT TGR, AND READ CLOCK.
PERFORM LRCR CHECK.
END OPERATION
FIGS. 181,182
RESET TAPE CONTROL CIRCUITS ACTIVATED TO PERFORM THE READ OPERATION. SET END TGR FIG. 181
TRANSMIT STATUS BYTE TO CHANNEL FIG. 182

FIG. 172A

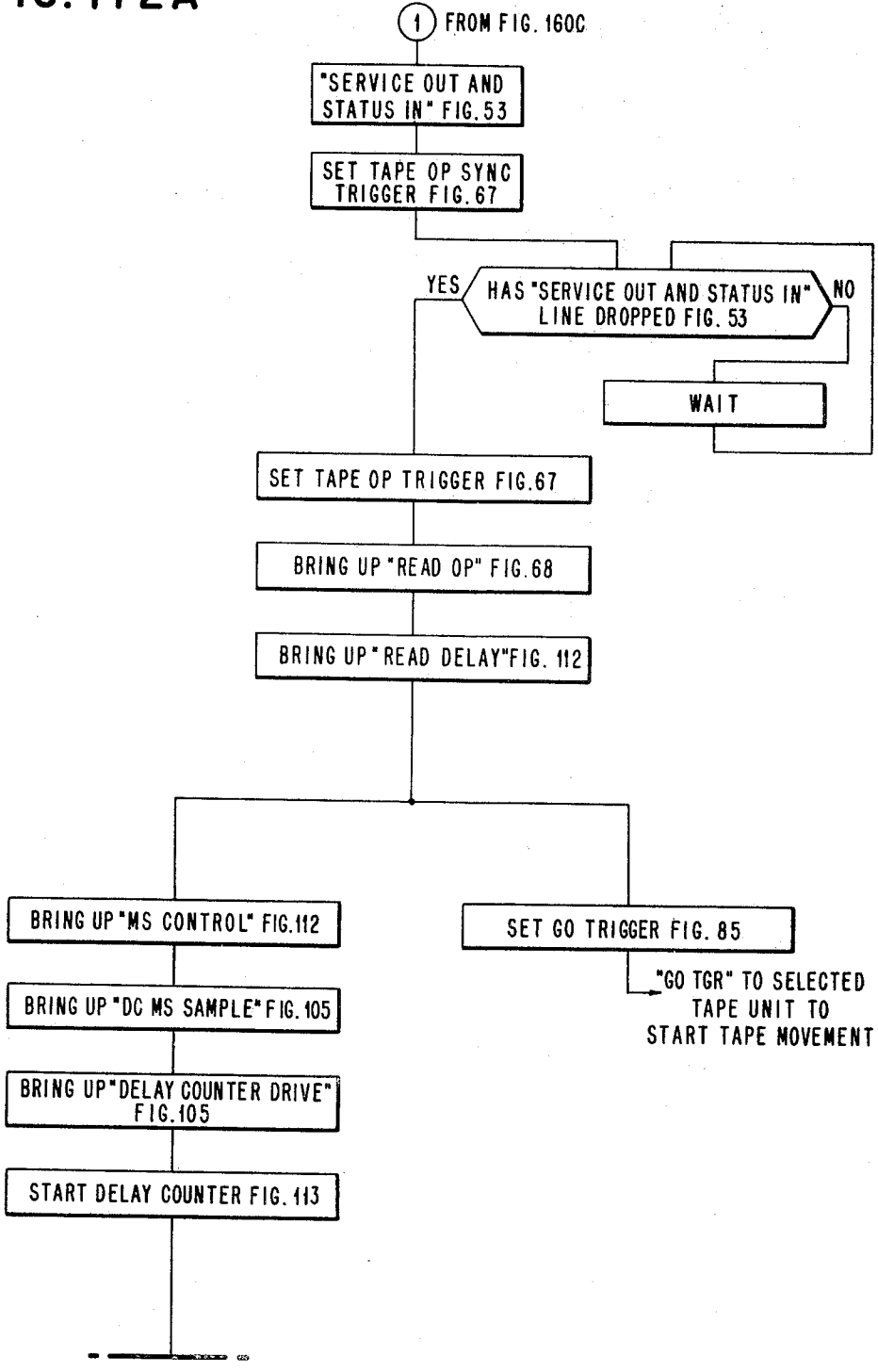

1 FROM FIG. 160C
"SERVICE OUT AND STATUS IN" FIG. 53
SET TAPE OP SYNC TRIGGER FIG. 67
YES
HAS "SERVICE OUT AND STATUS IN" LINE DROPPED FIG. 53
NO
WAIT
SET TAPE OP TRIGGER FIG. 67
BRING UP "READ OP" FIG. 68
BRING UP "READ DELAY" FIG. 112
BRING UP "MS CONTROL" FIG. 112
BRING UP "DC MS SAMPLE" FIG. 105
BRING UP "DELAY COUNTER DRIVE" FIG. 105
START DELAY COUNTER FIG. 113
SET GO TRIGGER FIG. 85
"GO TGR" TO SELECTED TAPE UNIT TO START TAPE MOVEMENT

FIG. 172B

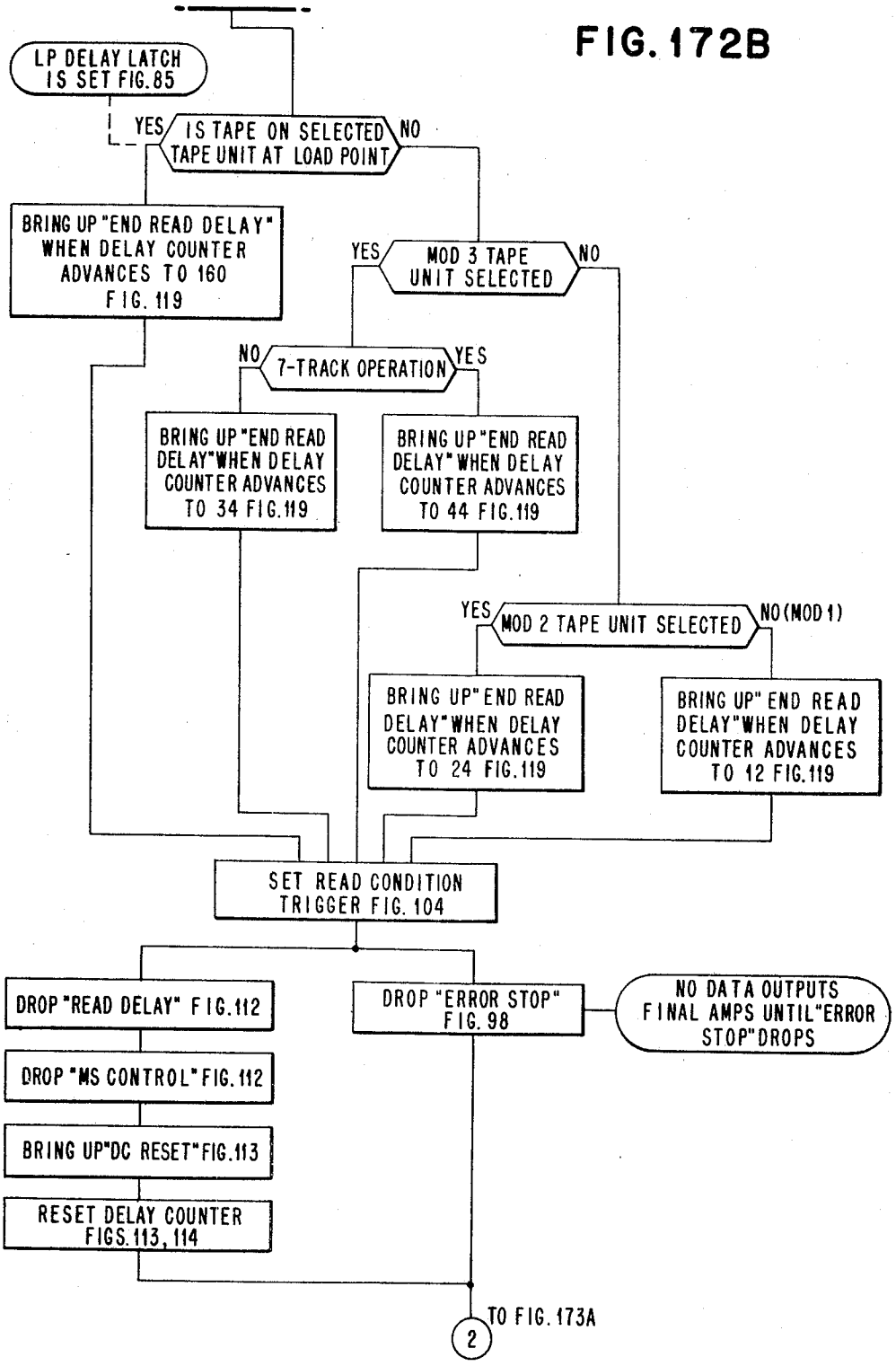

LP DELAY LATCH IS SET FIG. 85
IS TAPE ON SELECTED TAPE UNIT AT LOAD POINT
YES
NO
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 160 FIG. 119
MOD 3 TAPE UNIT SELECTED
YES
NO
7-TRACK OPERATION
NO
YES
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 34 FIG. 119
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 44 FIG. 119
MOD 2 TAPE UNIT SELECTED
YES
NO (MOD 1)
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 24 FIG. 119
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 12 FIG. 119
SET READ CONDITION TRIGGER FIG. 104
DROP "READ DELAY" FIG. 112
DROP "MS CONTROL" FIG. 112
BRING UP "DC RESET" FIG. 113
RESET DELAY COUNTER FIGS. 113, 114
DROP "ERROR STOP" FIG. 98
NO DATA OUTPUTS FINAL AMPS UNTIL "ERROR STOP" DROPS
2
TO FIG. 173A

FIG. 173A

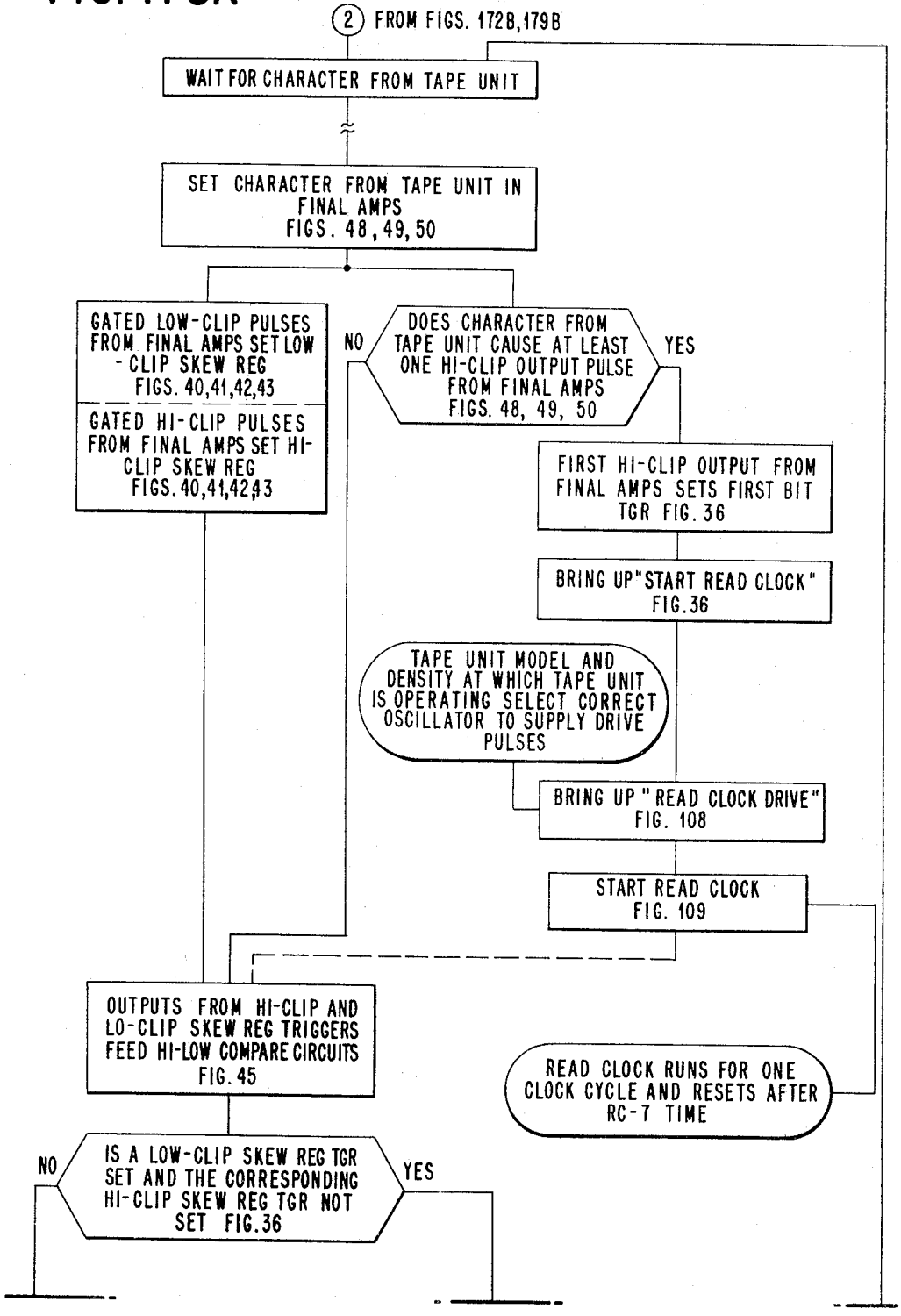
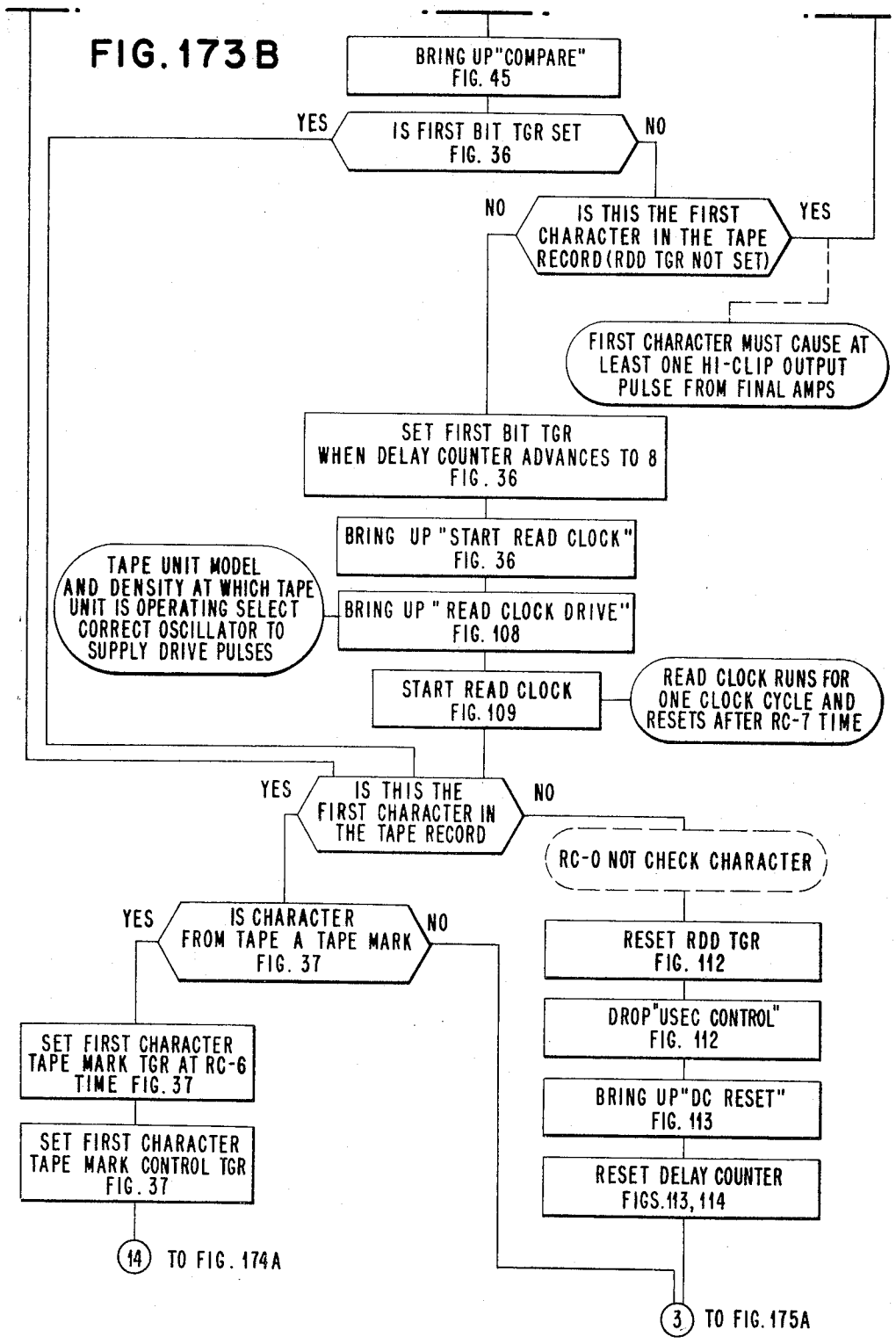

2 FROM FIGS. 172B,179B
WAIT FOR CHARACTER FROM TAPE UNIT
SET CHARACTER FROM TAPE UNIT IN FINAL AMPS FIGS. 48, 49, 50
GATED LOW-CLIP PULSES FROM FINAL AMPS SET LOW-CLIP SKEW REG FIGS. 40,41,42,43
GATED HI-CLIP PULSES FROM FINAL AMPS SET HI-CLIP SKEW REG FIGS. 40,41,42,43
DOES CHARACTER FROM TAPE UNIT CAUSE AT LEAST ONE HI-CLIP OUTPUT PULSE FROM FINAL AMPS FIGS. 48, 49, 50
NO
YES
FIRST HI-CLIP OUTPUT FROM FINAL AMPS SETS FIRST BIT TGR FIG. 36
BRING UP "START READ CLOCK" FIG. 36
TAPE UNIT MODEL AND DENSITY AT WHICH TAPE UNIT IS OPERATING SELECT CORRECT OSCILLATOR TO SUPPLY DRIVE PULSES
BRING UP "READ CLOCK DRIVE" FIG. 108
START READ CLOCK FIG. 109
OUTPUTS FROM HI-CLIP AND LO-CLIP SKEW REG TRIGGERS FEED HI-LOW COMPARE CIRCUITS FIG. 45
READ CLOCK RUNS FOR ONE CLOCK CYCLE AND RESETS AFTER RC-7 TIME
IS A LOW-CLIP SKEW REG TGR SET AND THE CORRESPONDING HI-CLIP SKEW REG TGR NOT SET FIG. 36
NO
YES

FIG. 173B


BRING UP "COMPARE" FIG. 45
YES
IS FIRST BIT TGR SET FIG. 36
NO
NO
IS THIS THE FIRST CHARACTER IN THE TAPE RECORD (RDD TGR NOT SET)
YES
FIRST CHARACTER MUST CAUSE AT LEAST ONE HI-CLIP OUTPUT PULSE FROM FINAL AMPS
SET FIRST BIT TGR WHEN DELAY COUNTER ADVANCES TO 8 FIG. 36
BRING UP "START READ CLOCK" FIG. 36
TAPE UNIT MODEL AND DENSITY AT WHICH TAPE UNIT IS OPERATING SELECT CORRECT OSCILLATOR TO SUPPLY DRIVE PULSES
BRING UP "READ CLOCK DRIVE" FIG. 108
START READ CLOCK FIG. 109
READ CLOCK RUNS FOR ONE CLOCK CYCLE AND RESETS AFTER RC-7 TIME
YES
IS THIS THE FIRST CHARACTER IN THE TAPE RECORD
NO
RC-0 NOT CHECK CHARACTER
YES
IS CHARACTER FROM TAPE A TAPE MARK FIG. 37
NO
RESET RDD TGR FIG. 112
DROP "USEC CONTROL" FIG. 112
BRING UP "DC RESET" FIG. 113
RESET DELAY COUNTER FIGS. 113, 114
SET FIRST CHARACTER TAPE MARK TGR AT RC-6 TIME FIG. 37
SET FIRST CHARACTER TAPE MARK CONTROL TGR FIG. 37
14 TO FIG. 174A
3 TO FIG. 175A

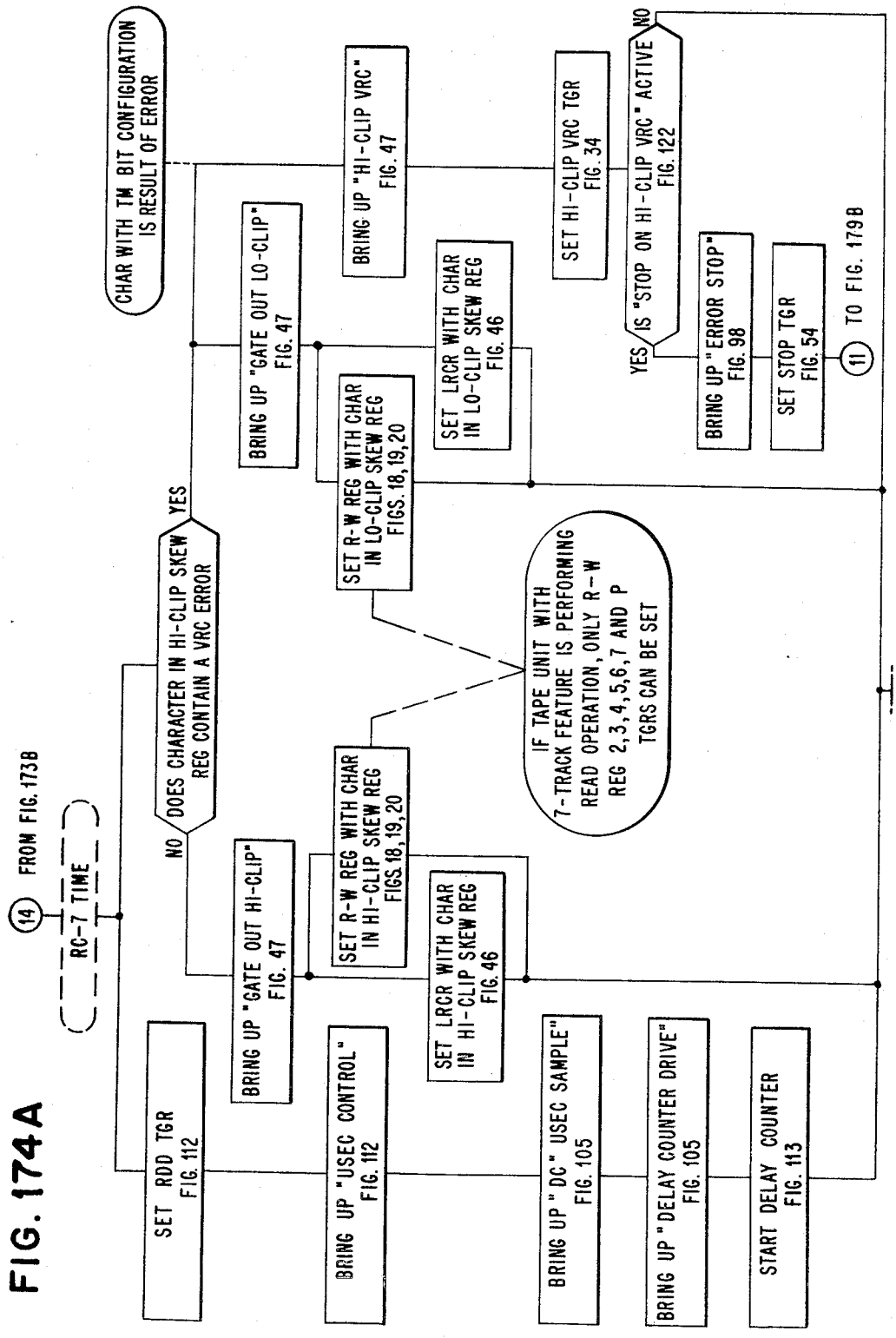

FROM FIG. 173B
14
RC-7 TIME
SET RDD TGR FIG. 112
BRING UP "USEC CONTROL" FIG. 112
BRING UP "DC" USEC SAMPLE" FIG. 105
BRING UP "DELAY COUNTER DRIVE" FIG. 105
START DELAY COUNTER FIG. 113
DOES CHARACTER IN HI-CLIP SKEW REG CONTAIN A VRC ERROR
NO
YES
BRING UP "GATE OUT HI-CLIP" FIG. 47
SET R-W REG WITH CHAR IN HI-CLIP SKEW REG FIGS. 18, 19, 20
SET LRCR WITH CHAR IN HI-CLIP SKEW REG FIG. 46
BRING UP "GATE OUT LO-CLIP" FIG. 47
SET R-W REG WITH CHAR IN LO-CLIP SKEW REG FIGS. 18, 19, 20
SET LRCR WITH CHAR IN LO-CLIP SKEW REG FIG. 46
IF TAPE UNIT WITH 7-TRACK FEATURE IS PERFORMING READ OPERATION, ONLY R-W REG 2, 3, 4, 5, 6, 7 AND P TGRS CAN BE SET
CHAR WITH TM BIT CONFIGURATION IS RESULT OF ERROR
BRING UP "HI-CLIP VRC" FIG. 47
SET HI-CLIP VRC TGR FIG. 34
IS "STOP ON HI-CLIP VRC" ACTIVE
NO
YES
BRING UP "ERROR STOP" FIG. 98
SET STOP TGR FIG. 54
11
TO FIG. 179B

FIG. 174A

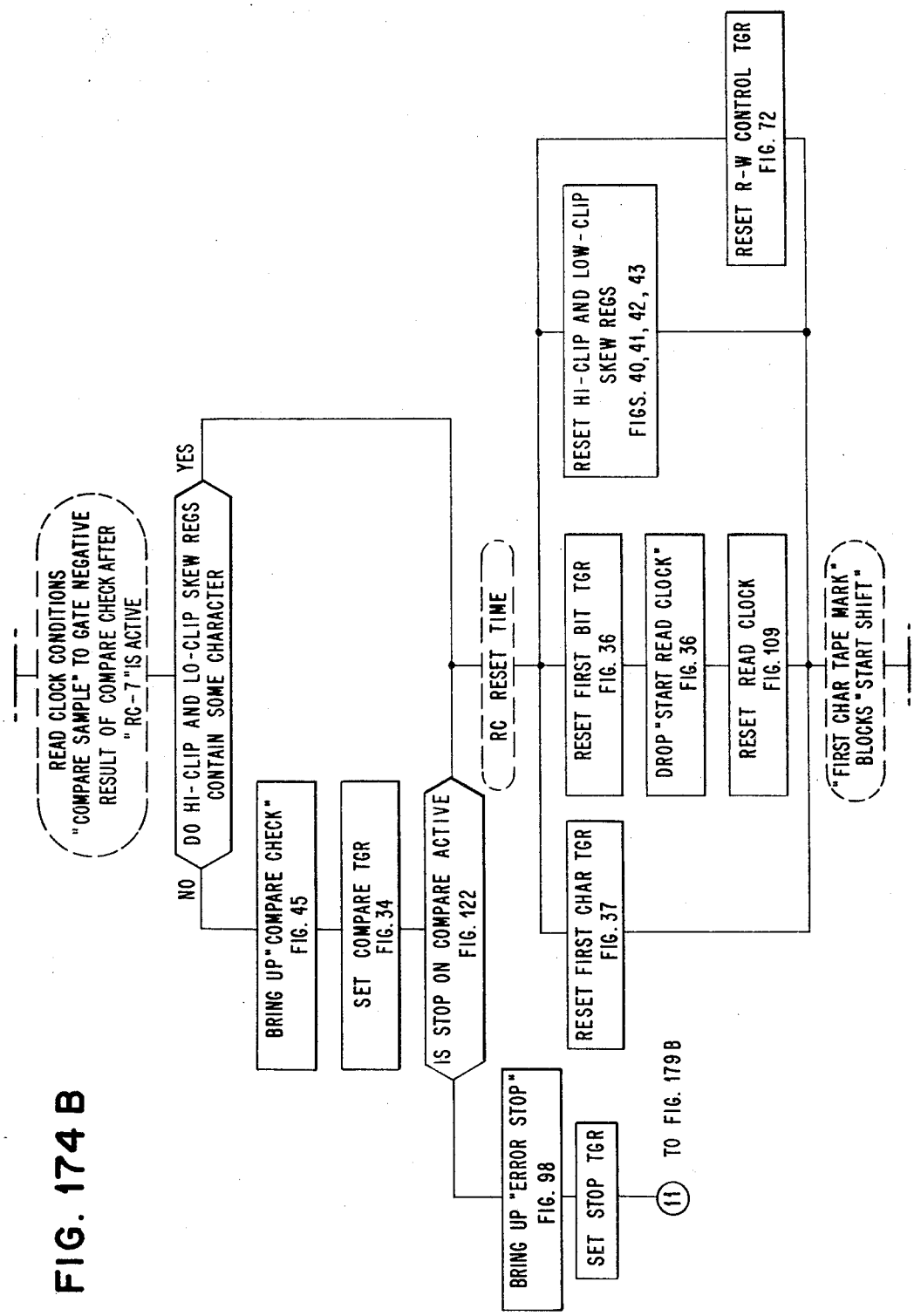

READ CLOCK CONDITIONS "COMPARE SAMPLE" TO GATE NEGATIVE RESULT OF COMPARE CHECK AFTER "RC-7" IS ACTIVE
DO HI-CLIP AND LO-CLIP SKEW REGS CONTAIN SOME CHARACTER
YES
NO
BRING UP "COMPARE CHECK" FIG. 45
SET COMPARE TGR FIG. 34
IS STOP ON COMPARE ACTIVE FIG. 122
RC RESET TIME
RESET HI-CLIP AND LOW-CLIP SKEW REGS FIGS. 40, 41, 42, 43
RESET R-W CONTROL TGR FIG. 72
RESET FIRST BIT TGR FIG. 36
DROP "START READ CLOCK" FIG. 36
RESET READ CLOCK FIG. 109
"FIRST CHAR TAPE MARK" BLOCKS "START SHIFT"
RESET FIRST CHAR TGR FIG. 37
BRING UP "ERROR STOP" FIG. 98
SET STOP TGR
(11) TO FIG. 179 B

FIG. 174 B

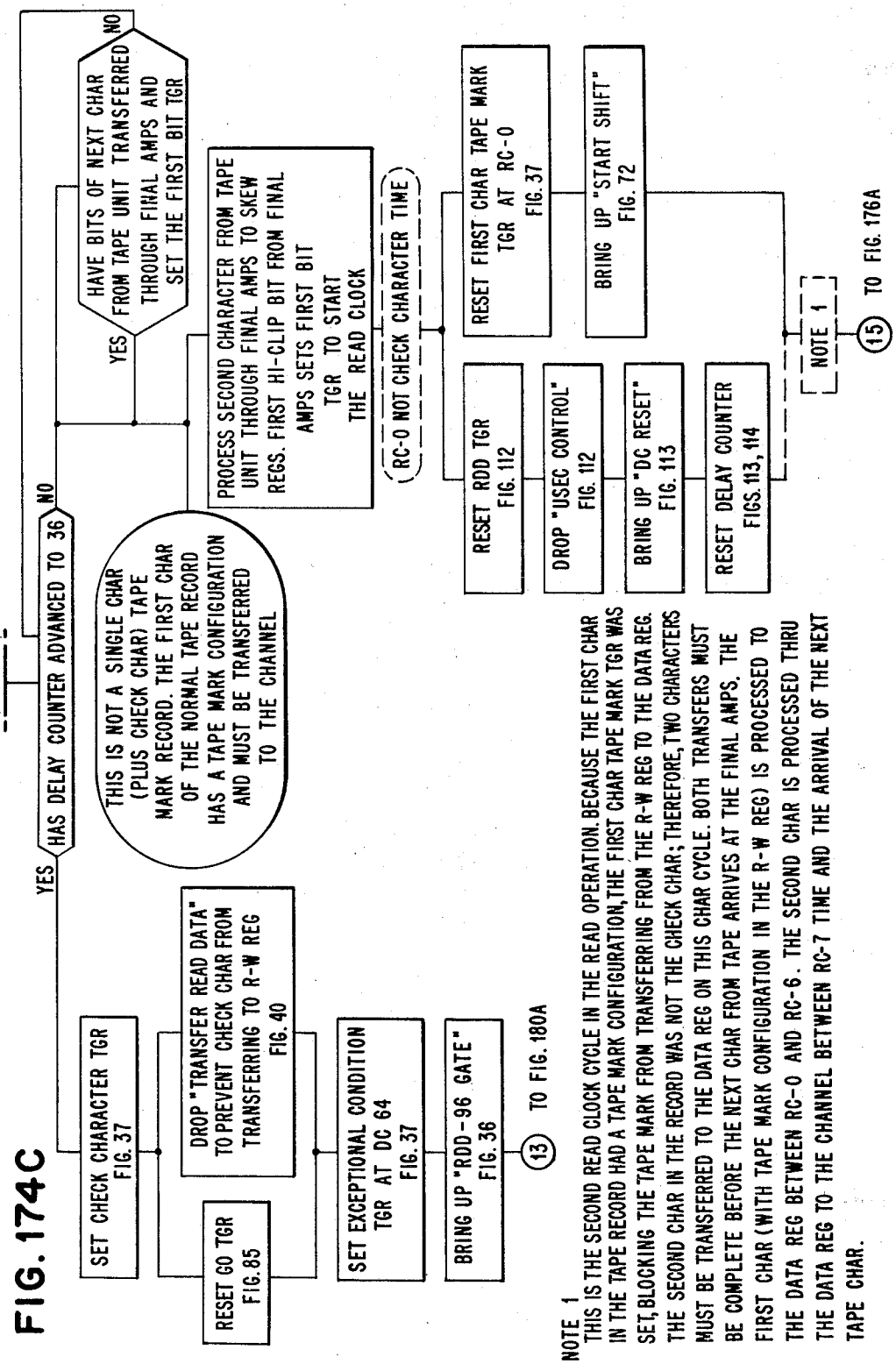



FIG. 174C

NOTE 1
THIS IS THE SECOND READ CLOCK CYCLE IN THE READ OPERATION. BECAUSE THE FIRST CHAR IN THE TAPE RECORD HAD A TAPE MARK CONFIGURATION, THE FIRST CHAR TAPE MARK TGR WAS SET, BLOCKING THE TAPE MARK FROM TRANSFERRING FROM THE R-W REG TO THE DATA REG. THE SECOND CHAR IN THE RECORD WAS NOT THE CHECK CHAR; THEREFORE, TWO CHARACTERS MUST BE TRANSFERRED TO THE DATA REG ON THIS CHAR CYCLE. BOTH TRANSFERS MUST BE COMPLETE BEFORE THE NEXT CHAR FROM TAPE ARRIVES AT THE FINAL AMPS. THE FIRST CHAR (WITH TAPE MARK CONFIGURATION IN THE R-W REG) IS PROCESSED TO THE DATA REG BETWEEN RC-0 AND RC-6. THE SECOND CHAR IS PROCESSED THRU THE DATA REG TO THE CHANNEL BETWEEN RC-7 TIME AND THE ARRIVAL OF THE NEXT TAPE CHAR.

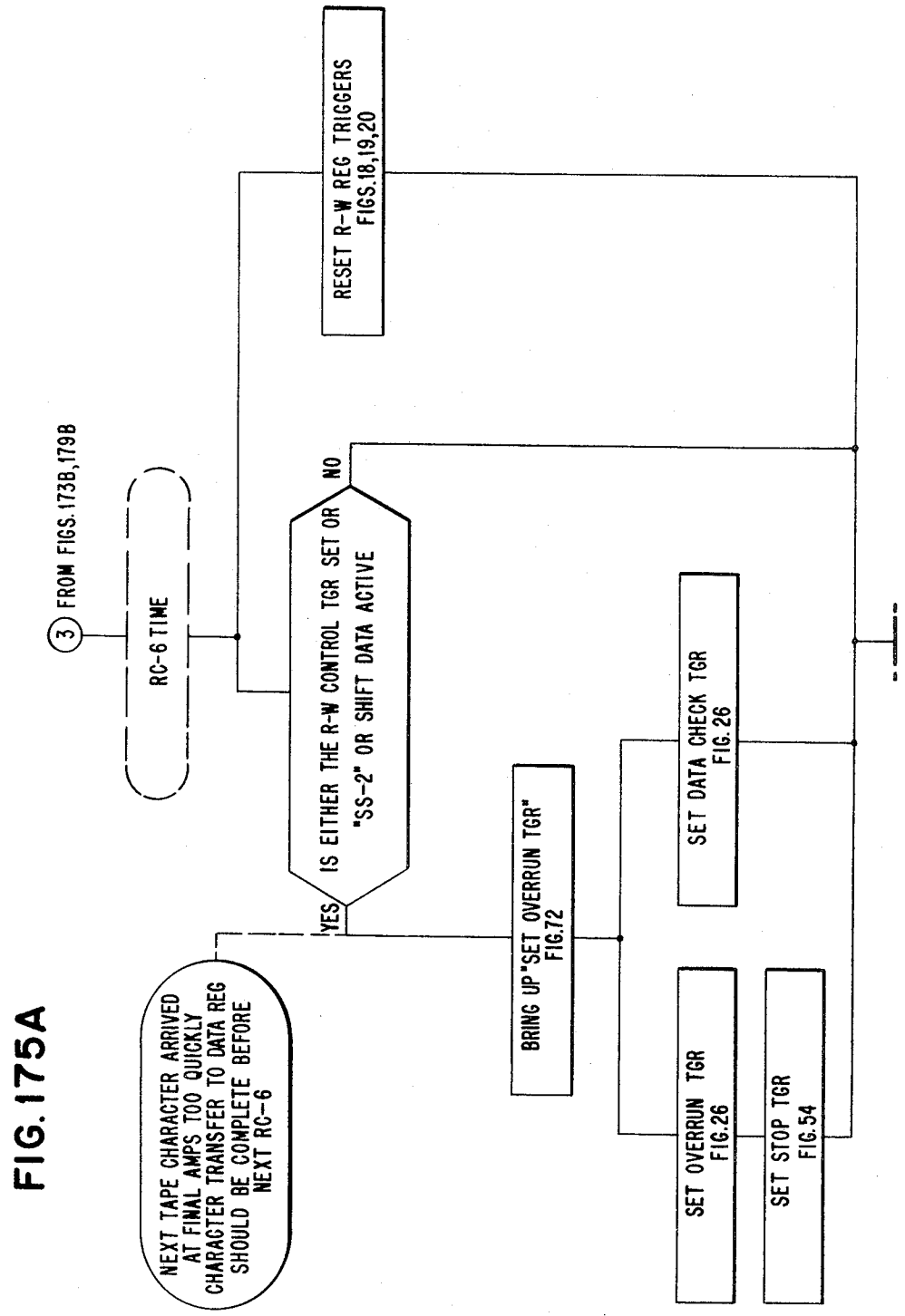

RESET R-W REG TRIGGERS FIGS.18,19,20
3 FROM FIGS.173B,179B
RC-6 TIME
IS EITHER THE R-W CONTROL TGR SET OR "SS-2" OR SHIFT DATA ACTIVE
NO
YES
NEXT TAPE CHARACTER ARRIVED AT FINAL AMPS TOO QUICKLY CHARACTER TRANSFER TO DATA REG SHOULD BE COMPLETE BEFORE NEXT RC-6
BRING UP "SET OVERRUN TGR" FIG.72
SET DATA CHECK TGR FIG.26
SET OVERRUN TGR FIG.26
SET STOP TGR FIG.54

FIG.175A

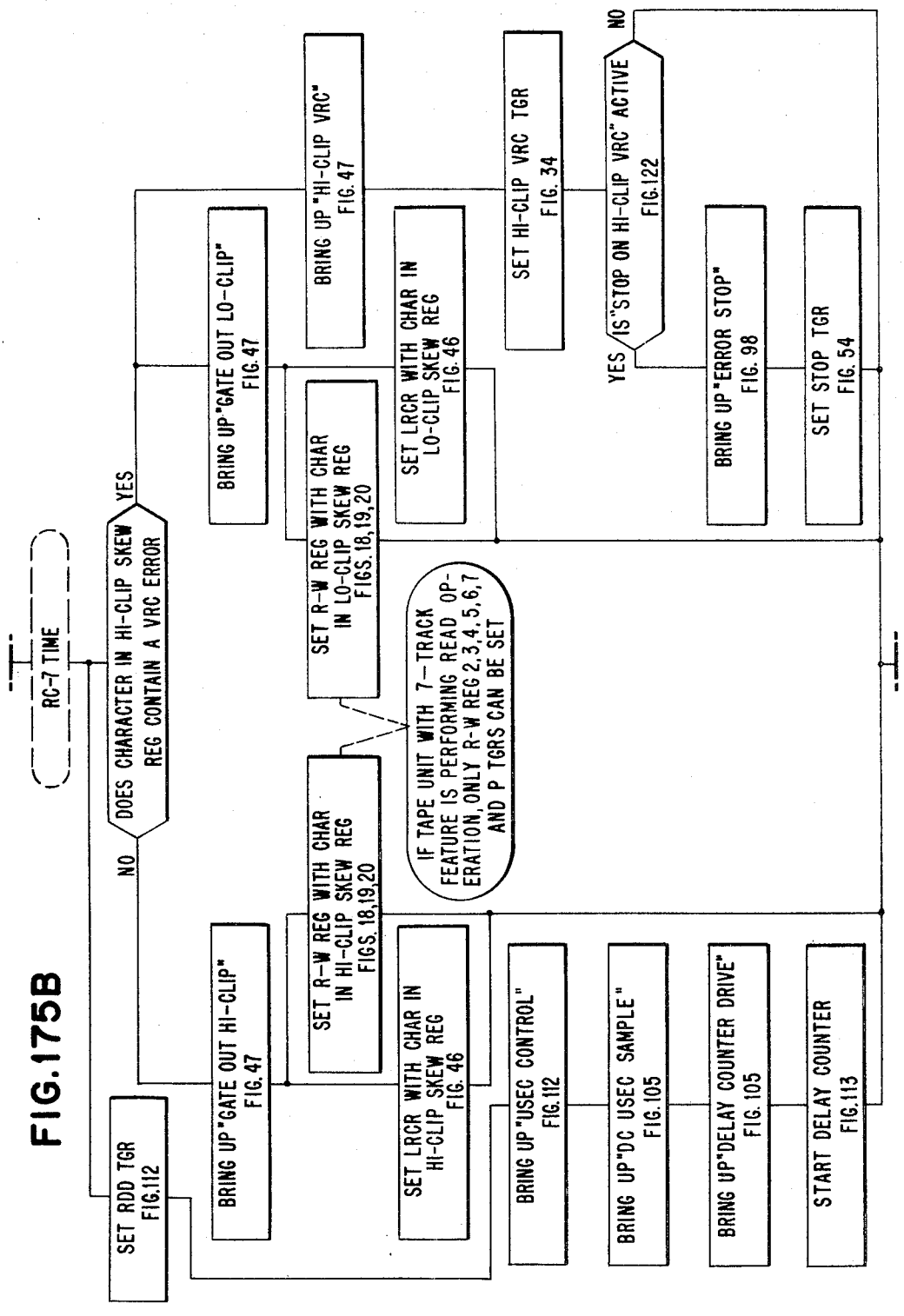

RC-7 TIME
DOES CHARACTER IN HI-CLIP SKEW REG CONTAIN A VRC ERROR
YES
NO
BRING UP "GATE OUT LO-CLIP" FIG. 47
BRING UP "HI-CLIP VRC" FIG. 47
SET R-W REG WITH CHAR IN LO-CLIP SKEW REG FIGS. 18,19,20
SET LRCR WITH CHAR IN LO-CLIP SKEW REG FIG. 46
SET HI-CLIP VRC TGR FIG. 34
IS "STOP ON HI-CLIP VRC" ACTIVE
NO
YES
BRING UP "ERROR STOP" FIG. 98
SET STOP TGR FIG. 54
IF TAPE UNIT WITH 7-TRACK FEATURE IS PERFORMING READ OPERATION, ONLY R-W REG 2,3,4,5,6,7 AND P TGRS CAN BE SET
SET R-W REG WITH CHAR IN HI-CLIP SKEW REG FIGS. 18,19,20
SET LRCR WITH CHAR IN HI-CLIP SKEW REG FIG. 46
BRING UP "GATE OUT HI-CLIP" FIG. 47
SET RDD TGR FIG. 112
BRING UP "USEC CONTROL" FIG. 112
BRING UP "DC USEC SAMPLE" FIG. 105
BRING UP "DELAY COUNTER DRIVE" FIG. 105
START DELAY COUNTER FIG. 113

FIG. 175B

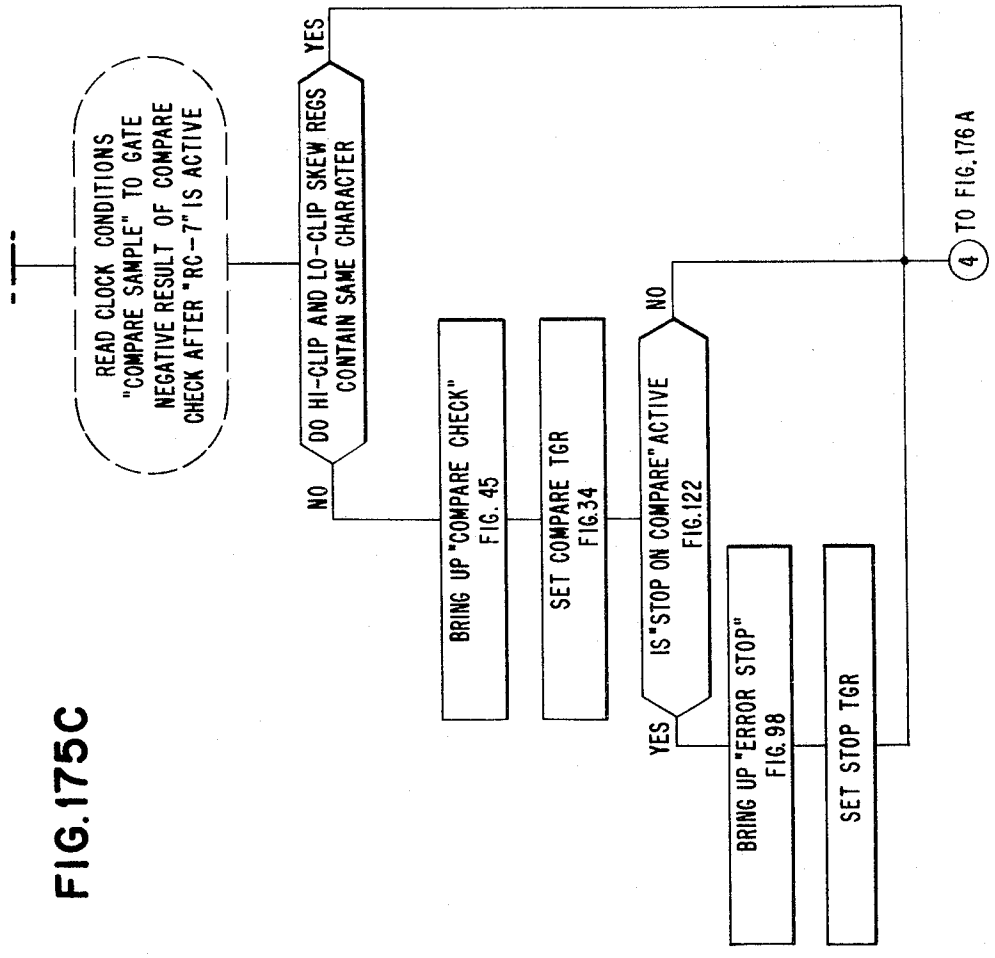

READ CLOCK CONDITIONS "COMPARE SAMPLE" TO GATE NEGATIVE RESULT OF COMPARE CHECK AFTER "RC-7" IS ACTIVE
DO HI-CLIP AND LO-CLIP SKEW REGS CONTAIN SAME CHARACTER
YES
NO
BRING UP "COMPARE CHECK" FIG. 45
SET COMPARE TGR FIG.34
IS "STOP ON COMPARE" ACTIVE FIG.122
NO
YES
BRING UP "ERROR STOP" FIG.98
SET STOP TGR
4 TO FIG.176A

FIG.175C

FIG. 176A

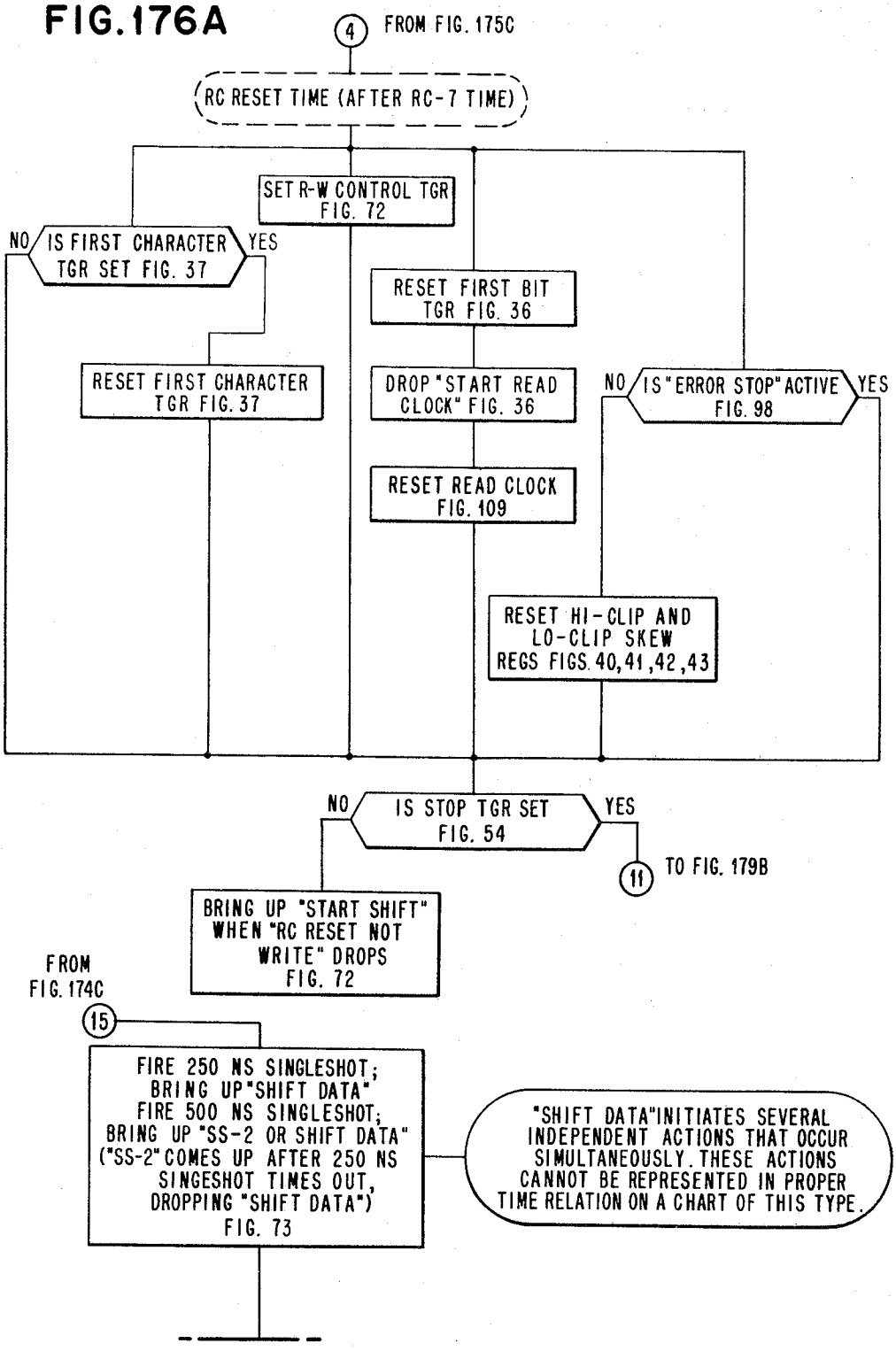
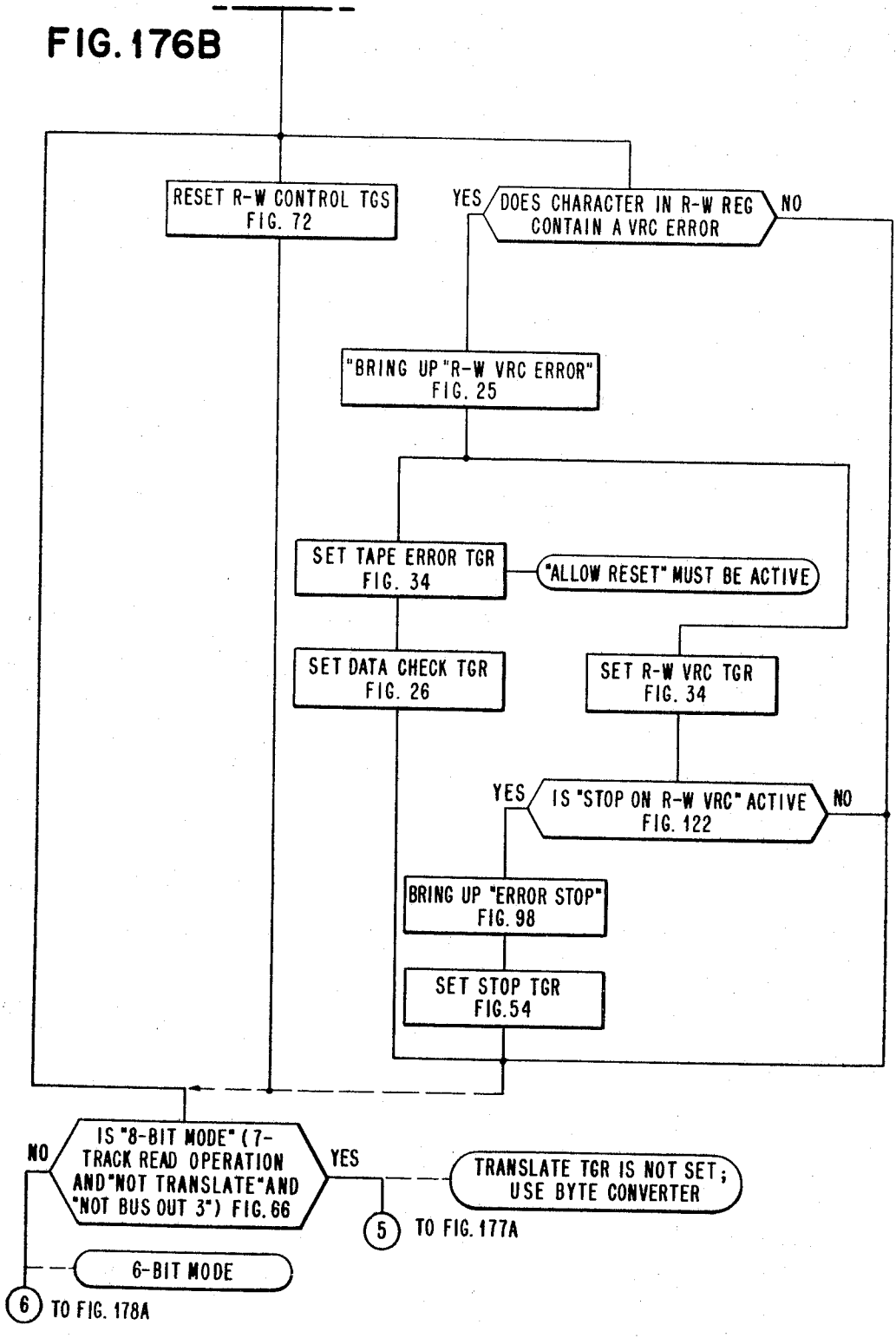

4
FROM FIG. 175C
RC RESET TIME (AFTER RC-7 TIME)
SET R-W CONTROL TGR FIG. 72
NO
IS FIRST CHARACTER TGR SET FIG. 37
YES
RESET FIRST CHARACTER TGR FIG. 37
RESET FIRST BIT TGR FIG. 36
DROP "START READ CLOCK" FIG. 36
RESET READ CLOCK FIG. 109
NO
IS "ERROR STOP" ACTIVE FIG. 98
YES
RESET HI-CLIP AND LO-CLIP SKEW REGS FIGS. 40,41,42,43
NO
IS STOP TGR SET FIG. 54
YES
11
TO FIG. 179B
BRING UP "START SHIFT" WHEN "RC RESET NOT WRITE" DROPS FIG. 72
FROM FIG. 174C
15
FIRE 250 NS SINGLESHOT; BRING UP "SHIFT DATA" FIRE 500 NS SINGLESHOT; BRING UP "SS-2 OR SHIFT DATA" ("SS-2" COMES UP AFTER 250 NS SINGESHOT TIMES OUT, DROPPING "SHIFT DATA") FIG. 73
"SHIFT DATA" INITIATES SEVERAL INDEPENDENT ACTIONS THAT OCCUR SIMULTANEOUSLY. THESE ACTIONS CANNOT BE REPRESENTED IN PROPER TIME RELATION ON A CHART OF THIS TYPE.

FIG. 176B


RESET R-W CONTROL TGS FIG. 72
YES
DOES CHARACTER IN R-W REG CONTAIN A VRC ERROR
NO
"BRING UP "R-W VRC ERROR" FIG. 25
SET TAPE ERROR TGR FIG. 34
"ALLOW RESET" MUST BE ACTIVE
SET DATA CHECK TGR FIG. 26
SET R-W VRC TGR FIG. 34
YES
IS "STOP ON R-W VRC" ACTIVE FIG. 122
NO
BRING UP "ERROR STOP" FIG. 98
SET STOP TGR FIG. 54
NO
IS "8-BIT MODE" (7-TRACK READ OPERATION AND "NOT TRANSLATE" AND "NOT BUS OUT 3") FIG. 66
YES
TRANSLATE TGR IS NOT SET; USE BYTE CONVERTER
5
TO FIG. 177A
6-BIT MODE
6
TO FIG. 178A

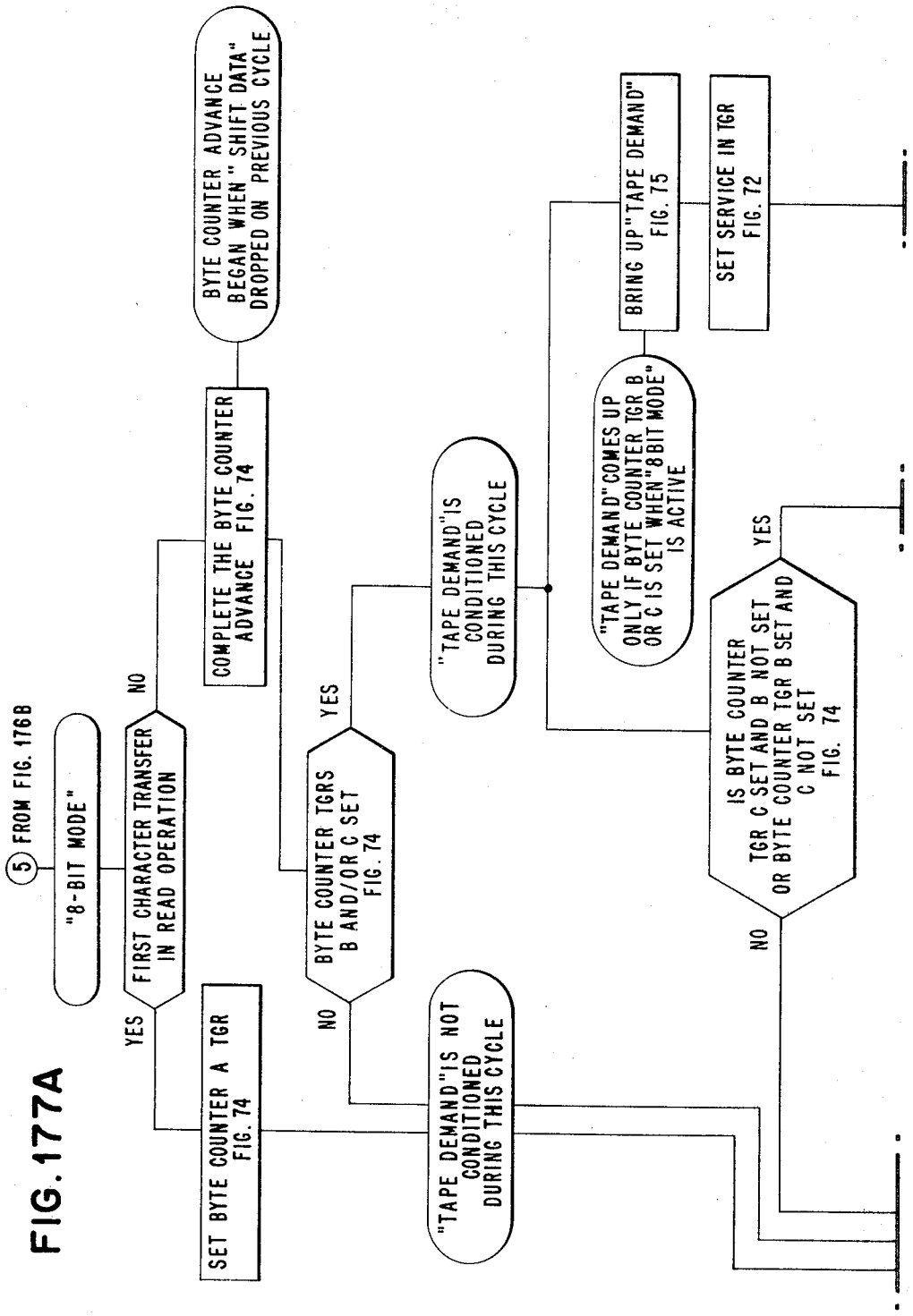

5 FROM FIG. 176B
"8-BIT MODE"
FIRST CHARACTER TRANSFER IN READ OPERATION
NO
YES
SET BYTE COUNTER A TGR FIG. 74
COMPLETE THE BYTE COUNTER ADVANCE FIG. 74
BYTE COUNTER ADVANCE BEGAN WHEN " SHIFT DATA" DROPPED ON PREVIOUS CYCLE
BYTE COUNTER TGRS B AND/OR C SET FIG. 74
YES
NO
"TAPE DEMAND" IS CONDITIONED DURING THIS CYCLE
"TAPE DEMAND" IS NOT CONDITIONED DURING THIS CYCLE
"TAPE DEMAND" COMES UP ONLY IF BYTE COUNTER TGR B OR C IS SET WHEN "8 BIT MODE" IS ACTIVE
BRING UP "TAPE DEMAND" FIG. 75
SET SERVICE IN TGR FIG. 72
IS BYTE COUNTER TGR C SET AND B NOT SET OR BYTE COUNTER TGR B SET AND C NOT SET FIG. 74
YES
NO

FIG. 177A

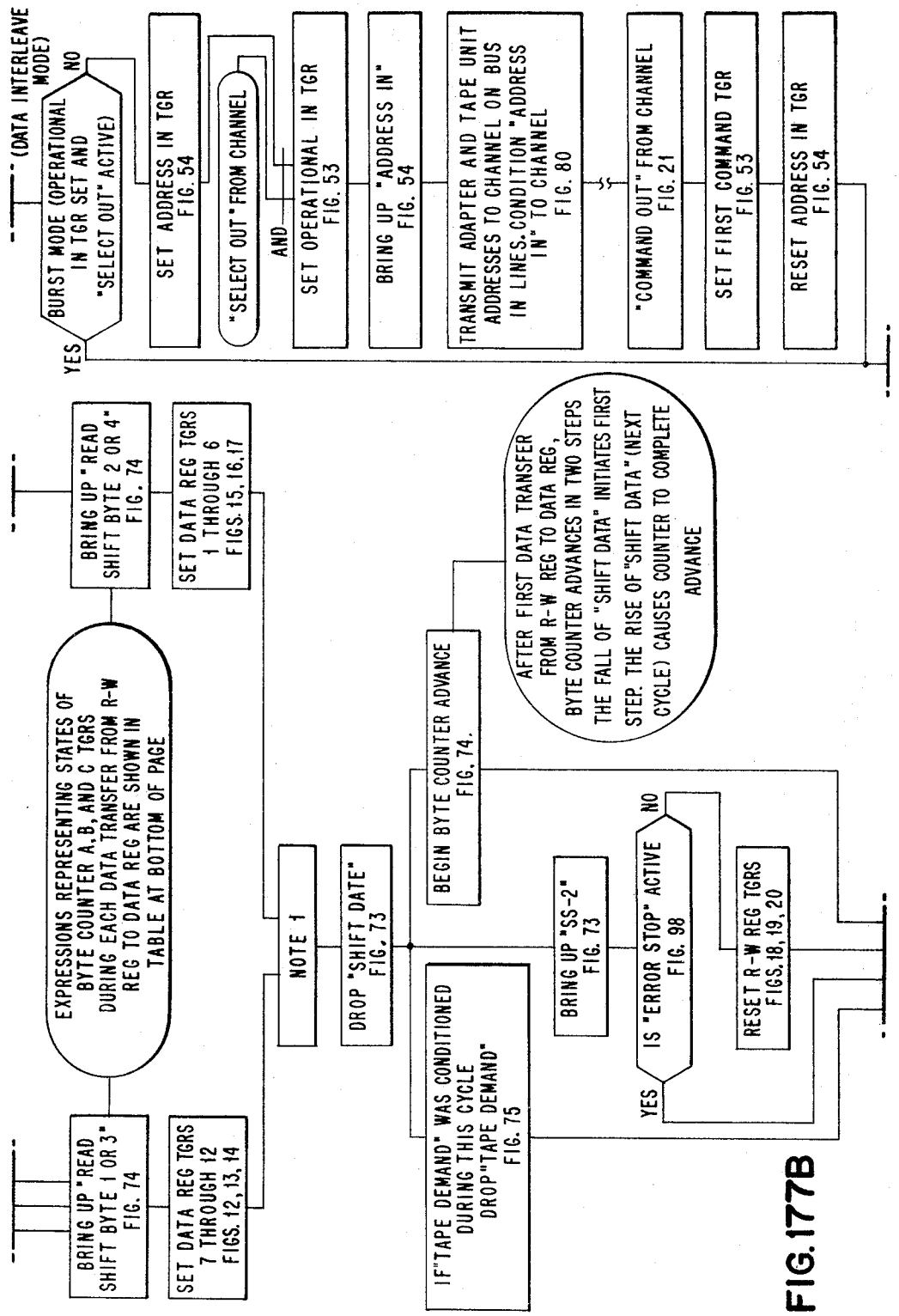

(DATA INTERLEAVE MODE)
BURST MODE (OPERATIONAL IN TGR SET AND "SELECT OUT" ACTIVE)
NO
YES
SET ADDRESS IN TGR FIG. 54
"SELECT OUT" FROM CHANNEL
AND
SET OPERATIONAL IN TGR FIG. 53
BRING UP "ADDRESS IN" FIG. 54
TRANSMIT ADAPTER AND TAPE UNIT ADDRESSES TO CHANNEL ON BUS IN LINES. CONDITION "ADDRESS IN" TO CHANNEL FIG. 80
"COMMAND OUT" FROM CHANNEL FIG. 21
SET FIRST COMMAND TGR FIG. 53
RESET ADDRESS IN TGR FIG. 54
BRING UP "READ SHIFT BYTE 2 OR 4" FIG. 74
SET DATA REG TGRS 1 THROUGH 6 FIGS. 15, 16, 17
EXPRESSIONS REPRESENTING STATES OF BYTE COUNTER A, B, AND C TGRS DURING EACH DATA TRANSFER FROM R-W REG TO DATA REG ARE SHOWN IN TABLE AT BOTTOM OF PAGE
AFTER FIRST DATA TRANSFER FROM R-W REG TO DATA REG, BYTE COUNTER ADVANCES IN TWO STEPS THE FALL OF "SHIFT DATA" INITIATES FIRST STEP. THE RISE OF "SHIFT DATA" (NEXT CYCLE) CAUSES COUNTER TO COMPLETE ADVANCE
BRING UP "READ SHIFT BYTE 1 OR 3" FIG. 74
SET DATA REG TGRS 7 THROUGH 12 FIGS. 12, 13, 14
NOTE 1
DROP "SHIFT DATE" FIG. 73
BEGIN BYTE COUNTER ADVANCE FIG. 74.
IF "TAPE DEMAND" WAS CONDITIONED DURING THIS CYCLE DROP "TAPE DEMAND" FIG. 75
BRING UP "SS-2" FIG. 73
IS "ERROR STOP" ACTIVE FIG. 98
NO
YES
RESET R-W REG TGRS FIGS. 18, 19, 20

FIG. 177B

FIG. 177C

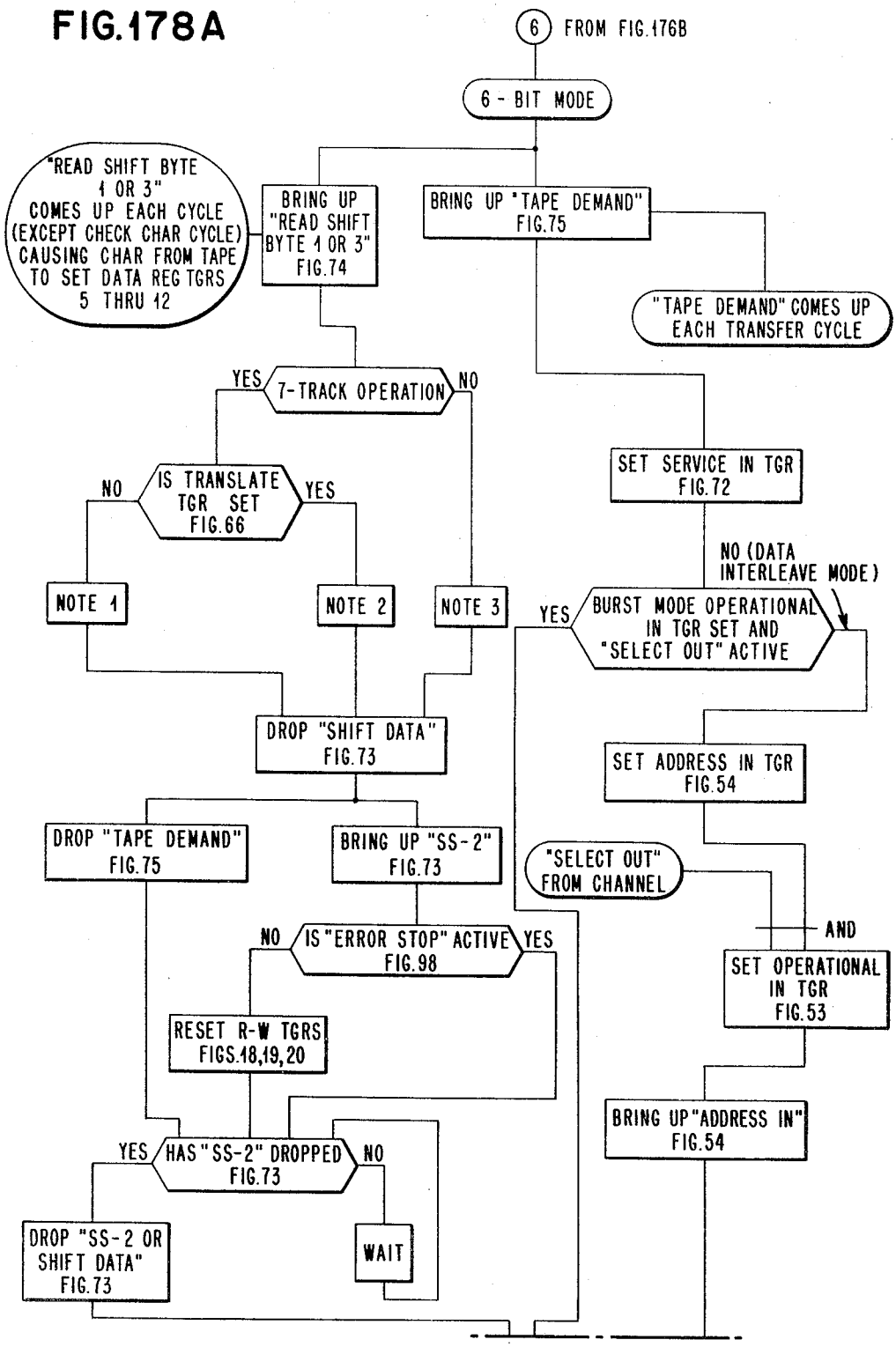



NOTE 1
ACTIVE OUTPUTS FROM R-W REG 5 AND 7 TGRS SET DATA REG TGRS 10 AND 12 ("RD SHIFT BYTE 1 OR 3") OR 4 AND 6 ("RD SHIFT BYTE 2 OR 4" RESPECTIVELY WITHOUT GOING THROUGH THE TRANSLATOR. ACTIVE OUTPUTS FROM R-W REG TGRS 2, 3, 4, AND 6 FEED THROUGH THE TRANSLATOR (TRANSLATION DOES NOT OCCUR) TO ACTIVATE READ TRANSLATE LINES 7, 8, 9, AND 11. "READ TRANSLATE 7" SETS DATA REG 7 TGR ("RD SHIFT BYTE 1 OR 3") OR DATA REG 1 TGR ("RD SHIFT BYTE 2 OR 4") RD TRANSLATE 12 "SETS DATA REG 12 TGR ("RD SHIFT BYTE 1 OR 3") OR DATA REG 6 TGR ("RD SHIFT BYTE 2 OR 4") THEREFORE, IF R-W TGRS 2 THROUGH 7 ARE SET, DATA REG TGRS 1 THROUGH 6 OR 7 THROUGH 12 ARE SET

| | | | |
|---|---|---|---|
| TRANSFER 1 | $A \cdot \bar{B} \cdot \bar{C}$ | TRANSFER 5 | $A \cdot \bar{B} \cdot \bar{C}$ |
| TRANSFER 2 | $\bar{A} \cdot B \cdot \bar{C}$ | TRANSFER 6 | $A \cdot B \cdot \bar{C}$ |
| TRANSFER 3 | $A \cdot B \cdot C$ | TRANSFER 7 | $A \cdot B \cdot C$ |
| TRANSFER 4 | $\bar{A} \cdot \bar{B} \cdot C$ | TRANSFER 8 | $\bar{A} \cdot \bar{B} \cdot C$ |

STATES OF BYTE COUNTER TRIGGERS AFTER ADVANCE IS COMPLETE

FIG.178A


6
FROM FIG.176B
6 - BIT MODE
"READ SHIFT BYTE 1 OR 3" COMES UP EACH CYCLE (EXCEPT CHECK CHAR CYCLE) CAUSING CHAR FROM TAPE TO SET DATA REG TGRS 5 THRU 12
BRING UP "READ SHIFT BYTE 1 OR 3" FIG.74
BRING UP "TAPE DEMAND" FIG.75
"TAPE DEMAND" COMES UP EACH TRANSFER CYCLE
YES
7-TRACK OPERATION
NO
NO
IS TRANSLATE TGR SET FIG.66
YES
NOTE 1
NOTE 2
NOTE 3
SET SERVICE IN TGR FIG.72
NO (DATA INTERLEAVE MODE)
YES
BURST MODE OPERATIONAL IN TGR SET AND "SELECT OUT" ACTIVE
DROP "SHIFT DATA" FIG.73
SET ADDRESS IN TGR FIG.54
DROP "TAPE DEMAND" FIG.75
BRING UP "SS-2" FIG.73
"SELECT OUT" FROM CHANNEL
AND
NO
IS "ERROR STOP" ACTIVE FIG.98
YES
SET OPERATIONAL IN TGR FIG.53
RESET R-W TGRS FIGS.18,19,20
BRING UP "ADDRESS IN" FIG.54
YES
HAS "SS-2" DROPPED FIG.73
NO
DROP "SS-2 OR SHIFT DATA" FIG.73
WAIT

# FIG.178B

NOTE 1

ACTIVE OUTPUTS FROM R-W REG 0,1,5 & 7 TGRS SET DATA REG TGRS 5,6,10 & 12 RESPECTIVELY. RECALL, HOWEVER, THAT DATA FROM TAPE DID NOT SET R-W REG 0 & 1 TGRS; ZEROS ARE INSERTED IN DATA REG POSITIONS 5 & 6. ACTIVE OUTPUTS FROM R-W REG TGRS 2,3,4 & 6 FEED THRU THE TRANSLATOR (TRANSLATION DOES NOT OCCUR) TO ACTIVATE READ TRANSLATE LINES 7, 8, 9 & 11. "READ TRANSLATE 7" SETS DATA REG 7 TGR. THEREFORE, IF R-W REG TGRS 2 THRU 7 ARE SET, DATA REG TGRS 7 THRU 12 ARE SET. THE CHAR GATED TO THE INTERFACE HAS ZEROS IN THE TWO HIGH ORDER POSITIONS. ONLY DATA REG 5 THRU 12 TGRS ARE USED IN THE 6-BIT MODE.

NOTE 2

"TRANSLATE" SETS DATA REG 6 TGR. ONLY ACTIVE OUTPUTS FROM R-W REG TGRS 5 & 7 SET DATA REG TGRS 10 & 12 RESPECTIVELY. OUTPUTS FROM R-W REG TGRS FEED THE BCD TO 8-BIT TRANSLATOR. THE TRANSLATOR CONVERTS DATA AND PRODUCES INPUT "READ TRANSLATE" LINES TO THE DATA REG. "READ TRANSLATE" LINES SET DATA REG 5,7,8,9 & 12 TGRS.

NOTE 3

GATE CHARACTER REPRESENTED BY STATES OF R-W REG TGRS TO DATA REG POSITIONS 5 THRU 12. (IF R W REG 0 TGR IS SET, SET DATA REG 5 TGR; IF R-W REG 7 TGR IS SET, SET DATA REG 12 TGR.)

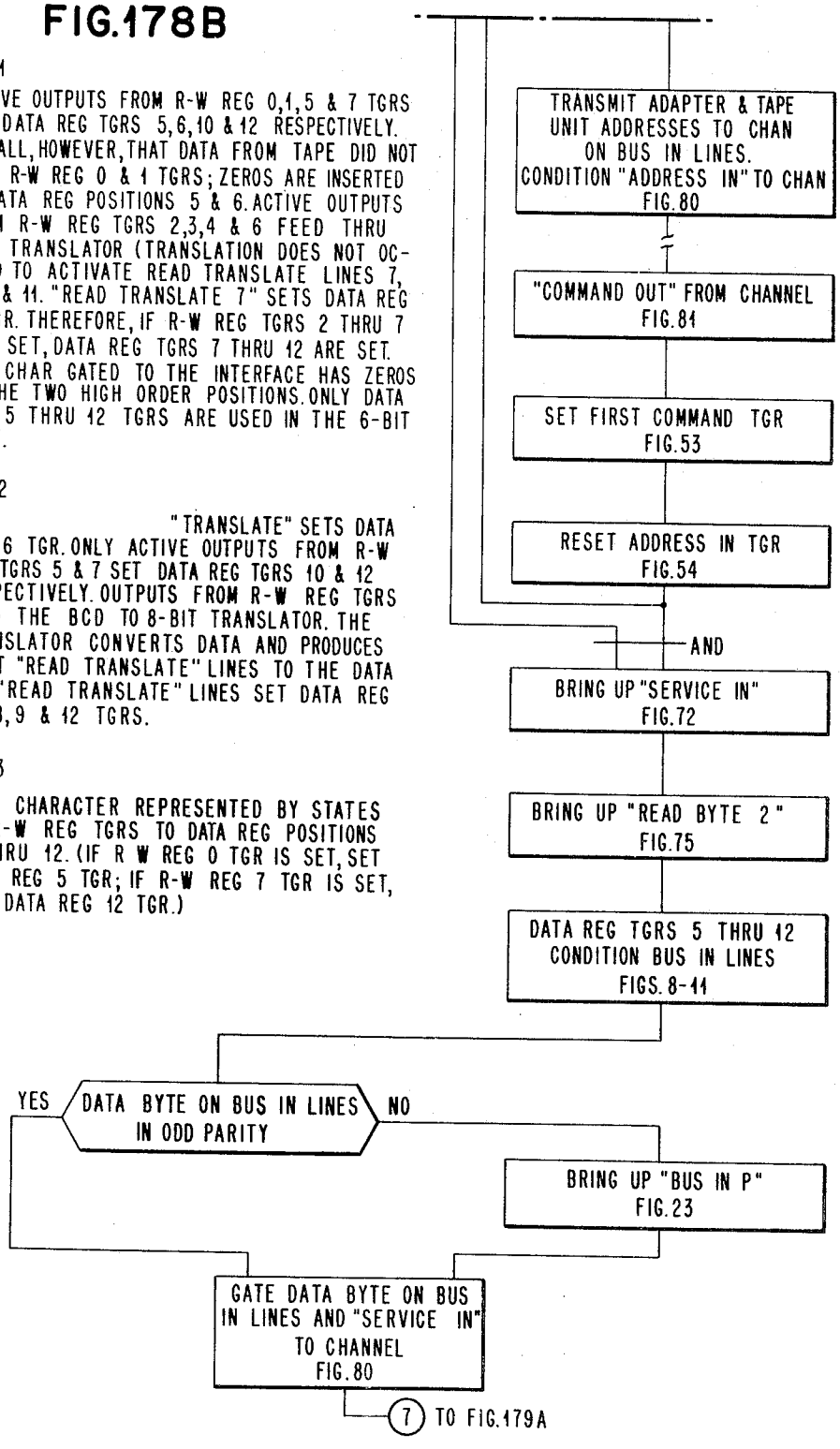


FIG. 179A

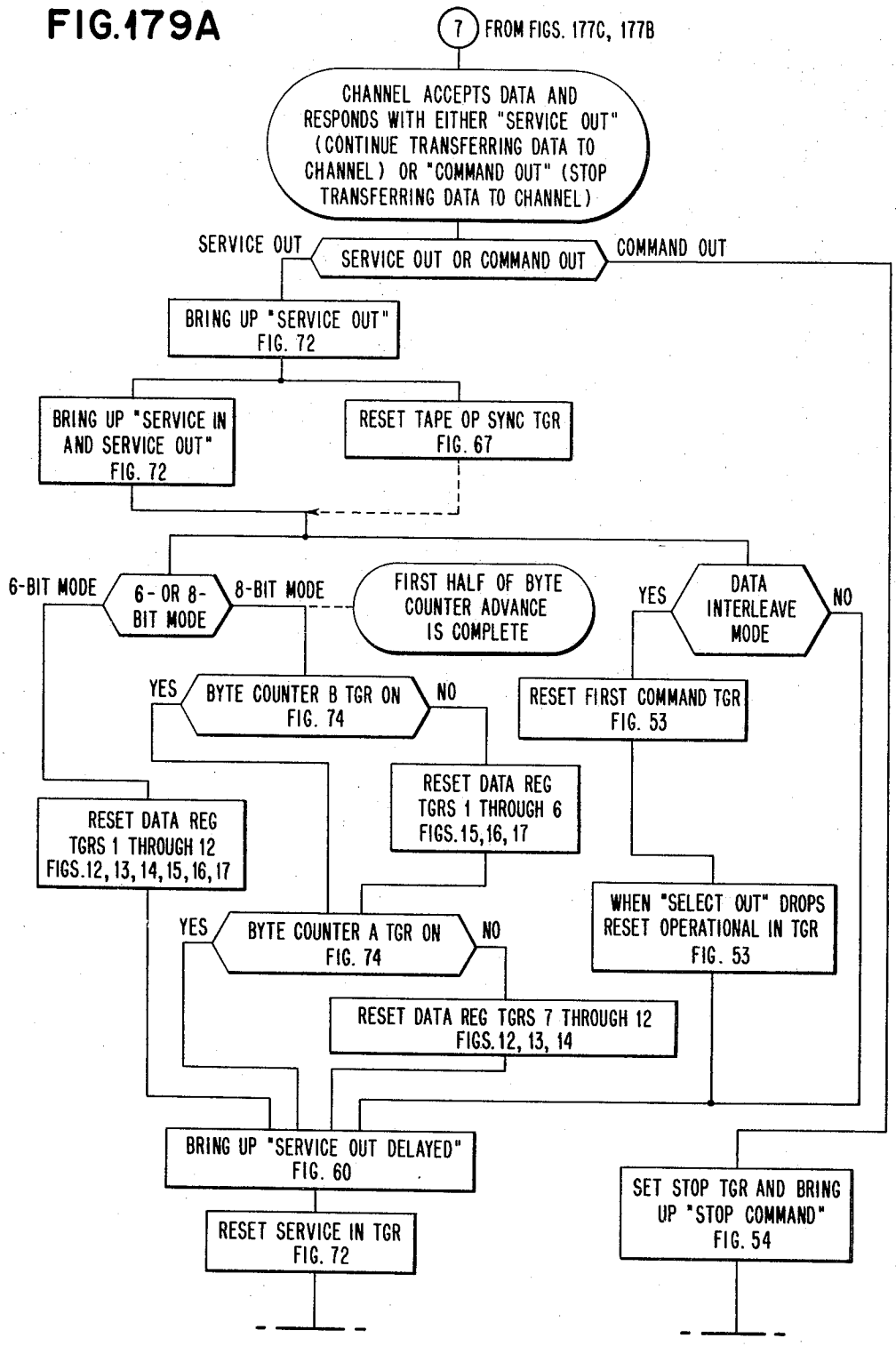

7
FROM FIGS. 177C, 177B
CHANNEL ACCEPTS DATA AND RESPONDS WITH EITHER "SERVICE OUT" (CONTINUE TRANSFERRING DATA TO CHANNEL) OR "COMMAND OUT" (STOP TRANSFERRING DATA TO CHANNEL)
SERVICE OUT
SERVICE OUT OR COMMAND OUT
COMMAND OUT
BRING UP "SERVICE OUT" FIG. 72
BRING UP "SERVICE IN AND SERVICE OUT" FIG. 72
RESET TAPE OP SYNC TGR FIG. 67
6-BIT MODE
6- OR 8- BIT MODE
8-BIT MODE
FIRST HALF OF BYTE COUNTER ADVANCE IS COMPLETE
YES
DATA INTERLEAVE MODE
NO
YES
BYTE COUNTER B TGR ON FIG. 74
NO
RESET FIRST COMMAND TGR FIG. 53
RESET DATA REG TGRS 1 THROUGH 6 FIGS. 15, 16, 17
RESET DATA REG TGRS 1 THROUGH 12 FIGS. 12, 13, 14, 15, 16, 17
WHEN "SELECT OUT" DROPS RESET OPERATIONAL IN TGR FIG. 53
YES
BYTE COUNTER A TGR ON FIG. 74
NO
RESET DATA REG TGRS 7 THROUGH 12 FIGS. 12, 13, 14
BRING UP "SERVICE OUT DELAYED" FIG. 60
RESET SERVICE IN TGR FIG. 72
SET STOP TGR AND BRING UP "STOP COMMAND" FIG. 54

FIG. 179B

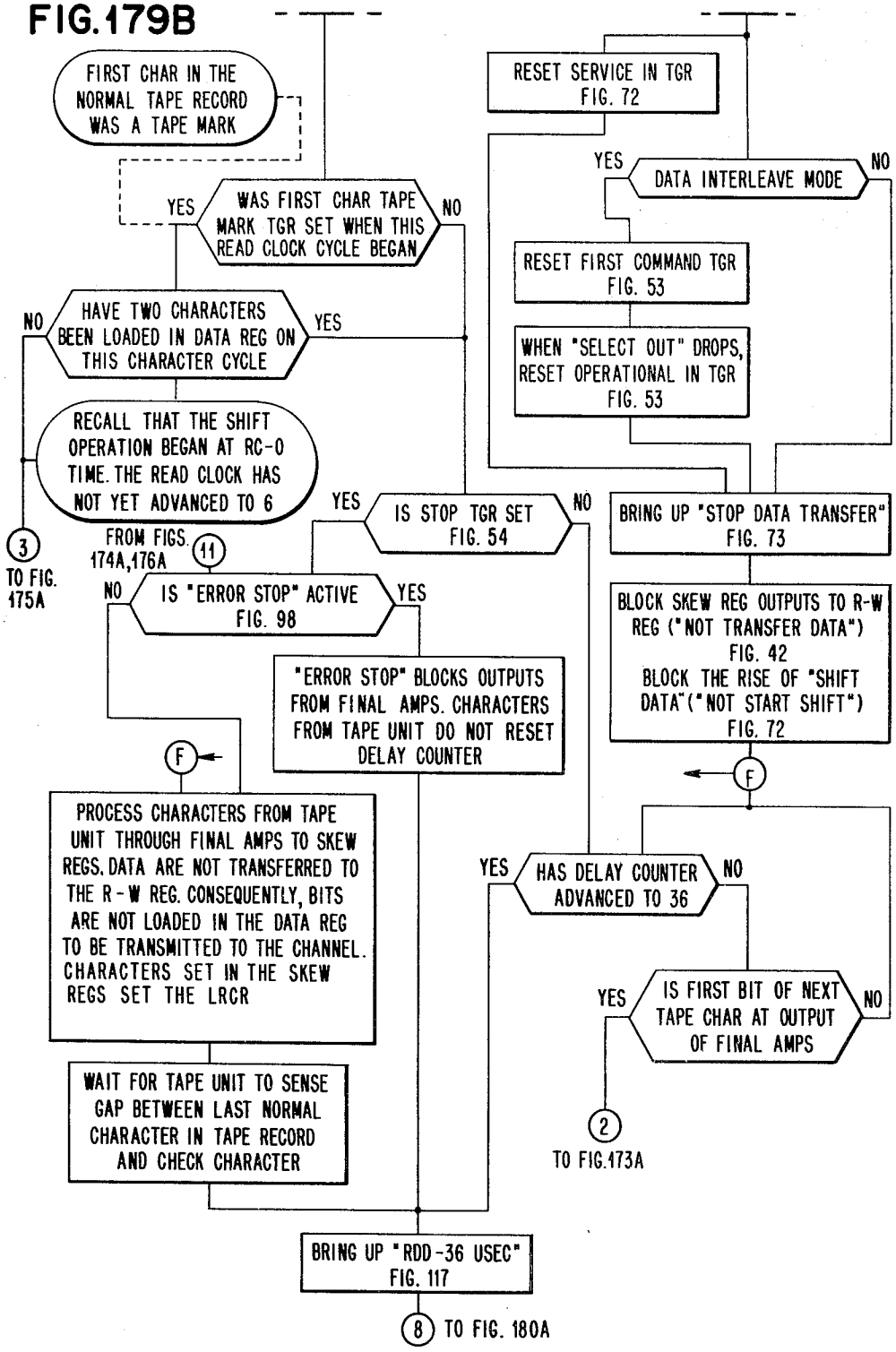

FIRST CHAR IN THE NORMAL TAPE RECORD WAS A TAPE MARK
YES
WAS FIRST CHAR TAPE MARK TGR SET WHEN THIS READ CLOCK CYCLE BEGAN
NO
NO
HAVE TWO CHARACTERS BEEN LOADED IN DATA REG ON THIS CHARACTER CYCLE
YES
RECALL THAT THE SHIFT OPERATION BEGAN AT RC-0 TIME. THE READ CLOCK HAS NOT YET ADVANCED TO 6
3
TO FIG. 175A
FROM FIGS. 174A,176A
11
NO
IS "ERROR STOP" ACTIVE FIG. 98
YES
"ERROR STOP" BLOCKS OUTPUTS FROM FINAL AMPS. CHARACTERS FROM TAPE UNIT DO NOT RESET DELAY COUNTER
F
PROCESS CHARACTERS FROM TAPE UNIT THROUGH FINAL AMPS TO SKEW REGS. DATA ARE NOT TRANSFERRED TO THE R-W REG. CONSEQUENTLY, BITS ARE NOT LOADED IN THE DATA REG TO BE TRANSMITTED TO THE CHANNEL. CHARACTERS SET IN THE SKEW REGS SET THE LRCR
WAIT FOR TAPE UNIT TO SENSE GAP BETWEEN LAST NORMAL CHARACTER IN TAPE RECORD AND CHECK CHARACTER
YES
IS STOP TGR SET FIG. 54
NO
RESET SERVICE IN TGR FIG. 72
YES
DATA INTERLEAVE MODE
NO
RESET FIRST COMMAND TGR FIG. 53
WHEN "SELECT OUT" DROPS, RESET OPERATIONAL IN TGR FIG. 53
BRING UP "STOP DATA TRANSFER" FIG. 73
BLOCK SKEW REG OUTPUTS TO R-W REG ("NOT TRANSFER DATA") FIG. 42
BLOCK THE RISE OF "SHIFT DATA" ("NOT START SHIFT") FIG. 72
F
YES
HAS DELAY COUNTER ADVANCED TO 36
NO
YES
IS FIRST BIT OF NEXT TAPE CHAR AT OUTPUT OF FINAL AMPS
NO
2
TO FIG. 173A
BRING UP "RDD-36 USEC" FIG. 117
8
TO FIG. 180A

FIG.180A

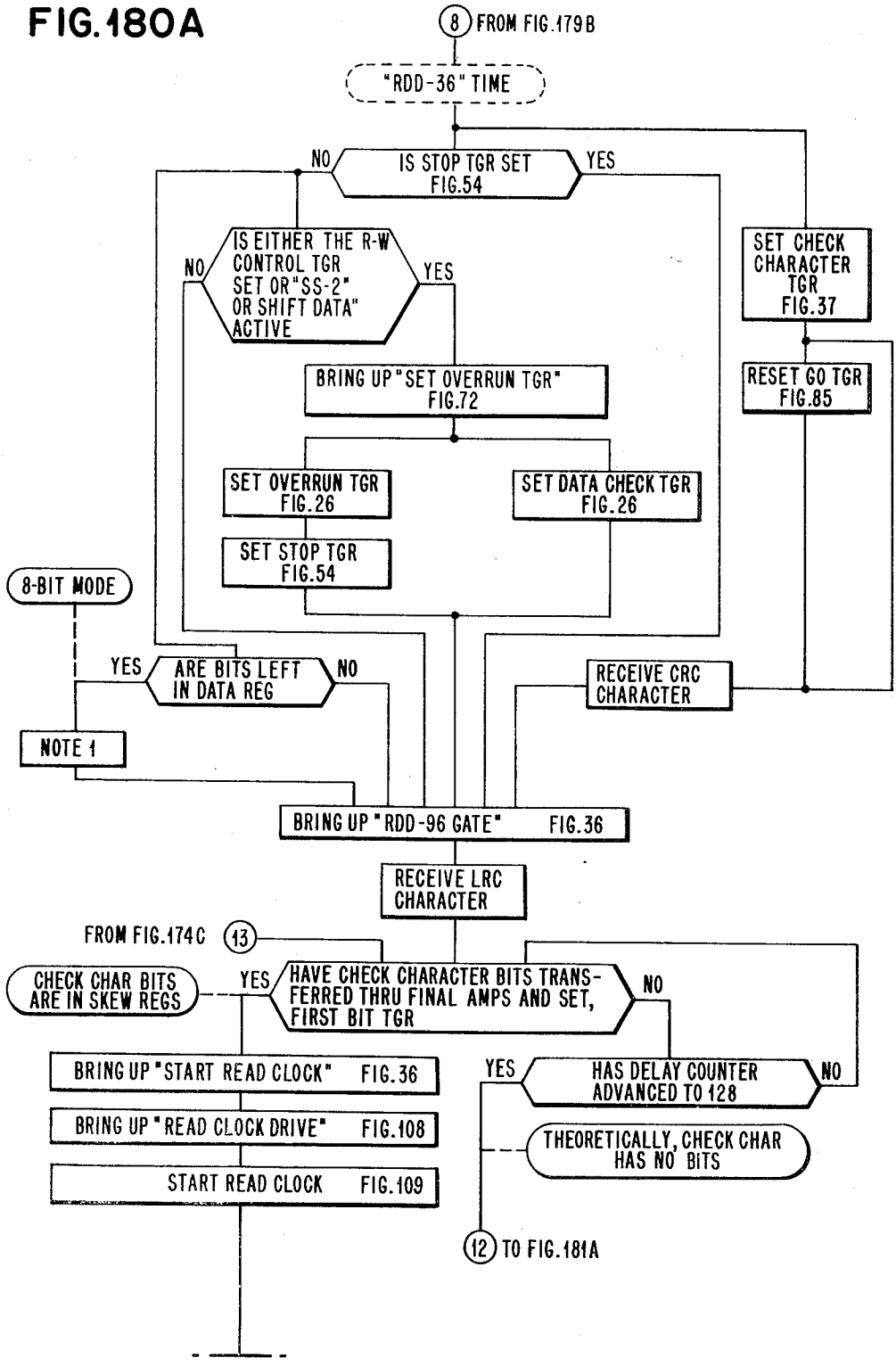
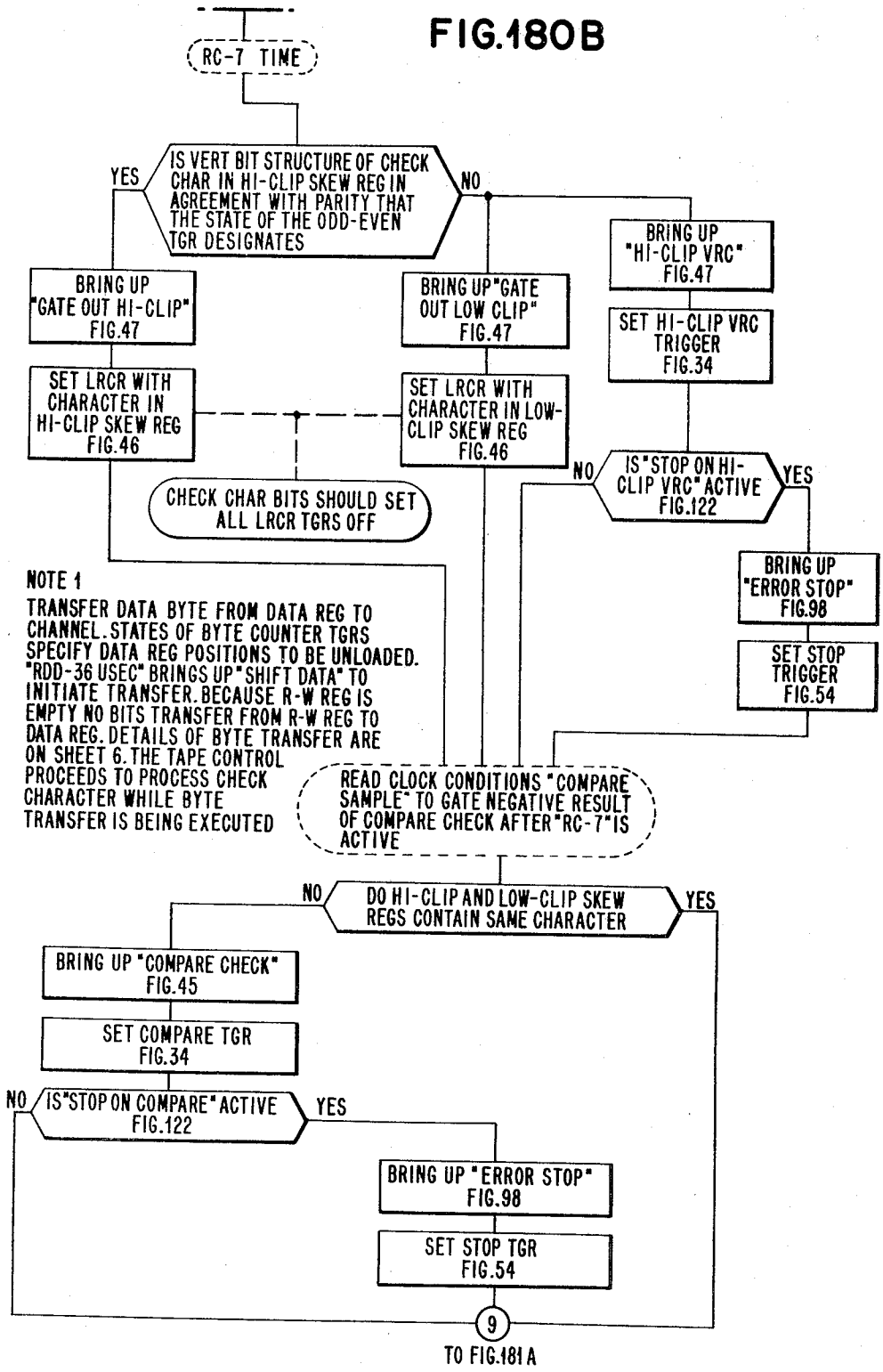

8 FROM FIG.179B
"RDD-36" TIME
NO
IS STOP TGR SET FIG.54
YES
IS EITHER THE R-W CONTROL TGR SET OR "SS-2" OR SHIFT DATA" ACTIVE
NO
YES
SET CHECK CHARACTER TGR FIG.37
RESET GO TGR FIG.85
BRING UP "SET OVERRUN TGR" FIG.72
SET OVERRUN TGR FIG.26
SET DATA CHECK TGR FIG.26
SET STOP TGR FIG.54
8-BIT MODE
YES
ARE BITS LEFT IN DATA REG
NO
RECEIVE CRC CHARACTER
NOTE 1
BRING UP "RDD-96 GATE" FIG.36
RECEIVE LRC CHARACTER
FROM FIG.174C 13
CHECK CHAR BITS ARE IN SKEW REGS
YES
HAVE CHECK CHARACTER BITS TRANSFERRED THRU FINAL AMPS AND SET, FIRST BIT TGR
NO
BRING UP "START READ CLOCK" FIG.36
BRING UP "READ CLOCK DRIVE" FIG.108
START READ CLOCK FIG.109
YES
HAS DELAY COUNTER ADVANCED TO 128
NO
THEORETICALLY, CHECK CHAR HAS NO BITS
12 TO FIG.181A

FIG.180B


RC-7 TIME
IS VERT BIT STRUCTURE OF CHECK CHAR IN HI-CLIP SKEW REG IN AGREEMENT WITH PARITY THAT THE STATE OF THE ODD-EVEN TGR DESIGNATES
YES
NO
BRING UP "GATE OUT HI-CLIP" FIG.47
BRING UP "GATE OUT LOW CLIP" FIG.47
BRING UP "HI-CLIP VRC" FIG.47
SET HI-CLIP VRC TRIGGER FIG.34
SET LRCR WITH CHARACTER IN HI-CLIP SKEW REG FIG.46
SET LRCR WITH CHARACTER IN LOW-CLIP SKEW REG FIG.46
CHECK CHAR BITS SHOULD SET ALL LRCR TGRS OFF
IS "STOP ON HI-CLIP VRC" ACTIVE FIG.122
NO
YES
BRING UP "ERROR STOP" FIG.98
SET STOP TRIGGER FIG.54
NOTE 1
TRANSFER DATA BYTE FROM DATA REG TO CHANNEL. STATES OF BYTE COUNTER TGRS SPECIFY DATA REG POSITIONS TO BE UNLOADED. "RDD-36 USEC" BRINGS UP "SHIFT DATA" TO INITIATE TRANSFER. BECAUSE R-W REG IS EMPTY NO BITS TRANSFER FROM R-W REG TO DATA REG. DETAILS OF BYTE TRANSFER ARE ON SHEET 6. THE TAPE CONTROL PROCEEDS TO PROCESS CHECK CHARACTER WHILE BYTE TRANSFER IS BEING EXECUTED
READ CLOCK CONDITIONS "COMPARE SAMPLE" TO GATE NEGATIVE RESULT OF COMPARE CHECK AFTER "RC-7" IS ACTIVE
DO HI-CLIP AND LOW-CLIP SKEW REGS CONTAIN SAME CHARACTER
NO
YES
BRING UP "COMPARE CHECK" FIG.45
SET COMPARE TGR FIG.34
IS "STOP ON COMPARE" ACTIVE FIG.122
NO
YES
BRING UP "ERROR STOP" FIG.98
SET STOP TGR FIG.54
9
TO FIG.181A

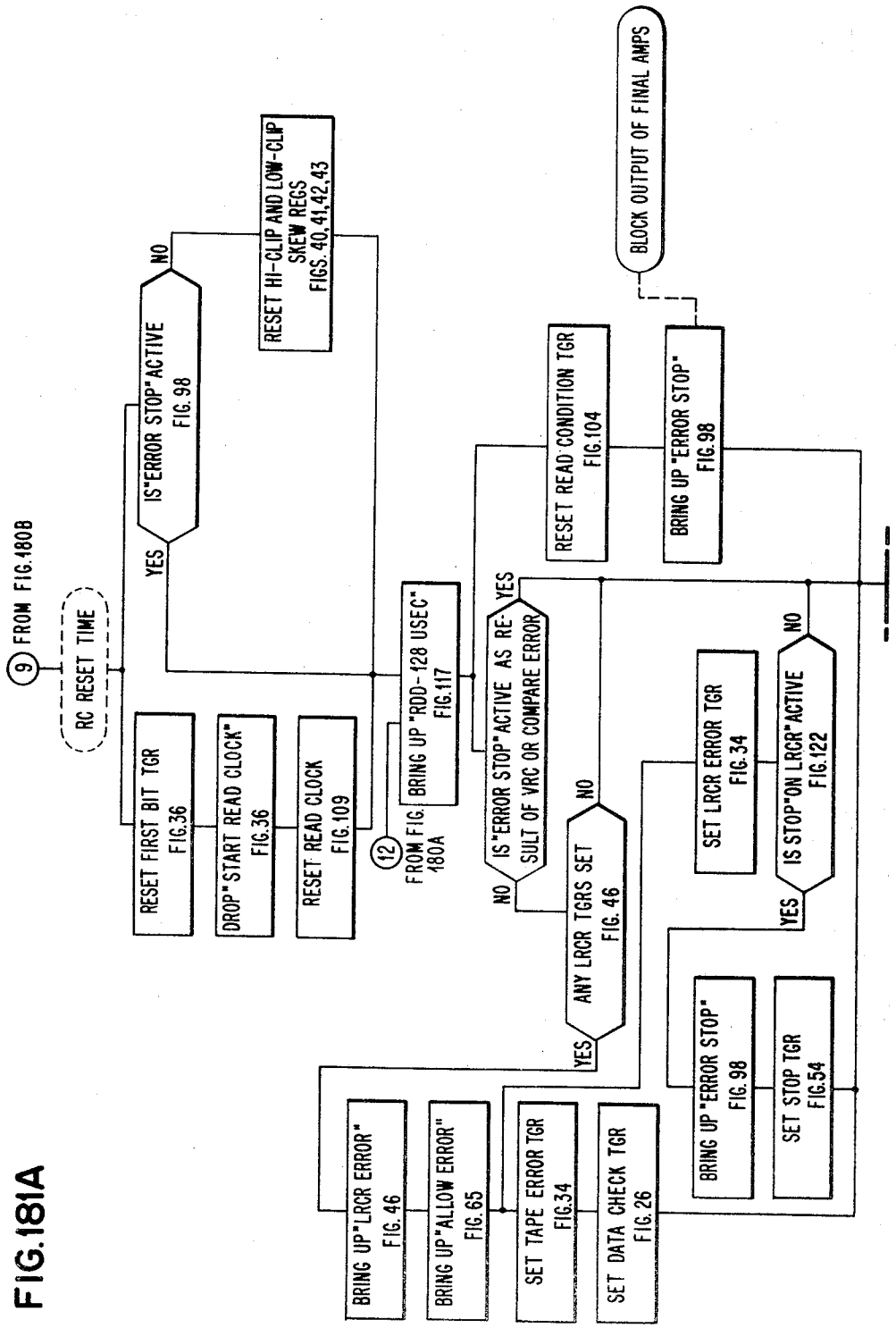

9 FROM FIG.180B
RC RESET TIME
RESET FIRST BIT TGR FIG.36
DROP "START READ CLOCK" FIG.36
RESET READ CLOCK FIG.109
IS "ERROR STOP" ACTIVE FIG.98
NO
YES
RESET HI-CLIP AND LOW-CLIP SKEW REGS FIGS.40,41,42,43
12 FROM FIG. 180A
BRING UP "RDD-128 USEC" FIG.117
IS "ERROR STOP" ACTIVE AS RESULT OF VRC OR COMPARE ERROR
NO
YES
ANY LRCR TGRS SET FIG.46
NO
YES
RESET READ CONDITION TGR FIG.104
BLOCK OUTPUT OF FINAL AMPS
BRING UP "ERROR STOP" FIG.98
SET LRCR ERROR TGR FIG.34
IS STOP "ON LRCR" ACTIVE FIG.122
NO
YES
BRING UP "LRCR ERROR" FIG.46
BRING UP "ALLOW ERROR" FIG.65
SET TAPE ERROR TGR FIG.34
SET DATA CHECK TGR FIG.26
BRING UP "ERROR STOP" FIG.98
SET STOP TGR FIG.54

FIG.181A

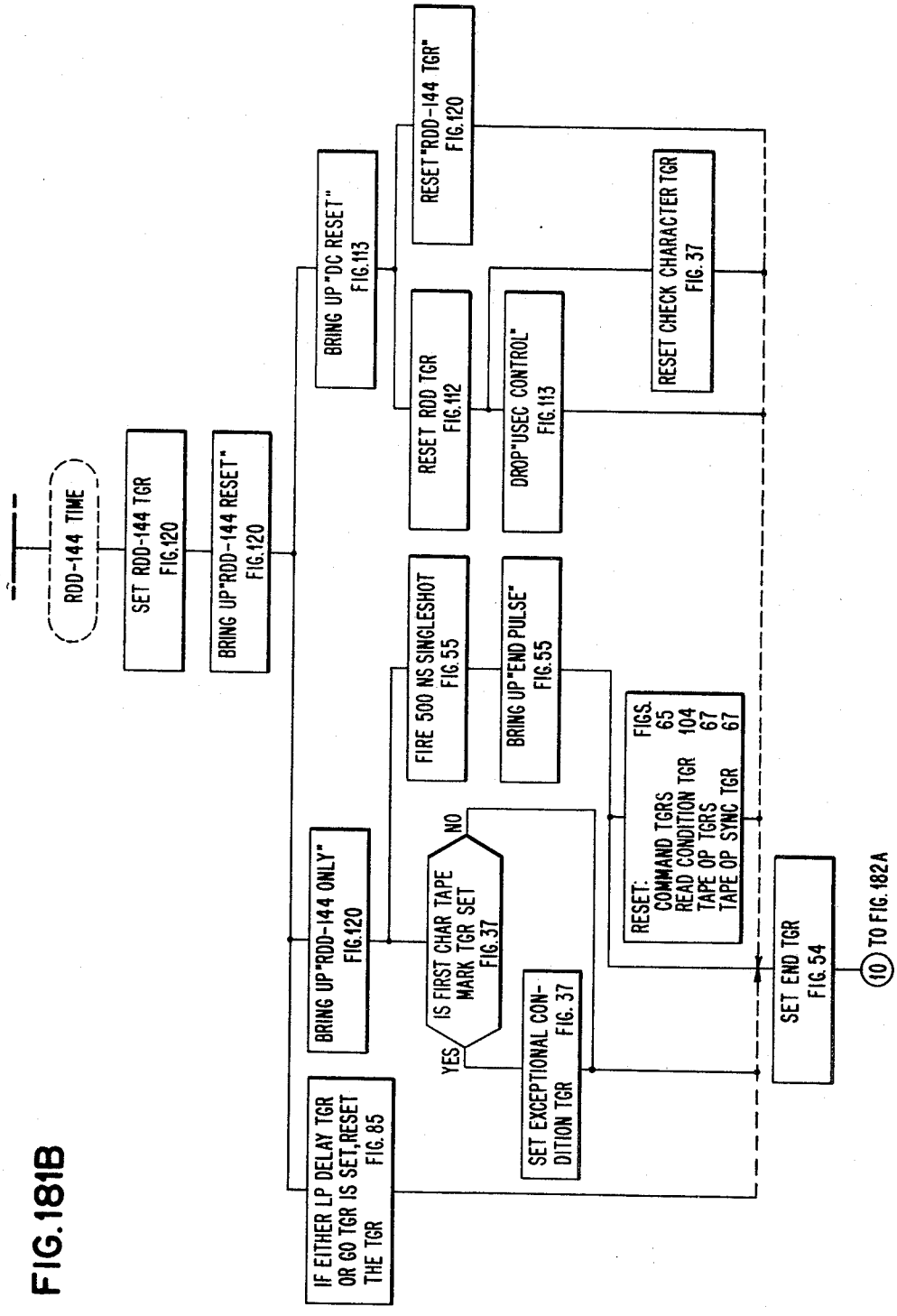

RDD-144 TIME
SET RDD-144 TGR FIG.120
BRING UP "RDD-144 RESET" FIG.120
BRING UP "DC RESET" FIG.113
RESET "RDD-144 TGR" FIG.120
RESET RDD TGR FIG.112
DROP "USEC CONTROL" FIG.113
RESET CHECK CHARACTER TGR FIG. 37
FIRE 500 NS SINGLESHOT FIG.55
BRING UP "END PULSE" FIG.55
RESET: COMMAND TGRS FIGS. 65
READ CONDITION TGR 104
TAPE OP TGRS 67
TAPE OP SYNC TGR 67
BRING UP "RDD-144 ONLY" FIG.120
IS FIRST CHAR TAPE MARK TGR SET FIG.37
NO
YES
SET EXCEPTIONAL CONDITION TGR FIG. 37
IF EITHER LP DELAY TGR OR GO TGR IS SET, RESET THE TGR FIG. 85
SET END TGR FIG. 54
(10) TO FIG. 182A

FIG. 181B

FIG. 182A

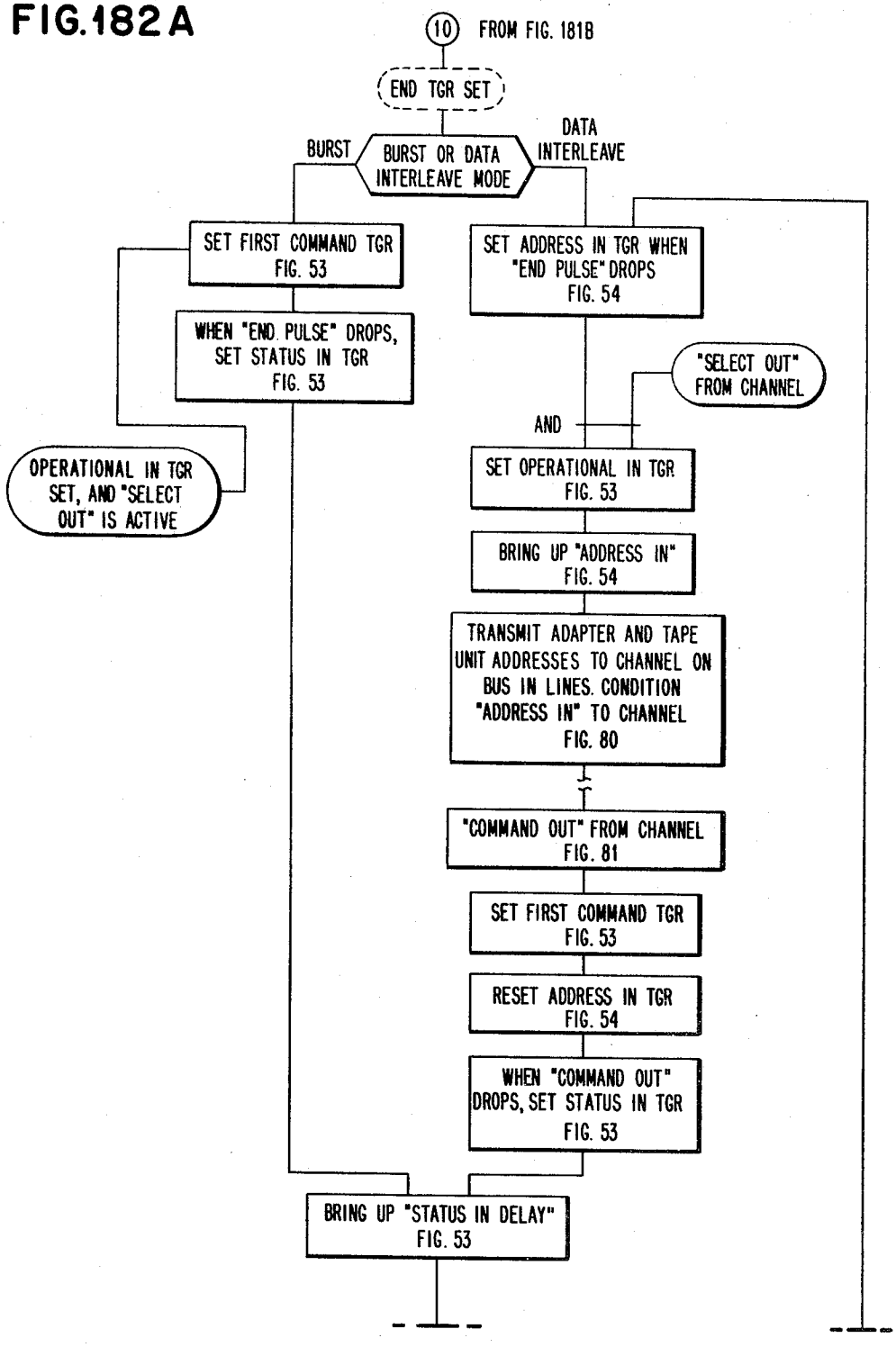
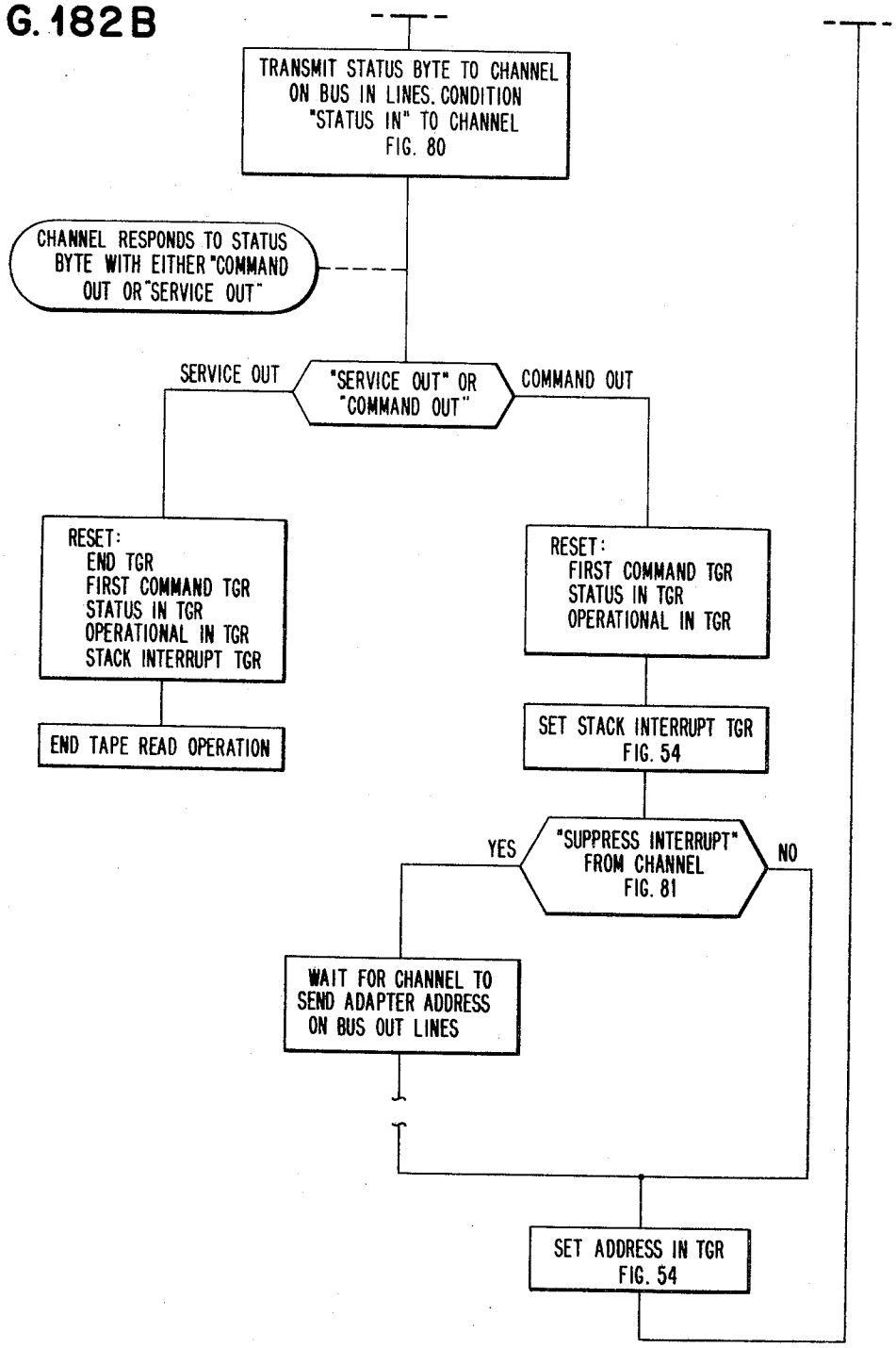

10
FROM FIG. 181B
END TGR SET
BURST
BURST OR DATA INTERLEAVE MODE
DATA INTERLEAVE
SET FIRST COMMAND TGR FIG. 53
WHEN "END PULSE" DROPS, SET STATUS IN TGR FIG. 53
OPERATIONAL IN TGR SET, AND "SELECT OUT" IS ACTIVE
SET ADDRESS IN TGR WHEN "END PULSE" DROPS FIG. 54
"SELECT OUT" FROM CHANNEL
AND
SET OPERATIONAL IN TGR FIG. 53
BRING UP "ADDRESS IN" FIG. 54
TRANSMIT ADAPTER AND TAPE UNIT ADDRESSES TO CHANNEL ON BUS IN LINES. CONDITION "ADDRESS IN" TO CHANNEL FIG. 80
"COMMAND OUT" FROM CHANNEL FIG. 81
SET FIRST COMMAND TGR FIG. 53
RESET ADDRESS IN TGR FIG. 54
WHEN "COMMAND OUT" DROPS, SET STATUS IN TGR FIG. 53
BRING UP "STATUS IN DELAY" FIG. 53

FIG. 182B


TRANSMIT STATUS BYTE TO CHANNEL ON BUS IN LINES. CONDITION "STATUS IN" TO CHANNEL FIG. 80
CHANNEL RESPONDS TO STATUS BYTE WITH EITHER "COMMAND OUT OR "SERVICE OUT"
SERVICE OUT
"SERVICE OUT" OR "COMMAND OUT"
COMMAND OUT
RESET: END TGR FIRST COMMAND TGR STATUS IN TGR OPERATIONAL IN TGR STACK INTERRUPT TGR
END TAPE READ OPERATION
RESET: FIRST COMMAND TGR STATUS IN TGR OPERATIONAL IN TGR
SET STACK INTERRUPT TGR FIG. 54
YES
"SUPPRESS INTERRUPT" FROM CHANNEL FIG. 81
NO
WAIT FOR CHANNEL TO SEND ADAPTER ADDRESS ON BUS OUT LINES
SET ADDRESS IN TGR FIG. 54

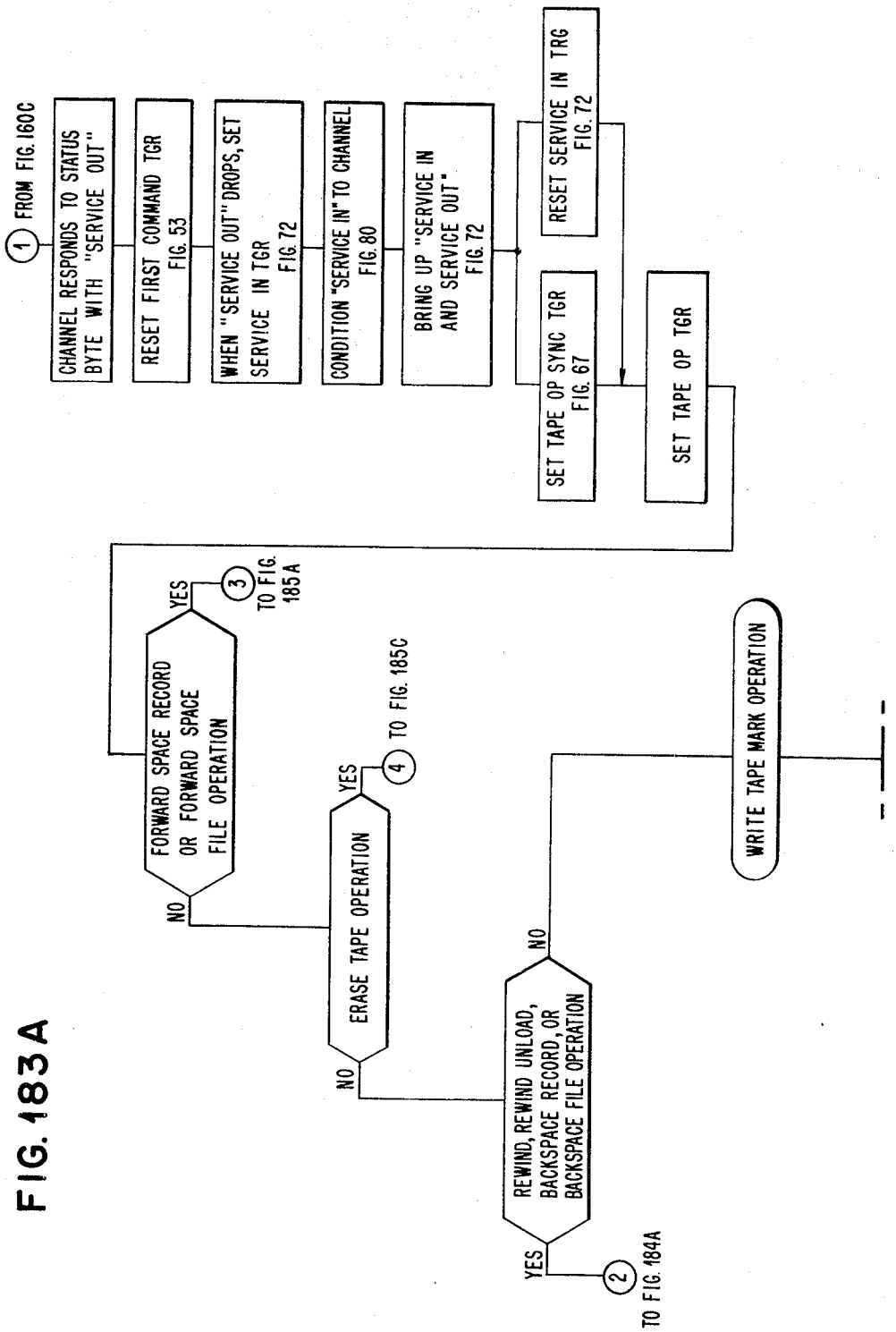

1 FROM FIG. 160C
CHANNEL RESPONDS TO STATUS BYTE WITH "SERVICE OUT"
RESET FIRST COMMAND TGR FIG. 53
WHEN "SERVICE OUT" DROPS, SET SERVICE IN TGR FIG. 72
CONDITION "SERVICE IN" TO CHANNEL FIG. 80
BRING UP "SERVICE IN AND SERVICE OUT" FIG. 72
RESET SERVICE IN TRG FIG. 72
SET TAPE OP SYNC TGR FIG. 67
SET TAPE OP TGR
FORWARD SPACE RECORD OR FORWARD SPACE FILE OPERATION
YES
3 TO FIG. 185A
NO
ERASE TAPE OPERATION
YES
4 TO FIG. 185C
NO
REWIND, REWIND UNLOAD, BACKSPACE RECORD, OR BACKSPACE FILE OPERATION
NO
YES
2 TO FIG. 184A
WRITE TAPE MARK OPERATION

FIG. 183A

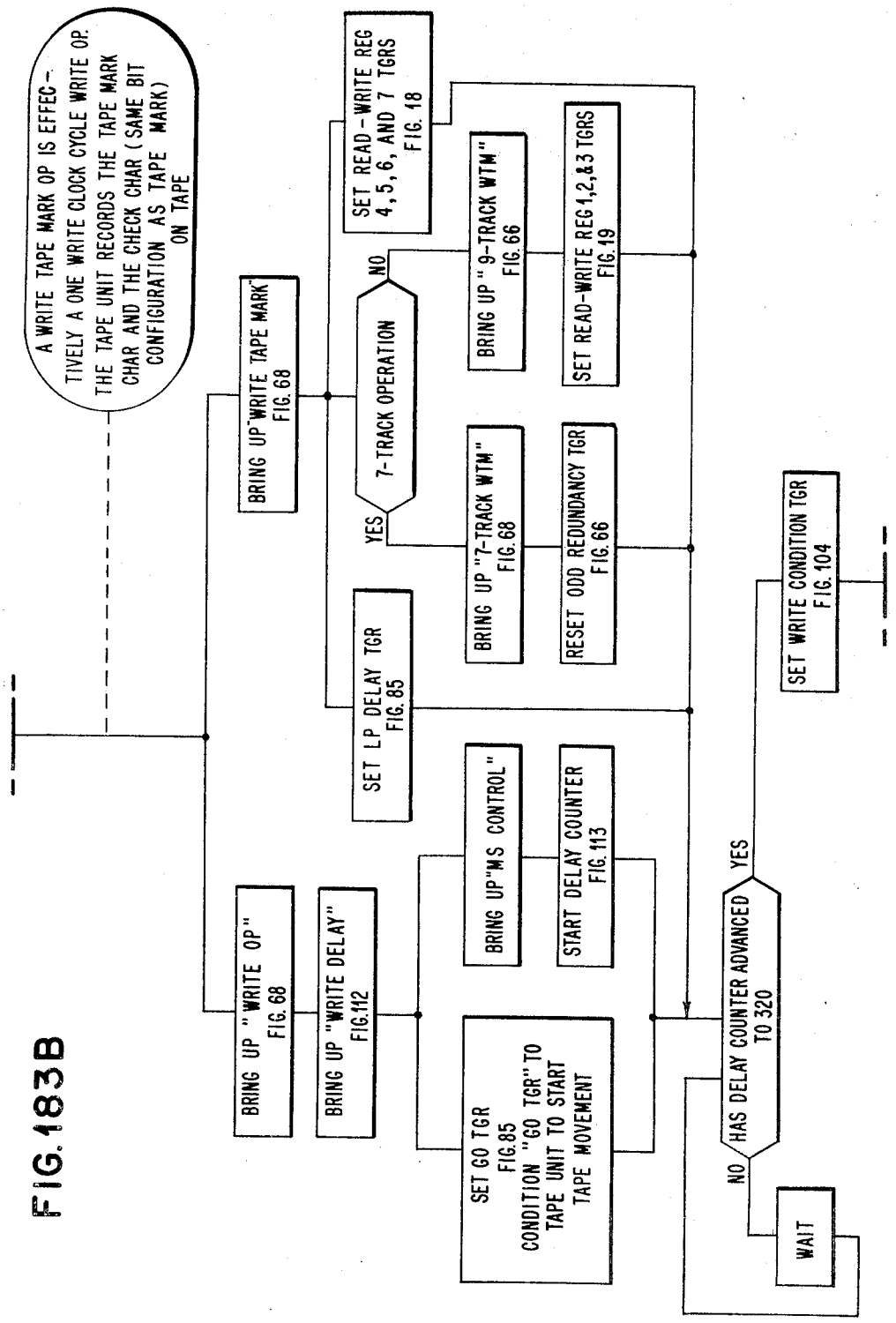

A WRITE TAPE MARK OP IS EFFECTIVELY A ONE WRITE CLOCK CYCLE WRITE OP. THE TAPE UNIT RECORDS THE TAPE MARK CHAR AND THE CHECK CHAR (SAME BIT CONFIGURATION AS TAPE MARK) ON TAPE
BRING UP "WRITE TAPE MARK" FIG. 68
7-TRACK OPERATION
NO
YES
SET READ-WRITE REG 4,5,6, AND 7 TGRS FIG. 18
BRING UP "9-TRACK WTM" FIG. 66
SET READ-WRITE REG 1,2,&3 TGRS FIG. 19
BRING UP "7-TRACK WTM" FIG. 68
RESET ODD REDUNDANCY TGR FIG. 66
SET LP DELAY TGR FIG. 85
BRING UP "WRITE OP" FIG. 68
BRING UP "WRITE DELAY" FIG. 112
BRING UP "MS CONTROL"
START DELAY COUNTER FIG. 113
SET GO TGR FIG. 85 CONDITION "GO TGR" TO TAPE UNIT TO START TAPE MOVEMENT
HAS DELAY COUNTER ADVANCED TO 320
YES
NO
WAIT
SET WRITE CONDITION TGR FIG. 104

FIG. 183B

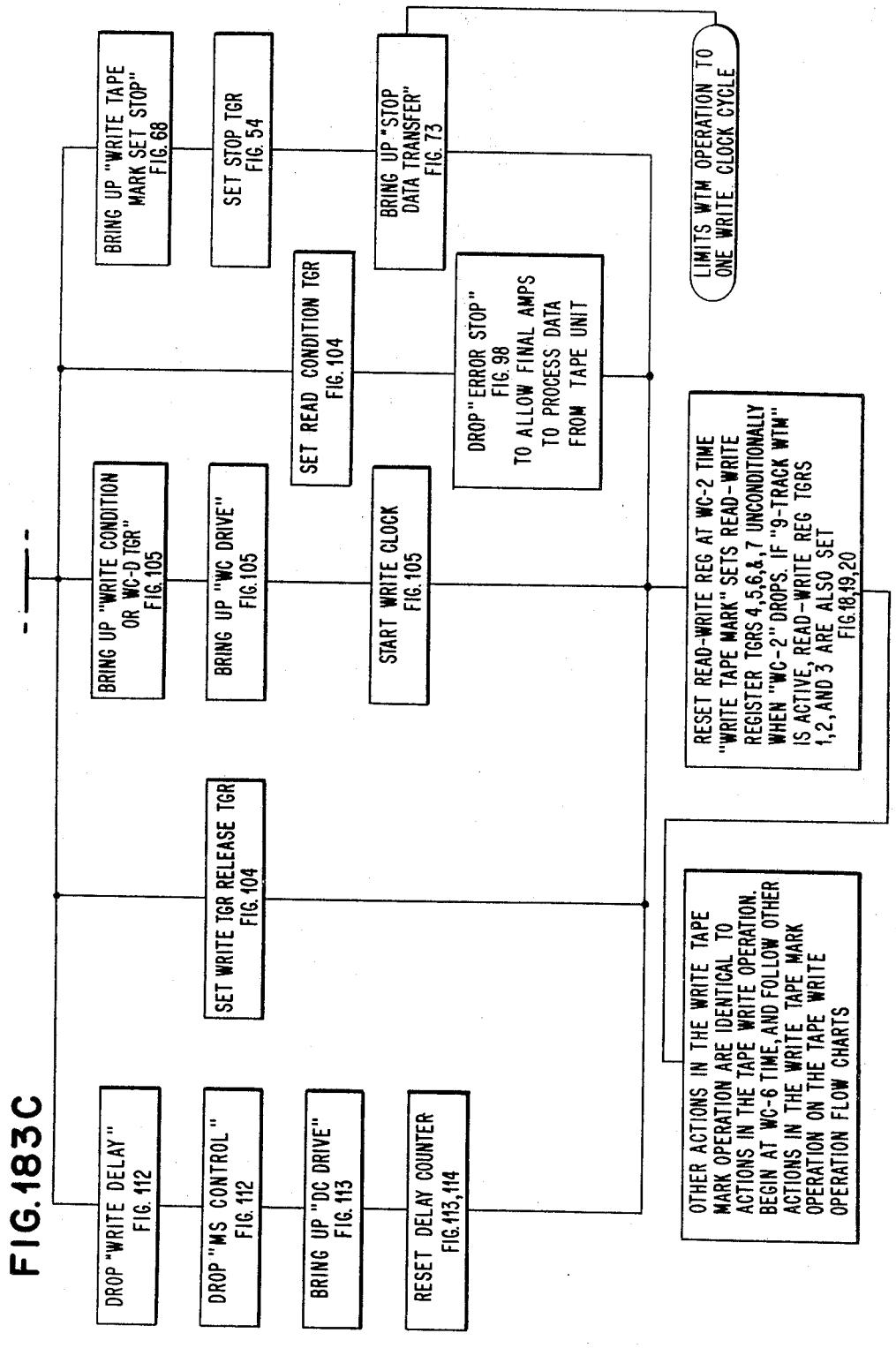

BRING UP "WRITE TAPE MARK SET STOP" FIG. 68
SET STOP TGR FIG. 54
BRING UP "STOP DATA TRANSFER" FIG. 73
LIMITS WTM OPERATION TO ONE WRITE CLOCK CYCLE
SET READ CONDITION TGR FIG. 104
DROP "ERROR STOP" FIG. 98 TO ALLOW FINAL AMPS TO PROCESS DATA FROM TAPE UNIT
BRING UP "WRITE CONDITION OR WC-D TGR" FIG. 105
BRING UP "WC DRIVE" FIG. 105
START WRITE CLOCK FIG. 105
RESET READ-WRITE REG AT WC-2 TIME "WRITE TAPE MARK" SETS READ-WRITE REGISTER TGRS 4,5,6,&,7 UNCONDITIONALLY WHEN "WC-2" DROPS. IF "9-TRACK WTM" IS ACTIVE, READ-WRITE REG TGRS 1,2, AND 3 ARE ALSO SET FIG. 18,19,20
SET WRITE TGR RELEASE TGR FIG. 104
OTHER ACTIONS IN THE WRITE TAPE MARK OPERATION ARE IDENTICAL TO ACTIONS IN THE TAPE WRITE OPERATION. BEGIN AT WC-6 TIME, AND FOLLOW OTHER ACTIONS IN THE WRITE TAPE MARK OPERATION ON THE TAPE WRITE OPERATION FLOW CHARTS
DROP "WRITE DELAY" FIG. 112
DROP "WS CONTROL" FIG. 112
BRING UP "DC DRIVE" FIG. 113
RESET DELAY COUNTER FIG. 113, 114

FIG. 183C

FIG.184A

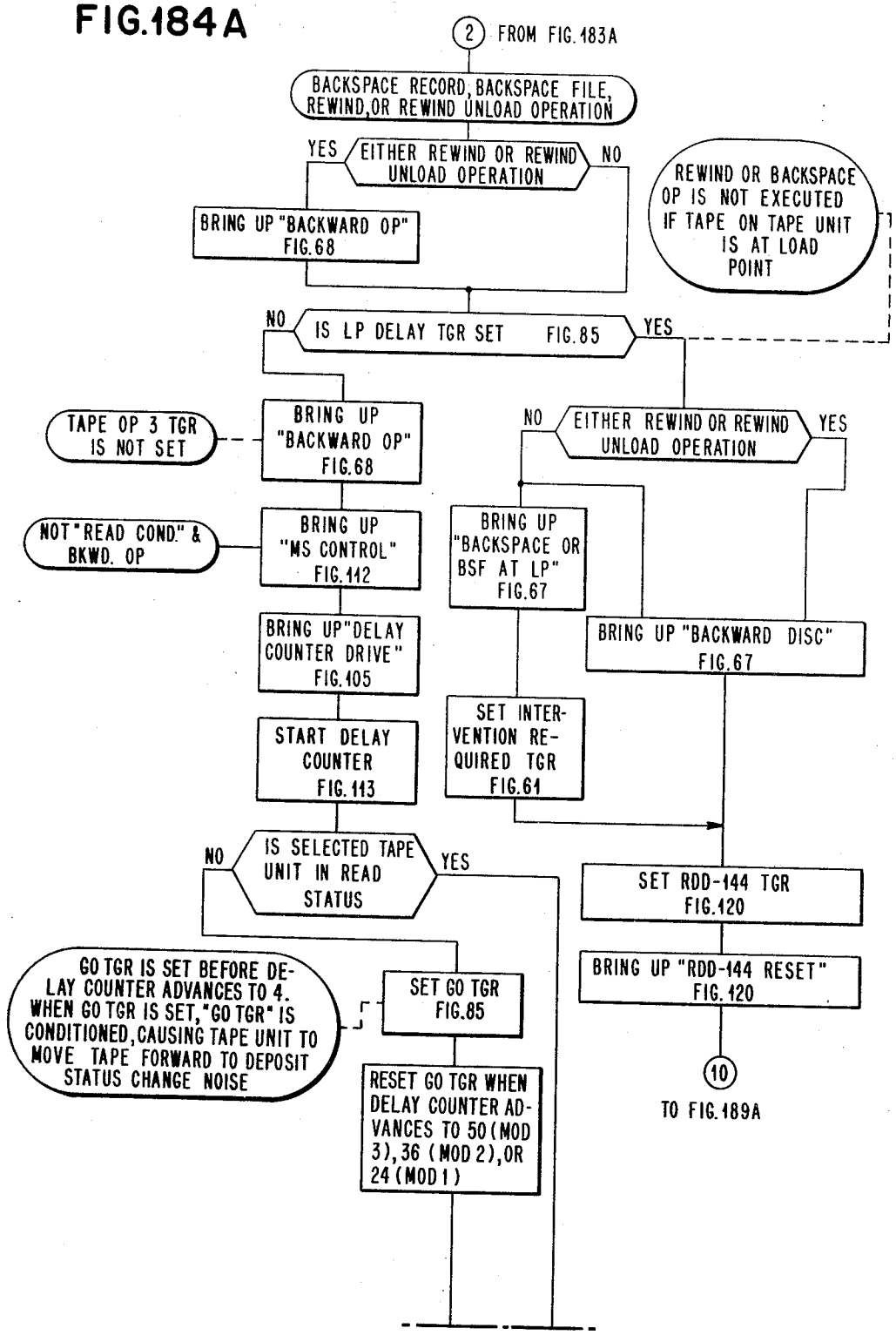
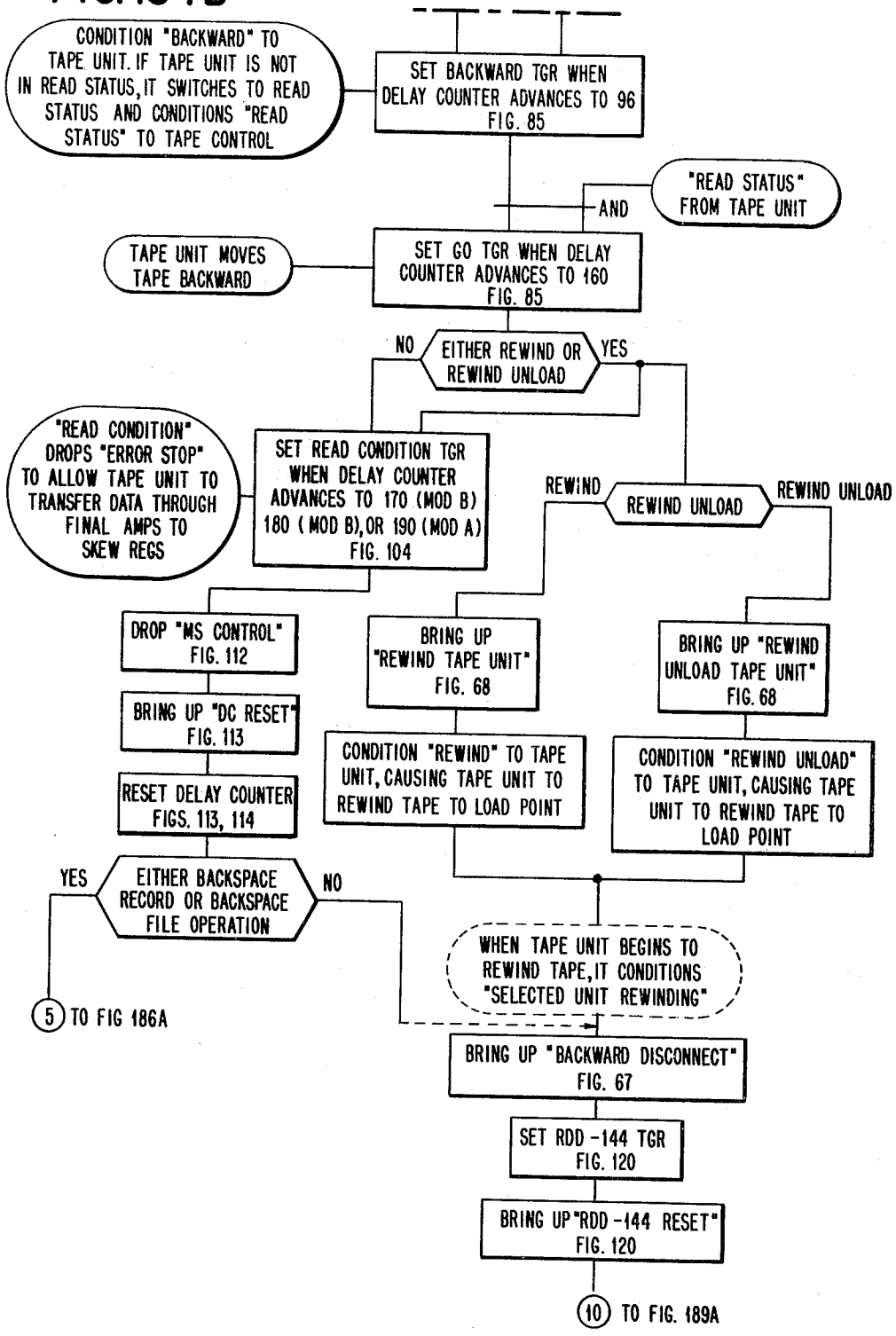

2 FROM FIG.183A
BACKSPACE RECORD, BACKSPACE FILE, REWIND, OR REWIND UNLOAD OPERATION
YES
EITHER REWIND OR REWIND UNLOAD OPERATION
NO
BRING UP "BACKWARD OP" FIG.68
REWIND OR BACKSPACE OP IS NOT EXECUTED IF TAPE ON TAPE UNIT IS AT LOAD POINT
NO
IS LP DELAY TGR SET FIG.85
YES
TAPE OP 3 TGR IS NOT SET
BRING UP "BACKWARD OP" FIG.68
NO
EITHER REWIND OR REWIND UNLOAD OPERATION
YES
NOT "READ COND." & BKWD. OP
BRING UP "MS CONTROL" FIG.112
BRING UP "BACKSPACE OR BSF AT LP" FIG.67
BRING UP "BACKWARD DISC" FIG.67
BRING UP "DELAY COUNTER DRIVE" FIG.105
START DELAY COUNTER FIG.113
SET INTER-VENTION RE-QUIRED TGR FIG.61
NO
IS SELECTED TAPE UNIT IN READ STATUS
YES
SET RDD-144 TGR FIG.120
BRING UP "RDD-144 RESET" FIG.120
GO TGR IS SET BEFORE DELAY COUNTER ADVANCES TO 4. WHEN GO TGR IS SET, "GO TGR" IS CONDITIONED, CAUSING TAPE UNIT TO MOVE TAPE FORWARD TO DEPOSIT STATUS CHANGE NOISE
SET GO TGR FIG.85
RESET GO TGR WHEN DELAY COUNTER ADVANCES TO 50 (MOD 3), 36 (MOD 2), OR 24 (MOD 1)
10
TO FIG.189A

FIG. 184B


CONDITION "BACKWARD" TO TAPE UNIT. IF TAPE UNIT IS NOT IN READ STATUS, IT SWITCHES TO READ STATUS AND CONDITIONS "READ STATUS" TO TAPE CONTROL
SET BACKWARD TGR WHEN DELAY COUNTER ADVANCES TO 96 FIG. 85
"READ STATUS" FROM TAPE UNIT
AND
TAPE UNIT MOVES TAPE BACKWARD
SET GO TGR WHEN DELAY COUNTER ADVANCES TO 160 FIG. 85
NO
EITHER REWIND OR REWIND UNLOAD
YES
"READ CONDITION" DROPS "ERROR STOP" TO ALLOW TAPE UNIT TO TRANSFER DATA THROUGH FINAL AMPS TO SKEW REGS
SET READ CONDITION TGR WHEN DELAY COUNTER ADVANCES TO 170 (MOD B) 180 (MOD B), OR 190 (MOD A) FIG. 104
REWIND
REWIND UNLOAD
REWIND UNLOAD
DROP "MS CONTROL" FIG. 112
BRING UP "REWIND TAPE UNIT" FIG. 68
BRING UP "REWIND UNLOAD TAPE UNIT" FIG. 68
BRING UP "DC RESET" FIG. 113
CONDITION "REWIND" TO TAPE UNIT, CAUSING TAPE UNIT TO REWIND TAPE TO LOAD POINT
CONDITION "REWIND UNLOAD" TO TAPE UNIT, CAUSING TAPE UNIT TO REWIND TAPE TO LOAD POINT
RESET DELAY COUNTER FIGS. 113, 114
YES
EITHER BACKSPACE RECORD OR BACKSPACE FILE OPERATION
NO
WHEN TAPE UNIT BEGINS TO REWIND TAPE, IT CONDITIONS "SELECTED UNIT REWINDING"
5 TO FIG 186A
BRING UP "BACKWARD DISCONNECT" FIG. 67
SET RDD -144 TGR FIG. 120
BRING UP "RDD -144 RESET" FIG. 120
10 TO FIG. 189A

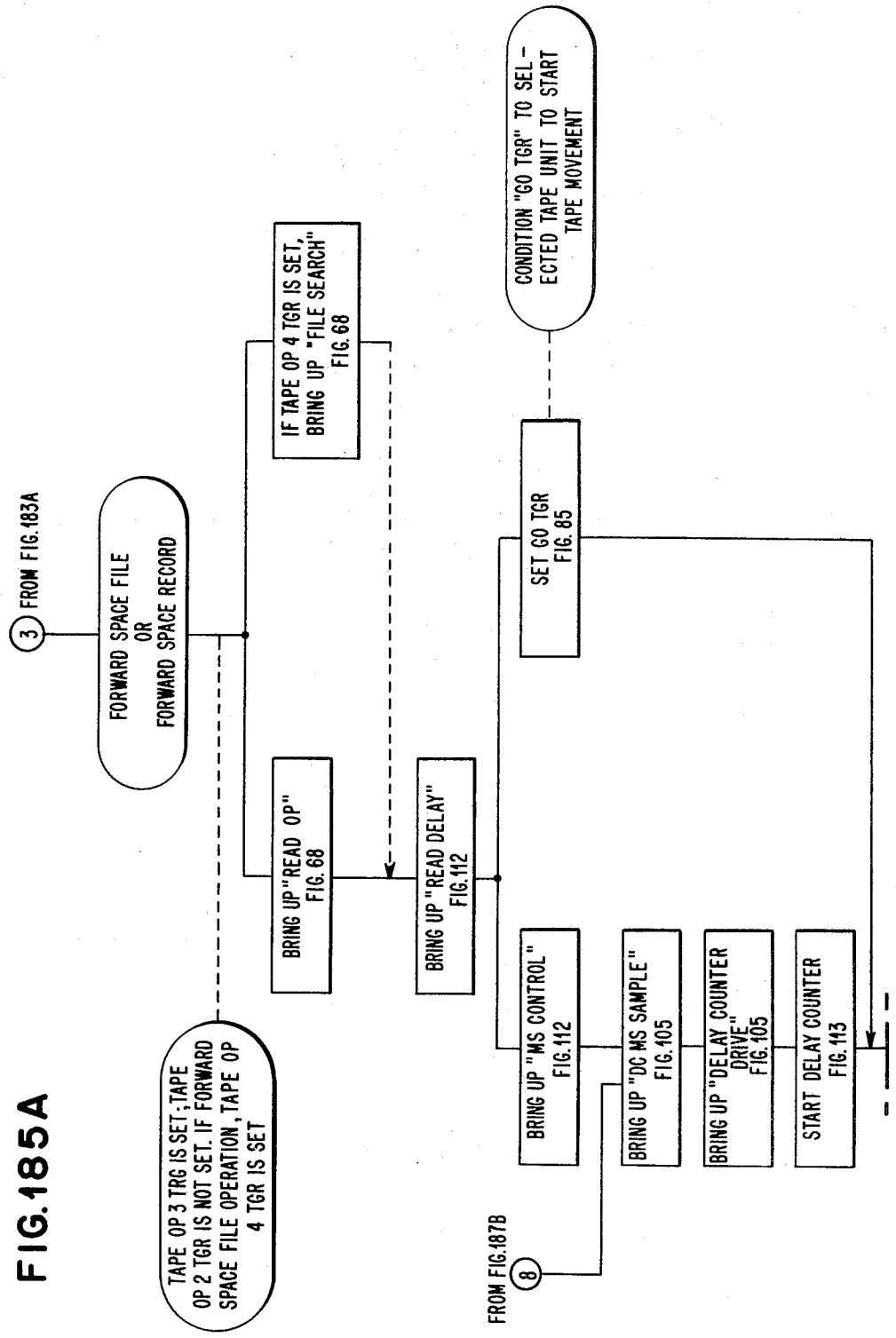

FROM FIG.183A
3
FORWARD SPACE FILE OR FORWARD SPACE RECORD
IF TAPE OP 4 TGR IS SET, BRING UP "FILE SEARCH" FIG.68
CONDITION "GO TGR" TO SELECTED TAPE UNIT TO START TAPE MOVEMENT
SET GO TGR FIG.85
BRING UP "READ OP" FIG.68
BRING UP "READ DELAY" FIG.112
TAPE OP 3 TRG IS SET; TAPE OP 2 TGR IS NOT SET. IF FORWARD SPACE FILE OPERATION, TAPE OP 4 TGR IS SET
BRING UP "MS CONTROL" FIG.112
BRING UP "DC MS SAMPLE" FIG.105
BRING UP "DELAY COUNTER DRIVE" FIG.105
START DELAY COUNTER FIG.113
FROM FIG.187B
8

FIG.185A

FIG. 185B

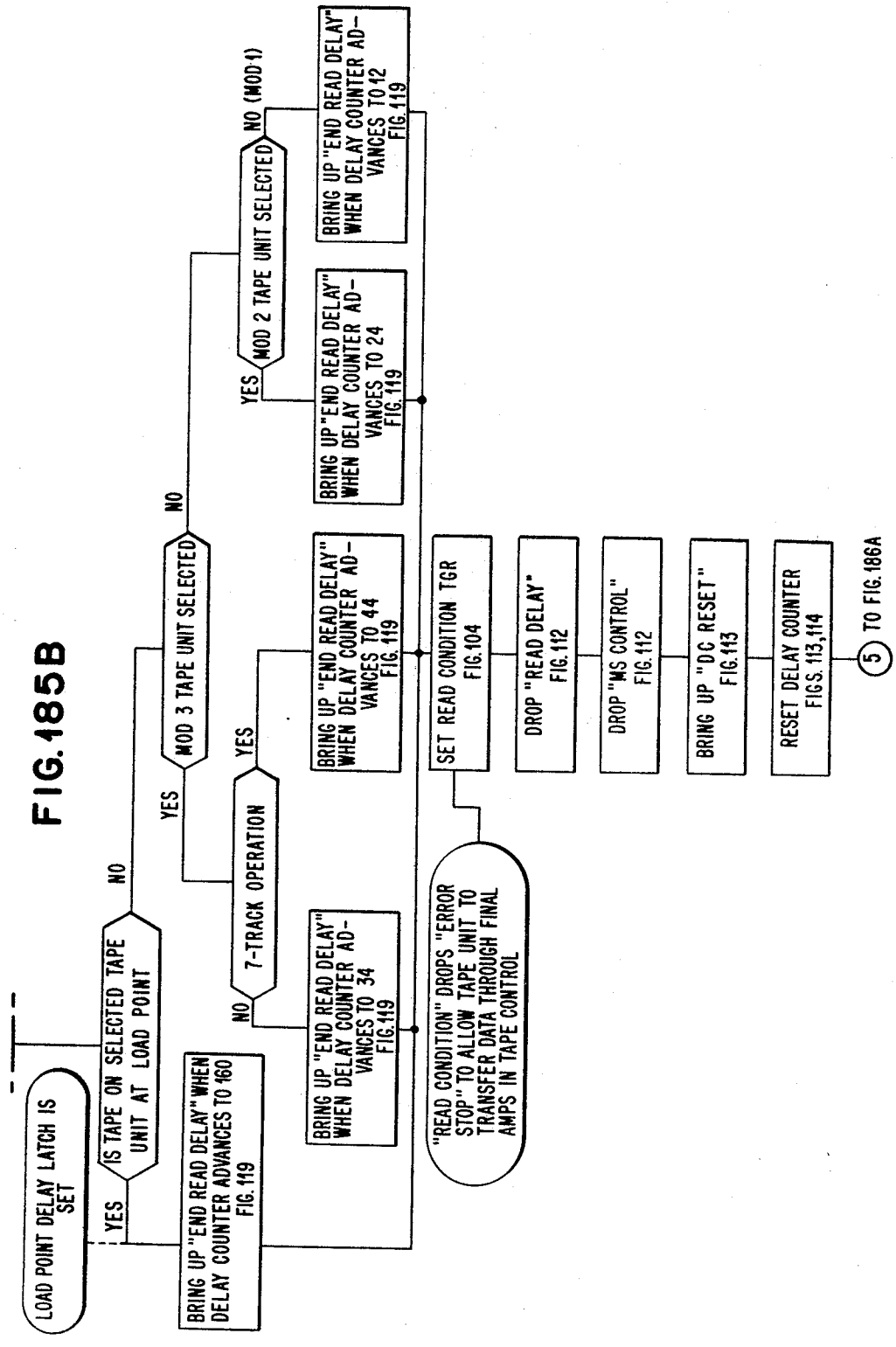

LOAD POINT DELAY LATCH IS SET
IS TAPE ON SELECTED TAPE UNIT AT LOAD POINT
YES
NO
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 160 FIG. 119
MOD 3 TAPE UNIT SELECTED
YES
NO
7-TRACK OPERATION
YES
NO
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 44 FIG. 119
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 34 FIG. 119
MOD 2 TAPE UNIT SELECTED
YES
NO (MOD 1)
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 24 FIG. 119
BRING UP "END READ DELAY" WHEN DELAY COUNTER ADVANCES TO 12 FIG. 119
"READ CONDITION" DROPS "ERROR STOP" TO ALLOW TAPE UNIT TO TRANSFER DATA THROUGH FINAL AMPS IN TAPE CONTROL
SET READ CONDITION TGR FIG. 104
DROP "READ DELAY" FIG. 112
DROP "MS CONTROL" FIG. 112
BRING UP "DC RESET" FIG. 113
RESET DELAY COUNTER FIGS. 113, 114
5
TO FIG. 186A

FIG. 185C

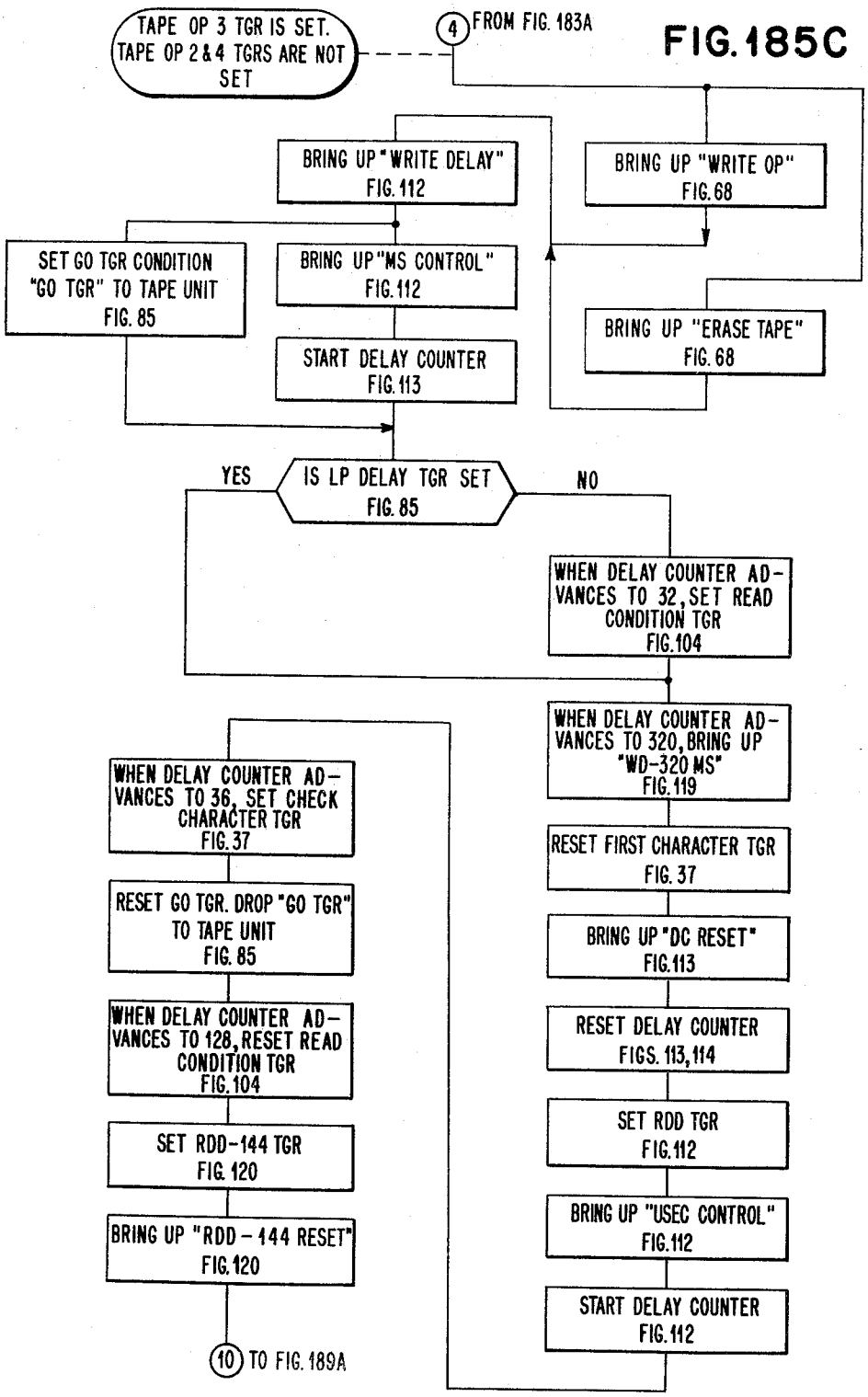

TAPE OP 3 TGR IS SET. TAPE OP 2 & 4 TGRS ARE NOT SET
4 FROM FIG. 183A
BRING UP "WRITE DELAY" FIG. 112
BRING UP "WRITE OP" FIG. 68
SET GO TGR CONDITION "GO TGR" TO TAPE UNIT FIG. 85
BRING UP "MS CONTROL" FIG. 112
START DELAY COUNTER FIG. 113
BRING UP "ERASE TAPE" FIG. 68
YES
IS LP DELAY TGR SET FIG. 85
NO
WHEN DELAY COUNTER ADVANCES TO 32, SET READ CONDITION TGR FIG. 104
WHEN DELAY COUNTER ADVANCES TO 320, BRING UP "WD-320 MS" FIG. 119
WHEN DELAY COUNTER ADVANCES TO 36, SET CHECK CHARACTER TGR FIG. 37
RESET FIRST CHARACTER TGR FIG. 37
RESET GO TGR. DROP "GO TGR" TO TAPE UNIT FIG. 85
BRING UP "DC RESET" FIG. 113
WHEN DELAY COUNTER ADVANCES TO 128, RESET READ CONDITION TGR FIG. 104
RESET DELAY COUNTER FIGS. 113, 114
SET RDD-144 TGR FIG. 120
SET RDD TGR FIG. 112
BRING UP "RDD-144 RESET" FIG. 120
BRING UP "USEC CONTROL" FIG. 112
START DELAY COUNTER FIG. 112
10 TO FIG. 189A

FIG. 186A

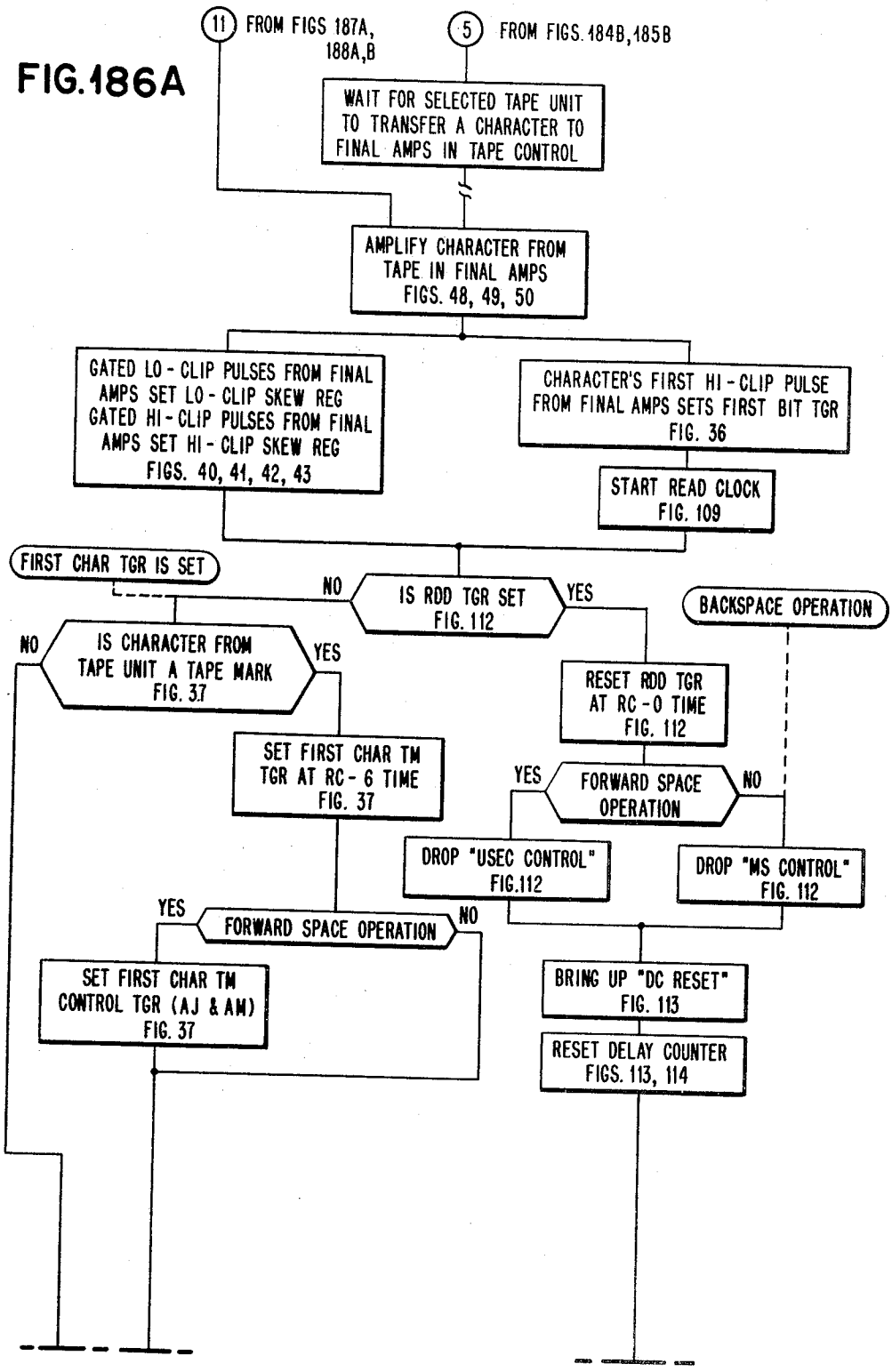

(11) FROM FIGS 187A, 188A,B
(5) FROM FIGS. 184B, 185B
WAIT FOR SELECTED TAPE UNIT TO TRANSFER A CHARACTER TO FINAL AMPS IN TAPE CONTROL
AMPLIFY CHARACTER FROM TAPE IN FINAL AMPS FIGS. 48, 49, 50
GATED LO-CLIP PULSES FROM FINAL AMPS SET LO-CLIP SKEW REG
GATED HI-CLIP PULSES FROM FINAL AMPS SET HI-CLIP SKEW REG
FIGS. 40, 41, 42, 43
CHARACTER'S FIRST HI-CLIP PULSE FROM FINAL AMPS SETS FIRST BIT TGR FIG. 36
START READ CLOCK FIG. 109
FIRST CHAR TGR IS SET
NO
IS RDD TGR SET FIG. 112
YES
BACKSPACE OPERATION
NO
IS CHARACTER FROM TAPE UNIT A TAPE MARK FIG. 37
YES
RESET RDD TGR AT RC-0 TIME FIG. 112
SET FIRST CHAR TM TGR AT RC-6 TIME FIG. 37
YES
FORWARD SPACE OPERATION
NO
DROP "USEC CONTROL" FIG. 112
DROP "MS CONTROL" FIG. 112
YES
FORWARD SPACE OPERATION
NO
SET FIRST CHAR TM CONTROL TGR (AJ & AM) FIG. 37
BRING UP "DC RESET" FIG. 113
RESET DELAY COUNTER FIGS. 113, 114

FIG.186B

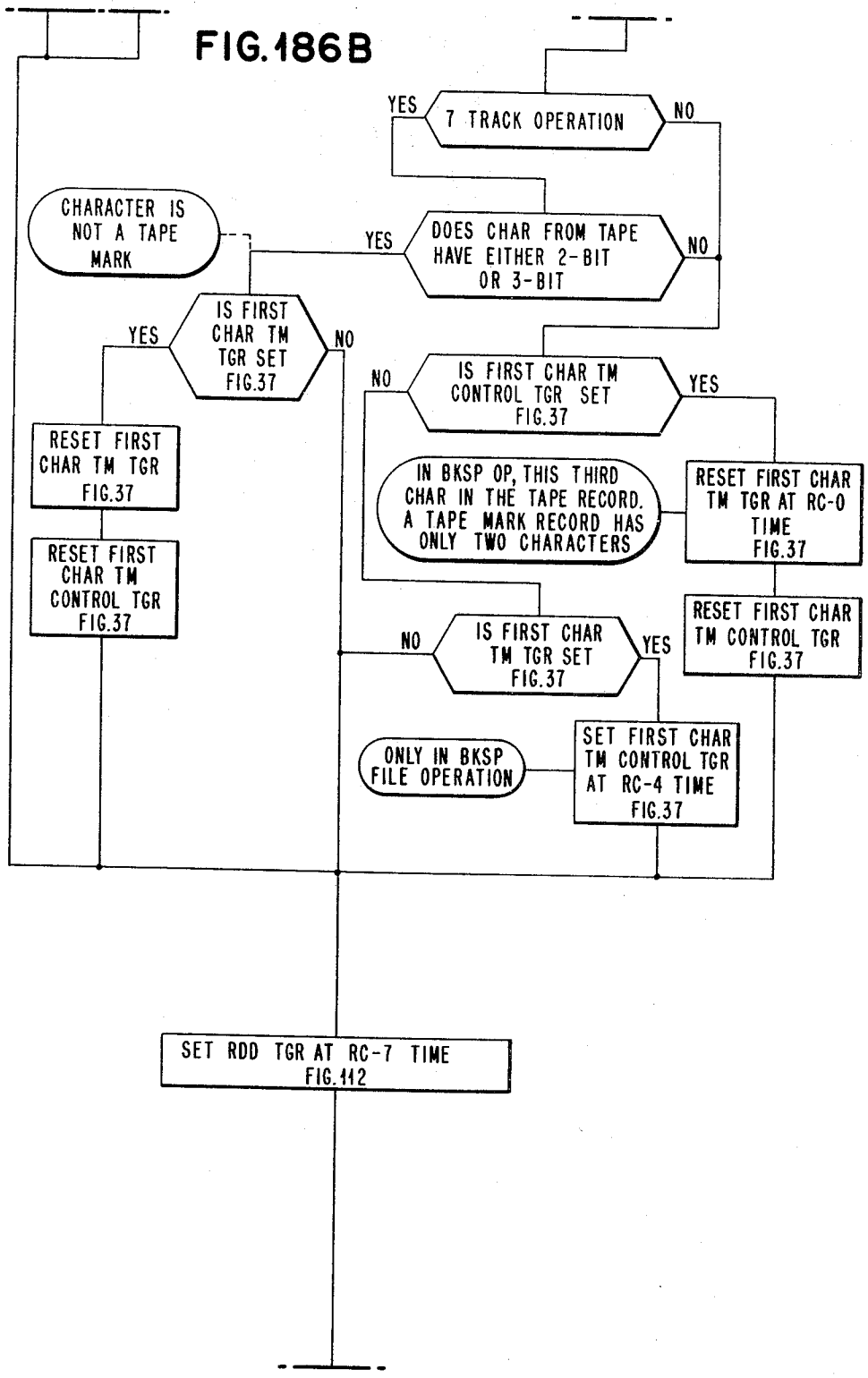
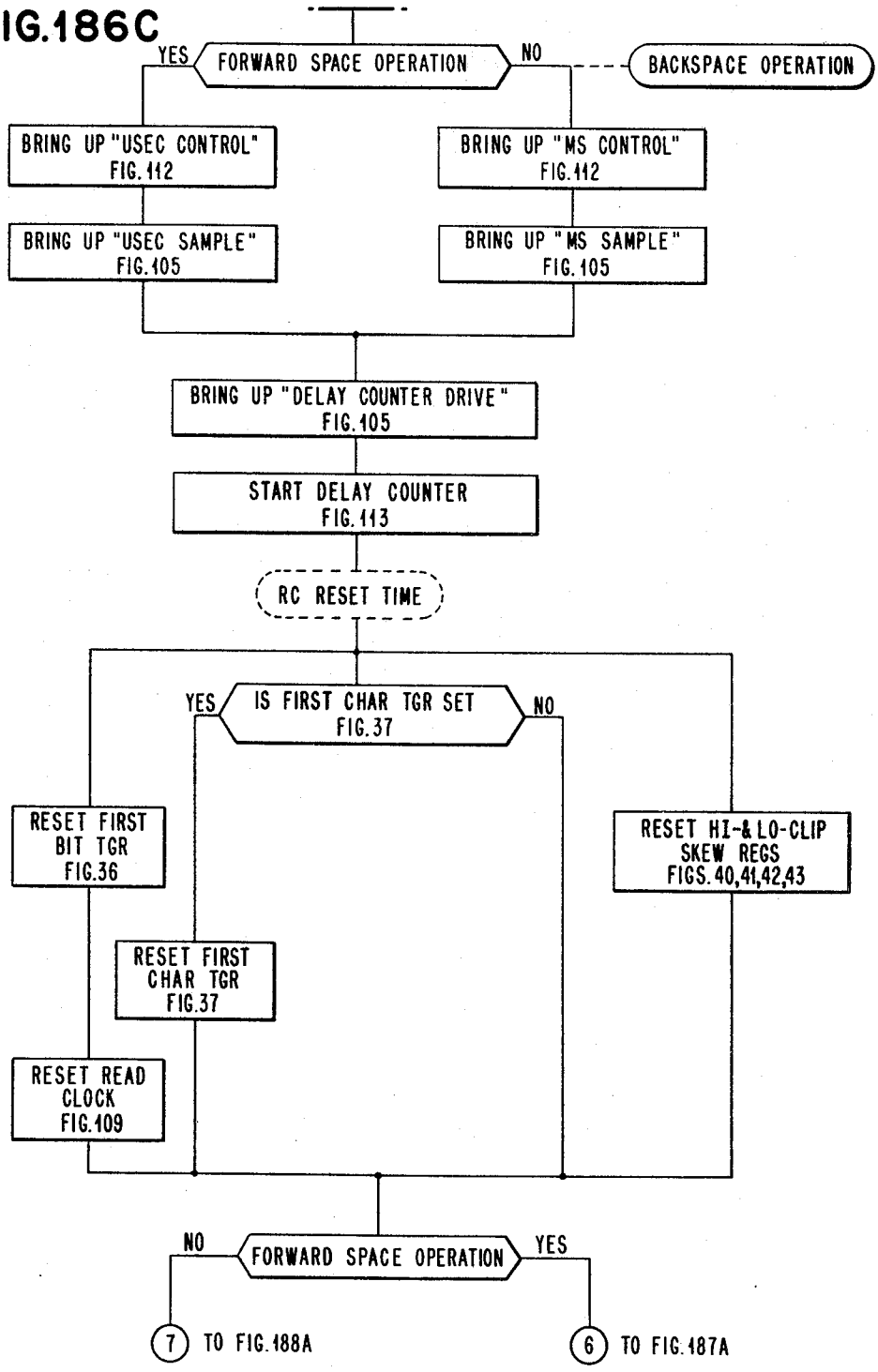

7 TRACK OPERATION
YES
NO
CHARACTER IS NOT A TAPE MARK
DOES CHAR FROM TAPE HAVE EITHER 2-BIT OR 3-BIT
YES
NO
IS FIRST CHAR TM TGR SET FIG.37
YES
NO
IS FIRST CHAR TM CONTROL TGR SET FIG.37
NO
YES
RESET FIRST CHAR TM TGR FIG.37
IN BKSP OP, THIS THIRD CHAR IN THE TAPE RECORD. A TAPE MARK RECORD HAS ONLY TWO CHARACTERS
RESET FIRST CHAR TM TGR AT RC-0 TIME FIG.37
RESET FIRST CHAR TM CONTROL TGR FIG.37
RESET FIRST CHAR TM CONTROL TGR FIG.37
NO
IS FIRST CHAR TM TGR SET FIG.37
YES
ONLY IN BKSP FILE OPERATION
SET FIRST CHAR TM CONTROL TGR AT RC-4 TIME FIG.37
SET RDD TGR AT RC-7 TIME FIG.112

FIG.186C


YES
FORWARD SPACE OPERATION
NO
BACKSPACE OPERATION
BRING UP "USEC CONTROL" FIG.112
BRING UP "MS CONTROL" FIG.112
BRING UP "USEC SAMPLE" FIG.105
BRING UP "MS SAMPLE" FIG.105
BRING UP "DELAY COUNTER DRIVE" FIG.105
START DELAY COUNTER FIG.113
RC RESET TIME
YES
IS FIRST CHAR TGR SET FIG.37
NO
RESET FIRST BIT TGR FIG.36
RESET HI-&LO-CLIP SKEW REGS FIGS.40,41,42,43
RESET FIRST CHAR TGR FIG.37
RESET READ CLOCK FIG.109
NO
FORWARD SPACE OPERATION
YES
7 TO FIG.188A
6 TO FIG.187A

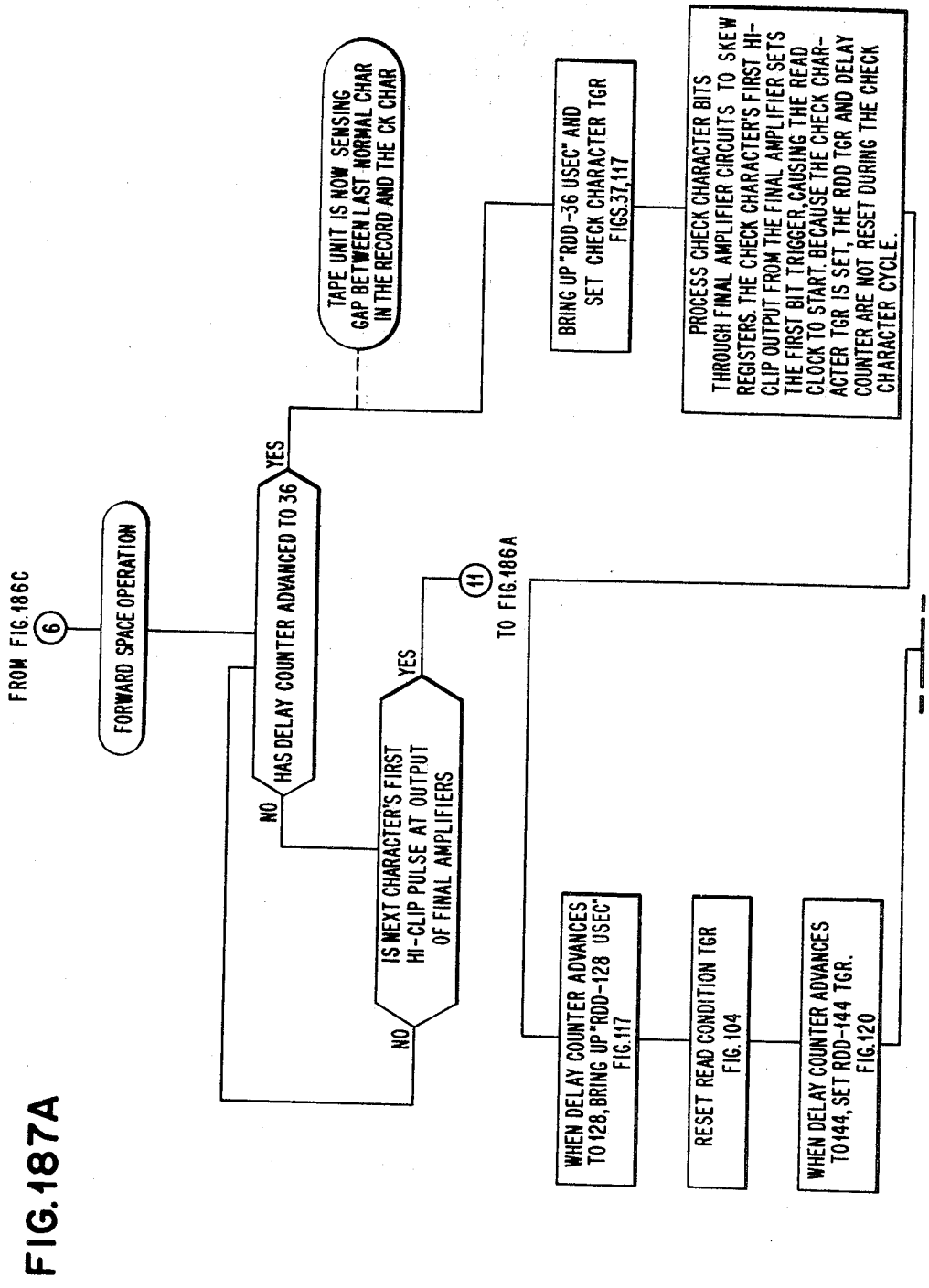

FROM FIG.186C
6
FORWARD SPACE OPERATION
HAS DELAY COUNTER ADVANCED TO 36
YES
NO
IS NEXT CHARACTER'S FIRST HI-CLIP PULSE AT OUTPUT OF FINAL AMPLIFIERS
YES
NO
11
TO FIG.186A
TAPE UNIT IS NOW SENSING GAP BETWEEN LAST NORMAL CHAR IN THE RECORD AND THE CK CHAR
BRING UP "RDD-36 USEC" AND SET CHECK CHARACTER TGR FIGS.37,117
PROCESS CHECK CHARACTER BITS THROUGH FINAL AMPLIFIER CIRCUITS TO SKEW REGISTERS. THE CHECK CHARACTER'S FIRST HI-CLIP OUTPUT FROM THE FINAL AMPLIFIER SETS THE FIRST BIT TRIGGER, CAUSING THE READ CLOCK TO START. BECAUSE THE CHECK CHARACTER TGR IS SET, THE RDD TGR AND DELAY COUNTER ARE NOT RESET DURING THE CHECK CHARACTER CYCLE.
WHEN DELAY COUNTER ADVANCES TO 128, BRING UP "RDD-128 USEC" FIG.117
RESET READ CONDITION TGR FIG.104
WHEN DELAY COUNTER ADVANCES TO 144, SET RDD-144 TGR. FIG.120

FIG.187A

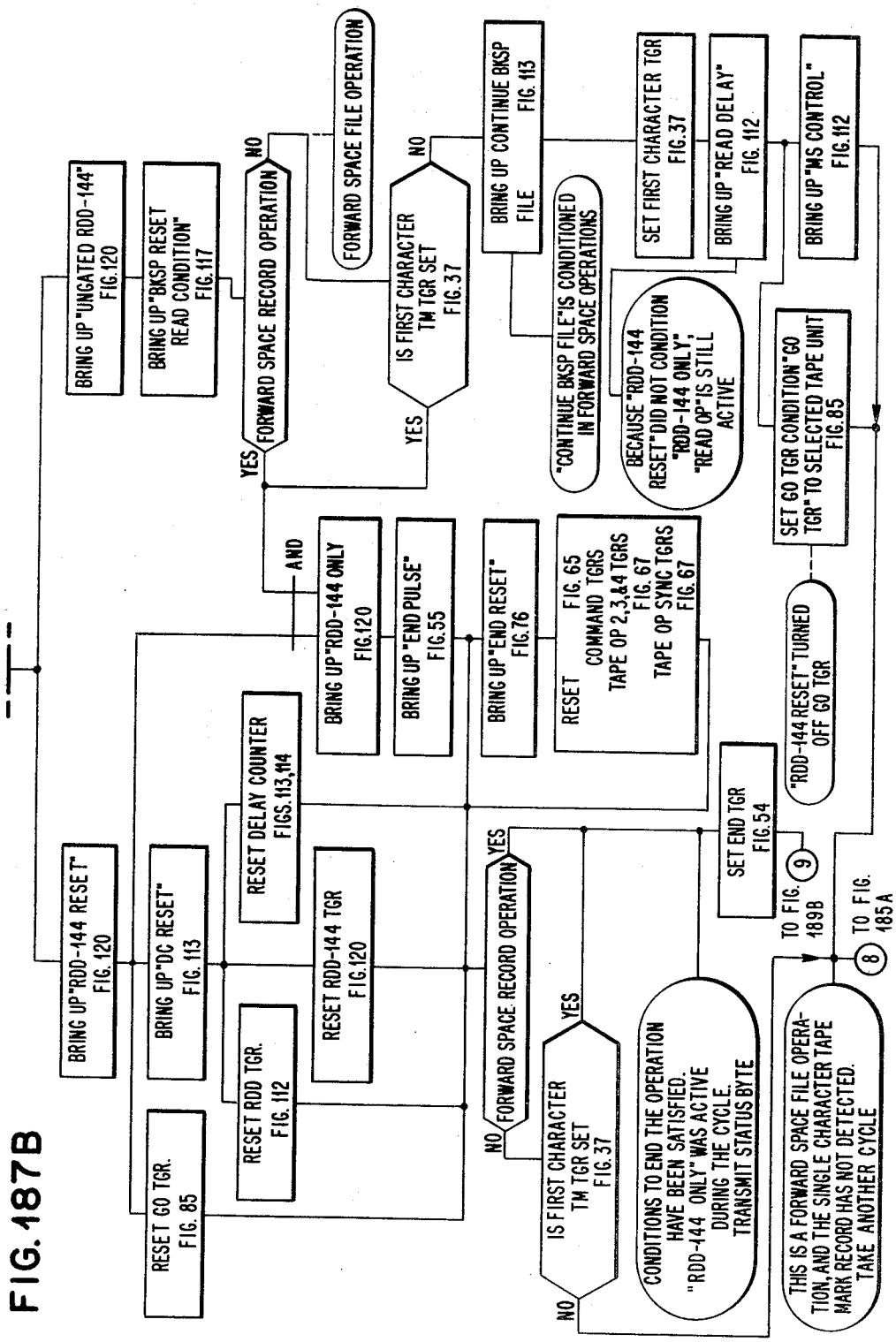

BRING UP "RDD-144 RESET" FIG. 120
BRING UP "DC RESET" FIG. 113
RESET DELAY COUNTER FIGS. 113, 114
RESET RDD-144 TGR FIG. 120
RESET RDD TGR. FIG. 112
RESET GO TGR. FIG. 85
BRING UP "UNGATED RDD-144" FIG. 120
BRING UP "BKSP RESET READ CONDITION" FIG. 117
FORWARD SPACE RECORD OPERATION
NO
YES
FORWARD SPACE FILE OPERATION
IS FIRST CHARACTER TM TGR SET FIG. 37
NO
YES
AND
BRING UP "RDD-144 ONLY" FIG. 120
BRING UP "END PULSE" FIG. 55
BRING UP "END RESET" FIG. 76
RESET FIG. 65 COMMAND TGRS TAPE OP 2,3,4,4 TGRS FIG. 67 TAPE OP SYNC TGRS FIG. 67
BRING UP CONTINUE BKSP FILE FIG. 113
"CONTINUE BKSP FILE" IS CONDITIONED IN FORWARD SPACE OPERATIONS
BECAUSE "RDD-144 RESET" DID NOT CONDITION "RDD-144 ONLY", "READ OP" IS STILL ACTIVE
SET FIRST CHARACTER TGR FIG. 37
BRING UP "READ DELAY" FIG. 112
BRING UP "MS CONTROL" FIG. 112
SET GO TGR CONDITION "GO TGR" TO SELECTED TAPE UNIT FIG. 85
"RDD-144 RESET" TURNED OFF GO TGR
SET END TGR FIG. 54
TO FIG. 189B
9
8
TO FIG. 185A
FORWARD SPACE RECORD OPERATION
NO
YES
IS FIRST CHARACTER TM TGR SET FIG. 37
NO
YES
CONDITIONS TO END THE OPERATION HAVE BEEN SATISFIED. "RDD-144 ONLY" WAS ACTIVE DURING THE CYCLE. TRANSMIT STATUS BYTE
THIS IS A FORWARD SPACE FILE OPERATION, AND THE SINGLE CHARACTER TAPE MARK RECORD HAS NOT DETECTED. TAKE ANOTHER CYCLE

FIG. 187B

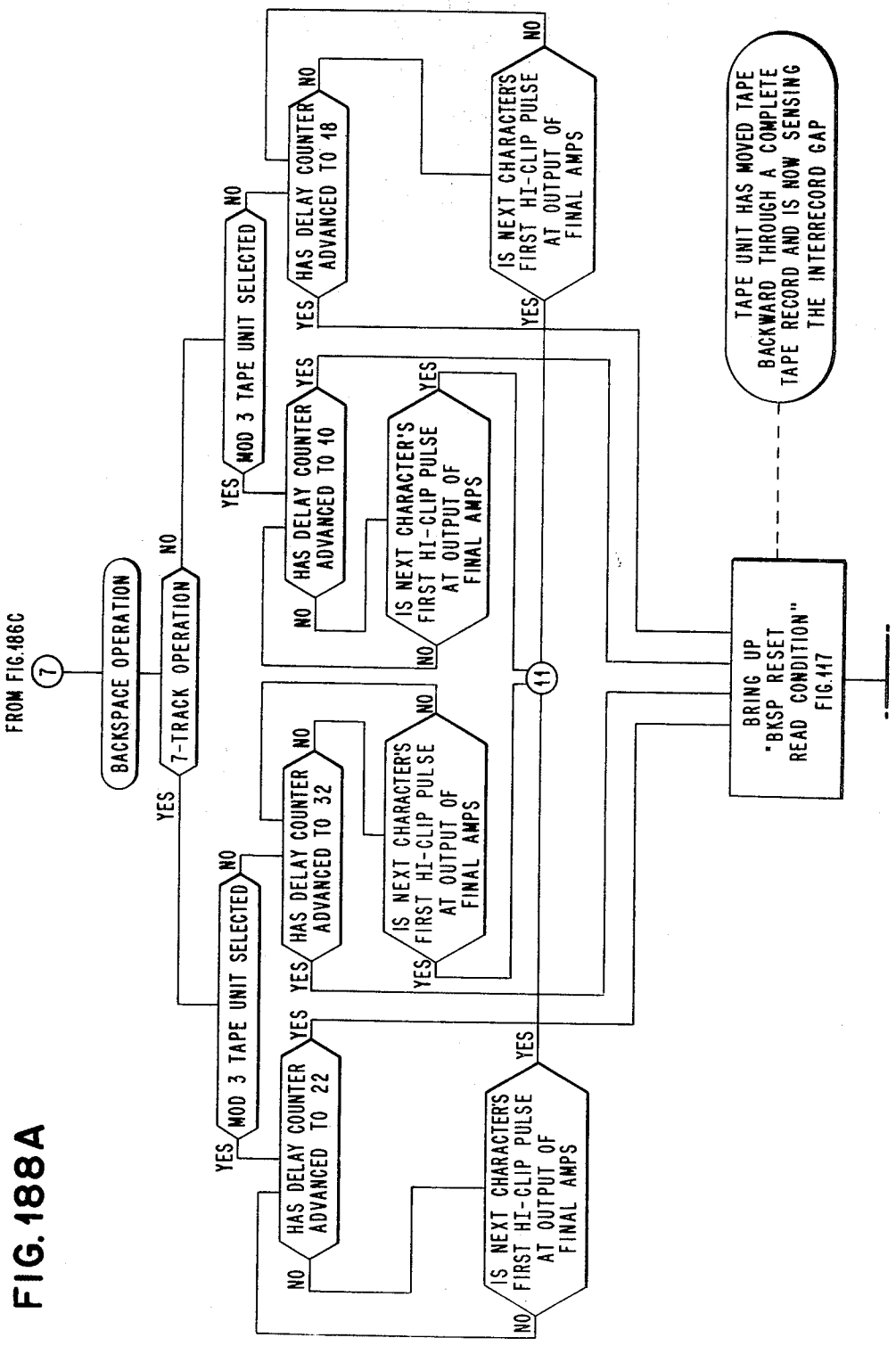

FROM FIG.186C
7
BACKSPACE OPERATION
7-TRACK OPERATION
NO
YES
MOD 3 TAPE UNIT SELECTED
NO
YES
HAS DELAY COUNTER ADVANCED TO 10
NO
YES
IS NEXT CHARACTER'S FIRST HI-CLIP PULSE AT OUTPUT OF FINAL AMPS
NO
YES
HAS DELAY COUNTER ADVANCED TO 18
NO
YES
IS NEXT CHARACTERS FIRST HI-CLIP PULSE AT OUTPUT OF FINAL AMPS
NO
YES
MOD 3 TAPE UNIT SELECTED
NO
YES
HAS DELAY COUNTER ADVANCED TO 32
NO
YES
IS NEXT CHARACTERS FIRST HI-CLIP PULSE AT OUTPUT OF FINAL AMPS
NO
YES
HAS DELAY COUNTER ADVANCED TO 22
NO
YES
IS NEXT CHARACTERS FIRST HI-CLIP PULSE AT OUTPUT OF FINAL AMPS
NO
YES
11
TAPE UNIT HAS MOVED TAPE BACKWARD THROUGH A COMPLETE TAPE RECORD AND IS NOW SENSING THE INTERRECORD GAP
BRING UP "BKSP RESET READ CONDITION" FIG.147

FIG. 188A

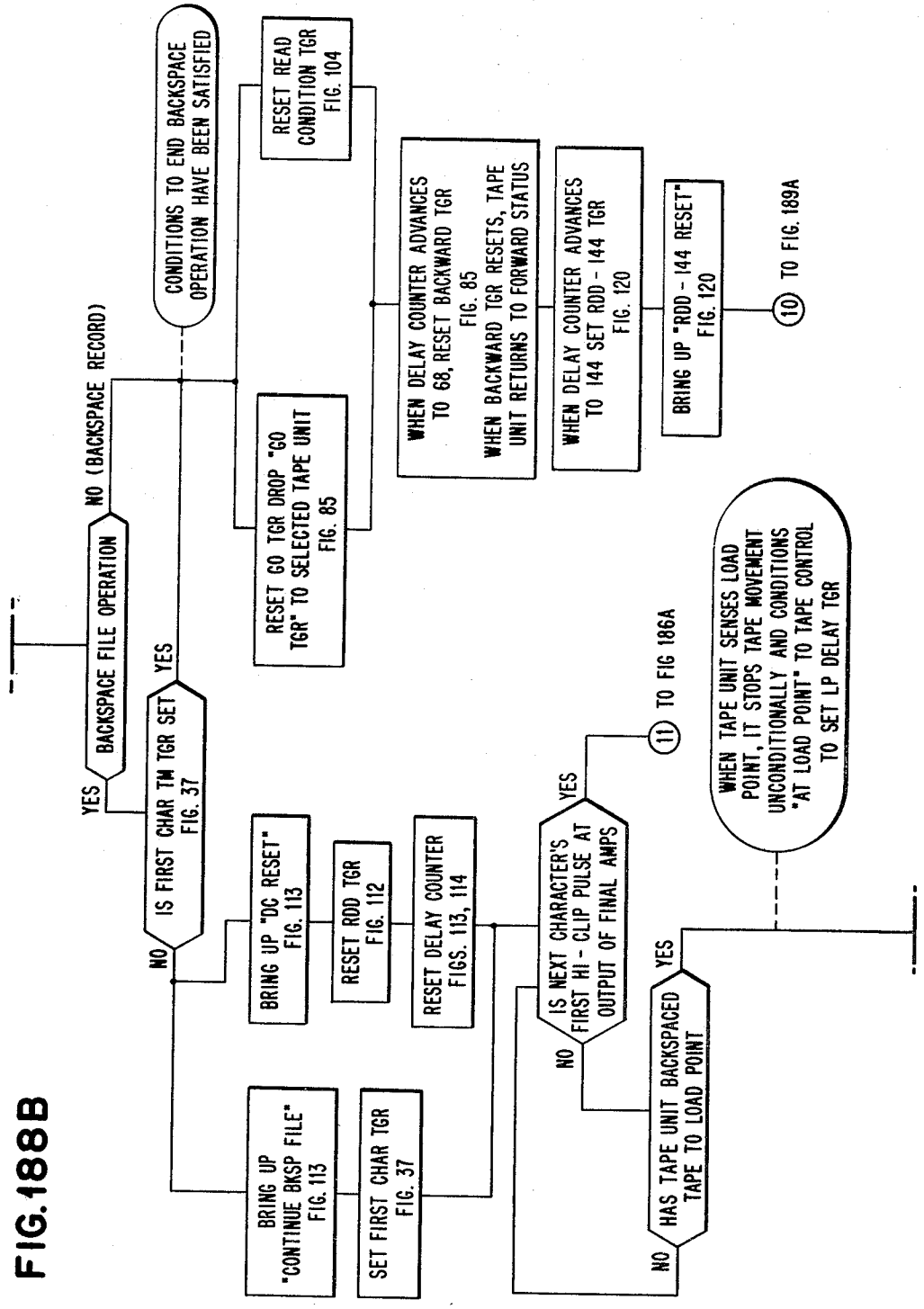

BACKSPACE FILE OPERATION
NO (BACKSPACE RECORD)
YES
IS FIRST CHAR TM TGR SET FIG. 37
YES
NO
CONDITIONS TO END BACKSPACE OPERATION HAVE BEEN SATISFIED
RESET READ CONDITION TGR FIG. 104
RESET GO TGR DROP "GO TGR" TO SELECTED TAPE UNIT FIG. 85
WHEN DELAY COUNTER ADVANCES TO 68, RESET BACKWARD TGR FIG. 85
WHEN BACKWARD TGR RESETS, TAPE UNIT RETURNS TO FORWARD STATUS
WHEN DELAY COUNTER ADVANCES TO 144 SET RDD - 144 TGR FIG. 120
BRING UP "RDD - 144 RESET" FIG. 120
10 TO FIG. 189A
BRING UP "DC RESET" FIG. 113
RESET RDD TGR FIG. 112
RESET DELAY COUNTER FIGS. 113, 114
IS NEXT CHARACTER'S FIRST HI - CLIP PULSE AT OUTPUT OF FINAL AMPS
YES
11 TO FIG 186A
NO
WHEN TAPE UNIT SENSES LOAD POINT, IT STOPS TAPE MOVEMENT UNCONDITIONALLY AND CONDITIONS "AT LOAD POINT" TO TAPE CONTROL TO SET LP DELAY TGR
BRING UP "CONTINUE BKSP FILE" FIG. 113
SET FIRST CHAR TGR FIG. 37
HAS TAPE UNIT BACKSPACED TAPE TO LOAD POINT
YES
NO

FIG. 188B

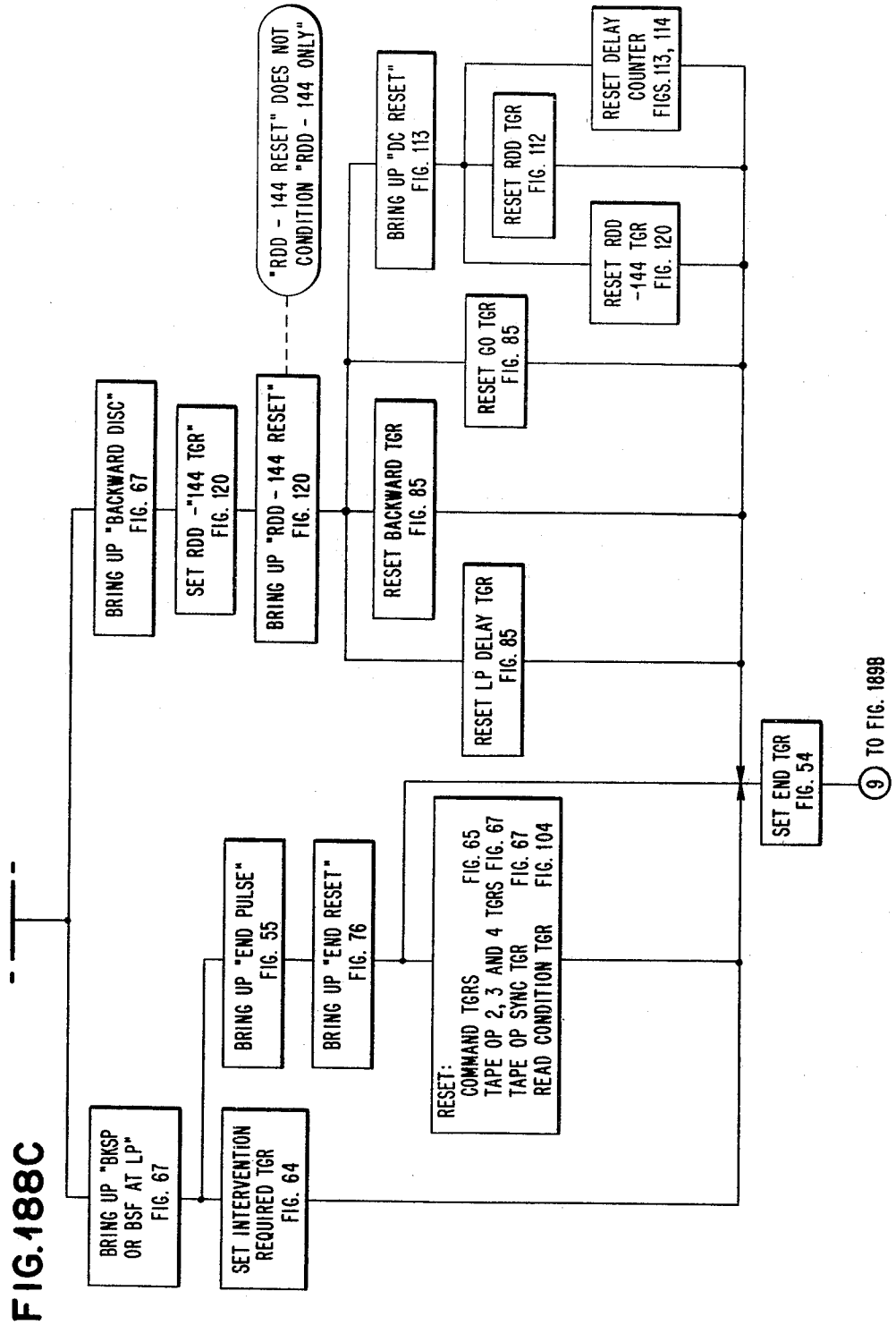

BRING UP "BKSP OR BSF AT LP" FIG. 67
SET INTERVENTION REQUIRED TGR FIG. 64
BRING UP "END PULSE" FIG. 55
BRING UP "END RESET" FIG. 76
RESET: COMMAND TGRS FIG. 65
TAPE OP 2, 3 AND 4 TGRS FIG. 67
TAPE OP SYNC TGR FIG. 67
READ CONDITION TGR FIG. 104
BRING UP "BACKWARD DISC" FIG. 67
SET RDD -"144 TGR" FIG. 120
BRING UP "RDD - 144 RESET" FIG. 120
"RDD - 144 RESET" DOES NOT CONDITION "RDD - 144 ONLY"
RESET BACKWARD TGR FIG. 85
RESET LP DELAY TGR FIG. 85
BRING UP "DC RESET" FIG. 113
RESET GO TGR FIG. 85
RESET RDD TGR FIG. 112
RESET RDD -144 TGR FIG. 120
RESET DELAY COUNTER FIGS. 113, 114
SET END TGR FIG. 54
9
TO FIG. 189B

FIG. 188C

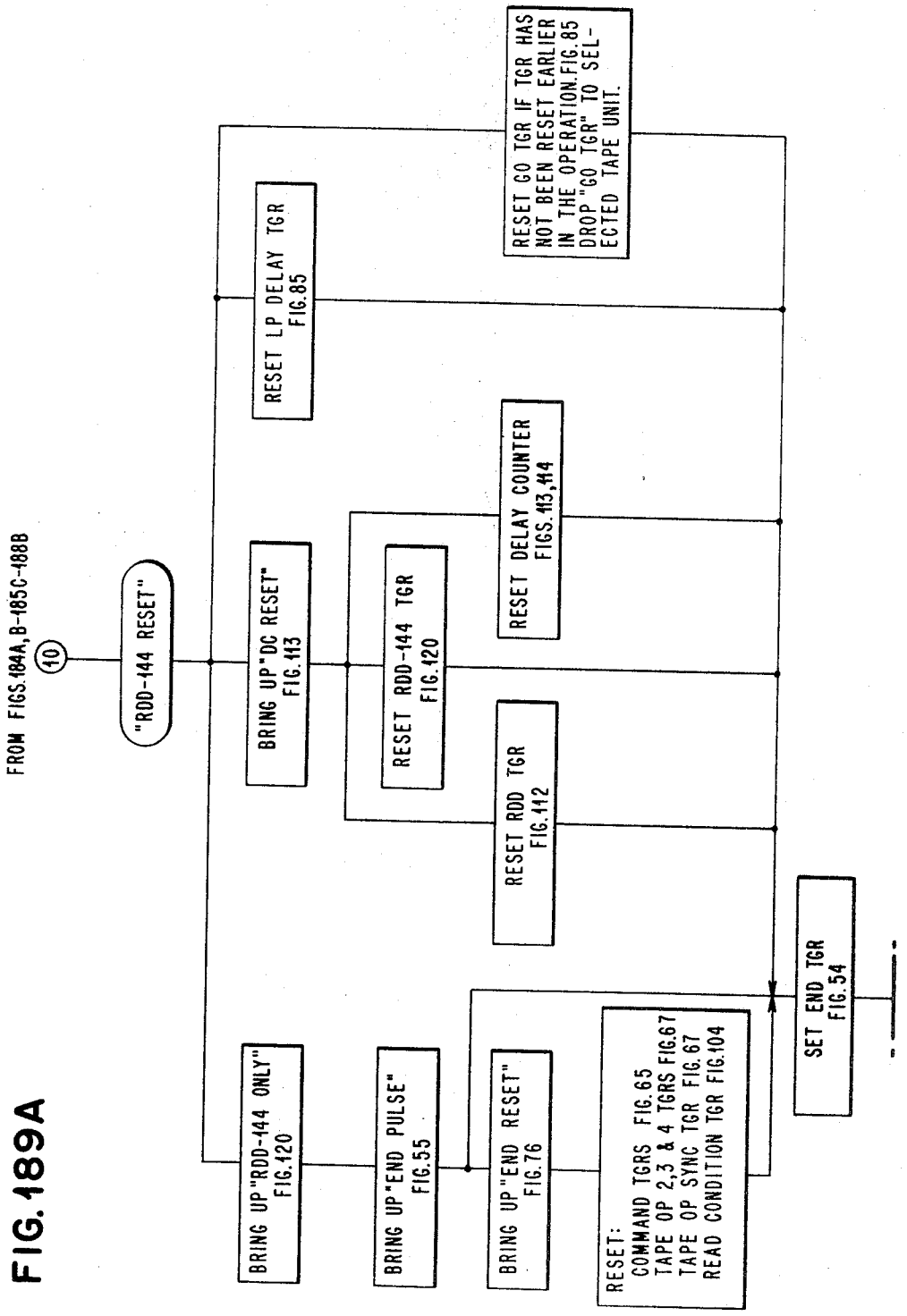

FROM FIGS.184A,B-185C-188B
10
"RDD-144 RESET"
BRING UP "DC RESET" FIG.113
RESET LP DELAY TGR FIG.85
RESET GO TGR IF TGR HAS NOT BEEN RESET EARLIER IN THE OPERATION.FIG.85 DROP "GO TGR" TO SELECTED TAPE UNIT.
RESET RDD-144 TGR FIG.120
RESET DELAY COUNTER FIGS.113,114
RESET RDD TGR FIG.112
BRING UP "RDD-144 ONLY" FIG.120
BRING UP "END PULSE" FIG.55
BRING UP "END RESET" FIG.76
RESET:
COMMAND TGRS FIG.65
TAPE OP 2,3 & 4 TGRS FIG.67
TAPE OP SYNC TGR FIG.67
READ CONDITION TGR FIG.104
SET END TGR FIG.54

FIG.189A

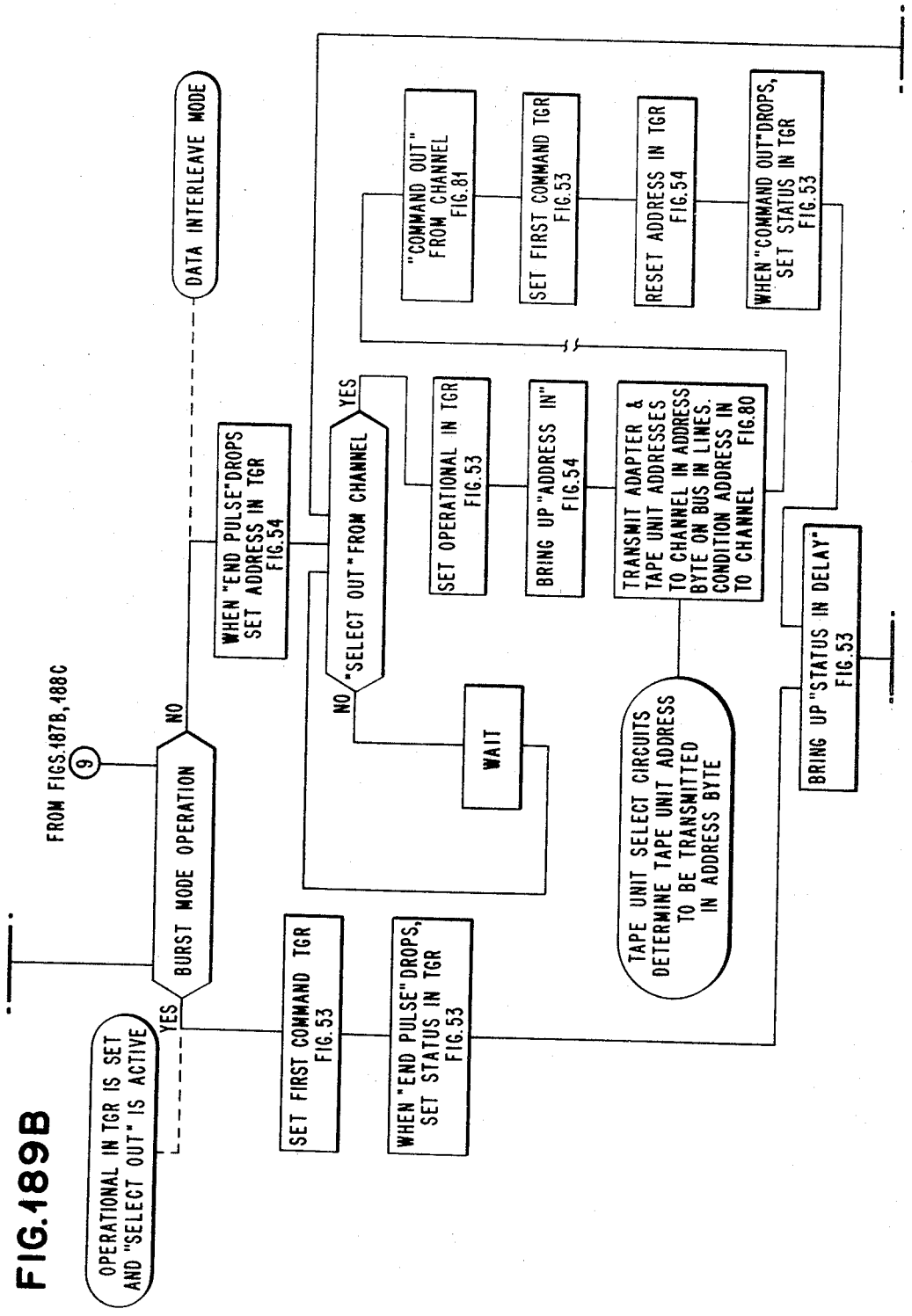

FROM FIGS.187B, 188C
9
BURST MODE OPERATION
NO
YES
OPERATIONAL IN TGR IS SET AND "SELECT OUT" IS ACTIVE
SET FIRST COMMAND TGR FIG.53
WHEN "END PULSE" DROPS, SET STATUS IN TGR FIG.53
DATA INTERLEAVE MODE
WHEN "END PULSE" DROPS SET ADDRESS IN TGR FIG.54
"SELECT OUT" FROM CHANNEL
YES
NO
WAIT
SET OPERATIONAL IN TGR FIG.53
BRING UP "ADDRESS IN" FIG.54
TRANSMIT ADAPTER & TAPE UNIT ADDRESSES TO CHANNEL IN ADDRESS BYTE ON BUS IN LINES. CONDITION ADDRESS IN TO CHANNEL FIG.80
TAPE UNIT SELECT CIRCUITS DETERMINE TAPE UNIT ADDRESS TO BE TRANSMITTED IN ADDRESS BYTE
"COMMAND OUT" FROM CHANNEL FIG.81
SET FIRST COMMAND TGR FIG.53
RESET ADDRESS IN TGR FIG.54
WHEN "COMMAND OUT" DROPS, SET STATUS IN TGR FIG.53
BRING UP "STATUS IN DELAY" FIG.53

FIG.189B

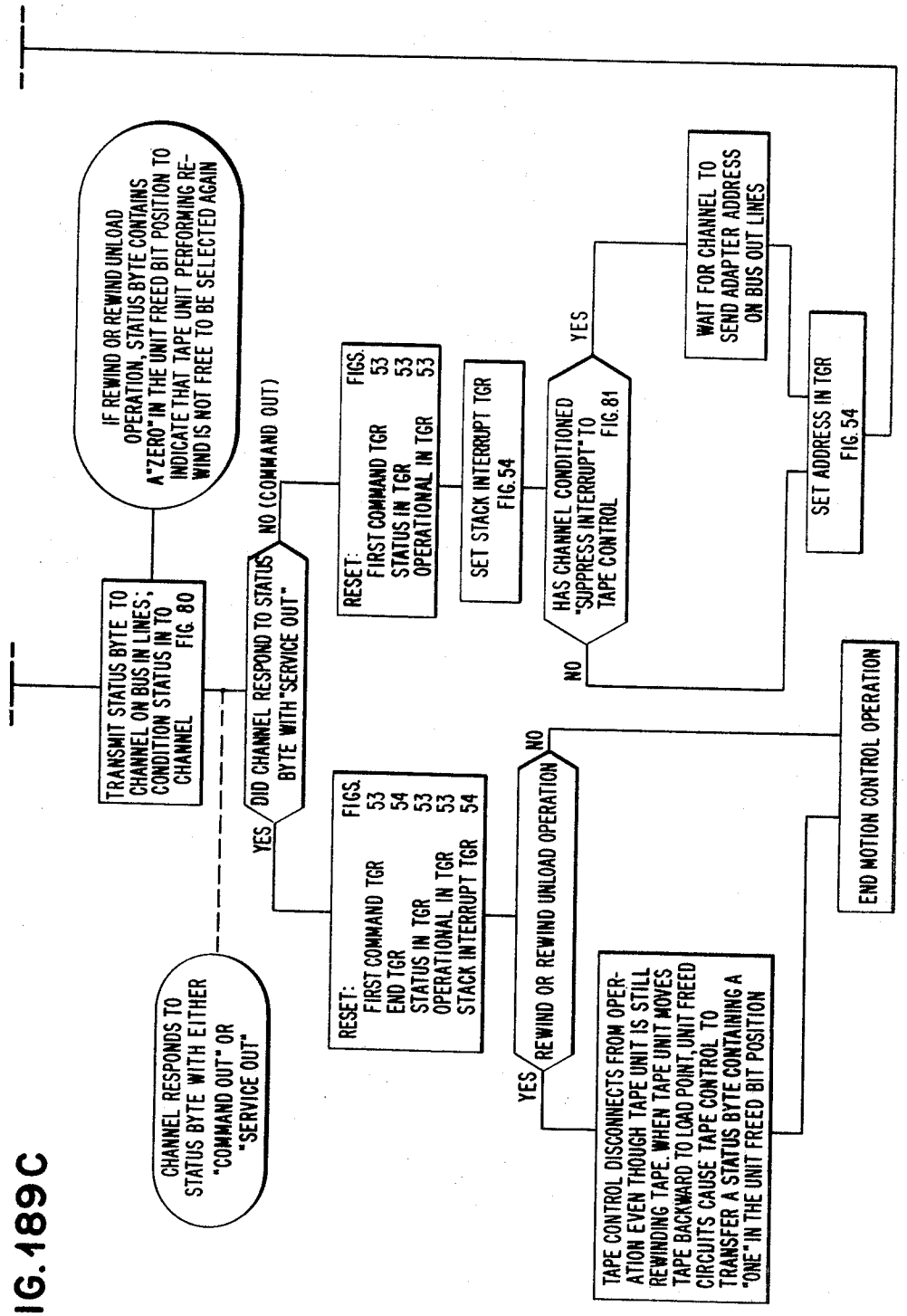

IF REWIND OR REWIND UNLOAD OPERATION, STATUS BYTE CONTAINS A "ZERO" IN THE UNIT FREED BIT POSITION TO INDICATE THAT TAPE UNIT PERFORMING REWIND IS NOT FREE TO BE SELECTED AGAIN
TRANSMIT STATUS BYTE TO CHANNEL ON BUS IN LINES; CONDITION STATUS IN TO CHANNEL FIG. 80
DID CHANNEL RESPOND TO STATUS BYTE WITH "SERVICE OUT"
NO (COMMAND OUT)
YES
RESET: FIRST COMMAND TGR FIGS. 53
STATUS IN TGR 53
OPERATIONAL IN TGR 53
SET STACK INTERRUPT TGR FIG. 54
HAS CHANNEL CONDITIONED "SUPPRESS INTERRUPT" TO TAPE CONTROL FIG. 81
YES
NO
WAIT FOR CHANNEL TO SEND ADAPTER ADDRESS ON BUS OUT LINES
SET ADDRESS IN TGR FIG. 54
CHANNEL RESPONDS TO STATUS BYTE WITH EITHER "COMMAND OUT" OR "SERVICE OUT"
RESET: FIRST COMMAND TGR FIGS. 53
END TGR 54
STATUS IN TGR 53
OPERATIONAL IN TGR 53
STACK INTERRUPT TGR 54
REWIND OR REWIND UNLOAD OPERATION
NO
YES
END MOTION CONTROL OPERATION
TAPE CONTROL DISCONNECTS FROM OPERATION EVEN THOUGH TAPE UNIT IS STILL REWINDING TAPE. WHEN TAPE UNIT MOVES TAPE BACKWARD TO LOAD POINT, UNIT FREED CIRCUITS CAUSE TAPE CONTROL TO TRANSFER A STATUS BYTE CONTAINING A "ONE" IN THE UNIT FREED BIT POSITION

FIG. 189C

FIG. 190A

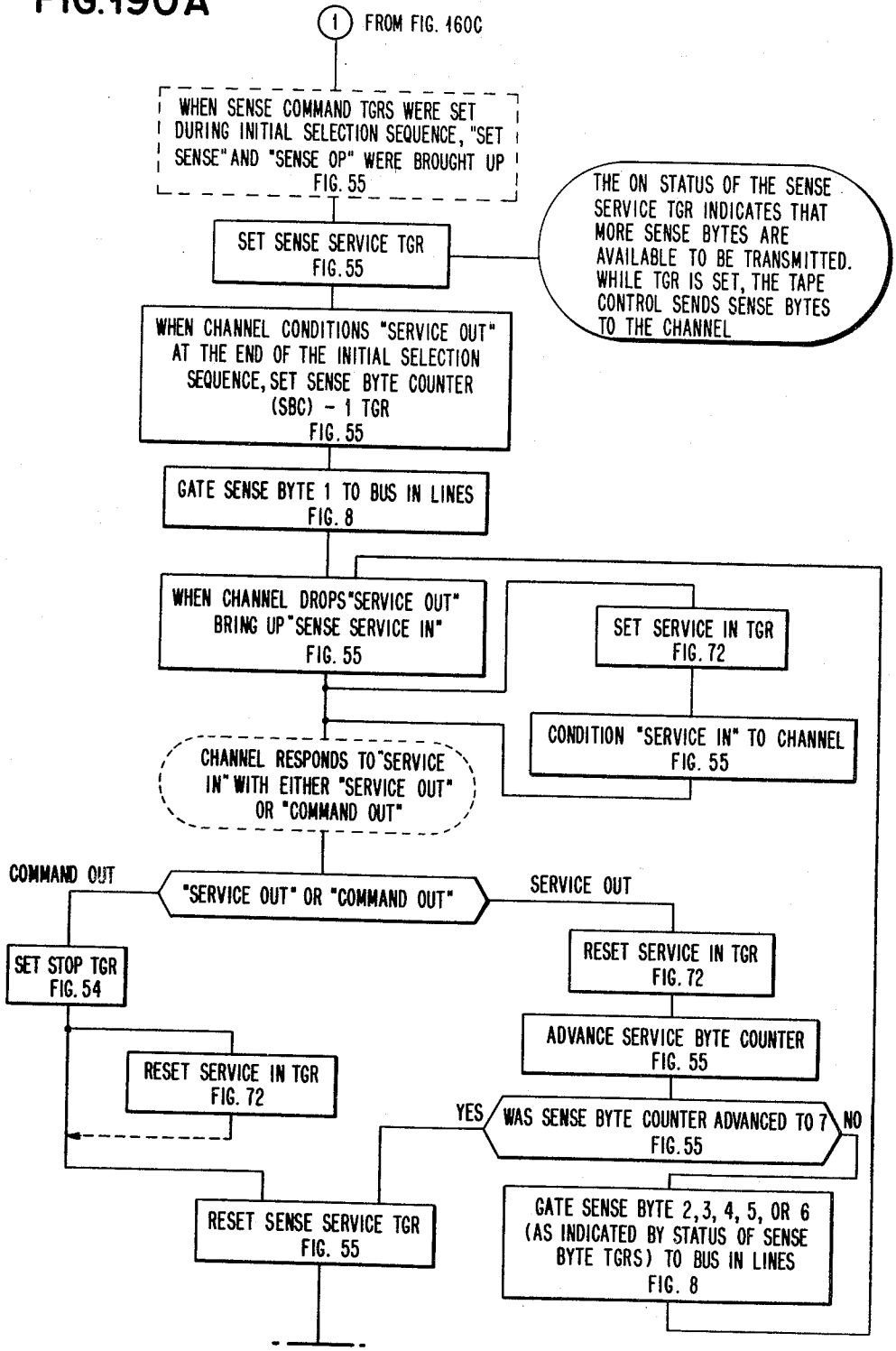
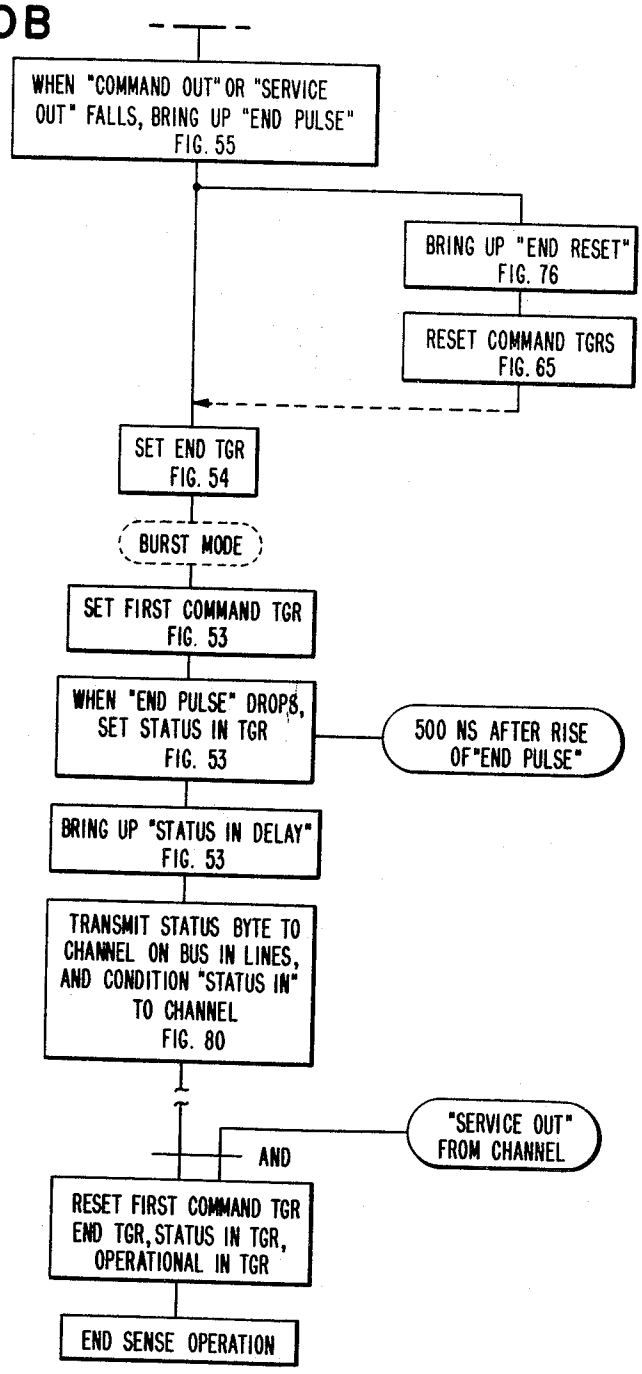

1 FROM FIG. 160C
WHEN SENSE COMMAND TGRS WERE SET DURING INITIAL SELECTION SEQUENCE, "SET SENSE" AND "SENSE OP" WERE BROUGHT UP FIG. 55
THE ON STATUS OF THE SENSE SERVICE TGR INDICATES THAT MORE SENSE BYTES ARE AVAILABLE TO BE TRANSMITTED. WHILE TGR IS SET, THE TAPE CONTROL SENDS SENSE BYTES TO THE CHANNEL
SET SENSE SERVICE TGR FIG. 55
WHEN CHANNEL CONDITIONS "SERVICE OUT" AT THE END OF THE INITIAL SELECTION SEQUENCE, SET SENSE BYTE COUNTER (SBC) - 1 TGR FIG. 55
GATE SENSE BYTE 1 TO BUS IN LINES FIG. 8
WHEN CHANNEL DROPS "SERVICE OUT" BRING UP "SENSE SERVICE IN" FIG. 55
SET SERVICE IN TGR FIG. 72
CONDITION "SERVICE IN" TO CHANNEL FIG. 55
CHANNEL RESPONDS TO "SERVICE IN" WITH EITHER "SERVICE OUT" OR "COMMAND OUT"
COMMAND OUT
"SERVICE OUT" OR "COMMAND OUT"
SERVICE OUT
SET STOP TGR FIG. 54
RESET SERVICE IN TGR FIG. 72
RESET SERVICE IN TGR FIG. 72
ADVANCE SERVICE BYTE COUNTER FIG. 55
YES
WAS SENSE BYTE COUNTER ADVANCED TO 7 FIG. 55
NO
RESET SENSE SERVICE TGR FIG. 55
GATE SENSE BYTE 2, 3, 4, 5, OR 6 (AS INDICATED BY STATUS OF SENSE BYTE TGRS) TO BUS IN LINES FIG. 8

FIG. 190B


WHEN "COMMAND OUT" OR "SERVICE OUT" FALLS, BRING UP "END PULSE" FIG. 55
BRING UP "END RESET" FIG. 76
RESET COMMAND TGRS FIG. 65
SET END TGR FIG. 54
BURST MODE
SET FIRST COMMAND TGR FIG. 53
WHEN "END PULSE" DROPS, SET STATUS IN TGR FIG. 53
500 NS AFTER RISE OF "END PULSE"
BRING UP "STATUS IN DELAY" FIG. 53
TRANSMIT STATUS BYTE TO CHANNEL ON BUS IN LINES, AND CONDITION "STATUS IN" TO CHANNEL FIG. 80
"SERVICE OUT" FROM CHANNEL
AND
RESET FIRST COMMAND TGR END TGR, STATUS IN TGR, OPERATIONAL IN TGR
END SENSE OPERATION

ERROR DETECTION AND CORRECTION SYSTEM

Filed April 6, 1964 316 Sheets-Sheet 311

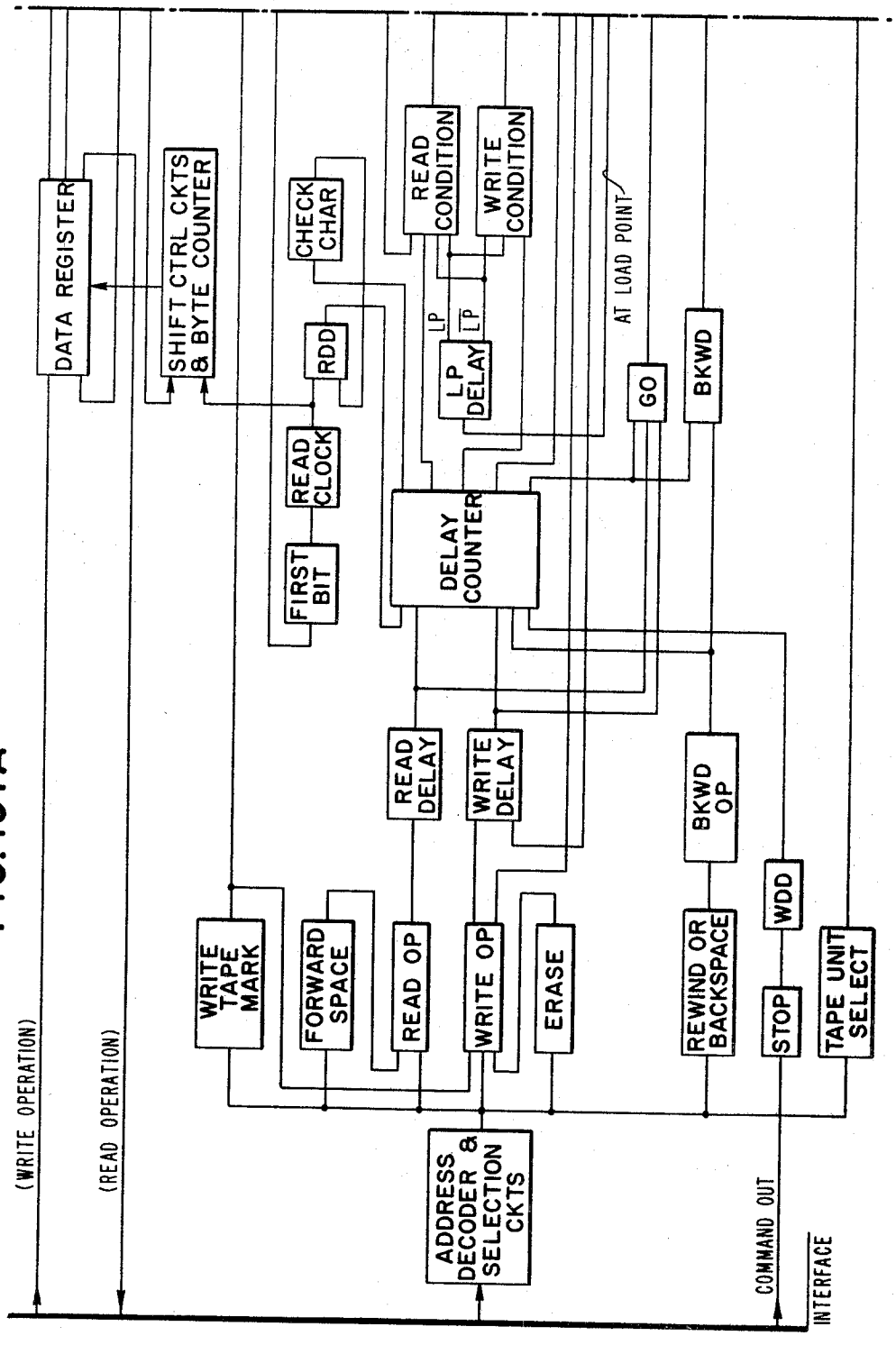



FIG. 191A

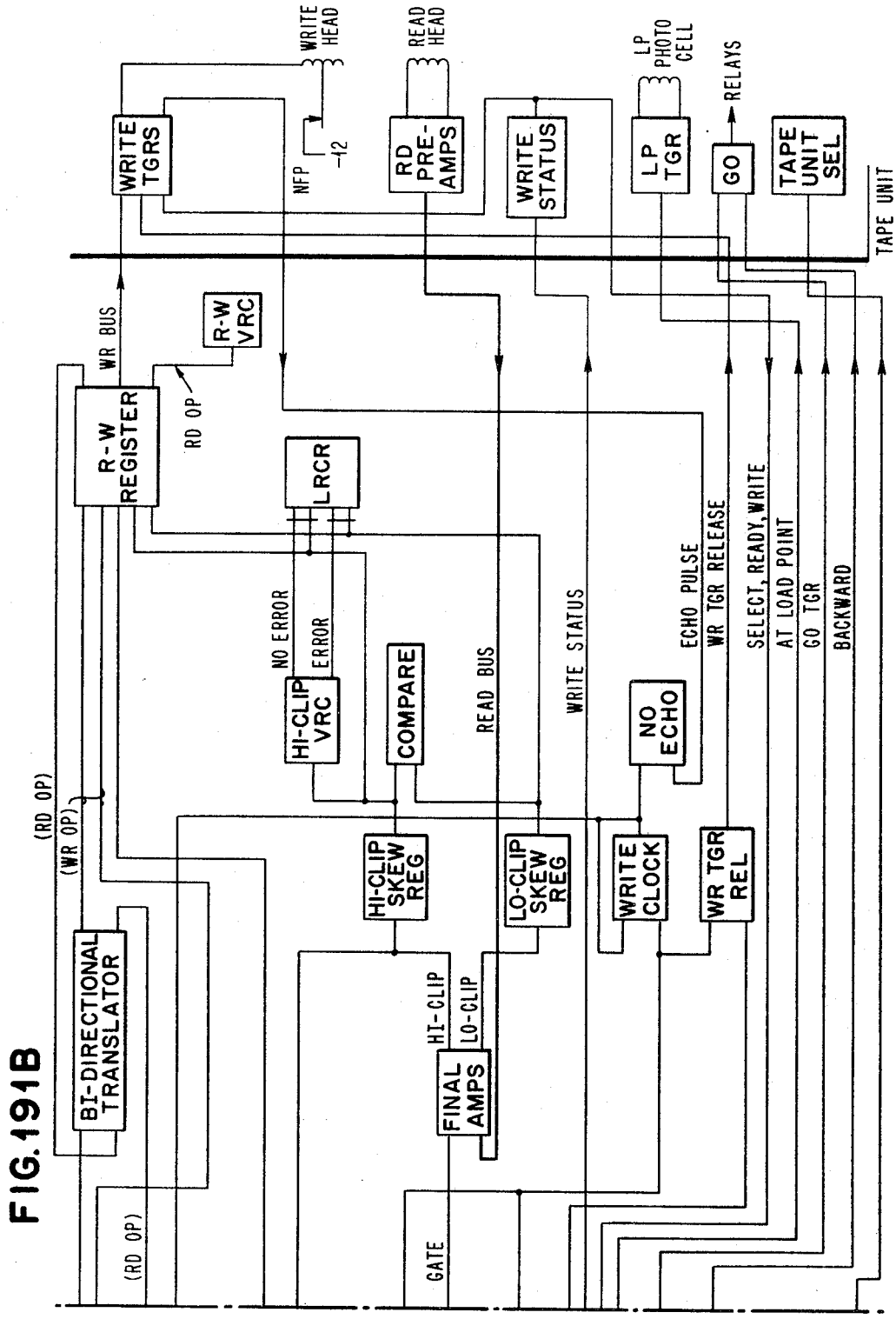

WRITE HEAD
READ HEAD
LP PHOTO CELL
RELAYS
WRITE TGRS
NFP
-12
RD PRE-AMPS
WRITE STATUS
LP TGR
GO
TAPE UNIT SEL
TAPE UNIT
WR BUS
R-W VRC
RD OP
R-W REGISTER
LRCR
NO ERROR
ERROR
HI-CLIP VRC
COMPARE
READ BUS
WRITE STATUS
ECHO PULSE
WR TGR RELEASE
SELECT, READY, WRITE
AT LOAD POINT
GO TGR
BACKWARD
NO ECHO
(RD OP)
(WR OP)
HI-CLIP SKEW REG
LO-CLIP SKEW REG
WRITE CLOCK
WR TGR REL
BI-DIRECTIONAL TRANSLATOR
HI-CLIP
LO-CLIP
FINAL AMPS
(RD OP)
GATE

FIG. 191B

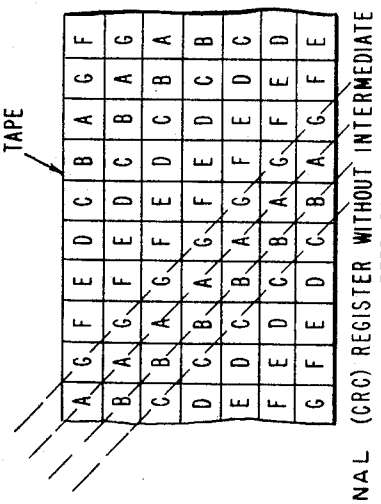



FIG. 194

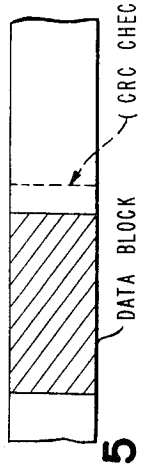



FIG. 195

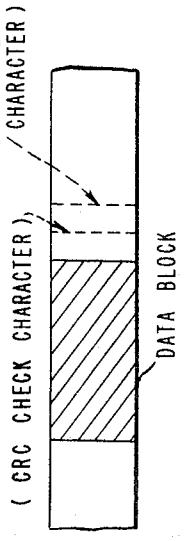



FIG. 196

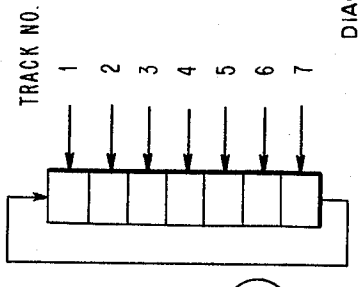



FIG. 192

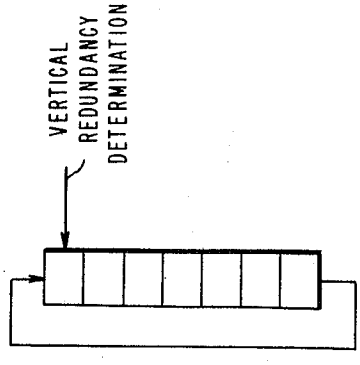



FIG. 193

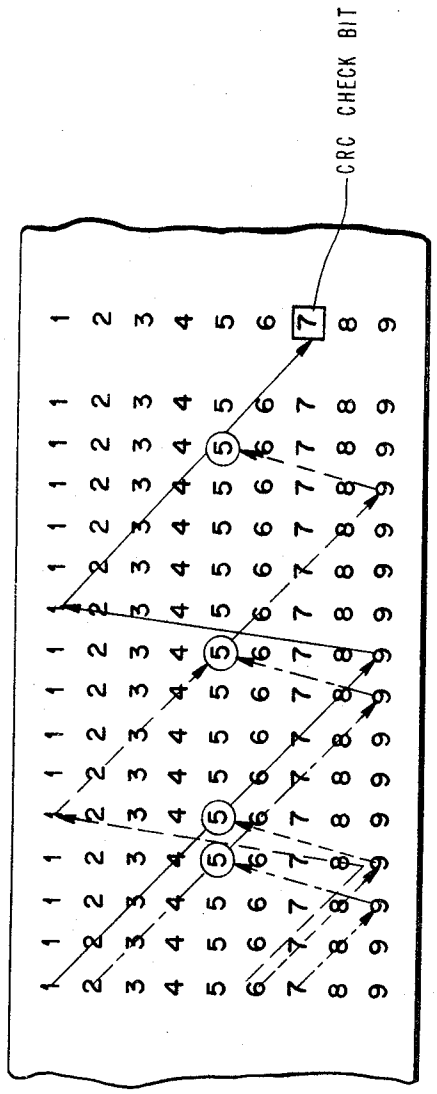



FIG. 197

CRC REGISTER IN FIG. 200
(WITH INTERMEDIATE FEEDBACK TO STAGE 5)

FIG. 200

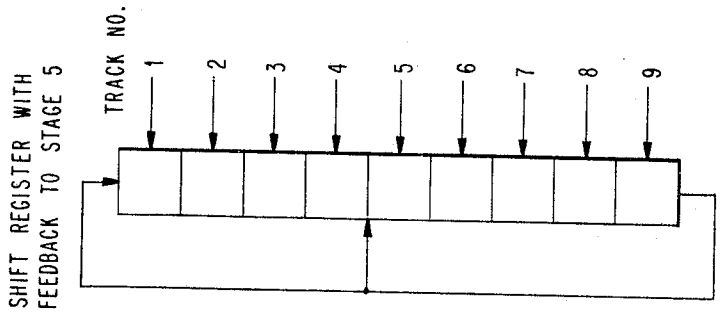



FIG. 199

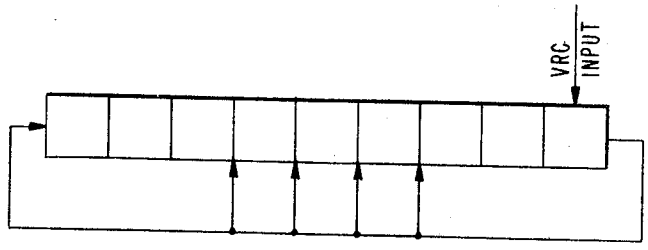
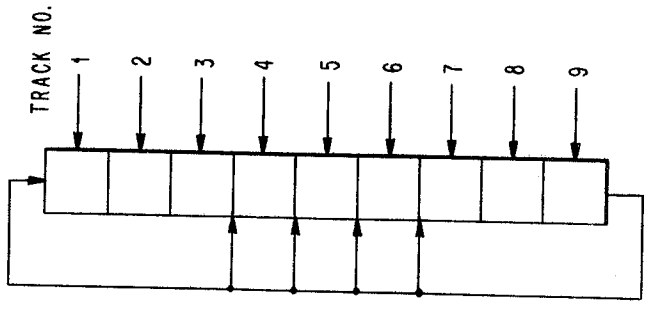



FIG. 198



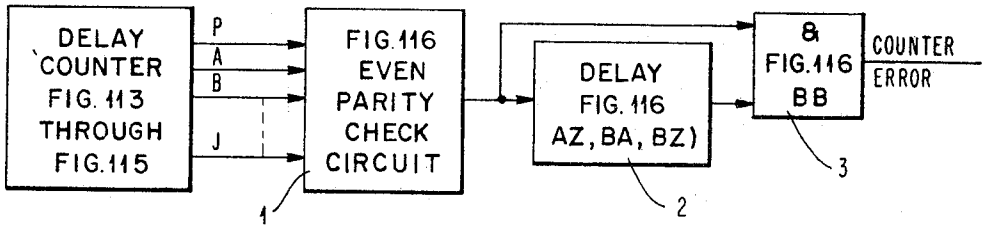



FIG. 201

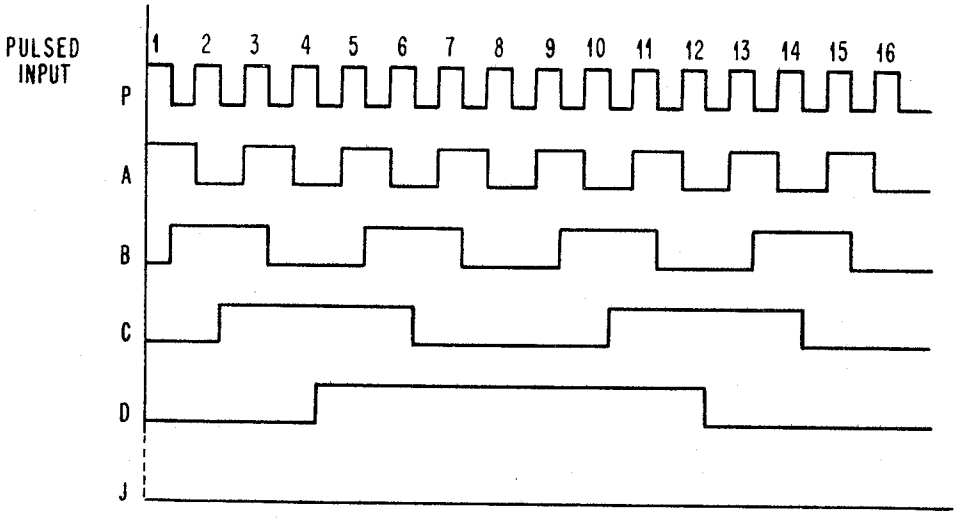



FIG. 202

United States Patent Office

3,508,194
Patented Apr. 21, 1970



3,508,194

# ERROR DETECTION AND CORRECTION SYSTEM

David T. Brown, Wappingers Falls, N.Y., assignor to International Business Machines Corporation, New York, N.Y., a corporation of New York

Filed Apr. 6, 1964, Ser. No. 357,368
Int. Cl. G08b *29/00*; G06f *11/00*
U.S. Cl. 340—146.1 10 Claims

## ABSTRACT OF THE DISCLOSURE

Locating any tape track having an error in a data block. The bit positions in a byte are in respectively different tape tracks. To locate any track having from one to all bit positions erroneous, a first cyclic storage means (CRC shift register) receives all of the bits of every byte in parallel, including a CRC redundancy byte, for an entire block. A second cyclic storage means (EP shift register) serially receives the output from a byte redundancy check (VRC) circuit which checks the parity for each byte in the block. Each storage means has at least end-around feedback. The CRC shift register is also used to generate and write the CRC byte at the end of a tape block. After a block is read, the two cyclic storage means should have bit patterns which correspond in a predetermined manner if no error occurred in the block. An error is indicated by non-correspondence. Then the pattern of one storage means is moved relative to the pattern in the other until the patterns correspond. The track in the block having the erroneous bits is determined by the number of bit positions of relative movement needed to obtain correspondence between the patterns. A diagonal redundancy byte is effective as the cyclic redundancy byte where the number of tape tracks is a prime number, such as 1, 2, 3, 5, 7, 11, 13, etc.

INDEX





This invention relates generally to error detection and correction systems. In particular, this invention relates to an error detection and correction system utilizing cyclic redundancy and is particularly pertinent to systems having a prime number of bits in a byte.

The term "byte" herein means a sub-word which is a fixed number of data bits handled as a matter of convenience by the hardware of a data system. Although the bits of a byte generally are transferred in parallel, they may be transferred serially in some situations.

This invention provides an error detection and correction system which has unique characteristics over prior error detection and correction systems. For example, it has been previously known to use a longitudinal redundancy check byte at the end of the data block and to use byte redundancy (character redundancy) for each byte in the block. This has been done for a number of years to obtain error detection for data read from magnetic tape. Also, it has been well known that either odd or even parity, or a fixed number of 1-bits check may be utilized as an effective form of byte redundancy (character redundancy).

Error check circuits do not provide error correction. Generally all error correction systems require error detection means in a form which can locate one or more erroneous bits. The location of any erroneous binary bit position permits error correction by inversion since a binary bit can only have two states; and if one state is determined to be erroneous, the other binary state must be correct.

Cyclic redundancy for error detection has been described generally in an article by D. T. Brown and W. W. Peterson, "Cyclic Codes for Error Detection," published in Proceedings of the I.R.E., No. 49 (January 1961). Cyclic error detection and correction is also treated in a book by W. W. Peterson "Error-Correcting Codes," Wiley & Sons (1961), and references cited in the book. The book and its references primarily are mathematical treatises on cyclic redundancy detection and correction for bit serial operation only. However, parallel bit operation for cyclic redundancy error detection is disclosed in other prior art which, however, does not disclose any error correction system utilizing cyclic redundancy. Error correction utilizing a combination of longitudinal redundancy and byte redundancy is described and claimed in the prior art such as Patent No. 3,273,120, issued Sept. 13, 1966 to R. V. McFadden, D. R. Dustin, C. P. Rauf, and G. G. Unger, entitled "Error Correction System By Retransmission of Erroneous Data." This last reference discloses how an er-

roneous bit in a data block read from tape can be corrected after the track in error has been determined by rereading the tape and utilizing the byte redundancy during the rereading to determine the particular byte having the error and reinserting a bit lost by drop out.

Prior error correction systems utilizing longitudinal redundancy to determine an erroneous bit position in a byte of a data block have the difficulty that longitudinal redundancy is only capable of detecting 50 percent of the possible error pattern which can exist in any common bit position of a block of data. This invention describes and claims means by which a common bit position having an error in a data block can be located for much greater than 50 percent of such error patterns. Depending upon the particular design of the invention, it can locate in excess of 95 percent of the possible error patterns that can exist in any given common bit position of a data block.

It is an object of this invention to provide means for controlling and communicating data with a high order of reliability.

It is an object of this invention to provide means for controlling a storage device to provide a high order of reliability in communicating data from the storage device.

It is another object of this invention to provide control means for a storage medium enabling error correction of data read from the storage medium.

It is another object of this invention to provide a control means for one or more tape drives capable of writing tape with a format which enables error correction of tape data.

It is another object of this invention to provide control means for one or more tape drives which can correct all of the bit errors occurring at least within any single track of a tape block.

It is a further object of this invention to provide cyclically-operating error detection during the reading of tape.

It is a further object of this invention to provide means for error correcting the data read from tape by utilizing cyclic-redundancy error-checking means.

It is a further object of this invention to provide a data format which utilizes a cyclic redundancy check character written either before, after, or within a data block.

This invention pertains to finding a common bit position having an error in any byte of a data block derived from any source, such as core memory, film memory, etc. This invention utilizes a storage means having wrap-around feedback for either generating or detecting cyclic redundancy with a data block; cyclic redundancy alone is used to locate an erroneous bit position in a byte of a block. Finding an erroneous bit position in a byte of a data block is a partial location of an error in a data block. In many magnetic tape systems, the bit positions in a byte are represented by the different tape tracks. The invention utilizes two cyclic storage means which are each capable of developing separate error patterns that have a particular bit correlation, in the absence of any error in a block. One cyclic storage means receives the data in the block, which may include redundancy. The other register receives an input from a byte redundancy check circuit which examines each byte in the block. Each cyclic storage means can be any one of number of circuits capable of moving data internally while receiving data, such as a delay line or a circuit capable of step-shifting bits such as a shift register. The two cyclic storage means are designated a cyclic redundancy check (CRC) storage means and error pattern (EP) storage means. Thus the CRC storage means can receive all of the bits of a data block, while the EP storage means receives the byte redundancy indication for each byte of the block. After a block is read with a cyclic redundancy byte, these two cyclic storage means should have patterns which correlate in a predetermined manner if no error occurred in the block. However, if an error did occur, the bit position in any byte having the error can be determined by moving the pattern of one of the storage means relative to the pattern of the other storage means until a particular predetermined correlation is obtained between the bit patterns. In this case the number of bit positions of relative movement needed to obtain a predetermined correlation relates to an erroneous bit position in any byte of the block. This particular means for determining the location of an erroneous bit position of any byte in the data block has been found to be particularly effective in cases where the number of bit positions in a byte is a prime number, such as 1, 2, 3, 5, 7, 11, 13, etc.

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of the preferred embodiments of the invention, as illustrated in the accompanying drawings.

DESCRIPTIVE TABLE OF DRAWINGS

| Title description of drawing: | Figure number |
|---|---|
| General Partial Error Location System | 1 |
| Error Correction System | 2 |
| Tape Control System | 3A & B |
| Interface In | 4 |
| Interface Out | 5 |
| Tape Unit Read Lines | 6 |
| Tape Unit Write Lines | 7 |
| Bus In 0 and 1 | 8 |
| Bus In 2 and 3 | 9 |
| Bus In 4 and 5 | 10 |
| Bus In 6 and 7 | 11 |
| Data Register Positions 12, 11 | 12 |
| Data Register Positions 10, 9 | 13 |
| Data Register Positions 8, 7 | 14 |
| Data Register Positions 6, 5 | 15 |
| Data Register Positions 4, 3 | 16 |
| Data Register Positions 2, 1 | 17 |
| R-W (Read-Write) Register Positions 4, 5, 6, 7 | 18A & B |
| R-W Register Positions 1, 2, 3 | 19A & B |
| R-W Register Positions OP | 20A & B |
| 8 Bit to BCD (Binary Coded Decimal) Translator 2, 3, 4, 6, Bits. | 21A & B |
| BCD to 8 Bit Translator 0, 2, 3, 4, 6 Bits | 22A & B |
| Bus In P | 23A & B |
| Bus Out Parity | 24A & B |
| R-W VRC (Vertical Redundancy Check) | 25A & B |
| C Compare Data Check and Overrun | 26A & B |
| R-W Correction Gate Bits 7-4 | 27A |
| R-W Correction Gate Bits 3. P | 27B |
| CRCR (Cyclic Redundancy Check Register) Bits 7-4 | 28A |
| CRCR Bits 3-P | 28B |
| EPR (Error Pattern Register) Bits 7-3 | 29A |
| EPR Bits 2-P | 29B |

DESCRIPTIVE TABLE OF DRAWINGS—Continued

| Title description of drawing: | Figure number |
|---|---|
| LRC (Longitudinal Redundancy Check) Sample | 30 |
| GRC (Cyclic Redundancy Check) Control 1 | 31 |
| EPR Twist to Read Backward | 32 |
| GRC Character Latches | 33A |
| CRC Pattern Test | 33B |
| GRC Control Latches | 33C |
| CRC Control 2 | 33D |
| Error Triggers | 34A & B |
| Error Triggers | 35 |
| First Bit | 36 |
| Read Recognition | 37A & B |
| Hi Clip Pulse Freeze Registers 7, 6, 5, 4, 3 | 38A & B |
| Hi Clip Pulse Freeze Registers 2, 1, O P | 39 |
| Skew Register Position 7, 6 | 40A & B |
| Skew Register Position 5, 4 | 41A & B |
| Skew Register Position 3, 2 | 42A & B |
| Skew Register Position 1, 0 | 43A & B |
| Skew Register Position P | 44A & B |
| Hi Lo Compare | 45 |
| LRCR and Echo Errors | 46A & B |
| Hi Clip Register VRC | 47A & B |
| Final Amplifier, 7, 6, 5, 4 | 48 |
| Final Amplifier 3, 2, P | 49 |
| Final Amplifier 0, 1 | 50 |
| Final Amplifier Controls | 51 |
| Write Bus | 52 |
| Interface Sequence | 53A & B |
| Interface Sequence | 54A & B |
| End Pulse + Sense Gating | 55A & B |
| Sense Bit 0, 1, 2 | 56 |
| Sense Bit 3, 4, 5 | 57 |
| Sense Bit 6, 7 | 58 |
| Address Compare | 59 |
| Address Decoder and CU (Control Unit) End | 60A & B |
| TU (Tape Unit) 0-7 Select | 61A & B |
| TU 8-15 Select | 62A & B |
| Tags Out | 63 |
| Command Register 0, 1 | 64A & B |
| Command Register 2, 3, 4, Mode Sets | 65A & B |
| Command Register 5, 6, 7 and Command Decoding | 66A & B |
| Tape Operation Sync and Backward Disconnect | 67 |
| Tape Operation Lines | 68A & B |
| TU Lines Valid and Chain TR | 69A & B |
| Invalid Command Decoding | 70 |
| Unit Check | 71 |
| Shift Control | 72A & B |
| Shift Control | 73A & B |
| Byte Counter | 74A & B |
| Byte Counter and Byte Counter Check | 75A & B |
| Resets | 76 |
| Freeze Generation | 77 |
| Interface Terminators 0 1 2 3 4 | 78 |
| Interface Terminators 5 6 7 P | 79 |
| Interface Drivers | 80A & B |
| Interface Terminators | 81 |
| Freeze Gate Bus Out 0010203+4 | 82A & B |
| Freeze Gate Bus Out 5 6 7 P | 83A & B |
| Freeze Gate Tags Out | 84A & B |
| Go Backward LP | 85A & B |
| Select and Ready Status | 86 |
| Unit Freed Register 0 1 | 87A & B |
| Unit Freed Register 2 3 | 88A & B |
| Unit Freed Register 4 5 | 89A & B |
| Unit Freed Register 6 7 | 90A & B |
| Unit Freed Register 8 9 | 91A & B |
| Unit Freed Register 10 11 | 92A & B |
| Unit Freed Register 12 13 | 93A & B |
| Unit Freed Register 14 15 | 94A & B |
| Tape Unit Select Triggers | 95A & B |
| Busy | 96A, B, C & D |
| Tape Unit Select Register and Unit Freed Scanner | 97A, B, C & D |
| Error Stop | 99 |
| Control Lines to Tape | 98 |
| Control Lines from Tape | 100A & B |
| Control Lines from Tape | 101 |
| Control Lines from Tape | 102 |
| USEC (microsecond) Pulse | 103A & B |
| Read + Write Condition | 104 |
| WC and DC Drive | 105A & B |
| Write Clock | 106A & B |



DESCRIPTIVE TABLE OF DRAWINGS—Continued

| Title description of drawing: | Figure number |
|---|---|
| Write Clock VRC | 107 |
| Density Selection | 108A & B |
| Read Clock | 109A & B |
| Read Clock VRC | 110 |
| RC 7+RC Reset Gate | 111 |
| Delay Counter Controls | 112A & B |
| Delay Counter A B C Triggers | 113A & B |
| Delay Counter D E F Triggers | 114A & B |
| Delay Counter G H J Triggers | 115 |
| Delay Counter VRC | 116 |
| Delay Counter Gating | 117A & B |
| Delay Counter Gating D43+45+53 WDD 60 End WDD | 118 |
| Delay Counter Gating End RD, Dly D96, D168, D170, D196 and WD 320 | 119 |
| Delay Counter Gating RDD 144 | 120 |
| Manual Control Switches | 121 |
| Off Line Tag Out Generation | 122A & B |
| Instruction Ring and Character Counter | 123 |
| Manual Controls Command Bits P 0 1 2 3 | 124 |
| Manual Controls Command Bits 4 5 6 7 | 125 |
| Manual Controls Data Bits P 0 1 2 3 | 126 |
| Manual Controls Data Bits 4 5 6 7 | 127 |
| Manual Controls Tape Unit Address and Stop Jacks | 128A & B |
| Off Line Start Stop and Record Length Controls | 129 |
| Off Line Control Unit Address Jacks Stop on Error Jacks | 130A & B |
| Sequence Indicators A, B | 131A & B |
| Sequence Indicator C | 132 |
| Simulate Select | 133 |
| Simulate Register 7 6 5 4 3 2 P | 134 |
| Simulate Register 1 0 | 135 |
| Multiple Control Unit Connection to Computer Channel | 136 |
| Backward Read Operation Timing | 137 |
| Write Operation Timing | 138A & B |
| Read Operation Timing | 139A & B |
| Tape Control—System Relation | 140 |
| Interface Sequences | 141 |
| Delay Counter | 142 |
| Read Clock | 143 |
| Write Clock | 144 |
| Final Amplifiers | 145 |
| Typical Skew Register Position | 146 |
| Read-Write Register | 147A & B |
| Data Registers | 148 |
| Data Registers | 149 |
| Byte Counter | 150A & B |
| Eight Bit Mode Data Register Operation | 151 |
| Eight Bit to BCD Translator | 152 |
| BCD to Eight Bit Translator | 153 |
| Eight Bit Code-BCD Relations | 154A & B |
| C-Compare | 155 |
| C-Compare Circuits in a Specific Write Operation | 156 |
| LRCR Circuits | 157 |
| Date Flow in Write Operation | 158 |
| Simplified Tape Control Write Operation | 159A, B & C |
| Initial Selection Sequence | 160A, B & C |
| Tape Write Operation (1 of 9) | 161A & B |
| Tape Write Operation (2 of 9) | 162A, B & C |
| Tape Write Operation (3 of 9) | 163A & B |
| Tape Write Operation (4 of 9) | 164A & B |
| Tape Write Operation (5 of 9) | 165 |
| Tape Write Operation (6 of 9) | 166A & B |
| Tape Write Operation (7 of 9) | 167A & B |
| Tape Write Operation (8 of 9) | 168A & B |
| Tape Write Operation (9 of 9) | 169A & B |
| Data Flow in Read Operation | 170 |
| Simplified Tape Control Read Operation | 171A & B |
| Tape Read Operation (1 of 11) | 172A & B |
| Tape Read Operation (2 of 11) | 173A & B |
| Tape Read Operation (3 of 11) | 174A, B & C |
| Tape Read Operation (4 of 11) | 175A, B & C |
| Tape Read Operation (5 of 11) | 176A & B |
| Tape Read Operation (6 of 11) | 177A, B & C |
| Tape Read Operation (7 of 11) | 178A & B |
| Tape Read Operation (8 of 11) | 179A & B |
| Tape Read Operation (9 of 11) | 180A & B |
| Tape Read Operation (10 of 11) | 181A & B |
| Tape Read Operation (11 of 11) | 182A & B |
| Motion Control (1 of 7) | 183A, B & C |
| Motion Control (2 of 7) | 184A & B |
| Motion Control (3 of 7) | 185A, B & C |

DESCRIPTIVE TABLE OF DRAWINGS—Continued

| Title description of drawing: | Figure number |
|---|---|
| Motion Control (4 of 7) | 186A, B & C |
| Motion Control (5 of 7) | 187A & B |
| Motion Control (6 of 7) | 188A, B & C |
| Motion Control (7 of 7) | 189A, B & C |
| Burst Mode Sense Operation | 190A & B |
| Unit Data and Control Diagram | 191A & B |
| Cyclic Redundancy Check Register Block Diagram | 192 |
| Error Pattern Register Block Diagram | 193 |
| CRC Register Operation without Internal Feedback | 194 |
| Data Block Format with CRC Byte | 195 |
| Data Block Format with CRC Byte and LRC Byte | 196 |
| Operation of CRC Register in FIGURE 200 | 197 |
| CRCR Block Diagram with Multiple Internal Feedback Paths. | 198 |
| EPR Block Diagram with Multiple Internal Feedback Paths. | 199 |
| CRCR Block Diagram with One Internal Feedback Path. | 200 |
| Parity Checked Counter Circuit | 201 |
| Waveforms for a Parity Checked Counter | 202 |

An overall reference designation for each block (box) may be obtained by combining the figure number (without the word "Figure") with the two-letter identification found in each block (box). For example "21BT" is the reference designation for the upper left hand box on FIGURE 21A containing the letters "BT." This reduces verbosity otherwise needed on the drawings and in the description of the detailed embodiment. Each input lead coming from a drawing having a different figure number is labeled with the overall reference designation of the box on the drawing from which the lead comes. Each output lead going to a figure having a different number is labeled with the two reference letters of the box on the drawing from which the lead comes. In addition, each such output lead has a functional label, and also is labeled with each figure number to which the lead goes.

The term "byte" defined in the second paragraph of this application is often used interchangeably with the term "character." However, "byte" has a broader meaning than "character," since the latter is generally associated with a binary coding defining a particular symbol, such as a letter of the alphabet or a decimal digit. The term "byte"" includes a "character" but also includes groups of bits which are not intended to have any common symbolic meaning. For example, bit positions 9 through 16 of a binary number can be a "byte" but they are not a "character." Unfortunately prior usage of the term "character" has permitted its usage for bytes that have no symbolic meaning, such as a "check character." The latter usage will be continued herein because of its conventionality, but "byte" will be used herein in most other situations.

(I) CRC error detection and correction

FIGURE 1 shows the outline of an error location system. Data is received in a data input bus **2**. The block data is broken into sub-blocks of equal size. There is a sub-block check circuit **4** which gives an output when the sub-block is detected to be in error. Each sub-block is written with redundancy in such a way that if an error occurs, the check circuit will detect the fact. The error indication is sent to the error pattern residue circuit **5** called the EPR. The data itself is sent to the cyclic redundancy residue circuit **3** called the CRCR. After the total block of data has been read, a check character, to be called the cyclic redundancy check or CRC, enters on the data input **2**. This is checked by a sub-block check circuit and is sent to the cyclic redundancy residue circuit. After the data block and the check character have been received, the digital comparing circuit **6** which has inputs both from the cyclic redundancy residue circuit and from the error pattern residue circuit now can determine the sub-block error position. This is entered into the sub-block position error indicator **7**.

In the embodiment of FIGURE 2 described herein a sub-block is a byte of 9 bits. The sub-block check circuit is a parity checker **31**. The CRCR **18** and the EPR **30** blocks are shift registers with feedback connections. The data input can be considered to be the read-write register **12**. The sub-block check circuit is the VRC checker **31**. The cyclic redundancy residue circuit is the CRCR **18**. The error pattern residue circuit is the EPR **30**. The digital comparing circuit is the find track test **41**. This is used in a conjunction with controls which shift the CRCR during the sub-block position error time. The sub-block error indicator in this case is the EPR **30** which is shifted with the CRCR **18**.

Tape control operation with CRC error correction

The following paragraphs describe a method of doing error correction on a magnetic tape and refer to FIGURE 2. The method of writing, reading, and reading with correction is described. Then variations on these basic operations are described for the read backward operation. There are a number of ways of implementing this, several are mentioned with the descriptions.

The operation of writing is broken into two parts. The first part consists of writing data in bytes. The data is received in a tape control unit and transmitted to the tape in bytes. Each byte has a parity bit appended for error detection. Other methods of writing characters which have error detection redundancy such as having a predetermined number of 1's and 0's are also suitable. As each character is sent to the tape drive from R/W register **12**, it is also sent to a register **18** in FIGURE 2 called the CRCR (cyclic redundancy check register). Data is entered in this register in such a way as to Exclusive-OR with the present contents of the register and to shift one position. FIGURES 192 and 198 represent a simplified block form of a CRCR. FIGURE 192 shows a diagonal redundancy register which has end-around feedback but no intermediate feedback, and it receives and Exclusive OR's the bits in the diagonal order shown in FIGURE 194. In FIGURES 198, 199, and 200, there also are intermediate feedback connections on the register to modify the contents during the shift operation to receive and Exclusive OR's the bits in a cyclic order, such as shown in FIGURE 197.

CRCR **18** is shown in detail in FIGURES 28A and B.

When the last byte of a data block is received in the tape control unit, it is sent to the tape and also to the CRCR. Then the CRCR is given one extra shift. The contents of the CRCR are then transferred to a read-write register **12** by use of AND gate **54**. This is done in such a way that certain bit positions of the CRCR are complemented. The contents of the read-write register are then written onto tape as a CRC check byte after the end of the data block (see FIGURES 194 or 195) at a time known as WDD **60**. A check character known as the LRC is written on the tape after the CRC byte in FIGURE 195 at time WDD **120**.

The LRC has the following property: It insures that there are an even number of ones written on tape in every track. For example, the 4 bit of the LRC character is chosen so that throughout the block there will be exactly an even number of ones in the four track. The LRC is calculated over the total block written and the CRC byte which was written. The purpose of complementing certain bits of the CRC as it was transferred to the read-write register is to insure odd redundancy for the LRC byte. This property is valuable during the read backward operation. The way this works is explained in a special section on the operation of the shift register.

During the write operation there is a read-while-write feature. Data is read on a tape head into the tape control and the block is checked for proper VRC (byte redundancy) and LRC. The LRC is calculated in **40** from data on line **14**. At the end of the block write operation if a VRC error was detected or if a LRC error was detected by circuitry **45, 46, 47** an error signal is sent to the computer signaling that the write operation was not done properly. In this manner, a number of information bytes, a CRC check character byte, and an LRC check character byte have now been written onto tape, as shown in FIGURE 195.

The operation of reading is done as follows. The read command is given and the tape starts to move. The data read from tape enters the tape control unit from the tape unit and is transferred to read-write register **12** in FIGURE 2 on line **4**. From the read-write register **12**, it is transferred through Exclusive OR **16** to the CRCR **18** in the same way as it was done during write. A VRC (Vertical Redundancy Check) is calculated on the data by VRC circuit **31**. If the data is calculated to be in error, a one is inserted into another register called the error pattern register or EPR **30** by use of AND **37**. The error pattern register (EPR) **30** is similar to CRCR **18** except that it has only one input for the read-write VRC error. See FIGURES 191A and B and 192 for simplified block representations of a CRCR and an EPR. The EPR is shifted at the same time the CRCR is. The shifting of the EPR is done by condition OR **26**. This feeds AND **38** which gives a pulse to shift the EPR. The CRCR is shifted by the output of OR **26** feeding OR **25** to AND **22** which gives a pulse out to shift the CRCR.

If there is a VRC error from circuit **31**, the AND gate **43** is sampled to send an error signal to the CPU.

Each character or byte of data read from tape actuates a read clock and is transferred under control of read clock timing pulses. At the end of the read operation, the read clock is reset and a delay counter called RDD is turned on. The RDD counter is reset and restarted each time a byte of data is read. If the counter called RDD is not reset by a following data byte, it reaches count RDD **36**, and a latch called first check character is turned on. If this latch is turned on, it is assumed that the next byte to be read will be the CRC character. When the CRC character is read into register **12**, it is sent to the CRC register as the data was. However, the redundancy (or VRC) of this character may not be the same as for the data. Hence, the VRC check is modified to take this possibility into account. This is done by the line check character correction. The CRC character is not sent to the CPU (computer). A short time later, the delay counter sequence provides a pulse at RDD **96**, and a second check character latch is sent. It is assumed the next character to be received will be the LRCR character. This character is not sent to the CRC register. Like the CRC it is not sent to the CPU. It is sent to the LRCR **40**. This is the end of the normal read operation. At this time the LRC register is sampled for error by circuits **45, 46, 47**, the CRC register is sampled by circuits **51** and **52** to determine if an error occurred, and the read-write VRC latch is sampled to determine if there was a read-write VRC error during the reading of this block. If any of these errors occurred, an error signal is sent to the CPU by means of a computer channel.

If an error did occur, the CRCR **18** and the EPR **30** will contain a pattern of ones and zeros related to the pattern of errors that occurred. Due to the special nature of these registers it is possible to use their contents to determine the track in which the error occurred. The normal mode of operating is to backspace over the block just read and to reread it, correcting those characters which have a VRC error by complementing the bit in those characters corresponding to the track which was determined to have the error. The time at which the track in error is determined could be anytime between the completion of the first block read operation, and the receiving of data during the next read operation.

In FIGURE 2 the calculation of track in error is done before the beginning of the second data read operation during the time in which the tape was picking up speed to again read the block. A variation on this technique would be to do the track calculation at the end of the first read operation and to store the track calculated to be in error. If the track in error were stored in the CPU the error detection track location circuitry could be used on other tape drives and/or other records on the same tape drive before correction was attempted on the original record in error. When it was desired to correct the original record in error, the track calculated to be in error could be transmitted back to the tape adapter along with status information which would set the tape adapter into an error correcting mode. The storage of the track in error could also have been done in the tape control in a special memory. The CPU would have special instructions to retrieve the track in error information when the record in error was reread. The track in error calculation may also be done during the backspace or during a special cycle between operations.

In the embodiment of FIGURE 2 track in error information is discarded if the correction cycle is not attempted immediately. That is, the sequence of operation must be—the record is read and error detected, a backspace operation is done and then a read operation is done on which the error correction is attempted. The latter is done through the use of several control latches. When a backspace operation is done after a block read with an error, a special latch **41** (called find track latch) is turned on. The contents of the EPR and the CRCR are retained. At the beginning of the next block read (which will be the read for error correction), read-write register **12** is reset, the contents of the EPR is transferred to the read-write register through use of gate **55**, and a one is inserted into the top position of the EPR register by means of AND gate **37**. At this time the CRCR and the EPR registers are shifted through use of AND circuit **34** into OR **26** into AND **38** to shift the EPR and OR **26** to OR **25** to AND **22** to shift the CRCR. After each shift the contents of the CRCR are compared bit position for bit position with the contents of the read-write register. The comparison is done by entering the CRCR **18** contents into the Exclusive OR gate **16** through use of AND gate **34** and OR **33**. The output of the Exclusive OR's **16** go to the find track test circuit **41** which gives an output if the CRCR and the contents of the read-write register **12** match bit for bit. If there is an exact match, a latch called the correcting latch (within block **41**) is turned on and the shifting is stopped. When the find track test **41** turns on, the inverter **42** will give a 0 output. This will degate AND **39** and the EPR and CRC will get no more shift pulses. Then the contents of the EPR will indicate the track in error by means of the position of the single one bit in the EPR. The CRC register is reset.

At this time a rereading of the tape data block can begin. Data is now entered into the read-write register under the control of the read clock just as in the normal read operation. However, if a read-write VRC error is found for a byte of data, the bit corresponding to the track which the EPR says is in error will be complemented. This is done as follows. The VRC **31** energizes AND **32** which gates the EPR **30** to the OR **33** to the correction Exclusive OR's **16**. Since the EPR has only one 1 in a position corresponding to the track in error the output of the Exclusive OR's **16** will be the same as the input except for the track in which the EPR has a 1. In this track the read-write register bit will be complemented. The byte as corrected is sent to the CPU **15** and to the CRC register **18**. Nevertheless, the input to the LRC register is the uncorrected data.

When all the data of the tape block has been read, the CRC character is also read. Its VRC may be different from that of the data. However, it is predictable. The CRC is sent to the CRC register but not to the CPU. Then the LRC character is read into the LRCR but not the CRCR. This is the end of the read with correction cycle. Now it is time to sample for error. The CRC register is tested by circuit **52** to be sure that it has the pattern corresponding to no error. The LRC register is sampled to see that it is all zeros except possibly in the position in which error correction took place. This is done by blocking the sampling of the track calculated to be in error (as held in the EPR **30**) using the ERROR track blocking gate **45**. This is sampled by AND **46** and zero tested by circuit **47**. If either of these indicate error, an error signal is sent to the CPU.

There are several exceptions to the general mode of operation. One exception is when no track is calculated to be in error. In this case at the beginning of the read, 8 shifts take place without a match between the CRCR and the contents of the read-write register which at that time is the previous contents of the EPR. Another possibility is that the CRCR register is either all zeros or a special setting. In both of these settings when the CRCR is shifted, its setting is identical to its previous setting. That is, if all zeros are shifted it will be all zeros or if the other special setting is shifted it will be identical to itself before shifting. No track calculation is attempted in these situations. For the case where no track in error is calculated, the CRC and the EPR are reset and the read proceeds as it does during a normal read, that is, the VRC errors are entered into the EPR and the data is entered into the CRC register. No correction, of course, being attempted.

It should be noted that the tape as written with these two check characters could be read on a tape control which was not fitted for error correction. During the check character reading time, it would read both check characters into its LRC register but ignore the information in the CRC character. If the time alloted for entry of check characters is specified to be long enough for the CRC and the LRC to enter the adapter there will be a compatibility between tapes written with and without CRC check character.

There is a tape read operation known as read backward or sometimes backward read. In this operation the tape is read in the opposite direction to that which it was written. In this case, the check characters are read first and then the data which comes in the reverse order of that which it was written. The tape error correction scheme just outlined will work about the same for read backward as it does for read. There are several special situations which must be considered, however.

First, since the data is going into the CRCR and shifted in a certain direction during the normal read and write, the data shifting must be modified during the backward read. This can be done by shifting the CRCR in the opposite of its normal shifting direction during backward read. Another way to accomplish the same result is to twist the data lines going into the CRCR so that those which normally went into the top go into the bottom and those which normally went into the bottom go into the top. A very important factor is that the CRCR itself must by symmetric, that is, feedback connections are symmetrically arranged around the center position of the register.

The input lines to the shift register are shown with circuitry for twisting in FIGURES 28A and B. When this is done, it is also necessary to twist the lines at the output of EPR for correction. This is on FIGURE 32. The reason for this is that when the EPR is matched with the CRC, the number of shifts necessary to get a match relates to the tracks in their twisted manner rather than the way they go to the CPU. Hence, in order to correct the proper track it is necessary to twist the output of EPR lines the same as the inputs to the CRC were twisted.

Another important factor in read backward is obtaining the check character properly. Since the LRC character is not sent into the CRC register, it is necessary to know which of the characters received is the LRC and which is the CRC. If only one check character is received, this is difficult to do. Hence, provision is made to guarantee the LRC character is never all zeros. This is done by changing the VRC of the CRC character during the write operation. This has the effect of causing the LRC character to have odd redundancy, thus the first character read during the read backward will be the LRC character. It is not entered into the CRC register. All other characters read will be read into the CRC register. During a read backward operation it is not possible to predict the VRC of the CRC character. Hence, either this must be given by the CPU or through memory in the tape control or this error signal must not be sampled. The latter course has been taken in this embodiment. The backward read is identical to the forward read except for these special considerations. Since the CRC character is not sampled for error, in read backward it is not possible to correct errors which include the CRC character. If the VRC of the CRC had been known, it would be possible to correct errors in the CRC character.

After a read backward operation in which errors are detected, a forward space operation is done. This operation will not be a read, it will be simply a forward space. At the beginning of the next backward read operation, the track in error is calculated. Error correction is attempted on the data read.

A number of variations on this general operating procedure is possible. One would be: when an error is found during read backward, the correction is automatically done during the next forward read. Another would be: to provide several kinds of instructions such as, a normal read command and a read with correction command. In this case a CPU would give a normal read command until an error is detected. Then the track in error would be located. The next time, the CPU would give a read with correction command; and data in the block would be corrected in the track calculated.

The shift register used in a described embodiment is one of a class of possible circuits which could do the same operation. The important requirements in these circuits are (1) they must be linear, that is, if a set of data called A is entered into the circuit, it will have a content after the entry we shall call F(A). If a set of data B is entered into the circuit it will at the end of the entry have a content we will call F(B). Now, if A and B, the data, are added together bit by bit Modulo 2 and are entered into the circuit, the contents of the circuit will be called $F(A+B)$. The circuit will be linear if $F(A)+F(B)$ where the $+$ means bit by bit addition modulo two is equal to $F(A+B)$. Second requirement: the circuit must cycle in such a way that for at least nine shifts (or the number of shifts equal to the number of tracks) the contents of the circuit will be different for every shift. Third requirement: in order to get a match when comparing the contents of the EPR to the contents of the CRC by shifting, it must be that if a pattern is entered into one input of the circuit contents will be the same as if the same pattern had been entered into another input position of the circuit and several extra shifts had been taken. Fourth requirement: for read backward, it is desirable that the VRC of the check character be predictable. Fifth requirement: for read backward, it is desirable that the circuit be symmetric.

One class of circuits which satisfies all these requirements are linear feedback shift registers. Typically one describes the shift registers by use of a polynomial. Where the polynomial has a one for a coefficient the shift register will have a feedback connection. If the polynomial cannot be factored into a product of smaller polynomials, there will only be one pattern, namely all zeros which will equal itself when shifted once. If a polynomial has $1+X$ as a factor, the VRC of the check character can be calculated. If the polynomial is self-reciprocal, that is, if it is equal to itself when the coefficients are switched, the highest to the lowest, then the shift register will be symmetric. A list of irreducible polynomials can be found in the book by Peterson, Error Correcting Codes. For the implementation illustrated here one of those polynomials of degree 8 was multiplied by $1+X$. The polynomial chosen was self-reciprocal. Naturally other polynomials could have been chosen. Though a particular type of shift register is shown here, a number of other possibilities exist including the use of delay lines, and the use of special sequential circuits which satisfy all the requirements listed.

In the detailed drawings it will be found that the read-write VRC error is entered into the bottom of the EPR register. This is equivalent to entering the pattern of errors only into the bottom of the EPR register. The data and the pattern of errors are entered into the CRC register, the data into all positions, and the pattern of errors into only the position equal to the track in error. This assumes only one track in error. To calculate the track in error the CRC is shifted to match its contents with the contents of the EPR. If, for example, the track in error is the second from the bottom, in one shift the contents of the EPR and the CRC will be identical.

If the EPR is shifted to try and match the CRC contents, the pattern of errors from the VRC should be entered into the top of the EPR. In that case the number of shifts to get a match would have a different correlation to the actual track in error.

If the polynomial describing the CRCR shift register is factorable into a factor of $1+X$ times another polynomial, there is a pattern of ones and zeros which will be itself when shifted. The pattern is 111010111 for the shift register of FIGURE 198. This pattern is important for two reasons. One, it must be recognized when trying to match the EPR with the CRC in order to not match erroneously on the first track. Second, this pattern can be used to change the VRC of the CRC character. This is done by writing the CRC character as calculated except complementing those positions in which this special character has ones. Since the number of ones is odd, this changes the VRC of the CRC character (byte). When this character is read back, normally the CRC register would go to all zeros. But since the check character is complemented in these particular positions, the CRC register will, in the case of reading without error, end up with this pattern of ones and zeros in it. This makes it possible to write the CRC character in the redundancy (rdncy.) which we desire. To calculate the VRC of the CRC character, refer to the following table:

| Rdncy. of Block | No. of bytes in block | VRC of CRC | VRC of LRC | VRC of CRC Complemented in selected positions | VRC of new LRC |
|---|---|---|---|---|---|
| Even | Even | Even | Even | Odd | Odd. |
| Do | Odd | do | do | Odd | Odd. |
| Odd | Even | do | do | Odd | Odd. |
| Odd | Odd | Odd | do | Even | Odd. |

In order for this system to work, it is necessary to have one shift on reading for every character written. If for some reason a character is completely dropped out, that is just not read at all, no shift will take place. The shifting will get out of sequence. In this case correction is impossible. Therefore, provision must be made to avoid the situation. One possibility is to write all characters with even redundancy and to not allow the all zeros character. Another solution is to automatically give a shift if it appears that a character has been lost. There are several ways to do this. One is to have a special shift instruction if the RDD counter, that is the counter which turns on at the end of every read clock cycle, goes beyond a certain point, say 16, and the read clock starts again. This situation would happen if more than a full character cycle time had elapsed without having had a character. That is, a special shift is inserted where a character was probably missing. A similar technique is described and claimed in U.S. patent application, Ser. No. 234,151 filed Oct. 30, 1962, entitled "First Bit Generation for Binary Systems."

## (II) Mathematical Description of the Cyclic Correction Process

In this section, familiarity with the polynomial description of cyclic codes is assumed as presented in "Cyclic Codes for Error Detection" by W. W. Peterson and D. T. Brown, Proc. of IRE, pp. 228–235, January 1961. The book Error Correcting Codes by W. W. Peterson, Wiley 1961 gives a more complete exposition. The algebraic operations of addition and multiplication are *NOT* the ones of high school algebra but rather those of a Galois field.

Consider each track in a record to contain a separate message. As an example, represent the message in the 3 track of an $m$ character block by the polynomial

$$M_3(X)=p_1X^{m-1}+p_2X^{m-2}+\ \ldots\ +p_{m-1}X+p_m$$

The $p_i$ are the bits in the message. $p_1$ is the first bit written on tape and the first bit to be read. $p_i$ can be one or zero.

If $M_3(X)$ is entered, high order first, into the CRCR, and all other CRCR inputs are zero, the final CRCR content will be the remainder on dividing $M_3(X)$ by G(X).

The form of G(X) is determined by the feedback connections in the CRCR. In the cases considered here

$$G(X)=1+X^7$$

which is shown on FIGURE 192

$$G(X)=1+X^3+X^4+X^5+X^6+X^9$$

which is shown on FIGURE 198.

Entry into the bit position one stage below the bit 3 input does the same thing except that each bit is advanced one shift time, which is equivalent to multiplying the input polynomial by X. This suggests that a data record be represented, for purposes of calculating the CRC, by

$$M(X)=M_1(X)+XM_2(X)+X^2M_3(X) + \ldots +X^8M_9(X)$$

this assumes the track numbering is as shown in FIGURE 198. Because of the linear nature of the shift register, the final contents of the CRCR after the record has been entered is

M(X) modulo G(X)

One additional shift of the CRCR gives

$$C(X)=XM(X) \text{ modulo } G(X) \quad (1)$$

which is the CRC character.

On reading, the data characters are entered and then the CRC character is entered on an additional shift so the result is

$$XM(X)+XM(X)=0 \text{ modulo } G(X)$$

A record with errors in one track is represented by $M(X)+X^iE(X)$ where $i$ gives the track in error, ($i=0$ for track 1 $i=1$ for track 2 etc.), and E(X) is a polynomial with a term for each bit in error. The final CRCR content on reading an erroneous record is

$$X^iE(X) \text{ modulo } G(X)$$

Because a one is entered in the last ($X^8$) position of the EPR whenever an error occurs, and because the EPR also reduces modulo G(X) the final result in the EPR on reading an erroneous message is

$$X^8E(X) \text{ modulo } G(X)$$

An additional $j$ shifts of the CRCR give

$$X^{i+j}E(X) \text{ modulo } G(X)$$

The track can be located by shifting the CRCR and comparing with the EPR if

$$X^{i+j}E(X) \neq X^8E(X) \text{ modulo } G(X) \quad (2)$$

for

$$i+j \angle 8$$

and

$$X^{i+j}E(X)=X^8E(X) \text{ modulo } G(X) \quad (3)$$

for

$$i+j=8$$

then we know the track in error by counting the number of shifts it takes to reach an exact match between the CRCR and the EPR. The count of shifts is $j$ in (2) and (3). For example, if 2 shifts are needed $j=2$ $i=8-2=6$ $i=6$ means the error is in track 7.

If no match is obtained after 9 shifts, the error is uncorrectable. Errors can, of course, occur in the CRC character. The correct parity (VRC) of the CRC character must be known to detect this error to enter a one in the EPR to calculate E(X).

To determine the correct CRC character parity, it is necessary to make $(1+X)$ a factor of G(X).

Rewrite Equation 1 as

$$C(X)+G(X)H(X)=XM(X) \text{ modulo } G(X)$$

and if $G(X)=P(X)(1+X)$ then G(X) H(X) has an even number of terms (i.e., even parity) [1]. It follows that, if G(X) has a $(1+X)$ factor, then the parity of C(X) equals the parity of M(X) because the sum of two even polynomials (polynomials with an even number of terms) is an even polynomial and the sum of an even and an odd polynomial is an odd polynomial.

As the parity of M(X) is just the parity of all the bits in the data, it can be calculated as follows:

(a) If the character parity (VRC) is even, the parity of M(X) will always be even.

(b) If the character parity is odd, the parity of M(X) equals the parity of the count of data characters in the message.

Thus, a modulo 2 character counter permits prediction of the CRC character parity.

Because G(X) must have the form $(1+X)P(X)$ to correct errors in the CRC character, more uncorrectable errors will result than if G(X) had no $(1+X)$ factor. It follows from (2) that a track cannot be located if

$$E(X)=0 \text{ modulo } P(X) \quad (4)$$

When an error is read, the final content of the CRCR is

$$E(X) \text{ modulo } G(X)$$

and if the error is uncorrectable, this final content will be

$$E(X)=P(X)K(X) \text{ modulo } G(X)$$

but in general the following relation is true

$$X^iP(X)=P(X) \text{ modulo } G(X)$$

So,

$$P(X)K(X)=0 \text{ modulo } G(X)$$

if K(X) has an even number of terms and

$$P(X)K(X)=P(X) \text{ modulo } G(X)$$

if K(X) has an odd number of terms.

To summarize: The two conditions when an error is uncorrectable are (1) when the final CRCR content is all zeros or (2) P(X) when $G(X)=(1+X)P(X)$.

The choice of the polynomial P(X) is important. The necessary conditions on G(X) are Equations 2 and 3. If $G(X)=P(X)$ or $(1+X)P(X)$.

P(X) irreducible is sufficient to insure the conditions are satisfied.

In terms of sequential machine theory, the CRCR and EPR must be multiple input linear machines with no cycles with periods less than 9 (unless such cycles are detected as all zeros and P(X) are for

$$G(X)=(1+X)P(X))$$

The inputs must have a linear relation between them. For example, if there is a binary input sequence A, let XA represent A shifted in time one bit position. Let the machine inputs be numbered 1, 2 . . . , $t$. If A is put into input 1, call the state of the machine $F_1(A)$. The relation among inputs must be

$$F_1(A)=F_2(XA)=F_3(X^2A)= \ldots o=F_t(X^{t-1}A) \quad (5)$$

the condition (2) and (3) will be fulfilled if

$$F_1(A) \neq F_1(XA) \neq F_1(X^2A) \neq F_1(X^{t-1}A)$$

and (5) hold and the machine is linear.

A number of shift register implementations will have these properties. Such devices as delay lines could be used in the embodiment. It should be noted that even though shift registers having a number of tracks equal to the number of bistable positions have been shown, this is not

a necessary condition. The shift register could have more or fewer bistable positions if its fulfills the condition listed.

CRC correction during read backward

In order to calculate the CRC on a read backward, it is necessary that the message be evenly divisible by the generator polynomial when it is received in reverse.

To investigate this situation let us write the message, as read in the forward direction, as a polynomial

$$M(X)=C_nX^0+C_{n-1}+ \ldots C_0X^n$$

Where

$$C_n=CRC=C_{n1}X^0+C_{n2}X^1+ \ldots C_{n9}X^8$$

and

$$C_0=1\text{st character}=C_{01}X^0+C_{02}X^1+ \ldots +C_{09}X^8$$

If the message is received in reverse it will be

$$M^R(X)=C_nX^n+C_{n-1}X^{n-1}+ \ldots C_0X^0$$

now if we reverse the order of the bits in the characters like this

$$C^+{}_n=C_{n1}X^8+C_{n2}X+ \ldots C_{n9}X^0$$

$$C^+{}_0=C_{01}X^8+ \ldots C_{09}X^0$$

then we can write

$$M^+(X)=C_n{}^+X^n+C_{n-1}{}^+X^{n-1}+ \ldots C_0{}^+X^0$$

(Note: The + means all the coefficients of the polynomial have reversed order.)

Now there is a theorem which says

Theorem: If $G(X)=G^+(X)$=the generator

polynomial for a code, then if M(X) is evenly divisible by G(X), $M^+(X)$ is also evenly divisible. (Peterson—page 161).

So in order to use the CRC in read backward, two things must be done

(1) Choose a generator polynomial for which

$$G(X)=G^+(X)$$

(2) Reverse the order of the bits entering the CRC register.

Now there is one more problem to consider. How do we calculate the track in error on a read backward operation?

The track location routine is exactly the same as for a forward read with one exception. The exception is that the number of shifts it takes to get an exact match of the EPR with the CRC will indicate a different track in error for forward location and backward location. This is because the bits in the CRC have been reversed in the backward case. Hence, the number of shifts to get a match will refer to the errors as they entered the CRCR not to the track in error as with the read forward condition. The table below gives the specific details

| No. of shifts to get a match | Track in error on read forward | Track in error on read backward |
|---|---|---|
| 0 | 9 | 1 |
| 1 | 8 | 2 |
| 2 | 7 | 3 |
| 3 | 6 | 4 |
| 4 | 5 | 5 |
| 5 | 4 | 6 |
| 6 | 3 | 7 |
| 7 | 2 | 8 |
| 8 | 1 | 9 |

(III) Embodiment Introduction

The Tape Control described herein is an intermediate unit connecting a plurality (such as eight) of Magnetic Tape Units to the Input-Output (I/O)=Channel Interface of a Digital Computer: (FIGURE 140). The interface is a plurality of transmission lines with precisely defined uses for communicating with and signaling I/O Control Units. The Tape Control controls all tape operations of the attached tape units and processes data transferred between the tape unit and the channel interface. The tape control executes read, write, sense, motion control, and tests I/O status for the interface.

The Tape Control consists of an interface adapter section and a tape adapter control section. The interface adapter has the following purposes:

(1) to convert interface signal sequences and coded commands into a form necessary to select and operate a tape unit.
(2) to convert tape adapter section timings to interface line sequences.
(3) to regulate data transfers between the interface and the tape adapter control section.
(4) to check parity of bytes either to be transmitted to or received from the interface and to force check bits when necessary to make the parity of all output data bytes to the interface odd.

The tape adapter control section has the following objects:

(1) to use line combinations from the interface adapter to supply required timings and controls necessary to operate a selected tape unit.
(2) to check all data that a selected tape unit writes.
(3) to provide signals that the interface adapter requires to communicate with the interface.
(4) to check the parity of bytes either to be transmitted to or received from the tape unit and to force check bits when necessary to make the parity of bytes to the selected tape unit have the parity that the tape command specifies.
(5) to dynamically correct one or more errors which may have been read from any tape track.

The described embodiment of the Tape Control can operate with either 9-track Tape Drives or 7-track Tape Drives. The 9-track Drive provides in parallel 8 information bits and one parity bit. The 7-track Drive provides in parallel 6 information bits and one parity bit.

The described embodiment of the Tape Control can process one 8-bit byte at a time. It can assemble bits from a 9-track tape unit into an 8-bit byte and transmit the byte to the channel interface, or it can accept an 8-bit byte from the channel interface and process the 8-bit byte directly to the selected tape unit.

When operating with a 7-track Tape Drive, a byte converter within the Tape Control can process the 6-information bits per tape byte into or from eight-bit bytes transmitted to or received from the channel interface. When using the byte converter, the tape control converts three 8-bit bytes to four 6-bit bytes or converts four 6-bit bytes to three 8-bit bytes per byte conversion cycle.

Also with a 7-track Tape Drive, the byte converter can be bypassed to (1) transmit 6-bit bytes directly to the interface, or (2) code translate a 6-bit byte from BCD code to 8-bit code. The code converter changes each 6-bit character to or from tape with a one-for-one correspondence into an 8-bit byte to or from the channel interface. Also with a 7-track Tape Drive, a direct transmission to tape can be provided for the six low-order bits in each 8-bit byte from the channel interface, in which the two high-order bits are discarded. Or, a direct transmission of each 6-bit byte from tape can be provided by adding two high order zeros to each BCD character received from the selected tape unit and transmitted to the interface (six-bit mode).

The basic tape unit used herein can basically be similar to the IBM 729 Tape Units which have been commercially used for a number of years. Nine track operation is obtained basically by substituting a nine track head assembly for a seven track head assembly. For 9-track operation, data rates are preferably changed by changing tape speed and having the same tape density and format used with each tape unit.

(A) I/O-channel interface

(1) *General interface line descriptions.*—Referring to FIGURE 136, interface lines are broadly classified as *out* or *in*. *Out* lines are directed from the channel to the attached control units. *In* lines are directed from the attached control units to the channel. *In* lines and *out* lines are connected to the attached control units in a parallel fashion with the exception of a *select-out* and *select-in* line sequence. This particular combination of lines requires special connection in view of the function performed.

(2) *Specific interface line descriptions.*—Interface lines are specifically classified according to the function they perform. The specific classifications follow.

Buses: *Bus-out and bus-in* lines are sets of nine lines (8 information and 1 parity) over which all addresses, commands, status information and bytes of data are transmitted between the attached control units and the channel. If the channel is addressing a particular control unit, it places the address of the control unit on the *bus-out* lines and signals all control units to decode the address. If the channel is transmitting command information or data to a selected control unit, the command information or the data is placed on the *bus-out* lines and the addressed control unit signalled accordingly. Conversely, if a control unit addresses the channel, it places its address on the *bus-in* lines and signals the channel to decode the address. If a control unit is transmitting status information or data to the channel, the status information or the data is placed on the *bus-in* lines and the channel signalled accordingly.

Tags: Tag lines are lines over which signals relating to bus information are transmitted. An *address-out* tag signals all control units that the channel has placed an address on the *bus-out* lines and that all units should attempt to decode the address. A *command-out* tag has the primary function of signaling a selected control unit that the channel has placed a command byte on the *bus-out* lines. It is furthermore used to respond to certain *in* tag lines. A *service-out tag* indicates to the attached control units that the channel has accepted information on *bus-in* lines or has provided previously requested data on *bus-out* lines. An *address-in* tag is used to signal the channel when a control unit has placed its address on the *bus-in* lines. Ordinarily the channel will respond to the *address-in* tag with the *command-out* tag. A *status-in* tag is used to signal the channel that a selected control unit has placed a byte of status information on the *bus-in* lines. The status byte generally describes the current status of the control unit. The channel will respond to the *status-in* tag with the *service-out* tag or the *command-out* tag. A *service-in* tag is used to signal the channel when the control unit has either placed a byte of data on the *bus-in* lines or is requesting a byte of data to be placed on the *bus-out* lines. The channel will respond to the *service-in* tag with either the *service-out* tag or the *command-out* tag.

Scan controls: *Select-out* is used by the channel to select an addressed control unit during the selection sequence. Selection begins when the channel places the address of a control unit on the *bus-out* lines and signals a decode operation by activating the *address-out* tag. Approximately 250 nanoseconds after the issuance of the *address-out* tag *select-out* is initiated and applied serially to all control units beginning with the control unit having the highest order of priority and ending with the unit having the lowest order of priority. The priority of units is determined by the rate at which they can transmit or receive data from the channel. Accordingly, a high speed magnetic tape control unit or magnetic disk file unit, for example, has lower priority than, for instance, a card control unit or a printer control unit. This allows the slower units an equal opportunity to be selected since selection is effected serially between control units. When a control unit receives *select-out* via a receiving device, TR, and if the unit for some reason cannot accept selection, it propagates the *select-out* signal via a driver TD, to the next lower priority control unit. This continues until the addressed control unit receives the *select-out* signal. The fact that the control unit had been addressed and its address is decoded allows it to inhibit further propagation of the *select-out* signal. Failure of the control units to inhibit propagation of the *select-out* signal results in its propagation from the lowest priority unit back to the channel, as *select-in*. *Select-in*, therefore, is the return path for the *select-out* signal and provides indication to the channel that none of the attached control units has responded to its *select-out* signal. The channel, depending upon its particular operational mode, to be described later, will respond to *select-in* in one of two ways. Either it will immediately initiate a new *select-out* or it will hold *select-out* until one of the attached control units specifically request service.

Interlocks: *Operational-out* is applied to all control units and is a necessary prerequisite to any operation. *Operational-out* is active over all operational sequences. If it should at any time become inactive for a period of 6 microseconds, all attached control units will reset. *Operational-in* is issued by a selected control unit to indicate to the channel that the control unit has accepted selection by inhibiting further propagation of *select-out*.

Special controls: This particular functional classification of interface lines includes all those lines required for data transfer and control which do not properly fit under any of the foregoing classifications. *Suppress-out* is a signal from the channel to all attached control units. *Suppress-out* is primarily the means whereby the channel maintains connection to a particular control unit. In the so called *byte* mode of operation, to be described later, the channel will alternately accept information or transmit information to several tape units in an interleaved fashion. If the channel desires to remain connected to a particular control unit in the *byte mode*, it can activate *suppress-out*. Thereafter any other control unit wishing to address the channel can attempt to interrupt but the existence of *suppress-out* will cause an interlock within the control unit that inhibits any further attempts at channel interruption until *suppress-out* is deactivated. Thereafter the various control units are again free to address the channel. *Hold-out* provides a means to accelerate the selection process. Consider the sequence where a *select-out* signal, having failed to select a unit, is propagated as a *select-in* signal to the channel, with a channel response initiating a new *select-out*. To avoid an erroneous selection, the new *select-out* must be inhibited until the decay of *select-in*. By way of example, if the delay of *select-out* through a control unit is 2.5 microseconds, and if 8 control units are attached to the channel, then 20 microseconds are required for the complete propagation of *select-out* to initiate *select-in*. Assuming the *select-in* signal to be received a negligible time after propagation from the last or lowest priority control unit, a 20 microsecond decaying period is necessary before the next *select-out* can be issued. *Hold-out* allows this period to be substantially reduced. *Hold-out* is the *select-out* signal applied to the attached control units in a parallel connection. In each control unit *hold-out* is applied to one input of an AND circuit with *select-out* being applied as the other input. Thus the *select-out* signal is gated through each unit by the *hold-out* signal. If selection fails to occur, *select-in* is initiated and its leading edge is used to reset the bistable device in the channel from which *select-out* and *hold-out* were derived. Since *hold-out* is applied in parallel to all control units and gates *select-out*, its deactivation immediately limits the delay of the decaying *select-in* signal to the delay through the lowest priority unit. Accordingly, the maximum decay time of the *select-in* signal can be no greater than the delay of *select-out* through the last propagating control unit. Since a 2.5

microsecond delay per control unit has been assumed, *select-in* will decay in 2.5 microseconds. As a result, *select-out* is reactivated after a 2.5 microsecond delay following reception of *select-in*. *Metering-out and-in* are special lines allowing for the registration of the timed use of a control unit and do not relate directly to the control or transfer of information over the interface lines. *Request-in* is a special line through which any attached control unit can specifically request service from the channel.

(3) *Operational modes.*—Two channel-I/O modes of operation are possible. The interface lines permit the channel to operate with the attached control units in either a *byte mode* or a *burst mode*. In the *byte mode* the channel communicates with several I/O control units in an interleaved fashion. In the *byte mode* each of the control units will have been selected through an initial selection sequence and placed on a standby or stacked status. In the *byte mode* the channel issues a *select-out* which is accepted by the highest priority selected control unit to allow, for instance, a byte of data to be transmitted on one of the buses. Each such transfer must be preceded by a new *select-out*. In the *byte mode* a control unit can accept only every other *select-out*. In this manner a high priority control unit cannot entirely dominate the selection process, thereby allowing lower priority units, at least on every other *select-out*, the opportunity to communicate with the channel. In the *burst mode*, the channel maintains communication with a single control unit until all required operations have been completed. After execution of an operation begins in the *burst mode*, the control unit need not identify itself prior to communicating with the channel. *Burst mode* is the priority mode. The channel designates *burst mode* by holding *select-out* active throughout an operation. In response, the control unit maintains *operational-in* active over the entire operation. A control unit can demand a *burst mode* operation by initiating an internal *force burst mode* which holds *operational-in* active even if the channel has deactivated *select-out*. This is a priority condition. If *force burst mode* is inactive, the channel can change from a *burst mode* to a *byte mode* after the operation begins in the *burst mode*.

(4) *Initial selection sequence.*—All communications between the channel and a control unit preparatory to the initiation of data transfer is referred to as an initial selection sequence. It is during the initial selection sequence that a particular control unit is addressed, selected, and commanded via a command byte to prepare to perform a particular operation. The initial selection sequence begins at the channel when the channel places the address of the control unit it wishes to select on the *bus-out* lines. The *bus-out* lines enter the Tape Control Unit (TCU) on FIGURE 5 via entrance blocks AD, AC, AB, AA, AH, AG, AF, AE, and AM to initiate *bus-out* lines **0** through **7** and P, respectively to FIGURES 78 and 79. *Bus-out* P is the partity bit bus. *Bus-out* lines **0** through **4** are applied to interface line terminators on FIGURE 78. *Bus-out* lines **5** through **7** and P are applied to interface line terminators on FIGURE 79. From the line terminators —*bus-out* lines **0** through **7** and P are initiated to FIGURES 5, 82A and B, and 83A and B. On FIGURE 5 —*bus-out* lines **0** through **7** and P exit the tape control unit via the indicated exit blocks to enter the next lower order priority control unit. On FIGURES 82A and B, and 83A and B —*bus-out* lines **0** through **7** and P are applied to freeze latches. The *bus-out* lines are applied to freeze latches to allow them to come under control of a freeze condition. A freeze condition is a TCU control allowing all significant conditions in the unit to be held static pending manual intervention. The ability of a freeze condition to effect a static condition is effected through a freeze latch for each significant condition. Freeze latches will, therefore, be encountered frequently throughout the following description. Freeze latches have been defined previously in the section on Drawing Element Definitions. Accordingly, a freeze latch is identified by the inverter ordinarily providing one output level and feedback to the latch itself. Where additional powering, due to output circuit loading, is required, as is the case with the freeze latches on FIGURES 82A and B, and 83A and B, the output may be taken through an additional inverter. This represents a slight departure from the normal definition but in no way changes the logical continuity of the output of the freeze latch. By way of example, —*bus-out* **0** on FIGURE 82B applied to the freeze latch designated by inverter AK initiates +*bus-out* **0** through the additional inverter BB whereas +*bus-out* **0**, were it not for output circuit loading, could logically have been obtained from inverter AK. *Bus-out* lines from FIGURES 82A and B, and 83A and B are wired to such general functional areas of the TCU as the data register contained on FIGURES 12, 13, 14, 15, 16, and 17, the command register on FIGURES 64A and B, and 66A and B, the address decoder on FIGURES 60A and B, and the tape unit select register on FIGURES 95A and B. The *address-out* tag from the channel is received via entrance block AK on FIGURE 5 to initiate —*address-out* to FIGURE 81. On FIGURE 81 —*address-out* is applied to line terminator AF. The negative output of line terminator AF is returned to FIGURE 5 to exit block BB for transmission to the next lower order priority control unit. The positive output of line terminator AF on FIGURE 81 completely conditions AND circuit AV to initiate —*address-out* to FIGURES 84A and B. On FIGURES 84A and B —*address-out* is placed under control of the freeze condition via the freeze latch designated by inverter AV to initiate +*address-out* to FIGURES 54A and 60A. +*address-out* on FIGURES 54A and B is applied to AND circuit AM to condition the reset input to the address in trigger AK, AN. +*address-out* on FIGURE 60A is applied to AND circuit BE. If the address on *bus-out* lines **0–4** has been successfully decoded by decode switches AB, AC, AD, AE, and AF and if the parity of the address on the *bus-out* lines is odd, AND circuit BE will be fully conditioned to initiate —*control unit addressed* to FIGURES 54A and B. The output of AND circuit BE is furthermore inverted via inverter BD to fully condition AND circuit AM and initiate —*adapter address decoded* to FIGURES 54A and B, 69A and 96A and C. —*adapter address decoded* on FIGURES 54A and B is applied to set the address in trigger AK, AN. —*adapter address decoded* on FIGURES 96A and C sets the initial selection trigger AU, AY and the tape unit select trigger AG, AM. The output of AND circuit AM applied to inverter AN is applied to one input of AND circuit AK. The reset condition of the stack interrupt trigger on FIGURES 54A and B produces a positive condition in —*stack interrupt trigger* to fully condition AND circuit AK to initiate —*set tape unit address* to FIGURE 69A and via inverter AL +*set tape unit address* to FIGURES 95A and B. On FIGURES 95A and B +*set tape unit address* conditions AND circuits AU, AP, AH and AB to allow the tape unit address on *bus-out* lines **4, 5, 6, 7** to be set into tape unit select triggers **4, 5, 6, 7**, respectively. The set outputs of the TU select triggers initiate +*tape unit select trigger* **4, 5, 6**, or **7** to FIGURES 61A and 62A and B. On FIGURES 61A and B, and 62A and B the tape unit address is decoded to effect tape unit selection. Returning to FIGURES 69A and B, —*set tape unit address* causes +*TU lines valid* to fall for a 10 microsecond period. The fall of +*TU lines valid* on FIGURE 76 via OR circuit AD and inverter AE initiates —*start reset* to reset the read-write register on FIGURES 18A and B, FIGURES 19A and B, and 20A and B, the byte counter on FIGURES 74A and B, the data register on FIGURES 12, 13, 14, 15, 16, and 17, the data parity check trigger on FIGURES 26A and B, the load point trigger on FIGURES 85A and B and set the 1st character trigger on FIGURES 37A and B. Further, on FIGURE 69A —*adapter address decoded* applies a reset hold to the chain trigger AS, AW.

250 nanoseconds after the *address-out* tag was initiated *select-out* and *hold-out* are activated. On FIGURE 5 —*select-out* and —*hold-out* enter the TCU via entrance blocks AQ and BH respectively to initiate —*select-out* and —*hold-out* to FIGURE 81. On FIGURE 81 —*select-out* and —*hold-out* are applied to line terminators AJ and BK respectively. The outputs of line terminators AJ and BK are applied in DOT AND connection BL to effect the previously described gate of *select-out* by *hold-out*. +*select-out* is initiated by DOT AND connection BL to FIGURES 54A and B. On FIGURES 54A and B +*select-out* conditions one input of AND circuit BG. If the control unit switches on FIGURES 60A and B had not decoded the address on *bus-out*, —*control unit addressed* would not have been initiated and the positive condition of —*control unit addressed* on FIGURES 54A and B would have fully conditioned AND circuit BG to initiate —*select-in* to FIGURE 80A. On FIGURE 80A, —*select-in* applied to OR driver AP would initiate —*select-out* to FIGURE 5 where, applied to exit block BE, it would be propagated to the next lower priority control unit. Since we have assumed a successful address decode on FIGURES 60A and B, AND circuit BG on FIGURE 54B is not conditioned and the TCU inhibits further propagation of *select-out*. Returning to FIGURE 81 the output of DOT AND connection BL is furthermore applied to previously conditioned AND circuit AY which through OR circuit BA initiates +*select-out* to FIGURE 84A. On FIGURE 84A +*select-out* is applied to the freeze latch designated by inverter AS to initiate +*select-out* to FIGURES 60A and B, and 63. On FIGURES 60A and B +*select-out* is applied as one input to AND circuit BF. If the TCU is already performing a command, then upon the rise of +*select-out* AND circuit BF previously conditioned by the set output of the performing command trigger AH, AL will be fully conditioned to initiate —*control unit busy* to FIGURES 80A and B. On FIGURES 80A and B, —*control unit busy* conditions OR driver AA, OR driver AC, and OR driver AM to initiate —*bus-in* **3**, **1** and P. Further, on FIGURES 80A and B OR driver AK initiates —*status-in* to FIGURE 4. On FIGURE 4 —*bus-in* **1** applied to exit block AB, —*bus-in* **3** applied to exit block AF, —*bus-in* P applied to exit block AK, and —*status-in* applied to exit block AM exit the TCU to the channel to indicate that the TCU is busy. Returning to FIGURE 60B, —*control unit busy* from AND circuit BF also sets control unit end trigger BJ, BK to condition one input to AND circuit BL preparatory to the initiation of —*control unit end status*. Upon the set of the channel end trigger AL, AP on FIGURE 54A, —*control unit end* is initiated to FIGURE 9 for transmission to the channel via FIGURES 80A and B and FIGURE 4 on —*bus-in* **2**. On FIGURE 63, +*select-out* from FIGURES 84A and B is applied to one input of previously conditioned AND circuit AT. The output of AND circuit AT inverted and powered by inverter AX activates +*select-out* to FIGURES 53A and B. On FIGURE 53A +*select-out* applied to previously conditioned AND circuit AQ sets the operational in trigger AR, AT to initiate +*operational-in trigger* to FIGURES 63, 54A and B, and 80A. On FIGURE 80A +*operational-in trigger* applied to previously conditioned AND circuit BD initiates, via OR driver AN, —*operational-in* to FIGURE 4. On FIGURE 4 —*operational-in* is applied to exit block AN to initiate *operational-in* to the channel. On FIGURE 63 +*operational-in trigger* conditions AND circuits AA and AJ whose other inputs are +*service-out* and +*command-out*, respectively. On FIGURES 54A and B +*operational-in* trigger is applied as one input to AND circuit AG. The other inputs to AND circuit AG are the set output of the *address-in* trigger and —*address-out* from FIGURE 84B. The channel will respond to *operational-in* by deactivating the *address-out* tag thereby establishing a positive condition on the —*address-out* line from FIGURE 84B to fully condition AND circuit AG and initiate —*address-in* to FIGURE 23A. On FIGURES 23A and B —*address-in* via inverter BC, DOT OR connection BH and inverter BB initiates a parity bit on +*bus-in* P to FIGURES 80A and B. On FIGURE 80B AND circuit BC and OR driver AM initiate —*bus-in* P to FIGURE 4 for exit to the channel. Returning to FIGURE 54B, the output of AND circuit AG is further applied to inverters AQ and BV to initiate +*address-in* to FIGURES 8, 9, 10 and 11. On FIGURE 8 +*address-in* is applied to AND circuits AA and AV to gate address switches **0** and **1** from FIGURE 60B respectively. On FIGURE 9 +*address-in* is applied to AND circuits AA and AP to gate address switches **2** and **3** from FIGURE 60B. The address switches AP, AQ, AR, AS, AT on FIGURE 60B are prewired switches to initiate the address of the tape control unit to the *bus-in* lines **0**, **1**, **2** and **3**, whenever they are gated as upon the issuance of the *address-in* tag. On FIGURE 10 +*address-in* conditions AND circuits AA and AQ to gate the outputs of tape unit select triggers **4** and **5** on FIGURES 96A, B, C and D to *bus-in* lines **4** and **5**, respectively. On FIGURE 11 +*address-in* conditions AND circuits AA and AV to gate tape unit select triggers **6** and **7** on FIGURES 95A and B to *bus-in* lines **6** and **7**, respectively. In this manner, the address of the tape control units and the previously selected tape unit are placed on the *bus-in* lines. The *bus-in* lines from FIGURES 8, 9, 10 and 11 are wired to FIGURES 80A and B where after gating by —*off line*, they condition OR drivers which drive them to the indicated exit blocks on FIGURE 4. Returning to FIGURES 54A and B, the output of AND circuit AG, in addition to initiating —*address-in* is applied to a delay circuit comprising inverters AT, AU and capacitor AW. The delay circuit defined by these blocks has been previously described in the Drawing Element Definition section. Its purpose is to delay the initiation of the *address-in* tag to the channel in order to allow time for transient conditions on the *bus-in* lines to disappear. This is commonly referred to as a skewed condition resulting from the fact that the circuits initiating the bits of the address byte are not entirely synchronous. Thus if the initiation of the *address-in* tag is delayed slightly, a more reliable transmission of the address byte is insured. Ordinarily the generation of any tag such as *status-in* or *service-in*, will be initiated through such a delay for the same reason. Thus delayed on FIGURE 54B, —*address-in* is applied to the freeze latch designated by inverter BF to initiate +*address-in dly* to FIGURE 80A. On FIGURES 80A and B +*address-in dly* through previously conditioned AND circuit BB via OR driver AL initiates —*address-in* to FIGURE 4. On FIGURE 4 —*address-in* is transmitted to the channel via exit block AJ. The channel responds to the *address-in* tag by activating the *command-out* tag. —*command-out* enters the TCU via entrance block AK to initiate —*command-out* to FIGURE 81. On FIGURE 81 —*command-out* is applied to line terminator AA. The negative output of line terminator AA initiates —*command-out* to FIGURE 5 where it exits the TCU via exit block BC for transmission to the next lower priority control unit. The positive output of line terminator AA through previously conditioned AND circuit AP initiates —*command-out* to FIGURES 84A and B. On FIGURES 84A and B —*command-out* is applied to the freeze latch designated by inverter AK. +*command-out* from the output of inverter AK is initiated to FIGURE 63. On FIGURE 63 +*command-out* is applied to AND circuit AJ previously conditioned by +*operational-in* trigger. The output of AND circuit AJ is applied to the delay circuit comprising inverters AK, BA and capacitor AG. Thus delayed +*command-out* is applied to the freeze latch designated by inverter BJ to initiate —*command-out delayed* to FIGURES 53A and B. The output of AND circuit AJ on FIGURE 63 is furthermore applied to inverter AH to initiate +*command-out* to FIGURES 53A and B, 64A, and 96C. On FIGURE 64A +*command-out* is applied as one input to AND circuit

AB previously conditioned by the set output of the address-in trigger on FIGURE 54B and the set output of the initial selection trigger on FIGURES 96B, C and D to initiate *—set command register* to FIGURE 55A. The output of AND circuit AB applied to inverter AC to initiate *+set command register* to FIGURES 65A, and 66A. In addition, the output of inverter AC is applied to AND circuits AE and AK to gate *+bus-out* lines **0** and **1** into command register positions **0** and **1** at locations AF, AH and AL, AQ, respectively. On FIGURE 65A *+set command register* is applied as one input to AND circuits AB, AL, AV to gate *+bus-out* lines **2**, **3** and **4** into command register positions **2**, **3** and **4** located at AC, AG; BC, AQ; and AW, BA respectively. On FIGURE 66A *+set command register* is applied as one input to AND circuits AA, AH and AR to gate *+bus-out lines* **5**, **6** and **7** into command register positions **5**, **6** and **7** at locations AB, AE; AJ, AN; AS, AP respectively. Upon initiation of the *command-out* tag to the TCU the channel will have placed a command byte on the *bus-out* line. Thus the command byte as described is entered into the command register for subsequent decoding in accordance with the command code. The following is a table of the command control code.

COMMAND, CONTROL CODE

[1. Command Format]

| Command | Command Byte (Bit Positions) 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Test I/O | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Sense | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Write | 0 | 0 | 0 | 0 | 0 | [1]Ø | 0 | 1 |
| Read | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Control | 0 | 0 | [2]C | [2]C | [2]C | 1 | 1 | 1 |
| Read Backward | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| No Operation (No P) | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

[2. Control Format]

| | Bit Positions 2 | 3 | 4 |
|---|---|---|---|
| Rewind (REW) equals | 0 | 0 | 0 |
| Rewind and Unload (RUN) equals | 0 | 0 | 1 |
| Erase Gap (ERG) equals | 0 | 1 | 0 |
| Write Tape Mark (WTM) equals | 0 | 1 | 1 |
| Backspace Record (BSR) equals | 1 | 0 | 0 |
| Backspace File (BSF) equals | 1 | 0 | 1 |
| Forward Space Record (FSR) equals | 1 | 1 | 0 |
| Forward Space File (FSF) equals | 1 | 1 | 1 |

[1] Ø=Dont's Care.
[2] C=Control format bit positions.

On FIGURE 96C *+command-out* from FIGURE 63 to previously conditioned AND circuit BC resets the not test IO trigger AX, BD. On FIGURE 53A *—command-out delayed* from FIGURE 63 conditions OR circuit AC. The output of OR circuit AC fully conditions AND circuit AD which has been partially conditioned by the set output of the address in trigger on FIGURE 54A. The output of AND circuit AD sets the first command trigger AC, AF and is further applied to one input of AND circuit AG. AND circuit AG is further conditioned by the positive condition of *—end pulse,* by the set output of the first command trigger AC, AF, by the positive conditions of *—service-out delayed* and *—service-in* trigger, and by *—command-out delayed.* AND circuit AG will not be fully conditioned until the fall of the *command-out* tag to initiate a positive condition on the *—command-out* line. The reset output of the first command trigger initiates *—first command trigger* to FIGURES 54A and B. On FIGURES 54 A and B *—first command trigger* resets the address in trigger AK, AN. The resulting negative condition of the set output of the address in trigger deconditions AND circuit AG. The resulting positive condition at the output of AND circuit AG applied to the delay circuit comprising inverters AT, AU and capacitor AW is delayed and applied to the freeze latch designated by inverter BF. This initiates a negative condition at the output of the inverter BF to establish a negative condition on *+address-in dly* to FIGURE 80A. On FIGURE 80 the negative condition of *+address-in dly* deconditions AND circuit BB. The resulting positive condition at the output of AND circuit BB through OR driver AL initiates a positive condition on *—address-in* to FIGURE 4. On FIGURE 4 the positive condition of *—address-in* exits the TCU via exit block AJ to deactivate the *address-in* tag to the channel. The channel responds to the deactivation of the *address-in tag* by deactivating the *command-out* tag. The resulting positive condition of *—command-out* on FIGURES 53A and B fully conditions AND circuit AG to set the status in trigger AH, AM. The reset output of the status in trigger AH, AM is applied to a delay circuit comprising inverters AZ, AW and capacitor AX. The delayed reset of the status in trigger AH, AM is applied to the freeze latch designated by inverter BH to initiate *+status-in delay* to FIGURE 80A. The reset output of the status in trigger AH, AM is also applied to inverter BD to initiate *+status-in trigger* to FIGURES 9, 10 and 11. On FIGURES 9, 10 and 11 the status bits are gated to the *bus-in* lines. On FIGURE 9 *+status-in trigger* conditions AND circuit AQ to gate *+busy status* via OR circuit AM to *bus-in* line **3**. On FIGURE 10 *+status-in trigger* conditions AND circuit AS to gate selected unit freed via OR circuits AR and AM to *bus-in* **5** line. It also conditions AND circuit AT to gate channel end trigger via OR circuit AU, DOT OR connection AZ, OR circuit AV to *+bus-in* **4** line. On FIGURE 11 *+status-in trigger* conditions AND circuit AD to gate unit check via OR circuits AC and AE to *+bus-in* **6** line. It also conditions AND circuit AS to gate unit exception via OR circuits AU and AQ to *+bus-in* **7** line. The status bits assigned to the various *bus-in* positions are listed in the following table. Control unit end has previously been described. It should be noted that the bus position identifies the bus on which the single one bit in the eight bit status byte appears.

Status Format

| Status: | Bus position |
|---|---|
| Not used | 0 |
| Status Modifier | 1 |
| Control Unit End | 2 |
| Busy | 3 |
| Channel End | 4 |
| Unit Freed | 5 |
| Unit Check | 6 |
| Unit Exception | 7 |

The status modifier bit in combination with a busy bit in an initial selection sequence, to be described later, indicates to the channel that the addressed tape unit is utilizing the TCU to perform an operation. The busy bit without a status modifier bit indicates that the addressed tape unit is performing an operation independent of the TCU (such as rewind) and that the TCU is available for the selection of another tape drive. On FIGURE 80A *+status-in dly* via previously conditioned AND circuit BA and OR driver AK initiates *—status-in* to FIGURE 4. On FIGURE 4 *—status-in* exits the TCU via exit block AM to activate the *status-in* tag to the channel. The channel decodes the status byte and, depending upon the information contained therein or upon the ability of the channel to continue communication with the TCU, the channel will respond to the status-in tag by initiating either the *command-out* or *service-out tag.* The activation of the *command-out* tag by the channel has the effect of putting the TCU in stacked status which is most generally a holding action. *Command-out* from the channel enters the TCU in the same manner as described previously when the *command-out* tag was initiated by the channel in response to the *address-in* tag. Accordingly, on FIGURES 53A and B *+command-out* from FIGURE 63 fully con-

ditions AND circuit AL to initiate —*status-in and command-out* to FIGURE 54A. On FIGURE 54A —*status-in and command-out* sets the stack interrupt trigger AA, AH. On FIGURE 53A —*command-out* deconditions AND circuit BF to remove the set hold input to the first command trigger AC, AF. It also deconditions AND circuit AG to remove the set hold input to the status in trigger AH, AM. The now positive condition at the output of AND circuit AG is applied to condition AND circuit AE that has been previously conditioned by the set output of the status in trigger AH, AM. —*command-out delayed* from FIGURE 63 via OR circuit AZ fully conditions AND circuit AE to reset the first command trigger AC, AF. The resulting negative condition on the set output of the first command trigger AC, AF is applied to reset the status in trigger AH, AM. The reset of the status in trigger AH, AM is applied to the delay circuit comprising inverters AV, AW and capacitor AX. The delayed reset is then applied to the freeze latch designated by inverter BH to initiate a negative condition on +*status-in dly* to FIGURE 80A. On FIGURE 80A the negative condition of +*status-in dly* deconditions AND circuit BA and via OR driver AK initiates a positive condition on —*status-in* to FIGURE 4. On FIGURE 4 the positive condition of —*status-in* from FIGURES 80A and B is applied to exit block AM to deactivate the *status-in* tag to the channel. As previously described, AND circuit AL on FIGURES 53A and B was fully conditioned by a +*command-out* to initiate —*status-in and command-out* to FIGURE 76. On FIGURE 76 —*status-in and command-out* via OR circuit AG and inverter AH initiates —*end reset* to FIGURES 64A and B. On FIGURES 64A and B —*end reset* via OR circuit AR and inverter AS resets the command register position **0** and **1**. The output of inverter AS furthermore initiates —*reset command register* to FIGURES 65A, and 66A. On FIGURE 65A command register positions **2**, **3** and **4** are reset by —*reset command register*. On FIGURE 66A command register positions **5**, **6** and **7** are reset by —*reset command register*. On FIGURES 53A and B the reset of the status in trigger AH, AM and the first command trigger AC, AF, the positive condition on —*address-in trigger* from FIGURE 54B, the positive condition on —*service-in trigger* from FIGURES 72A and B, and the deconditioned state of AND circuit AU are applied to condition AND circuit AS. At this time, —*select-out* from FIGURE 63 is in a positive condition from the prior receipt of the *operational-in tag* from the TCU thereby fully conditioning AND circuit AS to reset the operational in trigger AR, AT. This substantially restores the TCU to its condition prior to the initial selection sequence with the exception that the stack interrupt trigger AA, AH on FIGURE 54A is now set. One possible reason for the activation of *command-out* in response to the *status-in* tag is that the channel may be operating in a *burst mode* with some other control unit, in which case the combination of the set condition of the stack interrupt trigger AA, AH on FIGURE 54A and +*suppress-out* will condition AND circuit AB. The resulting negative condition at the output of AND circuit AB applied to one input of AND circuit AE deconditions AND circuit AE and holds it thus until the channel deactivates +*suppress-out*. The positive condition at the output of AND circuit AE fully conditions AND circuit AF. The negative condition at the output of AND circuit AF holds the address in trigger AK, AN in a reset condition. Accordingly, the TCU cannot under any circumstances initiate the *address-in* tag to interrupt the channel until after the channel has deactivated +*suppress-out*. Returning now to the point in the initial selection sequence where, in response to the receipt of the *status-in* tag, the channel could raise either the *service-out* or *command-out* tag, consider now that the *service-out* tag has been activated. —*service-out* enters the TCU via entrance block AJ on FIGURE 5 to initiate —*service-out* to FIGURE 81. On FIGURE 81 —*service-out* is applied to line terminator AB. The negative output of line terminator AB is returned to FIGURE 5 to be applied to exit block BD for transmission to the next lower order priority control unit. The positive output via previously conditioned AND circuit AS and DOT OR connection AT initiates —*service-out* to FIGURE 84A. On FIGURE 84A, —*service-out* is applied to the freeze latch designated by inverter AG. The output of inverter AG initiates +*service-out* to FIGURE 63. On FIGURE 63 +*service-out* is applied to AND circuit AA previously conditioned by the set output of the operational in trigger on FIGURES 53A and B. The output of AND circuit AA is applied to the delay circuit comprising inverters AB, AZ and capacitor AE. The delayed +*service-out* is then applied to the freeze latch designated by inverter BE to initiate —*service-out delayed* to FIGURE 53A. On FIGURE 53A, —*service-out delayed* via OR circuit AZ, as previously described for the activation of —*command-out delayed* will reset the first command trigger AC, AF and the status in trigger AH, AM. The reset outputs of these triggers, in combination with +*select-out* and the positive condition on —*address-in trigger* from FIGURE 54B and —*service-in trigger* from FIGURES 72A and B and the deconditioned state of AND circuit AU will fully condition AND circuit AS to reset the operational in trigger AR, AT. In response to the fall of the *operational-in* tag, the channel will deactivate the *service-out* tag. This completes the initial selection sequence. From this point on, the sequence will be entirely dependent upon the command previously issued on the activation of the *command-out* tag in response to the *address-in* tag and upon the operational mode, whether *burst* or *byte*. Hereinafter, execution of the command shall be referred to as the command sequence.

(5) *Interrupt sequence.*—Before proceeding to execute the command operation, an interrupt sequence will be described since it is for the most part equivalent to an initial selection sequence. An interrupt sequence is initiated whenever channel interruption by the TCU is required. After conditions requiring an interrupt occur, the interrupt sequence proceeds from the rise of the *address-in* tag and enters the initial selection sequence as if it were an initial selection sequence. Interrupt sequences are initiated during a read operation in the *byte mode* to transfer data, or to transfer unit free indications, or to transfer previously stacked status information to the channel. Whenever an interrupt sequence is indicated OR circuit AD on FIGURE 54A is conditioned. This partially conditions AND circuit AE. Since an interrupt is in progress, the operational in trigger on FIGURES 53A and B is reset. It is assumed that AND circuit AB is deconditioned by the stack interrupt trigger AH, AL via OR circuit AA. It is further assumed that no prior interrupt attempt has occurred which accounts for the reset condition of the stack interrupt trigger AH, AL. The *address-out* tag must be inactive for the control unit to interrupt since activation of the *address-out* tag indicates that the channel is trying to address some control unit. The *address-out* tag will take priority over any interrupt. A positive condition on —*select-out* is required to condition AND circuit AE. Since the *select-out* tag follows the *address-out* tag by 250 nanoseconds —*select-out* is positive for that period of time. Accordingly, AND circuit AE is fully conditioned and sets the address in trigger AK, AN and deconditions AND circuit AF, thereby trapping the *select-out* tag 250 nanoseconds later to effect selection by inhibiting propagation of the *select-out* tag to the next lower order priority control unit. The address in trigger AK, AN partially conditions AND circuit AQ on FIGURES 53A and B. When the *select-out* tag is activated, +*select-out* fully conditions AND circuit AQ to set the operational in trigger AC, AT to initiate +*operational-in trigger* via inverter BE on FIGURES 54A and B. On FIGURES 54A and B +*operational-in trigger* fully conditions AND circuit AG to initiate —*address-in* which through the same route described previously in the initial selection sequence, activates the *address-in* tag to

the channel. From this point on the interrupt sequence proceeds exactly as the initial selection sequence. If the interrupt were initiated for the purpose of transmitting status information to the channel, the *status-in* tag will occur exactly as previously described in the initial selection sequence. If the interrupt were initiated to transfer data, the *status-in* tag will not be activated. Instead, the *service-in* tag will be activated. If the interrupt were initiated to transfer data the service in trigger AF, AJ on FIGURE 72A would be set. —*Service-in* trigger on FIGURES 53A and B deconditions AND circuit at AG thereby preventing the status in trigger AH, AM from being set. The set output of the service in trigger AF, AJ on FIGURE 72A fully conditions AND circuit AA. The output of AND circuit AA is applied to the delay circuit comprising inverters AB, AC and capacitor AH. The delayed output of AND circuit AA applied to the freeze latch designated by inverter AD initiates +*service-in dly* to FIGURE 80A. On FIGURE 80A +*service-in dly* via previously conditioned AND circuit AZ and OR driver AJ initiates —*service-in* to FIGURE 4. On FIGURE 4 —*service-in* exits the TCU via exit block AL to initiate the *service-in* tag to the channel. The *service-in* tag indicates that a byte of data is available on the *bus-in* lines. In response to the *service-in* tag, the channel will activate the *service-out* tag. The TCU responds to the *service-out* tag by deactivating the *service-in* tag.

(6) *Command sequence.* — Hereafter information transfer will ensue in either a *byte mode* or a *burst mode*. *Burst mode* is the priority mode, that is, if the channel or control unit is forcing *burst mode*, neither unit can require an operation in *byte mode*. If neither unit is forcing *burst mode*, then either unit can operate in *byte mode* or either unit can arbitrarily switch to a *burst mode* depending upon conditions. For the TCU the necessary conditions can be defined as the conditioning inputs to AND circuit AC on FIGURE 55A. These are the set output of the sense service trigger AB, AD on FIGURE 55A, the reset output of the first command trigger on FIGURE 53A, an inactive *suppress-out* line, and positive conditions on —*service-out delayed* and —*command-out*. These conditions allow —*force burst-mode* to fully condition AND circuit AC to set the service in trigger on FIGURE 72A. Accordingly, an operation beginning initially as a *byte mode* can be changed to a *burst mode*. It is assumed that a *burst mode* is being forced by the TCU. It is furthermore assumed that in the initial selection the command byte indicated a sense operation. When the sense command byte on the *bus-out* lines was received, as described in the initial selection sequence, it was set into the command register on FIGURES 64A and B, 65A and B and 66A and B. On FIGURE 64B —*set command register* was initiated to FIGURE 55A. On FIGURE 55A —*set command register* sets the sense service trigger AB, AD. The command decoding circuitry on FIGURES 66A and B decoded the sense command to initiate +*sense command* to FIGURE 55A. On FIGURE 55A +*sense command* provides a hold input to the sense service trigger AB, AD so that at the end of the initial selection for a sense command, the sense service trigger AB, AD remains on. During the initial selection sequence, when the *command-out* tag in response to the *address-in* tag was received, —*command-out* deconditioned one input to AND circuit AX on FIGURE 55A. On the subsequent deactivation of the *command-out* tag, the positive condition on —*command-out* in combination with +*sense command* conditions AND circuit AX to apply a negative condition at the complement input of the sense byte counter 1 (SBC1) trigger AT. When the *service-out* tag is activated in response to the *status-in* tag, AND circuit AX is deconditioned. The resulting positive condition at the output of AND circuit AX, applied to the complement input of the sense byte counter 1 trigger AT, sets the sense byte counter 1 trigger AT. The set output of trigger AT via AND circuit AK and inverter AL initiates +*sense byte* 1 to FIGURES 56, 57 and 58. On FIGURE 56, for example, +*sense byte* 1 gates +*invalid command trigger* through AND circuit AA which via OR circuit AC and inverter AE initiates +*sense bit* 0 to FIGURE 8. On FIGURE 8 +*sense bit* 0 via AND circuit AB, conditioned by +*service-in*, OR circuit AW, the DOT OR connection AF, and inverter AE initiates +*bus-in* 0 to FIGURES 80A and B, and 23A. On FIGURES 80A and B +*bus-in* 0 via AND circuit AU and OR driver AD initiates —*bus-in* 0 to FIGURE 4 for transmission to the channel. On FIGURE 23A +*bus-in* 0 with similarly generated bits 1 through 7 are applied to a redundancy circuit that calculates the parity of *bus-in* bits 0 through 7 to initiate +*bus-in* P to make the entire bus parity odd. The sense byte counter steps once at each activation of the *service-out* tag for the remainder of the sense operation. Altogether five sense bytes are transmitted. These are contained in the following table.

Sense byte format

| | Byte 1 | | Byte 2 |
|---|---|---|---|
| 0 | Command Reject | 0 | Spare |
| 1 | Intervention Required | 1 | TU Status A |
| 2 | Bus Out Check | 2 | TU Status B |
| 3 | Equipment Check | 3 | 7-Track |
| 4 | Data Check | 4 | At Load Point |
| 5 | Overrun | 5 | Write Status |
| 6 | Word Count Zero | 6 | File Protected |
| 7 | Byte Converter Check | 7 | Tape Indicator |

| | Byte 3 |
|---|---|
| 0 | R/W Vertical Redundancy Check |
| 1 | Longitudinal Redundancy Check Reg. (LRCR) |
| 2 | Skew |
| 3 | Compare |
| 4 | Hi Clip Vertical Redundancy Check |
| 5 | Spare |
| 6 | Spare |
| 7 | C Compare |

| | Byte 4 | | Byte 5 |
|---|---|---|---|
| 0 | Echo Error | 0 | LRCR Reg. 0 |
| 1 | Reject Tape Unit | 1 | LRCR Reg. 1 |
| 2 | Read Clock Error | 2 | LRCR Reg. 2 |
| 3 | Write Clock Error | 3 | LRCR Reg. 3 |
| 4 | Delay Counter Error | 4 | LRCR Reg. 4 |
| 5 | Sequence Indicator C | 5 | LRCR Reg. 5 |
| 6 | Sequence Indicator B | 6 | LRCR Reg. 6 |
| 7 | Sequence Indicator A | 7 | LRCR Reg. 7 |

Note: 0 through 7 next to each byte indicate *bus-in* line on which the indicator bit appears.

Whenever AND circuit AC on FIGURE 55A becomes conditioned it initiates —*sense service-in* to FIGURE 72A. On FIGURE 72*A* —*sense service-in* sets the service in trigger AF, AJ. AND circuit AC is conditioned by the set output of the sense service trigger AB, AD previously set by +*sense command*, by the positive condition of —*first command trigger* from the reset output of the first command trigger on FIGURES 53A and B previously reset by the *service-out* tag in response to the *status-in* tag, and by the inactive condition of *suppress-out*. *Suppress-out* is employed to prevent the *service-in* tag from being initiated whenever *suppress-out* is active. A *suppress-out* during data transfer is the suppress data condition. The suppress data condition indicates to a control unit that is operating to stop transmitting data to the channel provided that the control unit can do so without overrunning. Overrunning refers to a condition where the control unit, if forced to stop, would still have data requiring transfer. —*Service-out delayed* delays the conditioning of AND circuit AC to prevent a *service-in* tag from being activated until the previous *service-out* tag is completely deactivated. In addition AND circuit AC is deconditioned by

the *command-out* tag to prevent the *service-in* tag from being activated while the *command-out* tag is active. AND circuit AC is further conditioned by AND circuit AA to allow sense bytes to be transferred during sense operations in the *byte mode* only on alternate rises of the *select-out* tag. If *—force burst mode* were active during the operation, AND circuit AC would be fully conditioned when the *service-out* tag in response to the *status-in* tag of the initial selection sequence was deactivated. AND circuit AC then initiates *—sense service-in* to FIGURE 72A. On FIGURE 72A *—sense service-in* sets the service in trigger AF, AJ. The set output of the service in trigger AF, AJ conditions AND circuit AA. The output of AND circuit AA is applied to the delay circuit comprising inverters AB, AC and capacitor AH. The delayed set of the service in trigger AF, AJ is applied to the freeze latch designated by inverter AD to initiate *+service-in delay*. *+Service-in dly* via conditioned AND circuit AZ and OR driver AJ initiates *—service-in* to FIGURE 4 to exit the TCU. The channel will respond to the *service-in* tag with the *service-out* tag. The service-out tag resets the service in trigger on FIGURES 72A and B to step the sense byte counter on FIGURES 55A and B. The sense service trigger AB, AD will remain reset until either a *command-out* tag is issued by the channel in response to the *service-in* tag or the sense byte counter steps to seven. If a *command-out* tag responds to the *service-in* tag, AND circuit AR on FIGURE 54A will be conditioned. The output of AND circuit AR will set the stop trigger AS, AV. The reset output of the stop trigger AS, AV via DOT OR connection AY initiates *—stop trigger* to FIGURE 55A. On FIGURE 55A *—stop trigger* resets the sense service trigger AB, AD. If the sense byte counter steps to 7, AND circuit AP on FIGURES 55A and B is conditioned. The output of AND circuit AP resets the sense service trigger AB, AD. With the sense service trigger reset, AND circuit AC on FIGURE 55A is deconditioned. The positive condition at its output imposes a positive condition on *—sense service-in* to FIGURE 72A. On FIGURE 72A the same positive condition holds the service in trigger AF, AJ reset. The reset output of the sense service trigger AB, AD on FIGURE 55A conditions AND circuit AE. Depending upon the condition causing the reset of the sense service trigger AB, AD, AND circuit AE will be fully conditioned when either the *command-out* tag that initiated the stop or the *service-out* tag that stepped the sense byte counter to a count of seven is deactivated. These conditions are exclusive of one another. AND circuit AE via OR circuit AF, inverter BP, and OR circuit AG fire the 450 nanosecond end single shot AH. Anytime the end single shot AH fires the end of an operation is indicated. The output of end single shot AH is applied to the freeze latch designated by inverter BM which through inverter BN initiates *—end pulse* to FIGURE 76. On FIGURE 76 *—end pulse* via OR circuit AG and inverter AH initiates *—end reset* to FIGURES 64A, 67, 85A, 104 and 120. On these pages *—end reset* resets the command register. On FIGURE 54A *—end pulse* sets the channel end trigger AL, AP. The set output of the channel end trigger AL, AP initiates *+channel end trigger* to FIGURE 10. On FIGURE 10 *+channel end trigger* is applied to one input of AND circuit AT. The reset output of the channel end trigger AL, AP via OR circuit AA on FIGURE 53A conditions AND circuit BF to set the first command trigger AC, AF. The set output of the first command trigger AG, AF via conditioned AND circuit AG sets the status in trigger AH, AM just as in the initial selection sequence. On FIGURE 10, therefore, *+status-in trigger* fully conditions AND circuit AT to gate the channel end status bit on *bus-in* 4 to the channel. The preceding execution of a sense command assumed the *burst mode* of operation. Assume now a *byte mode* and return to the point in the *burst mode* sequence where the *service-in* tags were effecting transmission of sense bytes. For the *byte mode* *—force burst mode* applied to OR circuit AA on FIGURE 55A will be in a positive condition. Accordingly, OR circuit AA can only be conditioned when the *select-out* tag is activated. Assume the *service-in* tag has been activated. The channel responds to the *service-in* tag with the *service-out* tag. *—Service-out dly* on FIGURE 55A deconditions AND circuit AC. When the *service-out* tag is deactivated, the positive condition of *—service-out dly* further conditions AND circuit AC. Since this is a *byte mode* operation, the *select-out* tag will have fallen during the previous byte transfer. On the next activation of the *select-out* tag, *—select-out* via OR circuit AA fully conditions AND circuit AC to initiate *—sense service-in* to FIGURE 72A. On FIGURE 72A *—sense service-in* sets the service in trigger AJ, AF to initiate *—service-in trigger* to FIGURE 54A. On FIGURE 54A *—service-in trigger* via OR circuit AD initiates an interrupt sequence. Since the service in trigger AF, AJ on FIGURE 72A is set during the preceding sequence, the *service-in* tag will be activated to the channel instead of the *status-in* tag. From this point, the *byte mode* transfer of information returns via the interrupt to the initial selection sequence as previously described. At the end of a byte transfer, the sense service trigger on FIGURES 55A and B is reset in the same manner in which it was reset during the *burst mode* operation to initiate an end pulse. Again the end pulse sets the channel end trigger AL, AP on FIGURE 54B. The channel end trigger AL, AP conditions AND circuit BS which via OR circuit AJ conditions one input of AND circuit AC. *—End pulse* from inverter BN on FIGURE 55B lasts for the duration of the 450 nanosecond end single shot AH on FIGURE 55B. Thereafter its positive condition conditions AND circuit AC which via OR circuit AD initiates another interrupt. Since the service in trigger is not set at this time, the *status-in* tag is initiated indicating an end status to the channel. During the data transfer portion of a sense operation in *byte mode* the service in trigger AF, AJ on FIGURE 72A can be set only when *select-out* is active. During an interrupt sequence the address in trigger AK, AN on FIGURE 54B can be set only when the *select-out* tag is inactive. Thus the operational in trigger AR, AT on FIGURE 53A can be set only when the address in trigger has been set by a previous deactivation of *select-out*. Accordingly, when operating in *byte mode* during a sense operation, one *select-out* will cause a byte to be transferred, the next *select-out* will cause the service in trigger to be set, the following *select-out* will cause a byte to be transferred, etc., thereby allowing data to be transferred only on alternate *select-outs*. This is done to ensure that a low priority control unit executing a sense operation in *byte mode* will not, for long periods, dominate the *select-outs*.

It is now assumed that a read command was decoded upon receipt of the *command-out* tag during the initial selection sequence. Initial selection for a read command proceeds in the same manner as initial selection for a sense command. Accordingly, AND circuit AK on FIGURE 67 is conditioned by *+status-in and service-out* from FIGURE 53B and *+command register b trigger* from FIGURE 66B. AND circuit AK via OR circuit AL sets the tape *op* sync trigger AM, AQ. When the *service-out* tag is deactivated, the positive condition of *—service-out* fully conditions AND circuit AJ to set the tape *op* trigger AN, AR to initiate *+tape op trigger* to FIGURE 68A. On FIGURES 68A and B the decoded read command initiates *—read command* which, via OR circuit AQ in combination with *+tape op trigger*, fully conditions AND circuit AR to initiate *—read op*. *—read op* starts the read operation in the TCU. When the TCU circuitry has a byte of data available for transfer to the channel, it will set the byte of data into the read-write register on FIGURES 18A and B, 19A and B, and 20A and B, and initiate *—read-clock reset not write* (*—RCRS not write*) pulse.

On FIGURES 72A and B —*RCRS not write* conditions OR circuit AN. OR circuit AN via inverter AP sets the read/write control trigger AR, AX and inhibits the start shift AND circuit AK. When the read/write control trigger AR, AX is set, its set output conditions the start shift AND circuit AK. When the —*RCRS not write* pulse falls, the start shift AND circuit AK is further conditioned. If at this time the service in trigger AF, AT is reset, the start shift AND circuit AK will be fully conditioned and will initiate —*start shift* to FIGURE 73A. On FIGURES 73A and B —*start shift* via OR circuit BB fires a 250 nanosecond single shot AA. The output of single shot AA is applied to the freeze latch designated by inverter AW to initiate +*shift data* via additional inverter BD to FIGURES 75A and B and —*shift data* via additional inverter AB to FIGURE 72A. When the 250 nanosecond pulse from single shot AA was applied to the input to the freeze latch, AND circuit AS, it created a positive condition at the output of AND circuit AS which fully conditions AND circuit AU. The output of AND circuit AU via DOT OR connection AV conditions OR circuit BC to fire a 500 nanosecond single shot AE. The output of single shot AE via inverter AF and previously conditioned AND circuit AC is applied to the freeze latch designated by inverter AD to initiate +*single shot* **2** to FIGURES 72A, and 20A. The output of single shot AE directly initiates —*single shot* **2** *or shift data* to FIGURE 72A. In a read operation in which the byte converter is not in operation AND circuit AC on FIGURES 75A and B is fully conditioned by +*shift data* to initiate —*tape demand* to FIGURE 72A. On FIGURES 72A and B —*tape demand* sets the service in trigger AF, AJ thereby deconditioning the start shift AND circuit AK. If the byte converter is in operation the service in trigger AF, AJ would not have been set and —*shift data* would reset the read/write control trigger AR, AX, thereby deconditioning the start shift AND AK to deactivate —*start shift* to FIGURE 73A. If a second byte of read data is entered into the read/write register FIGURES 18A and B, 19A and B, 20A and B before the first byte has cleared the data register FIGURES 12, 13, 14, 15, 16 and 17, the second byte will stay in the read/write register until the data register is ready to accept it. In this case the *service-out* tag response to the *service-in* tag will reset the service in trigger AF, AJ on FIGURES 72A and B but the negative condition of —*service-out* holds the start shift AND circuit AK deconditioned until the *service-out* tag is deactivated whereupon start shift AND circuit AK is conditioned and another data shift is taken. The byte of data in the read/write register will then be transferred into the data register. The operation is similar for a write operation except that prior to starting the operation a *service-in* tag is initiated by AND circuit BJ. AND circuit BJ is conditioned following initial selection for a data write by +*write command*. This sets the service in trigger AF, AJ to initiate the *service-in* tag in response to which the *service-out* tag signifies a byte of write data will be sent to the control unit. When the *service-out* tag is deactivated the start shift AND AK will be partially conditioned. A start shift will be generated as soon as the read/write control trigger AR, AX is set by —*write clock* **2** of the first write clock cycle. Therefore, the first byte of write data during the write operation will be set into the data register prior to the actual initiation of the write operation. Accordingly, transfer from the data register to the read/write register will not take place until after write clock **2** of the first write clock cycle. For the remainder of the write operation, the shifting proceeds similar to the shifting for a read operation except for direction of the transfer. An overrun condition caused by a buffer limitation is indicated by OR circuit AL and trigger AQ, AM on FIGURES 72A and B to initiate —*set overrun trigger* to FIGURES 26A and B. The two lower inputs to OR FF of trigger AQ are the conditions for overrun. The OR circuit AL is applied to trigger AQ, AM so that in the event the output of OR circuit AL is a sliver condition, the trigger AQ, AM will convert the sliver to a —*set overrun trigger* condition. On FIGURE 73B AND circuit AC is used to insure that the —*single shot* **2** does not rise before the fall of +*shift data*. The trigger comprising elements AJ and AL is used to control the rise of —*stop data transfer*. Consider the case where the byte converter is on. Write operation is initiated and a byte of write data is transferred to the control unit in response to the fist *service-in* tag. This byte of data will be sent into the data register and will be transferred to the read/write register following write clock **2** of the first write clock cycle. At this time a second *service-in* tag will be signaled. The channel can respond with the *command-out* tag to initiate a stop. However, even though a stop is indicated it is possible for there to be two bits of data remaining in the data register. These have to be shifted into the read/write register and written on tape. At this time +*stop data transfer* cannot be initiated since the OR FF portion of trigger AJ, AL has a positive condition at all its inputs. Following write clock **2** of the second write clock cycle these two bits of data will be shifted into the read/write register accompanied by zeros in the higher order positions of the byte. Upon initiation of the shift, —*start shift* via the 250 nanosecond single shot AA and the DOT OR connection AV of the freeze latch applies a negative input to trigger AL. This allows +*stop trigger* to set trigger AJ, AL to initiate +*stop data transfer*. This continues as long as the stop trigger AS, AV on FIGURE 54A is set. The same operation occurs if the stop is issued to a third *service-in* tag. If a stop is issued to a fourth *service-in* tag there will be no need to take an additional shift since the data register will have been cleared prior thereto. In this case both byte counter A and B triggers on FIGURES 74A and B are reset so AND circuit AK on FIGURE 73A conditions OR circuit AL to initiate +*stop data transfer* on the rise of +*stop trigger*. During a read operation or during the byte converter off mode OR circuit AL is continually conditioned by the negative condition on +*byte converter* and +*stop data transfer* will rise directly with the rise of the +*stop trigger*. During the read operation, the transfer of a byte of data produces +shift data on FIGURES 75A and B. Via conditioned AND circuit —*tape demand* is initiated to FIGURE 72A. On FIGURES 72A and B —*tape demand* sets the service in trigger AF, AJ. It also deconditions AND circuit AA. The set output of the service in trigger AF, AJ further conditions AND circuit AA so that when —*tape demand* is deactivated AND circuit AA will be fully conditioned to initiate +*service-in* via inverter BF for a *burst mode* read operation. For a *byte mode* read operation, the *service-in* tag will be initiated by means of an interrupt, as it was for the *byte mode* sense operation. Assuming this is a byte converter off read operation, +*service-in* will condition AND circuit AQ on FIGURES 75A and B which, via inverter AV, initiates +*read byte* **2** to FIGURES 8, 9, 10 and 11. On these pages +*read byte* **2** will gate data register positions to *bus-in* lines. For instance, +*read byte* **2** conditions AND circuit AC on FIGURE 9 which, via OR circuits AV and AE, initiates +*bus-in* **2** to FIGURE 80A. On FIGURE 80A +*bus-in* **2** via conditioned AND circuit AS and OR driver AB initiates —*bus-in* **2** to FIGURE 4 for exit to the channel. The transfer of read bytes continues until the tape unit controls recognize an end of record condition in which case —*RDD* **144** *only* pulse is initiated to FIGURE 55A. On FIGURES 55A and B —*RDD* **144** *only* via OR circuit AF, inverter BP and OR circuit AG fires the end single shot AH causing an end operation identical to the end operation described under the execution of the sense command.

Now assume that a write command was decoded upon receipt of the *command-out* tag during the initial selection sequence. The initial selection for a write operation is identical to the initial selection for a sense operation with

the exception that a write command is set into the command register on FIGURES 64A and B, 65A and B, and 66A and B. Following initial selection AND circuit BJ on FIGURE 72A sets the service in trigger AF, AJ to cause the first byte of write data to be transferred to the TCU. —*suppress-out* applied to AND circuit BJ provides means to inhibit the *service-in* tag from being signalled whenever the *suppress-out* tag is active. This constitutes the suppress data function. It is the only time during a write operation that suppress data can proceed without causing an overrun. When the *service-out* tag responds to the *service-in* tag +*write command* and +*service-in and service-out* condition AND circuit AP on FIGURE 67. The output of AND circuit AP via OR circuit AL sets the tape *op* sync trigger AM, AQ. When the *service-out* tag is deactivated AND circuit AJ is fully conditioned to set the tape *op* trigger AN, AR to initiate +*tape op trigger* to FIGURE 68A. On FIGURE 68A +*tape op trigger* conditions AND circuit AU which via inverter AV initiates +*write op*. This initiates the write operation in the tape unit controls. The remainder of the write operation proceeds exactly as the read operation except for the direction of data transfer. On FIGURE 75A +*service-out* and +*service-in dly* condition AND circuit AJ. The output of AND circuit AJ via inverter AK conditions AND circuit AG. The output of AND circuit AG via inverter AP initiates +*write set byte* **1** to FIGURES 14, 15, 16, and 17. On these pages +*write set byte* **1** gates data bytes from *bus-out* lines to the data register. For example, on FIGURE 14 +*write set byte* **1** gates +*bus-out* **7** via AND circuit AA, OR circuit AR, DOT OR connection AE and inverter AF to initiate +*data register* **8** *trigger* to FIGURES 18A, and 19A. On FIGURES 18A, and 19A +*data register* **8** *trigger*, under control of the byte counter on FIGURES 74A and B, will be gated into the read-write register. Depending upon the count in the byte counter, +*data register* **8** *trigger* will be gated on FIGURE 18A by +*data register to read write register BCF* via AND circuit BG and OR circuit AH into read write register position **7** trigger AD, AK or on FIGURES 19A and B by +*write shift byte* **2** or **4** via AND circuit AF and OR circuit AG into read write register position **3** trigger AH, AK.

Assume now a control command has been decoded. The control operation proceeds exactly the same as a read operation with the exception that no data transfer occurs. A control command decoded during the initial selection sequence conditions AND circuit AP on FIGURE 66A indicating that a control command has been received. Bits **2**, **3** and **4** of the command register are decoded on FIGURES 68A and B to determine which of 8 possible control codes have been received. The control code structure is contained in the command-control format table.

(7) *Tape unit selection and unit freed scanning.*—Tape unit selection occurs when the tape unit address on *bus-out* lines **4**, **5**, **6** and **7** is set into the tape unit select register on FIGURES 95A and B. Thereafter, the address of the tape unit is applied to the decoder circuit on FIGURES 61A and B, and 62A and B to activate the tape unit select line to the selected tape unit. The unit freed scanner is a device end scanner that will continually scan all tape units until it finds a unit freed indication (device end). When a unit freed indication is found, the scanner will stop scanning and indicate the address of the tape unit concerned. When the unit freed is reset, the scanner will resume scanning. Unit freed is an interface status bit indicating that an I/O device has completed a previously initiated command. For this and all following explanations, device end and unit freed are synonymous. The tape unit select register and the unit freed scanner have various functional elements in common. FIGURES 97A, B, C, and D are a composite drawing presenting all the elements from the various figures contributing to tape unit select register and the unit freed scanner. For the purpose of this drawing, each of the figure numbers from which a block or series of blocks was taken is included inside the block. By way of example, tape unit select trigger **7** comprises OR circuit AC and AND circuit AF, both of which have been shown on FIGURE 95A. The dual function of the circuit will become obvious from the following description. The circuit is operative as the tape unit select register during the initial selection sequence. When the tape unit address on bus-out lines **4**, **5**, **6** and **7** has been decoded —*adapter address decoded* from FIGURE 60B sets tape unit select trigger AG, AM on FIGURE 96A to initiate —*tape unit select trigger* to FIGURE 97D. On FIGURE 97D —*tape unit select* is applied to AND circuits **87B–AA**, **90B–AA**, **92B–AA**, **93B–AA**, **88B–AS**, **89B–AS**, **91B–AS**, **94B–AS**, **87B–AS**, **93B–AS**, **90B–AS**, **92B–AS**, **88B–AA**, **94B–AA**, **89B–AA**, and **91B–AA**. Other inputs to these AND circuits are from the unit freed registers on FIGURES 87A through 94B, inclusive. Accordingly, they provide inputs to the TU select register positions when the unit is being used for unit freed scanning. Initially it is assumed that the register is to be used for tape unit selection. Accordingly, —*tape unit select trigger* is in a negative condition which deconditions the scanner inputs to inhibit scanning. +*set tape unit address* from inverter AL on FIGURE 60B was initiated via AND circuit AK by the address decode condition to FIGURE 97A. On FIGURES 95A and B +*set tape unit address* conditions AND circuits AB, AE, AH, AL, AP, AS, AU and AW all from FIGURE 60B. *Bus-out* lines **7**, **6**, **5** and **4** via AND circuits AB, AH, AP, AU are applied as the set inputs to the tape unit select register positions **7**, AC, AF; **6** AJ, AH; **5** AQ, AT; and **4** AV, AX. The output lines of the tape unit select register are applied to a bus which returns them to the entry side of FIGURE 97D to be applied to the address decoder as indicated by the dotted box entitled "Address Decoder" and represented in complete detail on FIGURES 66A and B. The output of the address decoder initiates selection lines for tape units **0** through **15**. The address in the tape unit select register is maintained until either another tape unit is selected or until the following conditions are met: the adapter is not performing an initial selection, the adapter is not performing a command, the channel end trigger is reset, the stack interrupt trigger is reset, and a chain condition is not indicated by the channel. The chain condition is established by the channel through activation of *suppress-out* when the *status-in* and *service-out* tags are active, the channel trigger on FIGURES 54A and B is set or —*select unit freed* on FIGURE 96A is negative. The chain condition is a requirement placed on tape unit whereby the tape unit remains connected to the control unit having continuous chain of commands. When the aforementioned conditions have been met, the tape unit select trigger AG, AM on FIGURE 96A is reset. The resulting positive condition on —*tape unit select trigger* on FIGURE 97D conditions the aforementioned unit free input AND circuits. These AND circuits in combination with the tape unit select triggers **7**, **6**, **5** and **4** comprise a sequential ring device which begins with the address initially set into it from the *bus-out* lines and freely cycles through the following scanning sequence:

SCANNING SEQUENCE

| TU | 4 | 5 | 6 | 7 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 1 |
| 3 | 0 | 0 | 1 | 0 |
| 4 | 0 | 1 | 1 | 0 |
| 5 | 0 | 1 | 1 | 1 |
| 6 | 0 | 1 | 0 | 1 |
| 7 | 0 | 1 | 0 | 0 |
| 8 | 1 | 1 | 0 | 0 |
| 9 | 1 | 1 | 0 | 1 |
| 10 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 |
| 12 | 1 | 0 | 1 | 0 |
| 13 | 1 | 0 | 1 | 1 |
| 14 | 1 | 0 | 0 | 1 |
| 15 | 1 | 0 | 0 | 0 |

As the scanning sequence progresses, any tape unit or combination of tape units indicating a unit freed condition will activate —*unit freed.* The effect of this line is to decondition the unit freed AND circuit to which it is applied with the result that the sequencing ring stops sequencing to indicate the address of the freed unit. If there are multiple tape units indicating the unit freed condition, scanning stops on the first active unit freed line encountered during sequencing. Upon accommodation of that indicated unit, sequencing resumes to the next active line.

Unit freed indications are generated by 16 identical circuits, one for each tape unit. The unit freed circuitry for tape unit **0** is on FIGURES 87A and B. On FIGURES 87A and B —*tape unit* **0** *rewinding* is applied to line terminator AJ. The output of line terminator AJ is applied to the freeze latch designated by inverter AN. The tape unit rewinding indication will be frozen by —*freeze or initial select* either during initial selection or a freeze condition in order to prevent the rewinding indication from changing whenever the *status-in* tag is transmitted to the channel since the rewinding lines can affect the status bits. Such a change begins when the tape unit rewinding lines from the tape units on FIGURES 87A and B initiate —*selected unit rewinding* to FIGURES 96A and C. On FIGURES 96C and D these lines via OR circuit AR initiate *tape unit status B* to FIGURE 86. On FIGURE 86 +*tape unit status B* initiates +*ready and rewind or switch* to FIGURE 96C. On FIGURES 96A, B, C and D +*ready and rewind or switch* via AND circuit AZ initiates +*busy status* to FIGURE 9. Returning to FIGURES 87A and B, AND circuits AB, AG, OR circuit AC and inverter AD form the unit freed arming latch for tape unit **0**. This latch is set whenever OR circuit BJ is conditioned while tape unit **0** is selected. OR circuit BJ is conditioned each time an end pulse is initiated at the end of an operation or when the tape unit is not ready during initial selection. If tape unit **0** is not rewinding when the tape unit **0** arming latch is set, —*unit feed* **0** *input* will be indicated immediately. —*unit freed* **0** *input* is indicated via AND circuit AE. The two inputs to AND circuit AE are the unit freed arming latch and the freeze latch output. When the tape unit is not or has completed rewinding AND circuit AE deconditions AND circuit AA. AND circuit AA is one of the unit freed scanner input AND circuits previously described on FIGURES 97A and B. AND circuit AH is identical to AND circuit AE with the exception that it has an additional input, +*tape unit* **0** *select.* The output of AND circuit AH is +*selected unit* **0** *freed* and is OR'ed with the other selected unit freed lines from the remaining 15 unit freed circuits on FIGURES 96A, B, C and D to initiate the unit freed status bit. If tape unit **0** is rewinding when the tape unit **0** arming latch is set, unit freed will not be indicated immediately but rather will wait until —*tape unit* **0** *rewinding* becomes inactive. On FIGURES 96A and B the 16 individual selected unit freed lines are OR'ed together via OR circuit AA. The output of OR circuit AA via inverter AB initiates —*selected unit freed* to FIGURE 53A. On FIGURE 53A —*selected unit freed* via OR circuit AA will initiate the *status-in* tag to the channel if *operational-in* is active. This causes the unit freed status bit to be transmitted to the channel in an identical manner signalling channel end during *burst mode* operation. On FIGURE 54A —*selected unit freed* is also applied to OR circuit AD. This allows —*selected unit freed* to cause a status interrupt if *operational-in* is inactive. This is identical to the interrupt sequence for indicating channel end when operating in byte mode.

(8) *Sequence indication.*—The sequence indicators are a set of three triggers that are set and reset by AND circuits which recognize preassigned check points within a given operation. These check points are in series in time during the execution of any commanded operation. Recognition of the preceding checkpoint is a condition of the next checkpoint. The operation of the sequence indicators for a tape unit during a read operation follows. Referring to FIGURES 139A and B, the first checkpoint in a read operation is read delay, go, and first character are all active. AND circuit AE on FIGURE 131B detects this condition. AND circuit AE conditions OR circuit AF which conditions AND circuit BA. Sequence indicator C trigger should be off at this time since this is the first step in the operation. Therefore AND circuit BA is conditioned to set sequence indicator A trigger AD, AM. The second read operation checkpoint is read op, not read delay, first character, and start read clock active. This condition is detected by the AND circuit AT which will be fully conditioned if sequence indicator A trigger AD, AM is set indicating that the previous checkpoint had been recognized. AND circuit AT will set sequence indicator B trigger AP, AW. The third read operation checkpoint is read op, not first character, read condition and RDD active. This checkpont is detected by the AND circuit AF on FIGURE 132. AND circuit AF will be conditioned when this checkpoint is detected after the sequence indicator B trigger AP, AW on FIGURES 131A and B is set indicating that the previous checkpoint had been recognized. AND circuit AF will set sequence indicator C trigger AG, AN. The fourth checkpoint is read op, not go, read condition and check character active. This is detected by AND circuit AL on FIGURE 131B. When this checkpoint is detected indicator C trigger will have been set indicating that the previous checkpoint was detected. This fully conditions AND circuit A to reset sequence indicator A trigger AD, AM. The fifth checkpoint is not go, not first character, RDD **96** gate, RDD active. This is detected by AND circuit AV on FIGURE 131B. When the fifth checkpoint is detected sequence indicator A trigger AD, AM is off indicating that the previous checkpoint was detected. AND circuit AV will, therefore, reset sequence indicator B trigger AP, AW. The sixth checkpoint is not read delay, not read condition, and check character active and is detected by AND circuit AM on FIGURE 132. When this checkpoint is detected sequence indicator B trigger AP, AW is off, indicating that the previous checkpoint was detected. AND circuit AM wll therefore reset sequence indicator C trigger AG, AN on FIGURE 132. This was the last checkpoint in the operation and if no error conditions have occurred, all sequence indicators are off. At the end of the operation the sequence indicators are checked. If any sequence indicator triggers are on, an error is signalled. Upon the error indication, the code contained in the sequence indicators will indicate the last checkpoint that was detected in correct order. For instance, if at the end of a read operation, all sequence indicators are still on, this would indicate that the sequence indicators did not receive the third checkpoint, read op, not first character, read condition, and RDD after the second checkpoint.

(B) Non-cyclic error detection

The detailed embodiment described herein uses the following non-cyclic types of error detection items and circuits:

(1) C-compare circuit for byte converter checking.
(2) Odd-even character count trigger.
(3) Longitudinal redundancy check (LRC).
(4) Vertical redundancy check (VRC) for data bytes.
(5) Echo error.
(6) Skew error.
(7) Parity check for counters and clocks.

*C-compare circuit for checking byte converter.*—When the third bit of a read or write command specifies use of the byte converter, the C-compare circuit (FIGURE 155) detects errors that occur during byte conversion. In processing three 8-bit interface bytes or four BCD 6-bit tape bytes, the byte counter advances through one cycle, during which a single C-compare operation occurs. This is described and claimed in U.S. patent application No.

99,927, filed Mar. 31, 1961, and now Patent No. 3,200,242, by J. K. Crawford and C. M. Pietras.

Each odd parity byte (BCD or 8-bit) stored in or gated from the data register causes the binary C-compare control trigger (**26**B–AK, AQ) to change states. The C-compare control trigger should be pulsed an even number of times during a byte counter cycle to allow the trigger to return to its reset state when the last byte is transferred from the data register. If the control trigger is set when the byte counter completes the cycle, the trigger's ON output sets the device check trigger (**35**–AG, AN), which indicates the error. The following example illustrates C-compare circuit actions in an eight-bit mode write operation. Follow the diagram of the example in FIGURE 156.

If the 8-bit bytes 11110001, 11001101, and 01010100 transfer to the data register in a write operation, the 6-bit bytes 111100, 011100, 110101, and 010100 enter the read-write register. Because the first 8-bit byte contains an odd number of bits, it causes the C-compare control trigger to turn on. The first 6-bit byte leaving the data register contains an even number of bits and does not pulse the control trigger. The second input 8-bit byte contains an odd number of bits; it resets the control trigger. The second output 6-bit byte is odd; it sets the control trigger. The third 8-bit byte resets the trigger. The third and fourth output 6-bit bytes contain an even number of bits and do not cause the trigger to change states. At the end of the byte counter cycle, the C-compare control trigger is reset; the tape control does not indicate an error. No byte configurations in eight bit mode read or write operations should cause the C-compare control trigger's ON output to be active when the data register is empty.

*Odd-even character count trigger.*—The odd-even character count trigger (**47**A–BL) determines whether the tape block has an odd or even number of bytes preceding the check byte. Each input byte to the tape control starts the read clock through a cycle, as shown in the preceding Tape Control Timing Pulse Chart. At RC–0 time in all other read clock cycles except the one started by the check byte, the binary odd-even character count trigger is actuated and changes states. If the trigger is in the reset (OFF) state when the tape control receives the check byte (indicating that an even number of bytes precede the check byte), the vertical redundancy check circuit examines the check byte for even parity. If set, the (ON) state of the count trigger is active when the tape control receives the check byte (indicating that an odd number of bytes precede the check byte), the vertical redundancy check circuit examines the check byte for odd parity as specified by the state of the odd redundancy gate (**65**B–BD). The following examples illustrate this action.

If the odd redundancy gate is active and a two byte (plus check byte) block is written on tape, the second byte resets the character count trigger, showing that an even number of bytes precede the check byte in the block. If the bit structures of the BCD characters are 7, 6, 5, 4, 3, 2, P and 7, 5, 3, the bit structure of the check byte is 6, 4, 2, and P. Because the odd redundancy gate is activated, the first two bytes in the block are checked for odd parity. The character count trigger is off when the check byte transfers to the tape control; therefore, the vertical redundancy check circuit examines the check character for even parity.

If the odd redundancy gate is activated and a three byte block (plus check byte) is written on tape, the third byte sets the character count trigger, showing that an odd number of characters precede the check character. If the bit structure of the BCD characters are 7, 6, 5, 4, 3, 2, C; 7, 5, 3, and 7, 5, 4, the bit structure of the check byte is 7, 6, 5, 2, C. The first three characters of the block are checked for odd parity. Because the character count trigger is set when the check byte transfers to the tape control, the vertical redundancy check circuit examines the check byte for odd parity.

*Longitudinal redundancy check register (LRCR).*—The LRCR contains nine triggers, P, **0**–**7** found on FIGURE 46A, one for each of nine tape tracks. However, only seven of the triggers are with 7-track tapes. The LRCR checks each complete tape block.

Input bits to the LRCR from the skew register cause outputs from corresponding binary LRCR triggers to rise or fall to their opposite states (the first P-bit input sets the P trigger; the second P-bit input resets the P trigger, etc.). After the tape control processes the check byte, all LRCR triggers should be in their reset states. An active output from any LRCR trigger at the end of the operation (RDD–128 time) conditions "LRCR error," causing the LRCR error and data check triggers to turn on (FIGURE 157).

FIGURE 157 illustrates LRCR action when a 7-track tape control processes the simple tape record containing the characters I, B, M, and a check character. Assume that the tape command specified even parity. When a skew register transfers the record's check byte to the LRCR, all LRCR triggers in the ON state revert to their reset conditions. Because all LRCR triggers are off at RDD–128 time, no LRCR error is registered.

Assume that bits in dotted circles were dropped in transferring characters from the tape unit to the tape control. The bit structure of the check byte indicates that the bits were written on tape. Dotted lines in the sequence chart represent error conditions that the dropped bits cause. When the check byte transfers to the LRCR, all LRCR triggers are not set to their OFF states. Because ON outputs from two LRCR triggers are active at RDD–128 time, "LRCR error" sets the LRCR error and data check triggers.

*Vertical redundancy check (VRC) circuit.*—In read and write operations, bytes in the hi-clip skew register and the read-write register transfer to the vertical redundancy check (VRC) circuit, but only in read operation can the read-write register byte cause a VRC error. VRC circuits determine whether the byte contains an odd or even number of bits. All other bytes except the Tape Mark can contain either an odd or even number of bits without changing the byte.

During the I/O initial selection sequence, the tape command specifies that bytes processed to and from the tape unit should contain either an odd number of bits (odd parity) or an even number of bits (even parity). When the tape command designates odd parity, the tape control activates the odd redundancy gate. All other bytes read from tape in the operation except the check byte must contain an odd number of bits. If the tape command does not cause the tape control to activate the odd redundancy gate, all other bytes from tape except the check byte must contain an even number of bits.

In a write operation, the tape control sets the hi-clip VRC and data check triggers when:

(1) the odd redundancy gate is activated and any other byte except the check byte is found to contain an even number of bits.
(2) the odd redundancy gate is not activated and any other byte except the check byte is found to contain an odd number of bits.

In a read operation, the tape control sets only the hi-clip VRC trigger to indicate a parity error.

If bits are lost in transferring a byte from the selected tape unit to the tape control, VRC circuits indicate an error only if an odd number of bits are dropped. The following example illustrates this action. If the odd redundancy gate is activated and the tape control receives a byte containing the bits **1**, **2**, **3**, **4**, **5**, **6**, and P, no error is registered because the character's bit count is odd. If the P-bit is lost in transferring the byte from the tape unit, the tape control indicates an error because the byte then con-

tains an even number of bits. If both the P- and 1-bits are lost, the byte's bit count is still odd; consequently, the tape control does not detect the error.

In a write operation, the P-trigger in the read-write register is not used because the data register does not transfer a byte's parity bit to the read-write register. The tape control compares the result of the vertical redundancy check to the state of the odd redundancy gate. If VRC circuits determine that the byte in the read-write register contains an odd number of bits (exclusive of the parity bit) and the odd redundancy gate is not active, the tape control adds a parity bit to the byte transmitted to the tape unit. If the byte in the read-write register contains an odd number of bits and the odd redundancy gate is active, the tape control transmits the byte in the read-write register to the tape unit without adding the parity bit. The data register may transfer either odd or even parity bytes to the read-write register, but the byte that the tape control transfers to the tape unit must be in the parity that the odd redundancy gate designates.

*Echo error.*—In a write operation, the tape control transmits a byte to the selected tape unit to be recorded on tape during each write clock cycle. When the tape unit writes the byte, it conditions "echo pulse" to the tape control.

Each time that the write clock advances to 6, it generates a pulse to set the no echo trigger 46B–AA, AE. An "echo pulse" from the tape unit resets the trigger. The tape control checks the state of the no echo trigger when the write clock advances to 3 in the succeeding clock cycle. "Echo pulse" should reset the no echo trigger before this time. If, however, the no echo trigger is on, the tape control sets the echo error and data check triggers.

*Compare error.*—The byte processed through lo-clip final amplifier circuits sets the lo-clip skew register while the hi-clip final amplifier circuits set the hi-clip skew register. Therefore, the same byte might not be stored in both the hi- and lo-clip skew registers. The tape control compares the bytes stored in the skew registers. If the registers do not contain the same character in a write operation, the tape control sets the compare and data check triggers. In a read operation, the tape control sets only the compare trigger if the characters in the skew registers are unequal; and no data check occurs if proper vertical redundancy occurs for the byte in the lo-clip register.

*Skew error.*—The write head on the selected tape unit should write all bits in a byte at a 90° angle to the edge of the magnetic tape; when the byte passes the read head, the tape unit will then transfer all bits in the byte to the tape control simultaneously. Several conditions can cause improper alignment of bits across the face of the tape, preventing the simultaneous transfer of all bits in the byte to the tape control. The interval between the times that the byte's first and last bits are read and transferred to the tape control is called skew. Although it is not necessary that skew be completely eliminated, if the elapse in time between the arrival of the first and last bits in a byte exceeds a predetermined amount, the tape control indicates an error, since the bits may no longer be grouped with the proper byte.

The first bit in a byte transferred to the tape control starts the read clock. The clock generates an output to set the skew gate trigger when the clock advances to:

(1) RC–4 if the selected tape unit is not operating at 800 bits per inch.
(2) RC–5 if the selected tape unit is operating at 800 bits per inch.

Bytes from the tape unit can arrive at the tape control between the times that the read clock starts and the skew gate trigger is set. However, a bit from the tape unit while the skew gate trigger is set causes the tape control to set the skew error and data check triggers. An output from the read clock at the end of the read clock cycle resets the skew gate trigger and the read clock. The first bit of the next character from the tape unit causes the tape control to repeat the operation.

*Parity checked counters and clocks.*—(FIGURES 201 and 202.) The delay counter, read clock and write clock are each checked by a unique parity arrangement. Each of these counters has outputs which sequentially change in a Gray code sequence. The counter or clock outputs are provided in parallel along with the drive input of the counter as inputs to a vertical redundancy check (VRC) circuit **1** in FIGURE 201 operating in even parity. In a Gray code counter, only one variable changes between two sequenced outputs. The drive input P is up and then down for two sequential outputs which differ in only one bit position. Therefore a parity check of all the outputs of the counter and its drive yield an output having the same parity for each output from the counter or clock. This is illustrated by the timing diagrams of FIGURE 202 in which the drive input P and the outputs A, B, C, through J of the counter are illustrated. The drive input P and the outputs A, B, C through J provide inputs to even parity check circuit **1** which generates an output if there is a failure in any one of the inputs. The output of the parity check circuit feeds a delay circuit **2** which is inserted to filter out transients that occur when the counter is stepped. The length of delay should be less than one-half the period of the drive input wave P. The output of the delay circuit feeds one input of AND circuit **3**, its other input is received directly from the output of parity check circuit **2**. In the event any one of the inputs to the parity check circuit fails, the parity will no longer be even and an output will occur. Therefore, any single failure within the clock will produce an error signal which has a minimum width of the drive pulse width minus the delay time of delay circuit **2** in FIGURE 201.

An odd parity check circuit could be used in place of circuit **2** by replacing any VRC input with its complemented form.

### (C) Data transfer circuits

Data transfer in the tape control generally involves transferring data through a series of byte registers. A sequence of clock timing pulses is needed to transfer a byte through this series of registers. *Read clock* pulses control the transfer of a byte read from tape. Similarly, *write clock* pulses control the transfer through some of the same registers in transferring data bytes from the channel I/O interface to tape. Also a *delay counter* controls when data transfer begins, in response to a channel command to read or write a block of data. The delay counter also determines when the end of a data block is reached, and then determines the timing for writing or reading a CRC byte and an LRC byte. Hence the delay counter, read clock, and write clock in the Tape Control provide timing pulses to execute tape operations. FIGURES 142, 143, and 144 show diagrammed basic circuits and waveforms for each timing circuit. Characteristics of oscillators that supply drive pulses to the timing circuits are listed in the following chart:

TAPE CONTROL OSCILLATORS

| Function: | Oscillator (by frequency), kc. |
|---|---|
| Delay counter millisecond mode control | 5.0 |
| Read clock | 500.0 |
| Write clock and delay counter microsecond mode control | 500.0 |
| Write clock and delay counter microsecond mode control | 720.0 |
| Read clock | 700.0 |

The next chart, Tape Control Timing Pulses, lists in order the output pulses derived from the read clock, write

clock, and delay counter when each is driven by an oscillator defined in the prior chart:

TAPE CONTROL TIMING PULSE CHART

Output Pulse Designation (In Time Order)

| | |
|---|---|
| | Read clock cycle |
| RC **0** | RC Reset Read |
| RC **4** | Skew Gate and RC |
| RC **6** | Reset Write |
| RC **7** | Skew Gate Set |
| | Write clock cycle |
| WC **2** | Write Pulse Width |
| WC **8** and **9** | Stop and WC **14** |
| WC **11** | WC **2** |
| Write Pulse | |
| | Delay counter cycle (Microsecond mode) |
| RDD **36** | RDD **144** |
| RDD **96** Gate | WDD **60** |
| RDD **128** | WDD **120** |
| | Delay counter cycle (Millisecond mode) |
| Backspace Reset Read Condition | |
| RDD **68** | |
| RDD **144** | |
| End WDD | |
| End Read Delay Not at Load Point | |
| End Read Delay at Load Point | |
| WD **320** | |
| D **36** | |
| D **40** | |
| D **50** | |
| D **96** Backspace | |
| D **170** Backspace | |
| D **180** Backspace | |
| D **190** Backspace | |
| Start Read Condition **32** | |

The AND gates picking off these timing pulses from the binary triggers in the read and write clocks and delay counter are shown in detail on FIGURES 109A, 106A and B, 113A and 115.

*Final Amplifiers.*—(FIGURE 145 and in detail in FIGURES 48, 49 and 50) in the tape control detect, shape, and amplify input data signals from a tape unit. The final amplifiers include biasing circuits to control clipping levels for each of nine identical tracks. In tape controls equipped with the 7-track feature, only seven of nine amplifier tracks are used.

Biasing circuits within the final amplifiers compensate for:

(1) frequency differences in data signals from various tape units having different data rates;
(2) different input amplitude requirements for read and write operations. A higher minimum input level is required in write operation than in read operation is to insure that characters written on tape are of sufficient amplitude to be read in subsequent read operations after deterioration occurs to tape properties.

To eliminate noise transfers from final amplifiers, the tape control conditions gates to designate intervals in which final amplifier outputs are acceptable. By holding output gates inactive, the tape control minimizes the possibility of transferring noise from the final amplifiers while the tape unit spaces through an interrecord gap.

When characteristics of the incoming data signals from the tape unit satisfy input requirements, final amplifier tracks generate hi- and lo-clip outputs. This is described and claimed in U.S. Patent 3,078,448 to H. O'Brien titled "Dual Channel Sensing."

If the input signal is of insufficient amplitude, the track either produces a lo-clip, but no hi-clip output or does not yield a lo- or hi-clip output.

As represented by FIGURE 191B, lo-clip pulses from final amplifiers set a *lo-clip skew register*; hi-clip pulses set a *hi-clip skew register*. Both the hi- and lo-clip skew registers contain nine triggers. (When the 7-track feature is installed on a tape control, only seven of the available nine triggers are used.) Deskewing is done by the first bit of a tape byte setting a first bit latch, which begins the read clock cycle designated previously. The skew registers can receive any skewed bits of the byte until RC–7, which occurs less than one-half of a bit period after the first bit of the byte. At RC–7, the accumulated bits in the selected skew register are gated out to the read/write register. The hi-clip register is selected, unless an error is indicated therein by the connected VRC (vertical redundancy check) circuit, in which case the lo-clip register is gated instead.

In a read operation, outputs from the selected hi- or lo-clip skew register set a *longitudinal redundancy check register* (*LRCR*). The LRC byte is not transferred to the read/write register, because it does not go to the interface. When read checking during a write operation, the skew register outputs set only the LRCR; and data bytes are not transferred to the read/write register.

Final amplifier outputs load the skew registers with a byte:

(1) before the read clock advances to four (RC–4 time) during a read/while/write operation, (the skew tolerance is more stringent during a write operation than during a read operation);
(2) before RC–7 time in read only operations.

A Vertical Redundancy Check (VRC) circuit checks each byte stored in the hi-clip register for parity error. The status of an odd redundancy trigger designates the type of parity expected for each byte in a tape block.

A hi-clip skew register parity error:

(1) sets the hi-clip VRC and data check triggers in write operation;
(2) sets the hi-clip VRC trigger in read operation;
(3) causes the byte (character) in the lo-clip skew register to transfer to the LRCR in write operation;
(4) causes the byte in the lo-clip skew register to transfer to the LRCR and to the read/write register in read operation if the check trigger is not set (which blocks transfer of the LRC byte).

If no error is detected in a byte in the hi-clip skew register, the corresponding byte in the lo-clip skew register is not used.

Because low amplitude tape signals to the final amplifiers can produce lo-clip outputs without producing corresponding hi-clip outputs, the same set of byte bits might not be loaded into both the hi- and lo-clip skew registers. A *hi-lo* compare circuit compares the bits in the hi-clip skew register to the corresponding bits in the lo-clip skew register. If the bit configurations are not equal, the compare circuit sets a compare error trigger and, in write operation, the data check trigger.

FIGURE 146 shows a representative part of each of skew register circuit.

In write operations, the *read/write register* (FIGURES 147A and B) accepts interface data from the data register and/or the translator and transmits bytes to the selected tape unit. In read operations, the read/write register receives tape unit input data from the hi- or lo-clip skew register and transfers characters to the translator and/or data register. Although the read/write register contains nine data triggers, only seven triggers are used in tape controls equipped with the 7-track feature.

The read/write register is reset and then it is set by a data byte during each write clock cycle in a write operation, and during each read clock cycle except when

the check character trigger is sent to indicate the reading of an LRC byte.

As previously described, during the initial selection sequence, the command byte on *bus-out* lines **5**, **6**, and **7** was decoded in the command decode circuits on FIGURE 66A. The remaining read/write command bit positions from **0** through **4** bus out are all zeros for nine track operation, which does not use either the byte converter or the code translator. Had the decoding command cycle indicated a read or write operation in 7 track mode, bit positions **0** through **4** would contain additional command information as defined by the following Read and Write Command Table:

READ AND WRITE COMMAND* TABLE

| Bit Position 0 1 | 2 | 3 | 4 | 5 6 7 | |
|---|---|---|---|---|---|
| | | | | 0 0 1 | Write |
| | | | | 0 1 0 | Read |
| | | | | 1 1 0 | Read Search (Presently Unused) |
| | | | | 1 0 0 | Read Backward |
| | | | 0 | | Code Translator Inactive |
| | | | 1 | | Code Translator Active |
| | | 0 | | | Byte Converter Inactive |
| | | 1 | | | Byte Converter Active |
| | 0 | | | | Odd Parity |
| | 1 | | | | Even Parity |
| 0 0 | | | | | 800 BPI |
| 0 1 | | | | | 556 BPI |
| 1 0 | | | | | 200 BPI |

*Bit positions 0, 1, 2, 3, and 4 must contain a zero for 9-track operation.

CONTROL COMMAND** TABLE

| Bit Position 0 1 | 2 3 4 | 5 6 7 | |
|---|---|---|---|
| | | 1 1 1 | Control |
| | 0 0 0 | | Rewind |
| | 0 0 1 | | Rewind and Unload |
| | 0 1 0 | | Erase |
| | 0 1 1 | | Write Tape Mark |
| | 1 0 0 | | Backspace Record |
| | 1 0 1 | | Backspace File |
| | 1 1 0 | | Forward Space Record |
| | 1 1 1 | | Forward Space File |

**Bit positions 0 and 1 must contain zeros for 9-track operation.

SENSE, TEST I/O AND NO OP COMMANDS*** TABLE

| Bit Position 0 1 2 3 4 | 5 6 7 | |
|---|---|---|
| | 1 0 0 | Sense |
| | 0 0 0 | Test I/O |
| | 0 1 1 | No Operation |

***Bit positions 0, 1, 2, 3, and 4 must contain zeros.

In a write Tape Mark operation, normal inputs to the read/write register are not conditioned. "7-track Write Tape Mark" command sets read/write register triggers **4**, **5**, **6**, and **7** on FIGURES 18A–20B. If the 7-track feature is not installed on the tape control, the "9-track Write Tape Mark" command sets the read/write register triggers **1**, **2**, and **3** on FIGURES 18A–20B. The read/write register conditions write bus lines **-1**, **-2**, **-3**, **-4**, **-5**, **-6**, and **-7** (9 track operation) or write bus lines **-4**, **-5**, **-6**, and **-7** (7 track operation) to the selected tape unit.

The status of the command register trigger **2** (**65A–AC**, AG) determines whether the bytes of a block are to be checked for odd or even parity for 7 track operation. It is always set for odd redundancy for nine track operation. In write operation, the bytes interface parity bit does not transfer to the read/write register; and in neither read nor write operation does the read/write register condition the output parity line ("R–W reg P tgr") to the tape unit, data register, or translator. A parity error in a read/write register byte sets the read/write register VRC circuit and data check triggers.

The *data register* accepts 8-bit data bytes (without parity) from the interface in write operation and processes 8-bit bytes to the interface in read operation. However, the data register contains twelve triggers to accommodate byte conversion for the 7 track feature. For 9 track operation only eight of the triggers are used, since no byte conversion is needed. The data register does not store parity bits. In write operation the tape control

checks interface data bytes for correct parity and discards the byte's parity bit prior to transferring the byte to the data register. In read operation, the parity bit is not set in the data register, but the tape control checks each output data register byte and adds a parity bit when necessary to convert a byte to the correct parity. FIGURES 148 and 149 show data register triggers and input and output gates.

For 9-track operation, each byte transferred to the data register sets data register triggers **5** through **12** corresponding to the bits in the character; triggers **1** through **4** are not used. Bytes transfer unchanged from data register triggers **5** through **12** to either read/write register triggers **0** through **7** in write operation or to the channel interface in read operation. Each byte loaded in the data register must be unloaded before the next byte can be stored.

For 7-track operation, data can be handled to or from tape without or with byte conversion (six or eight bit mode).

Byte converter mode: The byte converter is used for 7 track operation, if command register trigger **3** (FIGURES 65A and B) receives a one bit from a read or write command. A command activating the byte converter should not activate a code translator.

In a write operation, the byte converter transfers four six-bit BCD characters to tape for every three 8-bit bytes received from the interface. The seventh bit read from tape with each six data bits is a parity bit, which is not transferred through the byte converter. A similar *byte converter* is described and claimed in U.S. patent application No. 162,788 filed Dec. 28, 1961 by J. K. Crawford and C. M. Pietras, titled "Data Transfer and Conversion Circuit." Also a "Byte-Converter Error-Check Circuit" of the same joint inventors is used in the subject tape control and is described and claimed in U.S. patent application No. 99,927 filed Mar. 31, 1961.

The byte converter operation includes the data register, a byte counter and a shift control.

The byte counter is represented in detail in FIGURES 74A and B, and 75A and B, its shift control on FIGURES 72A and B, and 73A and B, and data register on FIGURES 12 and 17.

The three stage *byte counter* (FIGURE 150) controls the byte converter operation. The byte counter:

(1) allows the tap control to signal the interface to (a) either request that a byte can be transferred to the tape control, or (b) to indicate that output data lines to the interface have been conditioned for transferring a byte from the tape control to the interface;
(2) generates input and output timing gates to allow the data register to store and unload data from different groupings of the twelve register positions.

The following example illustrates byte counter and data register actions in a write operation. FIGURE 151 diagrams an example of byte converter operation for read and write operations. In a write operation, the byte counter supplies three input timing gate signals ("write set byte **1**," "write set byte **2**," and "write set byte **3**"); each conditions a different group of 8 gates between the 8-bit interface and the 12 triggers compirsing the data register. Also, the byte counter supplies two output timing gate signals ("write shift byte **1** or **3**" and "write shift byte **2** or **4**") to condition groups of 6 gates connected to six data register positions to be unloaded as a 6-bit tape byte. When the interface transfers the first 8-bit byte to the data register, "write set byte **1**" is active, causing the byte to be stored in data register positions **5** through **12**. When the tape control brings up "write shift byte **1** or **3**," the six bits in data register positions **7** through **12** transfer to the read/write register; the tape control then requests another 8-bit byte from the interface. "Write set byte **2**" is active when the interface sends the second 8-bit byte to the data register; the byte is loaded in data register positions **1** through **4** and **9** through **12**. "Write shift byte **2** or **4**" allows six bits in positions **1** through **6** to transfer to the data register, after which the tape control signals the interface for a third 8-bit byte.

"Write set byte **3**" loads the third interface byte in data register positions **1** through **8**; at this time, all data register positions are filled. "Write shift byte **1** or **3**" is conditioned for the second time to transfer six bits in data register positions **7** through **12** to the read/write register. The tape control does not request another byte from the interface because data register positions **1** through **6** are loaded "Write shift byte **2** or **4**" is conditioned for the second time to transfer the last six bits in data register positions **1** through **6** to the read/write register; the data register is now empty and has completed a data register cycle. For every three input interface bytes, the data register and byte counter repeat the operation.

The 3-bit of a read or write command can activate a 6-to-8 bit code translator. The byte converter is not used when the code converter is activated. The translator is shown in detail in FIGURES 21A and B, and 22A and B. The data register and translator (see basic FIGURES 152 and 153) combine to convert either BCD (Binary Coded Decimal) characters to 8-bit code characters in a tape read operation, or 8-bit code characters to BCD characters in a tape write operation. FIGURES 154A and B shows bit configurations for characters in the BDC and the 8-bit code. In six bit mode, each eight-bit character transferred to the data register sets data register triggers **5** through **12** corresponding to bits in the character; triggers **1** through **4** are not used. Each character loaded in the data register must be unloaded before more data can be stored.

To effect the code translation in a write operation, each eight-bit character in the data register transfers through the translator to the read-write register; except bits in data register positions **10** and **12**, which transfer directly to the read/write register to set read/write register **5** and **7** triggers, respectively. The translator comprises gates that operate read/write register triggers **2**, **3**, **4**, and **6**.

In a read operation, a character in the read/write register transfers through the translator to the data register; except the ON states of the read/write register **5** and **7** triggers directly set data register triggers **10** and **12**, respectively. The translator gates operate data register positions **11**, **5** through **9**.

When read or write command bits **3** and **4** do not activate either the byte converter or code translator, the following occurs:

(1) in a write operation, the data register transmits only six of the 8-bits in the interface byte to the read/write register. Bits in data register positions **5** and **6** do not transfer to read/write register positions **0** and **1**. Hence, bits in data register positions **5** and **6** are lost, but the lost bits must be alike so that the character's parity does not change. (Bit positions **0** and **1** in the interface character must both contain bits or be void of bits.
(2) in a read operation the data register transmits 8-bit bytes to the interface, although the read/write register transfers only six bits to the data register. Data register positions **5** and **6** always contain zeros (bit positions **0** and **1** in the byte transferred to the interface are void of bits).

(D) Tape control write operation

In a *tape write operation,* a computer channel transfers one 8-bit byte at a time across the interface lines to tape control. The tape control continues all data transfer from channel and checks each input 8-bit byte for odd parity, discards the check bit, and stores the other bits in the byte in the *data register.* In processing

characters from the data register through the *read-write register* to a selected tape unit, the tape control either:

(1) transfers all bits in each 8-bit byte (9-track operation), or
(2) converts each 8-bit byte to a BCD character (7-track code converter mode), or
(3) converts three input 8-bit bytes to four 6-bit bytes (7-track byte converter mode).

One of these three conditions for a write operation is determined by the third and fourth bit positions in a write command given during the initial selection sequence. In all cases, however, the tape control transfers one byte (either 9 bit or 7 bit) to the tape unit during each write clock cycle. The initial selection sequence was explained in detail in the Channel-I/O Interface section of this application and is also represented by FIGURES 160A, B and C.

The selected tape unit writes each byte received from tape control on magnetic tape; later during the same block write operation, but in a separate checking operation, the tape unit reads each byte written shortly before and returns data to the tape control. The tape control checks each byte from the tape unit for recording errors that might have occurred within the block. Thus, tape control processes channel data to the tape unit to write a data block at the same time that it performs a check operation on data within the block that the tape unit has written earlier.

When channel transfers the last byte in the block, it answers the subsequent request for data from the tape control with an interface line effectively indicating that the last byte has been transmitted. The tape control causes the tape unit to write two more bytes in the block, the CRC and LRC check characters. After the tape unit reads the LRC check character and returns it to tape control, tape unit action in the write operation is ended. The tape control examines the check characters to see that no errors were made, resets circuits employed in the write operation, and transmits a status byte to channel. Channel must accept the status byte before tape control can disconnect from the write operation and execute another tape command.

FIGURE 158 shows data flow paths in the tape control during a write operation. FIGURES 159A, B and C and 161A through 169B show tape control circuit actions in the execution of a tape write operation. The figure number found in most of the blocks within FIGURES 161A–169B has the appropriate detailed circuits which perform the operation designated within the block. The writing of the Cyclic Redundancy Check (CRC) character is not represented in detail in FIGURES 161A–169B because it is explained in detail in other parts of this application.

*Conditioning tape control for write operation.*—During the initial selection sequence, the tape control receives a command byte designating the write operation and establishing the conditions under which the block is to be written. If tape control and the selected tape unit are free to write a tape block, the tape control activates "service-in" (tag line in the interface) to request data from the channel at the end of the initial selection sequence. The Channel responds to the "service in" request by transmitting a data byte on bus-out lines and conditioning the "service out" tag line. The tape control checks the parity of the input byte, then discards the byte's check bit and gates other bits in the byte to data register positions **5** through **12**.

The write command is decoded by AND circuit **66**A–AF, which sets a tape operation trigger **67**–AN, AR that causes the actuation of a write delay gate **112**A–AC to start a write delay.

*Write delay.*—The actuation of the write delay gate causes the following:

(1) sets the go trigger **85**B–AF, AP. The active go trigger output causes the selected tape unit to begin moving tape forward.
(2) conditions the millisecond delay counter drive circuits (**119**–AK). The delay counter (FIGURES 113A and B, 114A and B, and 115) advances in millisecond mode to time out the write delay.

While the "write delay" is active, tape on the selected tape unit accelerates to proper operating speed, but the tape control does not transfer any bytes to the tape unit.

If tape on the selected tape unit is not at load point when the write delay begins, a write condition trigger **104**–AK, AL is turned on when the delay counter advances:

(1) to 24 slow speed tape unit is selected.
(2) to 36 medium speed tape unit is selected.
(3) to 50 high speed tape unit is selected.

If tape on the selected tape unit is at load point when the write delay begins, the load point delay trigger **85**A–AV, AU is set which causes a longer write delay by blocking the set path to the write condition trigger unit until the delay counter advances to 320. This is because tape is not positioned at the same spot when the write head is at the reflective marker on the tape. The longer write delay then causes the tape unit to erase a section of tape approximately 3¾ inches long from a point on the reflective market to the first byte in the first tape block. This erasure ensures that all previously recorded data on tape are destroyed before the first block begins.

*Write condition.*—The write condition trigger's ON output terminates the "write delay" and stops the delay counter to permit actual writing of the tape block. However, between the time that the go trigger is set to start tape movement and the write condition trigger is set, tape on the selected tape unit has reached proper operating speed. In addition to ending the write delay, the on state of the write condition trigger further prepares the tape control to perform the write operation by:

(1) setting the write trigger release trigger **104**–AN, AM to allow write triggers in the selected tape unit to turn on and off as required to record bytes on tape. The write trigger release is reset after the tape control processes the last normal byte in the block to the tape unit, causing all tape unit write triggers in the on state at that time to switch to their off states. In returning to their reset conditions, write triggers record the LRC check (byte) character at the end of the tape block.
(2) setting the read condition trigger **104**–AC, AD when the delay counter advances to 32 during the write delay. The on state of the read condition trigger conditions output gates for the final amplifiers to allow the tape control to check each byte that the tape unit writes.
(3) starting the write clock FIGURES 106A and B by activating gate **105**A and B–CD. Write clock outputs establish timings that control data flow through the tape control and tape drive to tape. The write clock cycles continuously as long as the write condition trigger is set.

*Data flow to tape unit.*—When the write clock advances to 2, it generates the "WC–**2**" output to reset the read-write register (FIGURES 18A and 20B) and set the R–W control trigger **72**A–AR, AX. The R–W control trigger indicates the status of the read-write register; when the trigger is set, the read-write register is empty; when the trigger is off, the read-write register is either being loaded or already contains a byte.

On all other cycles except the first cycle in the write operation, the write clock output "WC–**3** or **4**" samples

the state of the no echo trigger. The no echo trigger's ON output combines with "WC–3 or 4" to set the data check trigger.

At this point in the write clock cycle, signal exchanges between tape control and channel required to transfer a data byte to the tape control are complete, and at least one byte is stored in the data register. The ON output of the R–W control trigger brings up a "start shift" AND gate 72B–AK; "start shift" conditions a "shift data" control on FIGURES 73A and B to initiate a transfer from the data register to the read-write register and to reset the R–W control trigger.

*Data register to read-write register character transfer: 9-track operation.*—During each write clock cycle, the tape control receives one 8-bit byte from channel and transmits the 8-bit byte to the selected tape unit. Each input channel byte sets data register positions **5** through **12**. "Shift data" initiates the direct transfer that causes the data register byte to set read-write register positions **0** through **7**. Because the byte in the data register does not have a check bit, the read-write register P position is not set.

*7-track operation.*—Tape controls equipped with the 7-track feature operate in either byte converter or code translator, or a 6-bit mode, which uses neither the converter or translator. The command byte received from channel during the initial selection sequence designates the mode.

In byte converter mode, the tape control converts three 8-bit bytes from the channel to four 6-bit characters, and transmits one 6-bit character to the tape unit during each write clock cycle. The three-stage byte counter generates input and output gates for the data register. The first input byte to the data register sets positions **5** through **12**; the second input byte sets positions **1** through **4** and **9** through **12**; the third input byte sets positions **1** through **8**. The first and third output bytes from the data register are taken from positions **7** through **12**; the second and fourth output bytes from the data register are unloaded from positions **1** through **6**. Each 6-bit byte transferred from the data register sets read-write register positions **2** through **7**; read-write register positions P, **1**, and **2** are not used in 7-track write operations. The tape control does not request a byte from channel after transferring the third byte from the data register to the read-write register. The byte counter advances after each byte is unloaded from the data register and resets after the fourth data register byte sets the read-write register. Therefore, the states of the byte counter triggers indicate data register positions to be loaded and unloaded and initiate a "service in" request to channel when a data transfer from channel is required.

In code translator mode, the tape control receives one 8-bit code character from channel, converts the character to BCD code, and transmits one BCD character to the tape unit during each write clock cycle; the byte counter is inoperative. All input characters from the channel set data register positions **5** through **12**. Bits in each 8-bit code character unloaded from the data register feed through an 8-bit code to BCD translator, then set read-write register positions **2** through **7**; read-write register positions P, **1**, and **2** are not used in 7-track write operations.

If the command byte did not activate either the byte converter or the code translator, bits in data register positions **7** through **12** transfer to read-write register positions **2** through **7**. Bits stored in data register positions **5** and **6** are discarded.

*Request byte from channel.*—The tape control receives a byte from channel and transmits a byte to the tape unit during each write clock cycle in 9-track operation or in 7-track operation without the byte converter. Each time that bits are unloaded from the data register, the tape control conditions a "tape demand" gate **75**A and B–AB and AC that sets the service in trigger **72**A–AF, AJ and activates the *service in* interface line to the channel to request that another byte be transferred across the interface. In 7-track byte counter mode, byte counter trigger states determine when another byte from channel is required. However, the sequence to condition the "service in" trigger is the same.

The channel responds to the "service in" signal either by transmitting a data byte on bus-out lines and activating the "service-out" line or by conditioning the "command-out" line to indicate that the previous byte transfer loaded the last byte in the block in the data register. A "command-out" response sets the tape control's stop trigger to end the operation when the present write clock cycle ends.

*Read-write register to tape unit byte transfer.*—The command byte designates that bytes recorded on magnetic tape be written in odd or even parity. If the command byte specified odd parity, the odd redundancy trigger is set at the beginning of the write operation. Vertical redundancy check (VRC) circuits examine the bit structure of the byte stored in the read-write register. If a byte in the read-write register contains an odd number of bits and the odd redundancy trigger is off, or if the byte contains an even number of bits and the odd redundancy trigger is set, the tape control adds a check bit (P-bit) to the character transferred to the tape unit.

When the write clock advances to 6, the tape control sets the no echo trigger. The tape unit transmits "echo pulse" to tape control when it records the byte transferred from the read-write register; an "echo pulse" resets the no echo trigger. The tape control must receive an "echo pulse" before WC–3 time of the next write clock cycle (before write disconnect delay (RDD) 60 time if this is the last write clock cycle) or indicate a data check error.

At write clock 12–15 time, the tape control activates a "write pulse" line (FIGURE 99) to the tape unit, causing the tape unit to write the byte transferred from the read-write register. Although output data lines (write bus lines) from tape control to the tape unit are conditioned when the read-write register is set, the tape unit does not accept the byte to record it on tape until "write-pulse" line is conditioned.

*End write clock cycle(s).*—If channel responded to the last "service in" demand for data by transmitting a byte and activating the "service out" line to tape control, the write clock advances to 15, resets, and begins another cycle to transfer the next byte to the tape unit. At least one byte is stored in the data register at this time.

If channel responded to the last "service in" demand for data by conditioning the "command out" line, the stop trigger **54**A–AS, AV is set. The write clock output "WC–**14**" resets the write condition trigger on FIGURE 104; the clock advances to 15, then resets. Because the write condition trigger is off, the clock cannot begin another cycle. The positive shift on the "write condition" line **104**–AL**4** brings up "write disconnect delay" **112**A–AE to condition microsecond delay counter drive circuits **112**A–AT; the delay counter on FIGURES 113A–115 steps in microsecond mode, wherein the delay counter is driven by a high frequency oscillator such as 720 kc. When the delay counter advances to 60 (WDD–**60**), it:

(1) resets the delay counter.
(2) resets the write trigger release trigger, causing the tape unit to write the check character at the end of the record.
(3) checks the state of the no echo trigger. If the no echo trigger is set, the tape control sets the data check trigger. When the tape unit wrote the last normal byte in the tape block, it should have conditioned the "echo pulse" line to tape control to reset the no echo trigger.
(4) resets the go trigger, dropping "write disconnect delay."

(If a tape unit with the 7-track feature is selected, "write disconnect delay" and "not write trigger release" condition the millisecond delay counter drive circuits; and the delay counter steps in millisecond mode. The go trigger resets, dropping "write disconnect delay" when the delay counter advances to 20 for a fast tape unit, or to higher count for a slower speed tape unit.)

The WDD–**60** delay causes the tape unit to space the check bytes further from the last data byte in the block than normal byte spacing controlled by the write clock. The delay after WDD–**60** allows the selected tape unit to move tape at full speed for a predetermined time after writing the check bytes before resetting the "go trigger" to the tape unit. This delay compensates for the different speeds at which various tape unit models move tape with regard to the distance between the write and read heads, since the read heads must check all the data in a newly written block.

*Read check of write operation.*—The selected tape unit writes bytes with the write head. As tape moves from the write head and passes the read head, the tape unit reads a previously recorded byte and transfers it to the tape control to be checked. The tape control examines each normal input byte from the tape unit to determine:

(1) whether the data written on magnetic tape is of sufficient strength to be read in subsequent read operations.
(2) whether each byte contains an odd or even number of bits and whether the bit count matches the state of the odd redundancy trigger.
(3) whether spacing errors (skew errors) occurred in recording the byte.
(4) the correct bit structure for the block's check bytes. When the tape unit transfers the check bytes to the tape control, the bytes must meet the specifications that the tape control (LRCR and CRCR circuits) has calculated.

Thus the tape control simultaneously executes operations to transfer bytes to the tape unit and receive bytes from the tape unit.

Each input byte from the tape unit enters final amplifiers in the tape control. The final amplifiers are nine (for 9-track tape units) or seven (for 7-track tape units) tracks, one for each used track position. Each final amplifier has two outputs, a hi-clip and a lo-clip output. Input bits to each final amplifier must meet predetermined minimum amplitude requirements to produce hi- and lo-clip outputs. The hi-clip output from each final amplifier sets the corresponding hi-clip skew register; the lo-clip output from each final amplifier sets the corresponding lo-clip skew register. Theoretically, all bits in a byte should arrive at the input to the final amplifiers simultaneously; however, all bits need not arrive at exactly the same time. The first hi-clip output from any final amplifier sets a first bit trigger **36**–AK, AL to operate a read clock control circuit on FIGURES 108A and B and start the read clock on FIGURES 109A and B.

The skew gate trigger turns on when the read clock advances to 4, 5 or 6 depending on the speed and bit density of the selected tape unit. All bits in the byte must be stored in skew registers before RC–**4**, –**5**, or –**6** time, because any hi-clip output from the final amplifiers after the skew gate trigger is set causes tape control to indicate a data check error.

Vertical redundancy check circuits examine the byte stored in the hi-clip skew register for parity error. If the hi-clip skew register byte contains an odd number of bits and the odd redundancy trigger is off, or if the byte contains an even number of bits and the odd redundancy trigger is set, the tape control:

(1) sets the hi-clip VRC and data check trigger at RC–**7** time.
(2) unconditionally gates the byte in the lo-clip skew register to the longitudinal redundancy check register (LRCR).

If the hi-clip skew register byte does not contain a parity error, the tape control gates that byte to the LRCR. Even though the byte in either the hi- or lo-clip skew register sets the LRCR, skew registers are not reset at this time.

The LRCR contains seven (tape controls equipped with the 7-track feature) or nine (tape controls without the 7-track feature) binary triggers, one corresponding to each bit position in the byte. Bits in input bytes to the LRCR cause corresponding LRCR binary triggers to switch to their opposite states. Tape control checks the LRCR at the end of the operation.

Recall that the tape control processes a byte from channel to be written on tape at the same time that it checks a byte written on tape earlier in the operation, The tape control completes actions involved in transferring bytes from the channel to the tape unit before it completes the read check operation. If the go trigger is reset when the read clock advances to 7, indicating that the tape unit has written the block's LRC check character, the tape control sets the read disconnect delay (RDD) trigger **112**B–AM,AN. "RDD" conditions microsecond delay counter drive circuits on FIGURES 117A and B, and the delay counter advances in microsecond mode. The ON state of the RDD trigger does not signal that this is the last read clock cycle, nor even the next to the last read clock cycle; it merely warns that only a few bytes in the block are left to be read and transferred to tape control, due to the spacing between the write and read head gaps.

Because different final amplifier clipping circuits set hi- and lo-clip skew registers, both registers might not contain the same bit configuration for the same byte. The tape control compares data stored in both skew registers; if bits stored in the hi- and lo-clip skew registers do not represent the same character, the tape control sets the compare and data check triggers when the read clock advances to 8.

The read clock output at RC–**10** time resets the first bit trigger and hi- and lo-clip skew registers. The first bit trigger's OFF output resets the read clock and the skew gate trigger.

If the RDD trigger was not set at RC–**7** time of the read clock cycle, no further read check actions occur until the tape unit transfers the next byte to final amplifiers in the tape control; the tape control does not establish a time for the byte to arrive. If the RDD trigger was set during the previous read clock cycle, the next byte from the tape unit should produce a hi-clip output from the final amplifiers before the delay counter advances to 36. The first hi-clip output from the final amplifiers sets the first bit trigger to start the read clock. The read clock output at RC–**0** time resets the RDD trigger, causing the delay counter to reset. The read clock output at RC–**7** time sets the RDD trigger starting the delay counter in microsecond mode.

Because any check character (byte) is spaced further from the last normal byte in the block than normal byte spacing, the delay counter advances to 36 after tape control checks the last normal byte in the record. "RDD–**36**" sets the check character trigger, indicating that the next character from the tape unit is the block's check character.

The first hi-clip output from the final amplifiers after the delay counter advances to 96 sets the first bit trigger to start the read clock. The tape block might not have an LRC check character (theoretically, this means a check character with no one bits), in which case, the delay counter output at RDD–**128** time initiates the LRCR check and actions to terminate the write operation.

Because the check character trigger is set, the read clock output at RC–0 time does not reset the RDD trigger, and the delay counter runs throughout the check character cycle. Check character bits cause the hi- and lo-clip outputs from the final amplifier tracks that set corresponding hi- and lo-clip skew registers. The tape control examines the check character bits stored in the hi-clip skew register for a parity error, but the state of the odd redundancy trigger does not unconditionally indicate whether the check character should contain an odd or even number of bits. Each input character from the tape unit before the check character trigger is set causes the binary odd-even character count trigger 47A–BL to change states. Therefore, if the block contains an even number of normal characters, the last normal character in the block resets the trigger; if the block contains an odd number of normal characters, the last normal character in the block sets the trigger. If the odd-even character count trigger is off when tape control receives the check character, the check character should contain an even number of bits. If the odd-even character count trigger is set when tape control receives the check character, the state of the odd redundancy trigger determines whether the check character should contain an odd or even number of bits.

If a parity error is detected in the check character in the hi-clip skew register, the tape control sets the hi-clip VRC and data check triggers and gates the character in the lo-clip skew register to the LRCR. If no error is found in the character in the hi-clip skew register, tape control gates that character to the LRCR.

Tape control compares check character bits stored in hi- and lo-clip skew registers. If bits in each register do not represent the same character, the read clock output at RC–8 time sets the compare and data check triggers.

The read clock output at RC–10 time resets the first bit trigger and skew registers. The first bit trigger's OFF output resets the read clock.

When the delay counter advances to 128 (recall that delay counter ran throughout check character cycle), the tape control resets the read condition trigger and checks the LRCR. When check character bits transferred to the LRCR, all LRCR triggers on at that time should have switched to their off states. If any LRCR trigger is set at RDD–128 time, the tape control sets LRCR error and data check triggers.

The delay counter advances to 144 and sets the RDD–144 trigger. The RDD–144 trigger's ON output activates "RDD–144 reset" to reset the delay counter and other tape control circuits activate in the write operation. "RDD–144 reset" also initiates actions to cause the tape control to transmit a status byte to the channel. The channel cannot begin another tape operation until it accepts the status data and responds with a "service out" interface tag signal.

### (E.) Tape control read operation without cyclic error correction

In a read operation, the tape control receives 6 or 8 bit bytes from the tape unit and transmits odd parity 8-bit bytes to channel. The tape control initiates all block transfers to channel, but it must take byte transfers from the tape unit as they are read. In processing a tape byte, the tape control strips the bytes parity bit before storing bits in the data register. Bytes transferred across the interface must contain an odd number of bits; therefore, tape control adds a P-bit to each byte unloaded from the data register that contains an even number of bits.

In processing tape data to channel, the tape control either:

(1) transfers to the channel all bits in each 8-bit byte received from the tape unit (9-track operation);
(2) converts each input BCD character to an 8-bit code character and transfers the 8-bit byte to channel (7-track code translator mode);
(3) converts four input 6-bit bytes to three 8-bit bytes for transmission to channel (7-track byte-converter mode);

A command byte received from a channel during the initial selection sequence designates the manner in which a tape control is to process data in the read operation.

The channel is not required to accept all bytes in a tape block. After accepting the first byte in the block, the channel can terminate output transfers from the tape control at the end of any cycle. However, the tape control and the tape unit are committed to the operation until the tape unit reads the complete block. If the tape unit has not transferred the last normal byte in the block when channel indicates that it will not accept more data bytes in the operation, the tape control does not initiate more data transfers to channel, but it receives, checks, then discards subsequent bytes received from the tape unit.

If the first byte transferred to tape control has the Tape Mark bit configuration and the second character in the block is not the check character, the tape control processes two characters to the data register on the same cycle.

At the end of the operation, the tape control transmits a status byte to channel. If the block was a Tape Mark record, the *exceptional condition* bit position in the status byte contains a "1." The channel must accept the status byte before tape control can disconnect from the read operation and execute another tape command.

FIGURE 170 shows data flow paths in the tape control in a read operation. FIGURES 171A through 182B show tape control circuit actions available in the execution of a tape read operation. Condition Tape Control for Read Operation: During the initial selection sequence, the tape control receives a command byte designating a read operation and establishing conditions under which a block is to be read and processed. If tape control and the selected tape unit are free to read a tape block, the tape control responds to the command by activating a "read delay" gate 112A–AA at the end of the initial selection sequence. The "read delay" gate output:

(1) sets the go trigger 85B–AF, AP. The active go trigger output causes the selected tape unit to move tape forward.
(2) Conditions millisecond delay counter drive circuits on FIGURE 119. The delay counter on FIGURES 113A–115 advances in millisecond mode at a rate determined by the speed and density of the tape unit.

While the "read delay" gate is active, tape on the selected tape unit accelerates to proper operating speed; because the read head on the tape unit was positioned either in an interrecord gap or in the gap between load point and the first tape record when the operation began, bytes do not transfer to the tape control immediately after tape movement begins.

If tape on the selected tape unit was not at load point when the read delay started, a read condition trigger 104–AC, AD turns on when the delay counter advances at a count 44 for a high speed tape unit and at a lower count for slower speed units. If tape on the selected tape unit is at load point when the read delay begins, a load point delay trigger 85A–AV, AU is set, blocking the set path to the read condition trigger until the delay counter advances to 160.

Regardless of the original position of tape on the selected tape unit, read delays are shorter than corresponding write delays to ensure that read circuits are conditioned soon enough to read the first byte in the tape block.

The read condition trigger's ON output drops the "read delay" gate, stopping the delay counter, and activates the final amplifier's output gate. With the output path from the final amplifiers cleared, the tape control can accept bytes from the tape unit and process the data to the chan-

nel. Between the times that the tape control sets the go and read condition triggers, tape accelerates to proper operating speed.

Each input byte from the tape unit enters the tape control's final amplifiers. A separate final amplifier exists for each used tape track. Each final amplifier has two outputs, a hi- and lo-clip output. Input bits to a final amplifier must meet predetermined minimum amplitude requirements to produce hi- and lo-clip outputs. The hi-clip output from each final amplifier sets the corresponding hi-clip skew register trigger; the lo-clip output from each final amplifier track sets the corresponding lo-clip skew register trigger. Theoretically, all bits in a byte should arrive at the input to the final amplifiers simultaneously; however, all bits need not arrive at exactly the same time. The first hi-clip output from any final amplifier track sets the first bit trigger **36**–AK, AL to start the read clock on FIGURES 109A and B, and begin a read clock cycle.

*Normal read clock cycle.*—(First character from tape unit does not have bit configuration of Tape Mark.) The read clock output "RC–**6**" resets the read/write register. Read clock output "RC–**7**" performs two functions:

(1) "RC–**7**" sets the read disconnect delay (RDD) trigger **112**B–AN, AM. "RDD" conditions the microsecond delay counter drive circuits on FIGURES 117A and B, and the delay counter on FIGURES 113A–115 steps in the microsecond mode at the rate determined by the selected tape unit. If the next byte from the tape unit is not the check character, the read clock output at RC–**0** time resets the RDD trigger, stopping the delay counter. The next tape byte must arrive at tape control soon enough to start the read clock before the delay counter advances to **36**, or tape control will process the byte as the check character.

(2) "RC–**7**" checks the byte in the hi-clip skew register for a parity error. The status of the odd redundancy trigger (fixed during the initial selection sequence) designates whether bytes in the tape block should contain an even or odd number of bits. If an error is detected in the byte in the hi-clip skew register, tape control sets the hi-clip VRC trigger and gates the byte in the lo-clip skew register to the LRCR and read/write register. If no error is found in the byte in the hi-clip skew register, tape control gates that byte to LRCR and read/write register.

The tape control compares the bits stored in the hi- and lo-clip skew registers; both registers should contain bits representing the same character. After the read clock activates "RC–**7**," it conditions "compare sample" circuit on FIGURES 109A and B activate the compare gate on FIGURE 45 AP if the skew registers do not contain identical characters.

After RC–**7** time, the read clock generates "RC reset" to:

(1) reset the first character trigger (only at the end of the first read clock cycle);

(2) reset the first bit trigger, blocking drive pulses to the read clock; the clock resets and cannot start again until the next character from the tape unit sets the first bit trigger;

(3) resets the hi- and lo-clip skew registers;

(4) set the R-W control trigger **72**A–AR, AX. The ON status of the R-W control trigger indicates that the read/write register contains a byte of data and that actions to unload the read/write register have not begun.

The R-W control trigger's ON output brings up "start shift" gate **72**B–AK. The "start shift" gate conditions a "shift data" latch **73**A and B–AS, AU to:

(1) reset the R-W control trigger.

(2) check the byte in the read/write register for a parity error. If the read/write register byte contains an even number of bits and the odd redundancy trigger is set, or if the byte contains an odd number of bits and the odd redundancy trigger is off, tape control sets the R-W VRC and data check triggers.

(3) initiate a byte transfer from the read/write register to the data register.

*Normal read/write register to data register byte transfers.*—(First character from tape unit does not have bit configuration of Tape Mark) *9 track operation.*—During each read clock cycle, an 8-bit code character sets read/write register positions P and **0** through **7**. At the end of the clock cycle, "shift data" causes the tape control to transfer bits in read/write register positions **0** through **7** to data register positions **5** through **12**. Because the data register does not store parity bits, the tape control does not gate the output of the read/write register P-position. The transfer from the read/write register to the data register is unconditional and direct; read/write register 7-position sets data register 12-position.

7-track operation: The command byte, received from channel during initial selection sequence, designates whether the byte converter, code translator or neither shall be used.

If the byte converter is used, the tape control transforms four 6-bit bytes to three 8-bit bytes. The three-stage byte counter controls the transformation by generating input and output gates for the data register. The first and third input characters to the data register set positions **7** through **12**; the second and fourth input characters set positions **1** through **6**. Therefore, bits stored in read/write register position **7** set data register position **12** or **6** depending on the status of the byte counter. Read/write register positions **0** and **1** are not used in byte conversion operations, and read/write register position P output does not set a data register position. The byte counter advances after each operation to load the data register.

If the code translator is used, the tape control converts each 6-bit BCD character stored in the read/write register to an 8-bit code character and transfers the 8-bit code character to data register positions **5** through **12**. Bits in the BCD character in the read/write register feed through the BCD to 8-bit code translator, then set data register positions **5** through **12**.

If the command byte did not activate the translator or converter, bits in read/write register positions **0** through **7** set data register positions **5** through **12**. Because read/write register positions **0** and **1** are not set in 7-track operation, data register positions **5** and **6** always contain zeros; and only data register positions **7** through **12** may contain bits. The byte gated from the data register to the channel is unloaded from positions **5** through **12**.

*Byte transfer from data register to channel.*—This section summarizes the read command sequence explained in detail in the preceding interface section. Without byte converter operation with 7 track tape, each shift data signal from latch **73**A and B–AS, AT conditions the tape demand gate **75**B–AC to set the service-in trigger **72**A–AF, AJ in order to initiate a byte transfer from data register to channel. Because the shift data latch is not activated either when the first character tape mark trigger is set or after the channel indicates that it will not accept more bytes from tape control, byte transfers to channel are not executed when one of these conditions exist. In all other cases, however, the tape control unloads an output byte from data register positions **5** thru **12** after each read clock cycle.

With byte converter operation, the byte counter controls the tape demand gate. "Tape demand" is conditioned only when the byte counter indicates that at least eight data register positions are loaded. For every four input 6-bit bytes to the data register, the byte counter conditions "tape demand" three times (after the second, third, and fourth 6-bit bytes are stored in data register

positions). The byte counter causes tape control to unload the first output byte from data register positions 5 through 12, the second byte from positions 1 through 4 and 8 through 12, the third byte from positions 1 through 7.

Because bytes transferred across the interface must be in odd parity, the tape control checks the bit count in each byte unloaded from the data register and adds a P-bit to the byte when necessary to make the parity of the output data odd.

Tape demand sets the service-in trigger. The ON status of the service-in trigger allows the tape control to transmit a data byte on bus-in lines and conditions the service in tag line to the channel 500 nanoseconds after the shift data latch is conditioned. Because the shift data latch also initiates a byte transfer from the read/write register to the data register, the 500 nanoseconds delay allows the tape control to complete the read/write register to data register transfer before signalling the channel.

The channel responds to the data byte and service-in signal with either a service-out signal (indicating that it has accepted the byte and will accept at least one more byte from tape control) or a command-out signal (indicating that it has accepted the byte but will not accept another data byte in the read operation). The command out signal sets the tape control's stop trigger 54A–AS, AV, holding the shift data latch inactive until the read operation is complete. Because the tape unit must read the complete block, the tape control accepts all bytes in the block from the tape unit and executes a read clock cycle for each input byte regardless of the stop trigger status. If the stop trigger is set, however, tape control does not transfer bytes to the read/write register or data register, and consequently, it does not attempt to transmit bytes across the interface to channel.

*Check character cycle.*—In a normal read clock cycle, the tape control sets the RDD trigger 112B–AM, AN at RC–7 time, allowing the delay counter on FIGURES 113A–115 to advance in microsecond mode; at RC–0 time of the following normal read clock cycle, the tape control resets the RDD trigger, stopping the delay counter. Normal character spacing in the tape block allows the tape unit to transfer at least one bit in a byte to the tape control to start the read clock and stop the delay counter before the counter advances to 36 to set the check character trigger.

Because the check character is spaced further from the last normal character (byte) in the tape block than normal character (byte) spacing, the delay counter advances to 36 after tape control processes the block's last normal character. Delay counter pulse RDD–36 sets the:

(1) check character trigger, indicating that the next byte that the tape unit transfers is the block's check character;

(2) go trigger, causing tape on the selected tape unit to coast to a stop.

The first hi-clip output from the final amplifiers after the delay counter advances to 50 sets the first bit trigger to start the read clock to read the CRC check character and then the LRC check character. The tape block might not have an LRC check character (theoretically, the check character has no one bits) in which case, the delay counter output at RDD–128 time initiates the LRCR check and actions to terminate the write operation.

Because the check character trigger is set, the read clock output at RC–0 time does not reset the RDD trigger, and the delay counter runs throughout the check character cycle. Check character bits cause hi- and lo-clip outputs from the final amplifier tracts that set corresponding hi- and lo-clip skew registers. The tape control examines check character bits stored in the hi-clip skew register for a parity error, but the state of the odd redundancy trigger does not unconditionally indicate whether the check character should contain an odd or even number of bits. Each input byte from the tape unit before the check character trigger is set causes the binary odd-even character count trigger to change states. Therefore, if the block contains an even number of normal bytes, the last normal byte in the block resets the trigger; if the block contains an odd number of normal bytes, the last normal character (byte) in the tape block sets the trigger. If the odd-even character count trigger is off when tape control receives the LRC check character, the check character should contain an even number of bits. If the odd-even character count trigger is set when tape control receives the LRC check character, the state of the odd redundancy trigger determines whether the check character should contain an odd or even number of bits.

If a parity error is detected in the check character in the hi-clip skew register, the tape control sets the hi-clip VRC trigger and gates the character in the lo-clip skew register to the LRCR. If no error is found in the character in the hi-clip skew register, the tape control gates that character to the LRCR. In either case, the tape control does not transfer the LRC check character to the read/write register.

The tape control compares check character bits stored in hi- and lo-clip skew registers. If bits in each register do not represent the same character, the read clock output "compare sample" causes the tape control to set the compare trigger.

At RC reset time, the tape control resets the first bit trigger and skew registers. The first bit trigger's OFF output resets the read clock. Because the LRC check character is the last character (byte) the tape unit transfers to the tape control, the read clock does not run again in the read operation.

When the delay counter advances to 128 (recall that the delay counter ran throughout the check character cycle), the tape control resets the read condition trigger and checks the LRCR. When check character bits are all transferred to the LRCR, all LRCR triggers on at that time should have switched to their off states. If any LRCR triggers are set at RDD–128 time, tape control sets the LRCR error and data check triggers.

The delay counter advances to 144 and sets the RDD–144 trigger. The RDD–144 trigger's ON output activates "RDD–144 reset" to reset the delay counter and other tape control circuits active in the read operation. "RDD–144 reset" also initiates actions to cause the tape control to transmit a status byte to the channel. The channel cannot begin another tape operation until it accepts the status data and responds with "service out."

*First character tape mark.*—The first character transferred to the tape control in a read operation causes the final amplifiers to produce hi- and lo-clip outputs that set corresponding hi and lo-clip skew register positions. The first hi-clip output from any final amplifier sets the first bit trigger, causing the read clock to start.

A first character trigger 37A–AA, AC is set during the initial selection sequence, and it is on during the first read clock cycle in the operation, allowing a Tape Mark decoding gate 37A–AF at read clock output RC–6 time to check the character (byte) in the hi-clip skew register for the tape mark bit configuration (1-, 2-, 4-, and 8-bits in BCD and 1-, 2-, 3-, 4-, 5-, 6-, and 7-bits in 8-bit code). If the character in the hi-clip skew register has the bit structure of a Tape Mark, the tape control sets a first character tape mark trigger 37A–AG, AL to determine if the block being read is a tape mark block (in which case no characters are transferred to channel) or if the character is the first in a normal block (in which case the character with the tape mark bit configuration is transferred to channel). Because a tape mark block contains only a tape mark character followed by an LRC check character, the tape control holds the first character in the read/write register until it receives the next character from the selected tape unit.

At RC–7 time in the first read clock cycle, the tape control:

(1) sets the RDD trigger. "RDD" conditions microsecond delay counter drive circuits, and the delay counter steps in the microsecond mode at the rate determined by the tape unit reading the record.
(2) checks the character in the hi-clip skew register for a parity error. The hi-clip skew register possibly contains a character with the structure of a tape mark as the result of an error. A 7-track tape mark is always an even parity character; a 9-track tape mark is always an odd parity character. If the block that the tape unit is reading is a tape mark block, the odd redundancy trigger should be set in 9-track operation, or reset in 7-track operation, to prevent tape control from setting the hi-clip VRC trigger. If a parity error is detected in the character in the hi-clip skew register, tape control gates the character in the lo-clip skew register to the read/write register and LRCR and sets the hi-clip VRC trigger. If no error is found in the character in the hi-clip skew register, tape control gates that character to the read/write register and LRCR.

The tape control compares the characters stored in the hi- and lo-clip skew registers; both registers should contain bits representing the tape mark character. If, however, the characters in the registers are not identical, the tape control uses the read clock output "compare sample" to set the compare trigger.

At RC reset time, the tape control:

(1) resets the first character trigger;
(2) resets the first bit trigger, causing the read clock to reset;
(3) resets the hi-and lo-clip skew registers;
(4) sets the R-W control trigger.

Because the first character tape mark trigger is set, the R-W control trigger's ON output does not condition "start shift" to initiate a read/write register to data register character transfer.

If the delay counter advances to 36 before the next tape character produces a hi-clip output from final amplifiers, the tape control sets the check character trigger and recognizes the tape block as a tape mark record. The tape mark character stored in the read/write register on the previous read clock cycle does not transfer. At RDD–36 time, the tape control resets the go trigger, causing tape on the selected tape unit to coast to a stop. When the delay counter advances to 64, it generates "DC–64" to set the exceptional condition trigger. The ON status of the exceptional condition trigger causes the tape control to transmit a "1" in the exceptional condition bit position in the status byte at the end of the read operation. At RDD–96 time, the tape control conditions "RDD–96 gate" to allow any hi-clip check character bit to start the read clock.

If the second character in the tape block arrives at the final amplifiers and causes the read clock to start before the delay counter advances to 36, the block is not a tape mark record. Although the first character in the block (now stored in the read/write register) has the bit configuration of a tape mark, the tape control must transfer that character to data register positions and, if this is a 7- track read operation, to channel during this read clock cycle. The tape control must also check and transfer the second character in the block (now in the skew registers) to LRCR and through the read/write and data registers to the channel before the tape unit transfers the next character to the tape control. (If this is a 9-track read operation, the tape control transfers only one byte to the channel during this character cycle. A character cycle extends from the beginning on one read clock cycle to the beginning of the succeeding read clock cycle. If this is a 7-track read operation, the tape control transfers two bytes to channel during this character cycle.) The tape control resets the first character tape mark trigger as quickly as it detects that the block is not a tape mark record, and begins actions to transfer the first character from the read/write register at RC–0 time. Before the read clock advances to 6 and conditions "RC–6 not write," the tape control:

(1) transfers the first character to the channel and receives the "service-out" reply in 7-track mode;
(2) transfers the first character to the data register in 9-track mode.
(3) resets the R-W control trigger.

The tape control processes the second and succeeding characters (bytes) in the tape block at normal times. The condition in which the first character (byte) in the record has the bit configuration of a tape mark but is not followed by a check character is the only case that requires tape control to execute two read/write register to data register character transfers in the same character cycle. Refer to FIGURES 174 A, B and C for details of this unique transfer.

(F) Tape motion control without data transfer

Tape motion control for a tape unit is determined by any of the eight channel commands designated in the Control Format part of a table titled Command-Control Code found in the Interface section of this application. The channel control command has a bit in positions **5, 6** and **7**. Bit combinations in positions **2, 3,** and **4** in the command byte designate a particular one of eight motion control operations. Decoding of motion control commands is done by circuits on FIGURES 67 and 68 A and B. The motion control commands are decoded in the tape control to designate one of eight tape operations:

(1) Erase
(2) Rewind
(3) Rewind Unload
(4) Write Tape Mark
(5) Backspace Record
(6) Backspace File
(7) Forward Space Record
(8) Forward Space File

No motion control operation requires data byte transfers between the channel and tape control, and only the write tape mark command causes the selected tape unit to write characters on magnetic tape, which is a tape mark character generated in the tape control and not transferred as a byte from the channel.

When the tape control completes the operation, it transmits a status byte to the channel. In all other motion control operations except rewind and rewind unload, the tape control actions in the operation end after tape unit functions are complete. Therefore, the status byte that tape control transmits after each of these operations indicates that both the tape unit and tape control are free to execute another tape command. In rewind and rewind unload operations, however, tape control disconnects from the operation before the selected tape unit rewinds tape to load point. In these operations, the tape control transmits a status byte when it completes actions in the operation and another status byte after the tape unit completes actions in the operation. The first status byte contains a "0" in the status modifier bit position, indicating that the tape unit is not free, and that the tape control can perform another operation. However, the second status byte contains a "1" in the status modifier bit position, indicating that the tape unit has completed the rewind or rewind unload operation, and that it can thereafter perform another operation.

FIGURES 183A–189C show detailed tape control actions in the execution of each motion control operation.

*Erase.*—Execution of an erase command conditions an erase tape gate **68**A–AP that causes the selected tape unit to erase approximately 3¾ inches of tape. The tape control starts the erase and write operations in a similar manner; both operations condition a write delay gate **112**A–AC to:

(1) set the go trigger **85**B–AF, AP. The active go trigger output causes the selected tape unit to move tape forward.
(2) condition millisecond delay counter drive circuits. The delay counter advances in the millisecond mode at a rate determined by the selected tape unit.

The first character trigger **37**A–AA, AC (set during the initial selection sequence) is reset by a write delay gate **37**A–AE when the delay counter advances to 320. The delay counter output at WD–320 time also conditions the DC reset latch **113**B–BG to reset the delay counter. The first character trigger's OFF output and erase set the RDD trigger, **112**A and B–AM, AN which conditions the microsecond delay counter drive circuits and the delay counter advances in the microsecond mode. The delay counter output at RDD–36 time sets the check character trigger. The trigger's ON output resets the go trigger, causing tape on the selected tape unit to coast to a stop. The delay counter advances to 144 and sets the RDD–**144** trigger **120**–AB, AC to activate the RDD–**144** reset line. RDD–**144** reset resets the erase operation and initiates actions to transmit a status byte to the channel.

*Rewind and rewind unload.*—Execution of a rewind command or a rewind unload command causes the selected tape unit to move tape backward to load point. At the completion of the rewind operation, the selected tape unit returns to ready status with the tape loaded in the columns and can perform another operation without manual intervention. At the completion of the rewind unload operation, tape on the selected tape unit is not loaded in the columns, and the head assembly is not positioned to write or read tape; a load operation is required to condition the tape unit for ready status.

If the load point delay trigger is set when tape control begins execution of a rewind or a rewind unload command, it conditions a backward disconnect gate **67**–AD to start the disconnect operation immediately. Because it is never necessary to move tape backward beyond load point, the tape control does not condition the go trigger to the selected tape unit to start tape movement, if initially at load point.

If the load point delay trigger is not set when tape control begins the rewind or rewind unload operation, it activates a backward *op* gate **68**A–AA to condition the millisecond delay counter drive circuits; and the delay counter steps in the millisecond mode. The tape unit can move tape backward only in read status. If the tape unit is not in read status, the backward *op* gate also sets the go trigger, causing the selected tape unit to move tape forward. The delay counter generates an output to reset the go trigger when the counter advances to 50 for a fast tape unit, or a lesser count for slower tape units. Thus the tape control causes the tape unit to move tape forward to erase a gap on tape before rewinding begins, and to deposit away from the end of a written block a noise pulse generated by turning off write current in changing to read status. If the tape unit is in read status when the operation begins, backward op does not set the go trigger to move tape forward.

When the delay counter advances to 96, the tape control sets the backward trigger **85**A–AS, AT to condition backward to the selected tape unit; if the tape unit is not in read status, it changes to read status immediately. At D–**160** time, the delay counter produces an output to set the go trigger. Because the backward trigger is set, the tape unit moves tape backward.

A delay counter output sets the read condition trigger when the counter advances to 170 for a high speed tape unit, or a higher count for a lower speed tape unit. The read condition trigger initiates actions to reset the delay counter.

The command activates a rewind gate **68**B–AE or a rewind unload gate **68**B–AK to the selected tape unit. The rewind gate allows the tape unit to return to ready status when it moves tape to load point. The rewind unload gate causes the tape unit to end the operation with the head assembly open and the tape unloaded from the vacuum columns.

When the tape unit begins to move tape backward, it conditions backward disconnect gate **67**–AD to the tape control to set the RDD–**144** trigger **120**–AB, AC. The trigger's ON output allows the tape control to disconnect, while the tape unit is executing the rewind or rewind unload operation. Upon disconnection, the tape control transmits a status byte to the channel containing a "0" in the unit freed bit position, indicating that the tape unit is not free to be selected again, but the tape control is available to the channel to perform another operation while a tape unit is rewinding, or rewinding and unloading. When the channel accepts the status byte, it can initiate a tape operation on any other tape unit attached to the tape control. When the tape unit performing the rewind or rewind unload operation senses load point, it causes the tape control to transmit another status byte to channel; this status byte differs from the status byte that the tape control transferred when the tape unit first began to move tape backward in that the second status byte contains a "1" in the unit freed bit position to indicate that the tape unit has completed rewinding to load, or has rewound and unloaded.

*Write tape mark.*—In a write tape mark operation, the tape control generates a tape mark character internally, transfers the character to the selected tape unit, then resets the tape unit's write triggers to write an LRC character and complete a tape mark block. A CRC check character is not written in a tape mark record, and CRC circuits are blocked. After writing the write tape mark and check characters, the tape unit continues moving tape to read check the newly written characters, then conditions data lines to the tape control. The tape control parity checks the newly written characters and ends the operation. The tape control actions in writing a tape mark are similar to the actions that it performs in the execution of a normal write operation. In a write tape mark operation, however, communication between channel and tape control after the initial selection sequence is required only in the status byte transfer; data are not processed across interface lines in executing a write tape mark operation.

After the initial selection sequence, the write tape mark command conditions write tape mark gate **68**A–AM and write delay gate **112**A–AC. The active write tape mark gate:

(1) sets the load point delay trigger **85**A–AU, AV. The load point delay trigger's ON output allows the selected tape unit to erase approximately 3¾ inches of tape before the tape control conditions data lines to the tape unit.
(2) unconditionally sets the read-write register **4**, **5**, **6** and **7** triggers.
(3) conditions 7-track write tape mark if tape control is equipped with the 7-track feature. Because a 7-track tape mark is an even parity character (1-, 2-, 4-, and 8-bits), 7-track write tape mark resets the odd redundancy trigger
(4) conditions 9-track write tape mark if tape control is not equipped with the 7-track feature. Because the 9-track tape mark has the bit configuration 1, 2,

3, 4, 5, 6 and 7, 9-track write tape mark also sets read-write register **1**, **2** and **3** triggers.
The active write delay gate:

(1) sets the go trigger, causing tape on the selected tape unit to move forward.
(2) conditions millisecond delay counter drive circuits; the delay counter advances in millisecond mode.

When the delay counter steps to 320, it generates an output to set the write condition trigger **105**A–AG. The active write condition trigger output:

(1) drops the write delay gate output, resetting the delay counter. The CRC character is written before the delay counter is reset.
(2) sets the write trigger release trigger **104**–AM, AN. When tape control resets the write trigger release trigger, the tape unit writes the LRC check character.
(3) starts the write clock on FIGURES 105A and B.
(4) sets the read condition trigger **104**–AC, AD to allow tape control to accept data from the tape unit.
(5) conditions write tape mark set stop gate **68**A–BD to set the stop trigger **54**A–AS, AV and limits the operation to two write clock cycles.

The tape control transfers the tape mark character in the read-write register to the selected tape unit and conditions the write pulse gate to the tape unit at WC–12 time; and the tape unit writes the character. Because the stop trigger is set, the write clock output at WC–14 time resets the write condition trigger to prevent the clock from advancing through another cycle.

Tape control actions after WC–5 time in a write tape mark operation are identical to the actions that it performs after WC–5 time in the last write clock cycle in a normal write operation. Tape control circuits that read check tape block, end the operation, and transmit the status byte, function in the same manner in write tape mark, as in a normal block write operation.

*Backspace record and backspace file.*—In a back space record operation, the selected tape unit moves tape backward through one block to the interrecord gap, or to the gap between load point and the first tape block. In a backspace file operation, the selected tape unit moves tape backward to the first interrecord gap after sensing a tape mark record, or to the gap between load point and the first tape record. The tape unit reads bytes recorded on tape and conditions data lines to the tape control. The tape control does not execute data transfers either to the channel or to the tape unit when performing the backspace record or backspace file operation.

If the load point delay trigger is set when the tape control begins execution of a backspace or backspace file command, it conditions:

(1) a backspace or backspace file at load point gate **67**–AB to set the intervention required trigger. The intervention required trigger's ON output causes tape control to store a bit in the intervention required bit position in the status byte transferred to channel at the end of the operation.
(2) the backward disconnect gate on FIGURE 67 to set the RDD–**144** trigger, initiating the disconnect operation.

Because it is never necessary to move tape backward beyond load point, the tape control does not condition the go trigger to the selected tape unit to start tape movement if the load point delay trigger is set.

If the load point delay trigger is not set when tape control begins the backspace record or backspace file operation, it activates the backward *op* gate **68**A–AA to condition the millisecond delay counter drive circuits; and the delay counter advances in the millisecond mode. The tape unit can move tape backward only in read status. If the tape unit is not in read status, the backward *op* gate also sets the go trigger, causing the selected tape unit to move tape forward. The delay counter generates an output to reset the go trigger when the counter advances to 50 for a high speed tape unit, or to a lesser count for a lower speed tape unit.

The tape control causes the tape unit to move tape forward to erase a gap on tape before rewinding begins, and to deposit away from a block the noise pulse generated in changing to read status. If the tape unit is in read status when the operation begins, the go trigger is not set to move tape forward.

When the delay counter advances to 96, the tape control sets the backward trigger to condition the backward line to the selected tape unit; if the tape unit is not in read status, it changes to read status immediately. At D–**160** time, the delay counter produces an output to set the go trigger. Because the backward trigger is set, the tape unit moves tape backward.

A delay counter output sets the read condition trigger when the counter advances to 170 for a high speed unit, or to a higher count for lower speed units. The read condition trigger **104**–AC, AD initiates actions to reset the delay counter and conditions the final amplifiers to accept data from the selected tape unit.

Bits in each character transferred from the tape unit enter corresponding a final amplifier with its hi- and lo-clip outputs. The hi-clip outputs set hi-clip skew register positions; and lo-clip outputs set the lo-clip skew register positions. The first hi-clip output from any final amplifier track in each character cycle sets the first bit trigger, causing a read clock to start. The tape control performs a normal read check operation on characters stored in the hi-clip skew register and sets the LRCR at RC–**7** time of each read clock cycle. However, results of the read checks in backspace operations are not significant. The block being read backward has been checked previously, and errors in the record have been detected in an earlier read or write operation.

The first character trigger, set during the initial selection sequence, is on during the first read clock cycle. If the first character received from the tape unit is a tape mark, the first character trigger's ON output allows the tape control to set the first character tape mark trigger at RC–**6** time, as explained previously for a write tape mark operation.

The read clock output at RC–**7** time sets the RDD trigger, causing the delay counted to advance in the *millisecond mode.*

The read clock output at RC–**7** reset time:

(1) resets the skew registers.
(2) resets the first bit trigger, causing the read clock to stop.
(3) resets the first character trigger.

In the second read clock cycle, the tape unit transfers a byte to tape control soon enough to produce a hi-clip output from a final amplifier, set the first bit trigger, and start the read clock before the delay counter advances to 10 for a high speed tape unit, or a higher count for a lower speed tape unit. The read clock output at RC–**0** time resets the RDD trigger to stop the delay counter.

Tape control examines the second character's bit configoration. If the character is a tape mark and the first character tape mark trigger was set during the first read clock cycle, the tape control does not reset the first character tape mark trigger. If the second character is not a tape mark and the first character tape mark trigger was set during the first character cycle, the tape control resets the trigger. If the second character is a tape mark and the first character tape mark trigger was not set during the first read clock cycle, the first character tape mark trigger is not reset (the first character trigger was reset at the end of the first read clock cycle).

If the tape unit transfers a third character to tape control before the delay counter advances to the point designated for the selected tape unit, tape control unconditionally resets the first character tape mark trigger if the trigger is set. Thus the first character tape mark trigger functions to detect a tape mark record. If the tape unit transfers three characters in the record, regardless of the bit structure of the characters, the record is not a tape mark record; a tape mark record contains only two bytes.

If the delay counter advances to the point designated for the selected tape unit, the tape control conditions a backspace reset read condition circuit **117B–AT**, indicating that the tape unit has backspaced through one block and has sensed either an interrecord gap or the gap between load point and the first tape record. In a backspace record operation, the tape control immediately resets the go trigger, to stop tape movement on the selected tape unit and resets the read condition trigger. Hence conditions to end the backspace record operation are satisfied when tape unit senses the extended gap between blocks. In a backspace record operation, the backspace reset read condition circuit resets the go and read condition triggers only if the first character tape mark trigger is set. With the go and read condition triggers reset in backspace record or backspace file operation, the delay counter output at D–68 time resets the backward trigger; the selected tape unit automatically returns to forward status. The delay counter advances to 144 and sets the **RDD–144** trigger to initiate actions to end the backspace operation and transmit the status byte to channel.

In a backspace file operation, the backspace reset read condition circuit activates a continue backspace file gate **113A–AF** if the first character tape mark trigger is not set. The continue backspace file gate sets the first character trigger and resets the delay counter. The tape control then waits for the tape unit to transfer the next character. If the tape unit is sensing an interrecord gap, the tape control processes the second block in the same manner that it processed the first block (the first character trigger is set). If, however, the tape unit is sensing the gap between load point and the first tape record, it will not transfer more characters to tape control, but will continue to move tape backward to load point (tape control has not reset the go trigger). When the tape unit senses load point, it stops tape movement automatically and activates an at load point line to the tape control even though tape control has not dropped the go trigger to the tape unit. The at load point signal sets the tape control's load point delay trigger **85A–AV, AU** to condition:

(1) the backspace or backspace file at load point gate **67–AB** to set the intervention required trigger and bring up end pulse.
(2) the backward disconnect gate **67–AC** to set the **RDD–144** trigger.

Outputs from an end pulse gate **55B–BK** and the set **RDD–144** trigger combine to initiate actions that reset the backspace file operation. The tape control then transmits a status byte to channel containing a "1" in the intervention required bit position.

*Forward space record and forward space file.*—In a forward space record operation, the selected tape unit moves tape forward through one tape block to an interrecord gap except at the end of tape. In a forward space file operation, the selected tape unit moves tape forward to the first interrecord gap that it senses after reading a tape mark record.

The tape unit reads bytes recorded on tape and conditions data lines to the tape control. But the tape control does not transfer the bytes to the channel. Thus this operation is similar to the backspace record or backspace file operation, except that tape moves forward instead of backward.

After the initial selection sequence which contains a command to forward space record or forward space file, the command is decoded, the read operation trigger is set, and read delay gate **112A–AA** is activated to:

(1) set the go trigger, causing tape on the selected tape unit to move forward.
(2) condition millisecond delay counter drive circuits; the delay counter advances in millisecond mode.

If the load point delay trigger is set, the delay counter output at **RD–160** time sets the read condition trigger **104–AC, AD**. If the load point delay trigger is not set, a delay counter output sets the read condition trigger when the counter advances to 44 if a high speed tape unit is selected, or a lesser count for a lower speed unit.

The read condition trigger resets the delay counter and allows the tape control to accept data that the selected tape unit transfers.

Bits in each byte transferred from the tape unit enter a corresponding final amplifier in the tape control. Each final amplifier has hi- and lo-clip outputs which set hi-clip skew register positions and lo-clip skew register positions. The first hi-clip output from any final amplifier in any character cycle sets the first bit trigger, causing the read clock to start. The tape control performs a normal read check operation on bytes stored in the hi-clip skew register and sets the LRCR at RC–7 time of each read clock cycle. However, results of read checks in forward space operations are not used to set any error indicator. The block being read has been checked previously, and errors in the record have been detected in an earlier read or write operation.

The first character trigger, set during the initial selection sequence, is on during the first read clock cycle. If the first character received from the tape unit is a tape mark, the first character trigger's ON output allows the tape control to set the first character tape mark trigger at RC–6 time.

The read clock output at RC–7 time sets the RDD trigger, causing the delay counter to advance in the *microsecond mode.*

The read clock output at RC–7 reset time:

(1) resets the skew registers.
(2) resets the first bit trigger, causing the read clock to stop.
(3) resets the first character trigger.

If the second character that the tape unit transfers to the tape control is not a check character, a final amplifier produces a hi-clip output to set the first bit trigger and start the read clock before the delay counter advances to 36. The read clock output at RC–0 time resets the RDD trigger; and the delay counter resets. If the first character tape mark trigger was set in the first read clock cycle, the tape control unconditionally resets the trigger in this second cycle. The first character tape mark trigger functions to detect a tape mark character followed by a check character. If the tape unit transfers two characters to tape control before transferring a check character, the record is not a tape mark record. The first character tape mark trigger has no purpose in the check operation on the record being read when tape control determines that the record is not a tape mark record.

Because the check character is spaced further from the last normal character in the record than normal character spacing, the extended delay before the tape unit reads and transfers the check character to tape control allows the delay counter to advance to 36 to set the check character trigger. The check character trigger's ON output blocks the read clock output at RC–0 time of the check character cycle from resetting the RDD trigger; consequently, the delay counter runs throughout the check character cycle.

At RDD–128 time, the delay counter generates an output to reset the read condition trigger, and at RDD–144 time, it sets the RDD–144 trigger **120–AB, AC.**

The RDD–144 trigger's ON output conditions activate an ungated RDD–144 trigger 120–AD, AE and the RDD–144 reset line. The ungated RDD–144 line activates the backspace reset read condition line, indicating that the tape unit has spaced forward through one tape block. In a forward record operation, the RDD–144 reset line and the backspace reset read condition line combine to initiate actions to reset the operation and cause tape control to transmit the status byte to channel.

In a forward space file operation, the RDD–144 reset line unconditionally resets the go, the RDD–144 trigger, and RDD triggers and the delay counter, but it can initiate actions to cause the tape control to transmit the status byte only if the first character tape mark trigger is set. Until the tape control transfers the status byte, it is committed to the operation. The reset status of the first character tape mark trigger at RDD–144 time indicates that the last block processed was not a tape mark record; and, therefore, conditions to end the forward space file operation are not satisfied. During the space forward file operation, the activation of the backspace reset read condition line conditions the continue backspace file gate 113A–AF to set the first character trigger 37A and B–AA, AC. The first character trigger's ON output causes the tape unit to read the next block by bringing up read delay to:

(1) set the go trigger.
(2) condition millisecond delay counter drive circuits.

When the delay counter advances to the point designated for the selected tape unit, it generates an output to set the read condition trigger. The tape control processes the succeeding tape blocks in the same manner that it processed to previous block until the selected tape unit transfers a tape mark record. In processing the tape mark record, tape control allows the first character tape mark trigger to remain set until RDD–144 time to terminate the forward space file operation.

(G) Cyclic error detection circuits

(1) *Description of cyclic data error detection.*—The CRC can be used for error detection. This is done by writing the CRC character when the record is written. When the record is read the CRC character is also read. If the CRC as read is not identical to the CRC as calculated on the data read, an error will be detected. The following description is of the operation of the CRCR as it is used to calculate the CRC character.

The method of operation of the CRCR is identical during read and write with the exception that at the end of write, a check character is written and at the end of read the CRCR is sampled for error. FIGURES 28A and B, 192 and 198 show the CRCR register. There are nine bistable circuits called flip-flops.

Referring to FIGURES 28A and B for the numbering of the blocks, when the *+shift CRC* line comes up, this line enters the figure near the top on the left, the output of the EXCLUSIVE OR block CA is shifted into the flip-flop block AA. This is done through two AND circuits, DA and EB. The *shift CRCR* line goes to both these AND circuits. The output of the EXCLUSIVE OR goes directly to AND DA and to the set side of the flip-flop. Hence if the output of CA was a 1, the flip-flop would be set. If the output of the EXCLUSIVE OR CA was a zero, it would be inverted by inverter DB and when the shift signal *+shift CRC* came, the two inputs of AND EB would be up, the output of that AND would then be up, and the flip-flop AA would be set at zero. Each of the 9 bits of the shift register is treated in the same way. The input to the EXCLUSIVE OR CA is the *+CRCR P-bit* and the output of an OR BA. If the output of BA is zero, the output of CA was a 1, the flip-flop would be set. If the output *CRCR P-bit*. That is, when a shift signal comes, the *CRCR P-bit* will go thru block CA where the output will be exactly the same CRCR P-bit. This will go thru the shift circuits thru the two AND circuits DA and EB. They will set or reset the flip-flop depending on whether the *CRCR P-bit* line was a 1 or a 0. The 6 bit is almost identical except that the *CRCR 7-bit* line takes the place of the *P-bit* line.

Hence if the output of the OR's BA, BB, BC, BD, BE, BF, BG and BH are all zeros, when a shift signal is given, the contents of the flip-flops will shift down one position. That is, the contents of the P or bottom flip-flop will go to the 7 bit flop-flop, the contents of the 7 bit flip-flop, that is flip-flop AA, will be shifted to flip-flop AB, which is the 6 bit flip-flop, the contents of that flip-flop will be shifted to the 5 bit flip-flop numbered AC; now at the 4 bit flip-flop there is a little variation. Instead of shifting the output of flip-flop AC into flip-flop AD directly, flip-flop AD, the 4 bit flip-flop, is set to the EXCLUSIVE OR of the setting of the P-bit flip-flop, and the 5 bit flip-flop. This is done with circuits CD and CE. Assume again the output of the OR circuit BD is a zero. Then the output of the EXCLUSIVE OR CD will be CRC P-bit. This enters EXCLUSIVE OR CE, which is also entered by the *CRCR 5 bit* from block *AC*. This output will be the EXCLUSIVE OR of the P-bit and the 5 bit the CRCR shift pulse enters AND block DG and EH. Note this configuration of AND's is the same as that described for the 7 bit. Namely, if the output of the EXCLUSIVE OR CE is a 1, the 1 will go to AND circuit DG and when the shift pulse occurs and the flip-flop AD will be set. If, however, the output of the EXCLUSIVE OR CE is a zero, it will be inverted by an inverter DH the AND circuit EH will be conditioned and when the shift pulse occurs and the flip-flop AD will be reset. Flip-flop bit 3, bit 2 and bit 1 blocks AE, AF, and AG respectively, operate in the same manner. These positions are called positions of feedback. FIGURE 198 shows the general feedback configuration showing where the feedback connections are made without reference to detailed configuration. That describes the shifting of the shift register when there is no data input.

If there is a data input, the line of FIGURE 28A called *+Forward* or the line on 28A called *+Backward* will be conditioned. The line called *+Forward* is on when the tape unit is going in the forward direction. At the time the shift CRC line is energized, there may be data on the line called *correct read/write bus*. There are 9 such lines coming into FIGURES 28A and B. Let us consider the operation of one of these lines, *+correct read/write bus 7 bit*. It enters AND AA along with the *+forward* line. The output of this AND will be a one if the *+correct read/write bus 7 bit* was a one, and a zero if the *+correct read/write bus 7 bit* line was a zero. At this time the line called *+backward* will be a zero, so the output of the AND AB will be a zero. The output of the OR BA will be exactly the same as the output of the AND AA. The OR BA feeds the EXCLUSIVE OR BA previously mentioned. The output of block CA will now be the EXCLUSIVE OR of the *+CRCR P-bit* and *+correct read/write bus 7 bit*. When the *+shift CRCR* line is energized, the flip-flop AA will be set to the state equal to the output of EXCLUSIVE OR CA as described previously. Bit 6, bit 5, bit 0, bit-P operate in the same manner. The feedback positions bit 4, bit 3, bit 2, and bit 1 are modified slightly. Consider bit 4 as an example. When the *+ forward* line is active, input to AND AG will be *+forward* and the line *+correct read/write bus 4 bit*. Since the *+forward* line is on, the *+backward* line will be off and the input to AND AH will be zero, so the output of OR BD will be the same as the output of AND AG. This will feed EXCLUSIVE OR CD. The output of this EXCLUSIVE OR will be *+CRCR P-bit* EXCLUSIVE OR'ed with *+correct read/write bus 4 bit*. This output will be sent to EXCLUSIVE OR CE. The output of this EXCLUSIVE OR will be now the *+CRCR 5 bit*, EXCLUSIVE OR *+CRC P-bit* EXCLUSIVE OR *+correct read/write bus 4 bit*. When the shift CRCR line is energized, this output will be set into flip-flop AD as described previously. Flip-flops 3 bit, 2 bit, and 1 bit operate in a similar manner.

This describes how the CRCR operates with data inputs during forward time.

During backward time the operation is almost identical with the exception that the entry into the CRCR 7 bit is the +*correct read/write bus P bit,* through AND AB, the input to the 6 bit is the +*correct read/write bus P-bit,* into the CRC 5 bit is the +*correct read/write bus 1 bit,* etc. This is a twist of the inputs necessary to do error detection during the read backward operation.

In order to illustrate the operation of the CRCR, reference is made to the following Tables 1–6, which are used in explaining the CRC and LRC operation:

TABLE 1

[5 characters of data]

| TR No. | CH 1 | CH 2 | CH 3 | CH 4 | CH 5 |
|---|---|---|---|---|---|
| 7 | 1 | 0 | 0 | 0 | 0 |
| 6 | 1 | 0 | 0 | 0 | 1 |
| 5 | 0 | 1 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 1 |
| 3 | 0 | 1 | 1 | 1 | 0 |
| 2 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| P | 0 | 1 | 0 | 0 | 0 |

TABLE 2

[CRCR contents when data of Table 1 shifted into it]

| TR No. | Initial | CH 1 | CH 2 | CH 3 | CH 4 | CH 5 | After last shift | CRCR INV. written on tape |
|---|---|---|---|---|---|---|---|---|
| 7 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| P | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

TABLE 3

[Message as written on tape]

| TR No. | CH 1 | CH 2 | CH 3 | CH 4 | CH 5 | CRC | LRC |
|---|---|---|---|---|---|---|---|
| 7 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 6 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 4 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 3 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 2 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | *1 | *1 | 1 | 1 | 1 | 0 |
| P | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

* Errors in these positions, see Table 5.

TABLE 4

[Contents of CRCR after block of Table 3 is read, without error]

| TR No. | CRCR | EPR | LRCR |
|---|---|---|---|
| 7 | 1 | 0 | 0 |
| 6 | 1 | 0 | 0 |
| 5 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| P | 1 | 0 | 0 |

TABLE 5

[Contents of CRCR and EPR after reading block of Table 3 with errors in * positions. INV. means position 7, 6, 5, 3, 1, 0, P inverted.]

| TR No. | CRCR | CRCR with INV. | CRCR shifted once with INV. | EPR |
|---|---|---|---|---|
| 7 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 1 | 1 |
| 4 | 0 | 0 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 2 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| P | 0 | 1 | 0 | 0 |

TABLE 6

[CRCR contents if data of Table 3 entered during read backward operation]

| TR No. | Initial | CH 1 | CH 2 | CH 3 | CH 4 | CH 5 | Final |
|---|---|---|---|---|---|---|---|
| 7 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 6 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| P | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

Assume a five character block as shown in Table 1. Table 2 is the contents of the CRCR after each of the bytes in Table 1 are entered into it. Note that the contents of the CRCR after character **1** is entered into it is identically character **1**. The contents after character **2** is entered into it is character **1** EXCLUSIVE OR bit by bit with character **2**, when character **1** had been shifted down 1 position. Note when there is a 1 in the P-bit position, the feedback paths are energized so that the pattern left is in the CRCR after character **3** is the pattern of the contents after character **2** EXCLUSIVE OR'ed with character **3** and EXCLUSIVE OR'ed with feedback connections.

During write after the last character is read into CRCR it is shifted once then 7 of the 9 bits are inverted, bits **7, 6, 5, 3, 1, 0,** and P, and this is writen on tape. Table 3 is the message as written on tape. If this message is read, each of the characters will be read to the CRCR register except for the LRC character. If there were no errors at the end of the reading, a CRC register will contain a pattern shown in Table 4. This table has one's in the inverted positions for the CRCR output when the CRC character (byte) was written. If there was an error (assume that the 2 positions starred in Table 3 are in error), then the CRCR will contain a pattern in general not the same as the pattern in Table 4, such as shown in Table 5, after reading the message. This pattern is different from the pattern of Table 4, its appearance will signal that there is an error in the record. On FIGURE 33B, AND circuit AB is sampled by signal +*RDD 140* if the pattern at the input of that AND circuit were not exactly the pattern of Table 4 an error signal called —*CRC error* is sent to the data error latch shown on FIGURE 35 (blocks labeled AB, AD).

Table 6 shows the contents of the CRCR for each character, if the characters of Table 3 on tape are fed into it in reverse order and the bits are reversed as described before. Note that the final contents of the CRCR are identical to the contents obtained on a normal forward read. That is the final contents shown in Table 6 are identical to the contents shown in Table 4.

(2) *Error detection after error correction.*—The CRCR is used for error detection for normal read. It is also used for error detection after error correction has been at-

tempted. On FIGURE 2 the corrected data appears on line **15** during a correction cycle. It also appears on line **17** which goes to the CRCR **18**. This is checked by the correct pattern test **51** at the CRL sample time. After error correction the LRCR **40** is sampled to assure that it is zero in all positions by the zero test sum **47** at LRC sample time gated in at **46**. However, sampling is inhibited on the track being corrected by the error track blocking gate **45**. Note that the input to the LRCR is tape read data line **14**, which is not corrected data. Rather, it is data before it has been corrected.

The LRC register is shown in FIGURES 46A and B. Each bit position is identical so description of the 6 bit to be given here will suffice to describe the LRC register operation. The input to the 6 bit is logically identical to the input to the read/write register. That is, it comes from the skew register, FIGURES 40A and B, output from OR block AZ. The input to the read/write register is the same function through inverter DA and then thru AND DB. The AND being a timing signal. The input to LRC register on 46A goes thru inverter AF and then to flip-flop AG, which is a binary complementing flip-flop. That is, each time the input goes from a 0 level to a 1 level, the output of the flip-flop goes to the opposite of its previous output. The output of the 6 flip-flop goes to FIGURE 30. On FIGURE 30 the *+LRC 6 trigger line* is the output of the 6 flip-flop. This goes to AND circuit AB. Note there is another input to this AND circuit called *+EPR 6 bit EX*. Note this is the type of AND circuit if there is a 1 on the top input, and a 0 on the bottom input, there will be an output. The arrow on the bottom input of the AND denotes this function. This means that if the LRC 6 trigger is a 1, and if the EPR 6 bit EX is a 1, there would be no output from AND circuit AB. If the EPR 6 bit EX were a 0, and the LRC 6 bit were a 1, there would be an output from AND circuit AB. The output of each bit of the LRCR is handled in the same way. These are all connected together in a configuration known as a DOT OR and sent to AND circuit BF which samples the output any time *+RDD 140*. The purpose of the special input on the AND circuit AB known as EPR 6 bit EX is to inhibit sampling of 6 bit trigger at a time when the *+EPR 6 bit EX* is a 1. The *EPR 6 bit EX* will be a 1 when errors are being corrected in the 6 bit track. Each bit of the LRCR is handled in the same way.

Both the LRC error detection and the CRC error detection circuits detect errors after correction was attempted. This feature of detection on corrected data is very important. One reason for this is that when data is reread from tape it is possible that the error pattern has changed from the first reading to the second reading. That is, the track in error may have changed or a character dropout may have occurred. Character dropout means a character was not read where one was written.

Another reason this is important is to provide protection if the situation where there is a multiple track error and a track is improperly calculated to be in error. In this case, correction will be attempted. The LRC check provides independent check after correction on the data. This check insures that the errors being corrected are not multiple track errors. This is important as the CRC check is not able to distinguish single track from multiple track errors if a track is calculated to be in error. If a multiple track error did occur, it will be detected by the LRC check with high probability.

(H) Cyclic error correction circuits

(1) *Description of error correction.*—There are three separate parts to the error correction cycle. The first is reading the record and calculating the CRC and EPR. Second part is calculating the track in error using the CRC and EPR. Third part is correcting the record during reread using the track in error information calculated in the second part.

(a) *Reading the block.*—The first part of the error correction cycle, that is reading the record and operating the CRC and EPR is very similar to reading with error detection. The CRCR is operated identically to the way it was described for use with error detection. The circuit known as the EPR is also operated. Referring to FIGURE 2 the data in read-write register **12** is set through the correction EXCLUSIVE OR's **16** thru line **17** to the CRCR as described before. The data is sampled for VRC errors on each byte by the VRC checker **31**. If there is a VRC error at the appropriate time, it is sampled by gate **37** and sent to EPR **30**. The EPR is a shift register similar to the CRC register. A block diagram of EPR is shown in FIGURE 199. Note there is only one input, the VRC input, at the bottom of the register.

The EPR is shown in FIGURE 29A and B. Note there are 9 latches; the output line from each latch identifies it. For example, FIGURE 29A, the top is EPR 7 bit, the next EPR 6 bit, the next EPR 5 bit. When *+shift* command entering the FIGURE 29A is energized, the data is shifted, that is the contents of the EPR 7 bit are shifted thru AND gate JM and DA to flip-flop CA, the EPR 6 bit. In detail it is done as follows: if the EPR 7 bit, that is block AB was set at 1, the top output would be a 1 and the bottom output would be a 0. Both of these outputs go to the AND circuits for the EPR 6 bit, top to block JM and bottom to block DA. When the *+shift* command is given, the *shift* command goes to both the blocks, the one with two one inputs will have an output. If the EPR 7 bit were a one, block JM would have a one output that would set the block CA which is the EPR 6 bit flip-flop. If the EPR 7 bit were a zero, the output DA could have a one which would reset CA to EPR 6 bit flip-flop. This operation is the same for the 7 bit, the 6 bit, the 5 bit, the 0 bit, and the P-bit. The 4 bit, 3 bit, 2 bit, 1 bit are modified by feedback connections. 4 bit will be described, the others are identical, in principle. There is an EXCLUSIVE OR JB. The output of this EXCLUSIVE OR is the EXCLUSIVE OR of the EPR 5 bit and the EPR P-bit. If that is a 1, when the shift command is given, the output of AND gate JP will set flip-flop HA, the 4 bit EPR. If the output of the EXCLUSIVE OR JB is a zero, it will be inverted by an inverter 3HHA when the shift command occurs and JQ will reset flip-flop HA the EPR 4 bit flip-flop. The 4 bit flip-flop will be set to the EXCLUSIVE OR of the previous contents of the 5 bit and the P bit after shifting. The EPR 3 bit, 2 bit, and 1 bit follow the same principle. After shifting they will be at the EXCLUSIVE OR of the previous contents of the flip-flop above them and contents of the EPR P-bit. EPR P-bit of FIGURE 29B has its next setting modified by an input. If there is a *read/write VRC* and the signal called *+read one* is turned on, the AND gate NC will be energized. Therefore EXCLUSIVE OR with the EPR 0 bit in EXCLUSIVE OR NF. The output of this EXCLUSIVE OR will then be set into EPR P-bit flip-flop when the *shift* pulse is given.

*The read/write register VRC* is generated on FIGURE 25A and B. On this page there are a number of ANDs and OR's that are connected in such a way that the output BF will be plus when the EXCLUSIVE OR function of all the read/write bit positions and the odd redundancy latch is on. This circuit is used to detect the parity or the VRC of the read-write register. This output is sent to the EPR, FIGURE 29A and B also to the latch known as the read/write VRC latch on FIGURE 34A and B, this is blocks AF and AJ and also to data check latch on FIGURE 35. The data check trigger is block AB and AD. This is set by the input to OR gate AC which is gated by appropriate conditions by AND gate AA.

At the end of the read operation, the CRCR is sampled as described in the section on the CRC used for error

detection, to determine if an error occurred. Also, the LRC register is sampled in the same way described previously. An error in either of these position will turn on the data check latch on FIGURE 35 thru OR gate AC if at the end of the operation the data check trigger is turned on by any of these errors; read/write VRC, LRC or CRC error, this information will be sent to the CPU.

If there are no errors, the EPR will be all zeros at the end of the read operation. If there is an error, the EPR register will not in most cases be all zeros. For example, if the data of Table 3 is read with errors in the starred positions, the contents of the EPR would be as shown in Table 5 at the end of the read.

(b) *Track in error determination.*—The second part of the error correction cycle, namely determining the tracking error, is also rather simple. The embodiment shown here was designed to minimize the amount of hardware necessary to do the calculation, hence, multiple use is made of some of the parts.

If an error is detected, it is expected that the CPU will order a backspace and then a reread operation. This positions the tape to reread the same record an error was just detected on. When the reread operation starts a control latch called Find Track Latch, blocks TB and TC of FIGURE 33C, will have been turned on. This latch will be turned on by the backspace operation after the forward operation. Referring to FIGURE 2 when the reread operation starts the EPR will be shifted to the read/write register by AND gate **55** when the signal called Find Track Set Up is turned on. Also a signal called Find Track Time will energize the AND gates **34** which sets the CRC output through AND gate **34** and OR gate **33** to the EXCLUSIVE OR gates **16**, the output of these EXCLUSIVE OR gates will then be 9 lines each one being an EXCLUSIVE OR of a bit of the CRC with a corresponding bit of the read/write register. The contents of the read/write register is now equal to the contents of the EPR register. Now the Find Track Set Up signal to gate **55** is turned off and a 1 is entered into the top position of the EPR **30** by AND gate **36** and the signal called Begin Find Track Time.

The EPR **30** is now going to be used as a counter. It will count the number of shifts necessary to get a match between the CRCR and the present contents of the read/write register. The block **41** of FIGURE 2 will be used to detect that the outputs of the nine EXCLUSIVE OR's (block **16**) are all in the zero position. When this occurs, there is a bit for bit match of the CRCR and the contents of the read/write register. When that occurs, the output of the Find Track Test **41** goes to inverter **42** which goes to AND gate **39**, the output of that AND gate which had been a 1 goes to 0, into OR gate **26** which then goes into AND gate **38** which is ANDing a DC signal at the clock to get shift entries to the EPR. In other words, the EPR can shift only until the output of the Find Tract Test indicates that there is all zeros, the output of EXCLUSIVE OR's block **16**. If this is true immediately, no shifts will be taken at all. When a shift pulse occurs, it enters the EPR and also the CRCR. If not comparison is found by circuit **41**, another shift occurs. This happens for 8 shifts, that is 9 comparisons. If at any one of those shifts a comparison had occurred, the shifting would have stopped. The latch called the correcting latch on FIGURE 33C blocks AG and XG would be turned on. The contents of the EPR would now be the one that was entered in the top shifted a number of times. This would indicate the track in error. The CRCR would be reset, the operation is now ready for the third part of the correcting operation, the actual correction of data. Before describing that, the several details of the EPR and the comparison will be described.

EPR is set to one in the 7 bit position as shown on FIGURE 29A. This is done by the signal called *Set One to EPR*. It enters OR gate JA. When this signal is set and a shift pulse occurs, the one goes through AND gate JK to set flip-flop AB. Hence, the EPR 7 bit is set to one.

This is a description of the usage of the EXCLUSIVE OR gates in FIGURE 2, block **16**. These are shown in FIGURES 27A and B. Each gate is identical to the 7 gate, which will now be described. When the gate called Gate CRCR to Read/Write Bus is energized, the *—CRCR 7 bit* enters AND gate DF. This goes through OR gate DG where inverter DH, AND DJ, AND DK, and OR DL constitute an EXCLUSIVE OR circuit.

As an example of the operation, assume that the *—CRCR 7 bit* is equal to a one, and the *+read/write register 7 trigger* is equal to a one. This means the *—read/write 7 trigger* will be equal to zero. Now, the input to AND gate DF will be a 1 on both legs so the output will be a 1. The output of AND gate DE will be a zero and is not being used at this time. The output of OR gate DG will be a zero. This inhibits the output of AND gate DJ so it will be a zero, inverter DH being fed by OR gate DG will be a one, but the input to AND gate DK will be a zero on the *—read/write reg. 7 trigger* input. Hence the output of that gate DK will be a zero. Now, both inputs to OR gate DL will be zeros, so the output will be one, which goes to inverter AA, which will now be a zero. In other words, if a *—CRCR 7 bit* was a one, and the *+read/write reg. 7 trigger* was a one, the output *+correct read/write bus 7 bit* will be a zero, that is, when the *CRCR 7 bit* and the *read/write reg. 7 bit* do not compare, the output of *corrected read/write 7 bit bus* will be a zero. This same relationship will hold for the 6 bit, the 5 bit, the 3 bit, the 1 bit, the 0 bit, and the P bit. Note, however, the input to the read/write gate **4** and the read/write gate **2** in FIGURES 27A and B, are +CRCR 4 bit and +CRCR 2 bit respectively. These outputs will be zero when the read/write reg. bit and the CRCR bit are identical. In other words, all 9 bit outputs will be zeros when the CRCR and the read/write register differ in the 7 bit, 6 bit, 5 bit, 3 bit, 1 bit, 0 bit, and P bit, and are identical in the 4 bit and 2 bit. This is done to compensate for the fact that the CRC character was written complemented in those bit positions.

(c) *Error correction of data block.*—Now the data will be corrected. This is done in a fairly simple manner. Refer to FIGURE 2. The data is read in a normal way to the read/write register **12**. If it is determined that the data byte has an incorrect VRC by the VRC checker **31**, the data is complemented by the EXCLUSIVE OR gate **16** in the position which the track was calculated to be in error, the track in error information is in the EPR **30**. The contents of the EPR **30** are gated to the error correction EXCLUSIVE-OR's **16** when there is a read/write VRC **31**. Since the contents of the EPR consist of all zeros except the 1 in the track in error position the data will emerge from the EXCLUSIVE OR's **16** identical to the way it entered except that the bit position corresponding to the track in error will be inverted. The corrected data is transmitted to the data register of the tape adapter through the translate circuits and hence to the CPU. It is also sent to the CRC register **18** to detect whether correction was done properly. The uncorrected data is sent to the LRC register **40**. At the end of the read with correction, both of these registers are sampled by circuits **51** and **47** as described in the section on error detection to make sure that they are in the proper state indicating no errors have occurred. The LRC is sampled by the circuit **47** as shown in FIGURE 32. It will be noted that the sample is inhibited in the position in which the EPR has a one by the error track blocking gate **45**, that is LRC is sampled to make sure it is all zeros except possibly in the position the EPR has a 1 bit, that is the position corresponding to the track calculated to be in error. That is the error correction cycle.

The important part of the correction hardware used in the error correction cycle just described are the correction gates shown on FIGURE 27A and B. The following is a description of the use of these correction gates to cor-

rect an error in a byte. As an example, assume there is an error calculated to be in the 7 bit. In this case, the *EPR 7 bit* will be a 1. This goes into AND gate DE. If there are no errors, the *read/write VRC error* will be a zero. Hence, the output of OR gate DG will be a one. If the *read/write 7 trigger* was a 1, the output of AND gate DJ will be a one. So, the output of OR gate DL will be a zero. Then the output of inverter AA will be a 1. In this case, the data is fed through the read/write gate without change. However, if there had been a *+read/write VRC error,* both inputs to AND gate DE would be a one. The output of OR DG would be a zero. Hence, the output of inverter DH would be a one. Again, if the *+read/write reg. 7 trigger* was a one, *—read/write 7 trigger* would be a zero. Neither the AND gates DJ or DK would be energized. Hence, the output of OR gate DL would be a one. Then, the output of inverter AA would be a zero. In other words, in this case the contents of the *read/write reg. 7 trigger* were inverted. The output called *correct read/write bus* is then sent to the translate circuit, under normal control to the CPU. This output is also sent to the CRC register.

It is necessary to twist the outputs of the EPR during read backward while doing correction. This is to compensate for the fact that the track in error during read backward was calculated on the data in the CRCR which is twisted in respect to the tracks in the read/write register. FIGURE 32 shows where the EPR outputs are twisted during read backward. Consider output *EPR 7 bit EX* from block BB. During forward time the signal called *forward* is energized so the EPR P-bit output goes to AND gate AC. This then goes to OR gate BB to give the *EPR 7 bit EX* signal. This is the signal used for the Error Correction FIGURE 27A and B. During backward, however, the *EPR 7 bit* goes to AND gate AB, which then goes to OR gate BB. This then causes the corrections signal *EPR 7 bit EX*. This compensates for the position of the EPR which corresponds to the track in error.

(J) Read operation with cyclic error correction

(1) *Description of a typical read operation.*—Let us assume that a block such as that shown in Table IV is written on the tape and the tape is positioned so that the next block read will be that record. First thing that happens is a read command is sent through the interface sequence by the CPU. This energizes the controls in the tape control unit and these in turn energize the tape drive which starts to move forward. After the appropriate time-out the amplifier circuits are degated and the byte of data is expected.

(a) *Initial block reading operation.*—The first byte of data arrives and goes into the amplifier circuit as shown in FIGURES 48, 49, and 50. To illustrate the flow of data, consider the flow of the 0 bit on FIGURE 50, which enters on the left as a signed called *read bus 0 bit.* This enters the OR block numbered AW. The output of that goes to the amplifier block labeled AM. The output of that goes to the amplifier block labeled AN. The output of that amplifier goes to two blocks, one labeled AP, the other labeled AS. The output of these two blocks go to another two blocks labeled AQ and AT. There are two outputs called *high clip 0 pulse,* the other called *low clip 0 pulse.* These are directed to the high clip register and the low clip register, respectively. The *high clip pulse* goes to FIGURE 39, where it goes to the pulse freeze registers, enters on the left called *+high clip 0 pulse* into block AL where it is ANDed with the signal called *freeze condition.* This goes into the latch made up of the AND, AM, the OR, AR, and the inverter AS. On the right hand side of the page there are two outputs called *—high clip 0 pulse gated* and *+high clip 0 pulse gated.*

The *+high clip 1 gate* goes to the skew register 0 position FIGURE 43A and B. It enters on the right and goes into the AND labeled AV, this goes to the OR labeled AW. This goes to the inverter labeled AX. The inverter feeds back to the AND labeled BC, back to the OR labeled AW. This makes a latch circuit called the skew register high clip latch. The *low clip pulse* enters the left hand side of the same page and enters AND circuit BM. This goes to OR circuit BF which goes to inverter circuit BG which feeds back to AND circuit BE and hence to OR circuit BF. This makes a latch called the skew register low clip latch. The *0 read data* output in generated as follows: if a signal called *gate out high clip* coming in on the right is energized into AND AU, the output of high clip register AX will go to OR circuit AZ to the inverter BA to the AND circuit BB which generates the signal *—0 read data* which is leaving the page on the right hand side. If on the other hand, the signal called *+gate out low clip* were energized that signal would go to AND circuit BH which is ANDed with the low clip output from BG that will go to the OR circuit AZ to the inverter BA to the AND circuit BB which will generate the *—0 read data output.* The other bit positions of the skew registers will be found on pages 40A and B, 41A and B, and 42A and B.

If any of the high clip pulses occur, they will be gated in FIGURE 36. They can be seen entering the OR gate AF, then to the AND AG, then to the OR AH which then turns on the first bit trigger which is made of blocks AK and AL. The output of this trigger is a signal called *start read clock* which can be seen on the right hand side of this page. The signal called *start read clock* enters the FIGURE 108A and B on the left hand side. It goes to clocks labeled BC, AY, AZ, AU, AS, and AP. These are the gate inputs to the oscillators. These inputs are turned on, the oscillators will start oscillating. The other logic on the page selects which oscillator is used, depending upon the density and the model of the tape drive that is selected. The output signal on this page is the signal known as *read clock drive,* there are two *+read clock drive* and *—read clock drive.* These are the signals which cause the read clock counter to count. They enter FIGURE 108A and B on the left hand side. The read clock shown in FIGURE 108A and B and 109A and B counts for 7 counts, then resets itself. These counts are to control the feeding of data through the skew registers, to the read/write register and signals the controls to take the data from the read/write register through the translate circuits to the interface onward to the CPU.

Contents of the skew register are sent to the read/write register at RC 7. This is done as follows: on FIGURE 47A and B, *+RC 7 gated* goes into blocks AJ and AT. These go through inverters AK and AU. These give the *gate out high clip pulse* and the *gate out low clip pulse.* These go to the skew register pages for example, skew register 7 FIGURE 40A and B. The *—7 read data* output comes from block AH. This output will have a pulse on it when either the *gate out high clip pulse* hits AND circuit AA or the *gate out low clip* hits AND circuit AP. In either case the signal will go through OR AF, through inverter AG to AND circuit AH to give the *read data* signal. This will go to the read/write register 7 bit position on FIGURE 18A where it will enter block AD OR half of the flip-flop made of AD and AK. This is the read/write register 7 bit position.

The next thing that happens is the read back VRC is sampled. The read/write VRC appears on FIGURE 25A and B. It has been described previously and is the EXCLUSIVE OR of all the bit positions of the read/write register and a signal called *odd redundancy.* This is sampled at a time called *shift data.* A signal called *shift data* can be seen entering on the left hand side of the page into block BJ. This signal is generated on FIGURE 74A and B, where it comes out of blocks BD and AB. It is originally generated by a signal coming in the left hand of that figure called *—start shift.* It goes through OR circuit BB to single shot AA to AND AS which goes to AND/OR AU which goes to OR AV which goes to inverter AW which goes to inverter BD. The AND/OR AT is a part of a latch which will latch up when the

—*freeze condition* comes. This is part of the error detection circuitry and is not used during a normal operation. The signal called *start shift* is generated on page 73A and B. On the left hand side of the page a signal called —*RC RS not write*. This is the read clock reset. It goes to OR AM, to inverter AP to AND AK where is generates the signal called —*start shift*. The RC reset comes with the time right after RC 7. The start shift then samples the VRC. If there is a VRC error, the error triggers are set. The read/write VRC error trigger is on FIGURE 34A and B, blocks AF and AJ. Also the data check trigger is on FIGURE 35, data blocks AB and AD. This is set by the read/write VRC error signal coming into OR block AC, then to the AND block AA, then to data check trigger previously mentioned.

The +*shift data* signal also is used to shift the EPR and the CRCR. The +*shift data* signal enters FIGURE 33C on the left, enters AND block XH then goes to OR block BJ which goes to AND block TG which goes to inverter CD which generates the +*shift EPR* signal which AND gate TG also goes to FIGURE 33D or enters OR block AF and generates the +*shift CRCR* signal. These shift signals shift the EPR and CRC respectively, in the way previously described.

If the read/write VRC had been wrong, a 1 would have been shifted into the bottom of the EPR register in the manner previously described.

The read clock and the first bit latch are now reset by the RC reset signal which is used to generate the shift data command.

Another counter which is in the tape adapter is a delay counter. During this part of the cycle it is referred to as the RDD counter. RDD stands for read-disconnect-delay. The counter is controlled in such a way that at RC 7 it starts to count but RC 0 comes the counter is reset and will not start the count again until RC–7 time. This control is accomplished as follows. On FIGURE 112A and B, signal +*RC 7* gated from the read clock goes into AND/OR AR, goes into OR AL. This goes into OR AM, this is one half the flip-flop known as the RDD flip-flop. The other half is AN. When this flip-flop is on the AND AV will be on. This will go to the OR AW and this will turn on the signal +*microsecond control* of FIGURE 113A and B. *Microsecond control* enters AND gate AA. When +*microsecond control* went on the —*microsecond control* entering AND gate AA went off so the gate is no longer active. Since OR AB is no longer active, this OR went into OR BF and BC which went into OR BD which went into OR AE which went into inverters BG, AX, and BL which generate the delay counter reset signals called DC reset. That is, the reset signal will be turned off. The counter will now start counting. The counter itself, is made up of triggers called the DCA trigger, DCB trigger, DCC trigger, these triggers being on. FIGURES 113A and B, blocks AH, AP, and AV, respectively. FIGURE 114A and B has three more triggers, DC, DE, and F being blocks AG, AP and AV respectively. FIGURE 115 has three more blocks of the trigger, the DCG, H, and J triggers being AC, AF and AH. These triggers, together with their associated circuitry, make up the delay counter.

The next byte of the block will be treated in the same way as will each of the succeeding bytes of the block.

*Check character cycle.*—In a normal read clock cycle, tape control sets the RDD trigger at RC 7 time allowing the delay counter to advance in microsecond mode. At RC 0 time of the following read cycle, tape control resets the RDD trigger, stopping the delay counter. Normal character spacing in the tape record allows the tape unit to transfer at least one bit in a character to tape control to start the read clock and to stop the delay counter before the counter advances to RDD 36 to set the CRC check character latch. This latch shown on FIGURE 33A is blocks BA and BB is set by RDD 36 going through AND circuit AA. When first high clip output from the final amplifier after the delay counter reaches 36, the first bit trigger starts the read clock. Because the CRC check character trigger is set, the read clock output at RC 0 does not reset the RDD trigger. The delay counter runs through the check character cycle. The check character which is this time the CRC check character will go through to the CRC register and if it had a VRC error, a 1 will be inserted into the EPR register as for normal data.

Each input character from the tape unit before the check character trigger is set causes the binary even-odd character counter trigger on FIGURE 47A and B, this is block BL, to change states. Therefore if the record contained in even number of normal characters, the last normal character in the record resets the trigger. If the record contains an odd number of normal characters, the last normal character in the tape record sets the trigger. If the odd-even character counter trigger is off when the tape control receives the check character, the CRC character should contain an odd number of bits. If the odd-even character count trigger is set when the tape control receives the check character, the state of the odd-even odd redundancy trigger determines whether the check character should contain an odd or even number of bits. The output of the check character trigger is therefore used during CRC character time to modify the calculation of the read/write VRC. This is done on FIGURE 25A and B by use of the EXCLUSIVE OR gate BE.

When the delay counter reaches count 96, it will turn on the LRC latch on FIGURE 33A. This is done by the signal +*RDD 96* into AND gate AD into the AND gate BC and BD which is the LRCR latch. At RDD 96 a shift signal is given to the CRC and the EPR. This is generated on FIGURE 33C by AND gate XK which goes to OR gate BJ which goes to AND gate TG which goes to inverter CD which gives the +*shift EPR* signal. The output of AND gate TG also goes to FIGURE 33D where the signal goes to OR gate AE which gives +*shift CRC*. This is the last CRC shift in the cycle. No shift is given for the LRC character which arrives sometime after RDD 96. When the delay counter advances to 140, recall that the RDD counter ran throughout the check character cycle, gate control resets the read condition trigger and samples the LRC and the CRC. The LRC is sampled on FIGURE 30 by the signal +*RDD 140* which goes into AND gate BF. The CRC is sampled on FIGURE 33B by the signal +*RDD 140* which goes to AND gate AB. The EPR and the CRCR are not reset. If the error sample signals have resulted in an error signal, the check character latch would have been set as described previously in the section in error detection. This signal would be sent to the CPU for further action.

If there had been no read/write VRC error during the operation, a signal would be generated at time 144 on FIGURE 33A through AND gate AB which would reset the CRCR check character latch and would reset the LRC check character latch, both on the same page, and would reset the read-forward latch on FIGURE 33C. This latter latch made of blocks XC and XD is set when a read command is received. Its purpose is to store the fact that read command was received if there was an error during that read. We shall now describe control actions which would occur if there had been an error during this first read.

If an error signal is sent to the CPU at the end of a read operation, it is supposed to send back a backspace signal. If the read forward latch was on and a backspace record signal came, the find track latch shown on FIGURE 33C would be turned on by AND gate BH which would be conditioned by the read forward latch and *backspace record* signal titled +*BSR* on FIGURE 33C. The find track latch is the two AND gates labeled TB and TC. The read forward latch will be reset during the backspace operation by a signal generated on FIGURE 33A by the AND gate AG, the input to that being +*TA 96*.

*TA 96* means turn around delay count **96**. The turn around latch is turned on under these conditions (the turn around latch is on FIGURE 31). The signal *backward memory* from the tape drive and the *backward operation* turn on the turn around latch shown as the OR AA and the AND AG. This latch is reset in the backspace operation. Delay count **96**, the fact the turn around latch is on and *delay count 96* occur energizes AND gate AB generating signal *TA 96*.

(b) Subsequent block reading with error corrections: If the next operation is a read operation, the following actions will be done to find the track in error. The contents of the EPR will be gated to the read/write register (the delay counter has reached count 7 and 8). This is done on FIGURE 33D by AND gate AF which generates the signal *+gate EPR to read/write register*. This signal conditions gates on the input of the read/write register which gate the EPR to the read/write register. For example, on FIGURE 19A and B, the read/write register 2-bit will be turned on if the EPR 2-bit is a 1 by AND gate BA which will go to OR gate AB which will go to OR gate AC and AND gate AE which is the read/write register 2-bit. The other bits of the read/write register are treated in a similar fashion.

The next EPR is reset. This happens at DC **11** and **12**. The signal to do this is generated on FIGURE 33D by AND gate AG. The signal *+reset EPR* goes to all the EPR latches and resets them all. Then the EPR is set to a 1, that is a 1 is put into the top or 7 position of the EPR. This is done with the signal set *EPR to 1* generated on page 33D. It is generated by AND gate AH. All during this time the signal called *+gate CRC to read/write bus* is on. This signal is generated on FIGURE 33D. It will be on if the find track latch is on and the turn around latch is on and this is not a backspace record or forward space record operation. This signal is generated by AND gate XP which goes to inverter AC. *+gate CRC to read/write bus* goes to the correction EXCLUSIVE OR's shown on FIGURE 27A and B. These have been described in detail in the section on error correction. At DC **25, 26, 27, 28, 29, 30,** and **31** shift pulses are sent to the EPR and CRCR. These pulses are generated on FIGURE 33A by AND gate AJ. They go through inverter BE to FIGURE 33C, the AND gate BJ, then to the OR gate BJ, then to the AND gate TG which goes to the inverter CD which gives the signal *shift EPR*. The signal from the AND gate TG also goes to trigger **33D**, OR gate AF which gives the signal *+shift CRCR*. If the output of the correction read/write bus, that is the output of the correction EXCLUSIVE OR's, FIGURE 27A and B, goes to all zeros, this will be detected by the OR circuit on FIGURE 33B, **09**. This will go through inverter BG to the AND circuit BF. This AND circuit is inhibited with the two conditions that the CRCR is not all zeros detected by OR circuit **10** and that the CRCE is not at the setting 111010111 detected by the AND circuit **09**. If the AND circuit is conditioned, the signal goes to the correction latch on FIGURE 33C and sets it. Correction latch is made up of the two AND circuits AG and XG. When the correcting latch turns on it inhibits the input to AND circuit TG and this stops the shifting of the EPR and the CRCR. The find track latch is now turned off at DC **99**. This will turn off the signal *gate CRCR to read/write bus* generated by XP, the signal *find track latch* being the top input to that AND gate. It will also reset the CRCR through AND gate XQ to AND gate YC, giving the *reset CRCR* signal which goes to each of the flip-flops of the CRC register.

Note that if no track location had been found, that is the correction latch had not been turned on, the find track latch still would have been turned off. In this situation data received during this read would have gone to the EPR and the CRC in the normal fashion for a read operation. Since the correcting latch is on, the EPR will remain in its present condition having one 1 in it, corresponding to the track in error. Data from the read/write register will be corrected by the correction EXCLUSIVE OR's as outlined in the section on error correction. The EPR is gated to the correction EXCLUSIVE OR's by the signal *read/write VRC error*. This signal is generated on FIGURE 25A and B block BK which is the AND of the correcting latch being on and there being a read/write VRC error. Each of the bytes of the data will be now read exactly in the same way as they were during the normal operation with the exception that the data will be corrected by the correction EXCLUSIVE OR's. The CRC character is read as usual and corrected if necessary. Then the LRC character is read only into the LRC register and the read/write register.

The LRC and the CRCR error detection is described in the section on error detection. The bit position of the LRC corresponding to the track being corrected is not sampled by the LRC error sample. The way this is accomplished was described in the previous section. If either the LRC or the CRC have an error, the check latch on **35** will be turned. Note, however, that the read/write VRC error generated on FIGURE 25A and B is inhibited by the correcting latch. Hence read/write VRC errors which are used during the correcting cycle will not turn on an error signal during this cycle. At RDD **144** the correcting latch will be reset. This is the end of the correction cycle.

## (IV) Drawing Element Definitions

A logical circuit unit is represented by a rectangular block, which performs a logical function, and is connected by lines representing actual hardware line connections. A figure (such as FIGURES 61A and 61B) provides a subcombination of hardware within an embodiment of the invention. The input and output lines to a figure generally fan out into a network and respectively connect to other figures. Thus the input and output lines to a figure are called *net lines*, and each is uniquely labeled with a reference designation, such as **12–AB** comprising a figure number (**12**) and a block identification (AB) that represents the logic block from which the line derives as an output.

Input lines enter a figure or a block at the left; logical outputs leave at the right.

Information at the Bottom of the Block is provided in each figure.

The Block Identification in a figure is a two-letter unique designation in a rectangular box in a figure. Block Identifications are assigned in each figure beginning with AA. In the A and B sections comprising a figure, the Block Identification is unique to a block.

A hollow triangular wedge in the edge of a block, in line with an input or output line, indicates that the more negative of two D.C. voltages may be expected at the indicated input or output line when the circuit represented is performing the indicated function. For example, see boxes AH and AJ on FIGURE 95B. The absence of a wedge indicates that the more positive voltage may be expected.

All branches of a line going to several figures are represented on the source figure. Branches are listed at the right of the figure by the sink figure number, which is the figure to which a line goes.

Every logical line is assigned a unique Reference Number. The reference number is composed of the *source block* figure number, and the identification of the block (2 characters). The two character combination of the source block serial identification is known as the line origin, and is printed at the end of every line leaving the figure.

Drawing inputs must come into the left hand side of the block and all outputs leave the right side of the logic block. This means an input to one block cannot be tied to the input of another block without first coming from a right side source. The only exception to this rule is when inputs are being tied to a voltage on the same block.

On an Input Line entering a figure is printed the Line Name and the reference number of the block which provides the line as an output.

At the right edge of a figure on an Output Line leaving a figure is printed the sink figure number (figure to which line goes), the line name, and the line origin. Multiple sink figures are listed to designate all figures to which line branches go. The same line name applies to all branches.

Line names are shown with plus (+) or minus (−) signs representing the voltage level that provides the active state for a line, e.g., +START.

Cross-referencing sends the signal name that is on the source figure of any given net to the sink figures on which the net appears. The *source signal* name is the master name.

A functional symbol appears in each structural block found in the figures and in some cases has associated with it an optional suffix. The functional symbol denotes the primary use for the circuit represented by the block, which, unless stated otherwise, is a circuit available in the prior art performing the defining function. The optional suffix defines the environmental usage for a particular circuit. The block input is on the left side of a block and the block output is on the right side. A hollow wedge (▷) appearing in the edge of block defines a polarity inversion for an input or output in reference to the operation of a logic block.

The circuits represented by a *functional symbol* in a block follow:

AND Device is represented by the symbol A. See box AB on FIGURE 95A. The output of the AND device is in its more positive condition when, and only when, all of the inputs are in their more positive condition.

AND INVERT Device is the output of an AND that is INVERTED into a more negative condition when, and only when, all of the inputs are in their more positive condition.

OR Device is represented by the symbol OR. See box AA on FIGURE 95A. The output of the OR Device is in its more positive condition when, and only when, one or more of the inputs are in their more positive condition.

OR INVERT Circuit. The output of the OR INVERT device is in its more negative state when only when one or more of the inputs are in their more positive state.

GATE is represented by the symbol G in a box. This is a device which performs the basic logical AND operation but whose inputs are especially designed or are not interchangeable. It has a specified control input which passes or rejects a signal at the second input.

INVERTER device is represented by the symbol N. See box AR on FIGURE 95A. This is a device whose output is in the more positive condition as a result of the input being in the more negative condition and vice-versa. This symbol will be used when a logical AND/OR invert is used with one and only one input.

OSCILLATOR is represented by the symbol OSC. See FIGURE 108A. This is a self-excited device which produces a uniform repetitive output.

A DELAY device is represented by a symbol TD in a box. This is a device whose primary function is the time delay of a signal without intentional distortion of the signal. The time delay symbol must always be accompanied by its time delay, shown by a value inscribed next to the block.

INDICATOR device is represented by the symbol IND in a box. Neon or incandescent lamp.

DRIVER device is represented by a symbol D in a box. This is a power amplifying device whose fundamental purpose is to provide adequate driving energy and appropriate impedance matching to other devices.

AMPLIFIER device is represented by a symbol AR in a box. This is a device which increases one or more of the essential characteristics of the input such as level, signal swing, current, or improves rise and/or fall time.

CLIPPER is represented by a symbol L in a box. This is a device that limits one or both extremes of a waveform to a predetermined level without intentional alteration of the remaining waveform.

CLAMP device is represented by a symbol S in a box. This is a device that sets one extremity of a waveform at a predetermined level without intentional alteration of the waveform. It is sometimes referred to as a DC restorer.

CONVERTER device is represented by a symbol C in a box. This is a device that provides the necessary conversion or translation between two types of logic implementation; e.g., N line to P line, P line to N line, voltage mode to current mode, etc.

SINGLE-SHOT device is represented by the symbol SS in a box. This is a device that is characterized by a transistion from the prescribed stable state into a quasi-stable state where it remains for a predetermined time before returning to the stable state. The single-shot symbol is accompanied by the time duration of its quasi-stable state, which is indicated next to the block.

PULSE GENERATOR is represented by a symbol PG in a box. A device that is characterized by a variation from the prescribed state of such short duration that it cannot be thought of as switching to a new or second state. The pulse generator symbol may or may not be accompanied by its pulse timing.

FLIP-FLOP device is represented by a symbol FF in one or more related boxes. This is a device which has two stable conditions and has any one or any combination of the following possible inputs: inputs which set the two stables; inputs which change or complement the existing state; and inputs which hold the existing state regardless of signals on the other inputs. It normally has two outputs. These outputs are always at opposite polarities.

FLIP-FLOP LATCH is represented by the symbol FL in a box. A flip-flop latch has outputs which are not necessarily opposite in polarity. The application of simultaneous significant levels to the set and clear inputs causes both outputs to assume the same level (both assume the significant level or both assume the reference level, depending on the design of the circuit).

The FREEZE LATCH has a FREEZE signal input to cause the latch to hold whatever bistable condition existed in it just prior to the FREEZE signal. The reference symbol GB in a box represents a FREEZE LATCH.

LOAD RESISTOR device is represented by the symbol R in a box. Normally the collector or emitter load associated with a given circuit, when the load has its own circuit number.

Time duration indicated beside a block by N, U, M denotes NANO, MICRO, MILLI seconds respectively.

When the inputs to a logic block are not symmetrical, an asterisk is used to signify specific line positions for these points. More than likely the asterisk will be associated with the following symbols: G, D, SS, PG, FF, FL.

The following suffixes are used with the functional symbol (hereafter referred to as F.S.):

| | | |
|---|---|---|
| A | AND | Used with F.S. OR or A to denote a block which performs a wired AND function. |
| OR | OR | Used with F.S. OR or A to denote a block which performs a wired OR function. |

The following suffixes may be used with F. S. "D":

| | | |
|---|---|---|
| I | Indicator | Denotes a driver indicator. |
| T | Terminator | Denotes a driver terminator, a device which terminates a transmission line. |
| SP | Sample Pulse | Denotes a sample pulse driver. |
| L | Line | Denotes a line driver, which is a device used to drive a transmission line. |
| R | Relay | Denotes a device which drives a relay. |
| C | Core | Denotes a core driver. |

Example of how a functional symbol (F.S.) and a suffix are combined to form a suffix:

The logical And Inverter is used to drive a lamp. The block would be represented as: F.S. "D" combined with "I" to form suffix "DI."

Example of F.S. used as a suffix:

Two logical OR/AND Inverters may be used to implement a flip-flop. For example see boxes AC and AF on FIGURE 95A.

Special components or mechanically actuated components can be represented when certain ground rules are followed. The types of special components included in this section are switches, relays, fuses, resistors, capacitors, RC networks, thermals and indicators.

The use of an asterisk "*" as the last character of a functional symbol indicates a special component block representation.

The logic blocks for special components are identified as: switch, relay, fuse, resistor, RC, indicator, thermal.

Timings or values in the blocks can have the following multiplication indicators:

P=Pico=$10^{-12}$
N=Nano=$10^{-9}$
U=Micro=$10^{-6}$
M=Milli=$10^{-3}$
K=Kilo=$10^{3}$
MEG=Meg=$10^{6}$

The following block representations are also available. SWITCHES are represented by:

SW or SW1:

PUSH—pushbutton or toggle switch.
MICRO—microswitch.
TELEP—telephone switch.
ROT—rotary switch.

RELAYS are represented by:

RLY or RLY 02:

P–pick H–hold.
WIRE—wire contact or permissive make.
DUO—duo contact relay.
REED—high speed reed relay.

FUSES are represented by:

FUSE or FUSE 01:

3.5 a. fuse value.
4 a.s.b. fuse value, slow blow type.

RESISTORS are represented by:

R or R111:

10K value (use standard abbreviation) blank indicates the value in ohms.

CAPACITORS are represented by:

CAP:

100 U value (use standard abbreviation).

RC NETWORKS are represented by:

RC:

180 N delay value.

THERMALS are represented by:

THER:

120 F. operating temperature.

INDICATORS are represented by:

IND:

LAMP—type of indicator.
NEON

An asterisk (*) after the functional symbol in a logic block signifies a special block. On a special block a four (4) character mnemonic code identifies the type of special block and its processing, such as:

SERV—causes wiring only.

In addition to the wedge ( ▷ ) which indicates a negative voltage level, symbols P (positive) and N (negative) may be used to indicate inputs which are activated by voltage shifts, such as obtained from a capacitor:

Generally, two outputs cannot go to the same input. An exception is a box with a DOT symbol, which is used to represent a connection of two ouput lines.

The logic function performed by the dot block is identified in each block connected to a dot block, as well as in the dot block.

There are several cases in which output lines not performing a logical function are tied together, such as where *output* (or input) lines on the same block may be tied together; or *non-logical outputs* on different blocks may be tied together, such as a load resistor sink.

If outputs for plural blocks are tied together on the same figure, the net number for the output line is the reference number of one of the blocks. In the edge of all the other blocks having outputs in the same net, a K appears in line with each of the commoned outputs.

If all of the outputs tied together are not on the same figure, the outputs tied together on one figure show an output to the right side of the figure. The outputs in the same net on other figures return to the left of their respective figures and are referenced to the first figure in the normal manner. The net number will include the line origin of one of the commoned outputs on the first figure. In the edge of all the other blocks having outputs in the same net, a K will appear in line with each commoned output.

Asterisks are placed at the outputs of all blocks which have connectors associated with them. The connectors may be listed above the line, slanting towards it.

Connectors are usually shown on outputs, and are always associated with net sources. Connectors may not be shown on individual branches. The asterisk symbolizing the presence of connectors appears at the source of the net next to the pin designation. All the connectors in the net are listed together at the bottom of the figure and referenced to the net source by line origin.

Connectors generally are put on the source of a net and are printed on the figure where the source originated.

ENTR and EXIT blocks may be used whenever a line connects to a different machine type. This is a convenient way of generating net No.'s when connector wiring is necessary. The figure number where the line goes to or comes from is shown only within the block. It should not be shown in the usual way as an input net number or as a sink figure number associated with a line name.

Machine Type:

For EXIT blocks, the machine type is where the signal line is going.
For ENTR blocks, the machine type is from where the line is coming.
ADAPT means Tape Control Unit.
CHAN means Computer Channel.

While the invention has been particularly shown and described with reference to a preferred embodiment thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and details may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

1. Means for use in locating an error in a data block having plural groups of bits comprising,
first cyclic means for receiving parallel groups of bits from said block and performing a part of a cycling operation for each of said groups of bits received within the block,
means for generating an error status indication for each parallel group within the block,
and second cycling means receiving the error status indication for each of said parallel groups of bits and performing a cycling operation with each parallel group of bits received by said first cycling means, and
means for synchronizing the cycling of said first and second cycling means with each parallel group transfer within the block, whereby both said first and second cycling means are synchronized for the same parallel groups of bits while the block is being received.

2. Means for developing an error pattern for a block of data having a plurality of sequential subgroups of data, and each subgroup having at least one redundancy checking bit comprising,
means for generating an error status indication for each of said subgroups,
and cyclic means for receiving said error status indication from said generating means for each of said subgroups to develop an error pattern for said block of data while receiving said block.

3. A system as defined in claim 2 in which, said means for generating an error status indication for each subgroup is a byte redundancy check circuit.

4. A system for generating information which can be used to determine an erroneous bit posiiton in an information block, comprising
means for providing at least one subgroup redundancy bit for each information subgroup in said block having plural subgroups,
means for transferring the bits of each subgroup and said subgroup redundancy bit in a plurality of channels,
means for generating a cyclic redundancy subgroup for data of said block transferred by said plural channels and receiving the output of said subgroup redundancy means,
and means for storing said subgroup redundancy and said cyclic redundancy subgroup with said block using said transferring means, whereby said block of information has stored with it a form of redundant information later usable for locating any number of erroneous bit positions in any one of its channels.

5. A system for generating and storing redundancy for determining an erroneous bit position in any part of an information block comprising,
first means for utilizing subgroup redundancies of said block for generating a first cyclic error pattern for said block,
second means for utilizing cyclic redundancy with respect to all bits in said block for generating a second cyclic error pattern for said block,
and means for utilizing the patterns generated by said first and second means for locating an erroneous bit position in any subgroup of said block.

6. A system for locating an error in a block of bytes of data, having a cyclic redundancy byte included with said block of data, and byte redundancy being included with each byte of said block, comprising,
means for reading said block, and first cyclic computing means receiving the bits being read from said block including said cyclic redundancy byte,
means for checking the byte redundancy of the bytes being read from said block to provide an error indication upon a byte error,
second cyclic computing means receiving any said error indication from said checking means,
and means for comparing the output states of said first and second cyclic computing means to determine the location of an erroneous bit in any byte of said block.

7. Means for partially locating an erroneous bit in a data block having first and second forms of redundancy, said first form being a cyclic redundancy byte associated with said data block, and said second form being a redundancy associated with each byte in said block, comprising
means for detecting an error in any data byte using said second form of redundancy,
first and second cyclicly-operating storage means, and including means for moving data within said storage means synchronously with received data bytes,
said first storage means receiving an output of said error detecting means,
said second storage means receiving parallel bits of each byte,
means for comparing the bit patterns in said first and second storage means after the data block and its cyclic redundancy byte have been received,
and means dependent upon the operation of said comparing means for recognizing the partial location of any bit error in said block.

8. Means for partially locating an erroneous bit in a data block as defined in claim 7 in which:
said first form of redundancy being a cyclic redundancy byte predetermined to obtain a predetermined residue pattern in said second storage means after being processed with the data of said block,
said second form of redundancy being a parity redundancy for each byte in said block,
said error detecting means being a parity check circuit and providing an input to said first storage means for it to develop a predetermined residue pattern after receiving all bytes of said block,
said first and second storage means having end-around feedback paths,
said comparing means including means for moving the bit pattern of one of said storage means relative to the bit pattern of the other storage means and making a comparison after each relative bit movement,
said means for recognizing including means for counting the number of bit-pattern shifts required to obtain a particular comparison, with said counting means thereby recognizing the partial location of an erroneous bit as a bit position in a byte of said block.

9. Means for partially locating an erroneous bit in a data block as defined in claim 8 in which:
said first and second storage means are simalar shift registers.

10. Means for locating an error in a block of data bytes having a byte redundancy check comprising,
means for reading said block of data,
a diagonal redundancy check register having an output feedback to the input of said register,
an error-pattern register having an output feedback to its input,
means for shifting each of said registers once per byte read from said block,
means for entering the bits of each byte read from said block into different stages of said diagonal redundancy check register,
means for generating a redundancy determination for each byte read from said block and entering it into an input of said error-pattern register,
means for comparing the respective positions in said two registers after an entire block has been read, including a diagonal redundancy check character,

and means for shifting one of said registers until a match is obtained if a match is not initially obtained, whereby the number of shifts by said last means determines the bit position in a byte being in error.


**References Cited**

UNITED STATES PATENTS

| | | | |
|---|---|---|---|
| 3,069,657 | 12/1962 | Green et al. | 340—146.1 X |
| 3,291,972 | 12/1966 | Helm | 340—146.1 X |
| 3,311,878 | 3/1967 | Melas | 340—146.1 |
| 3,051,784 | 8/1962 | Neumann | 178—23 |
| 3,222,643 | 12/1965 | Klinkhamer | 340—146.1 |
| 3,308,429 | 3/1967 | MacWilliams | 340—146.1 |
| 3,315,228 | 4/1967 | Futerfas et al. | 340—146.1 |

OTHER REFERENCES

W. W. Peterson, Error-correcting Codes, Scientific American, vol. 206, No. 2, February 1962, pp. 96–108.





MALCOLM A. MORRISON, Primary Examiner

C. E. ATKINSON, Assistant Examiner

U.S. Cl. X.R.

235—153