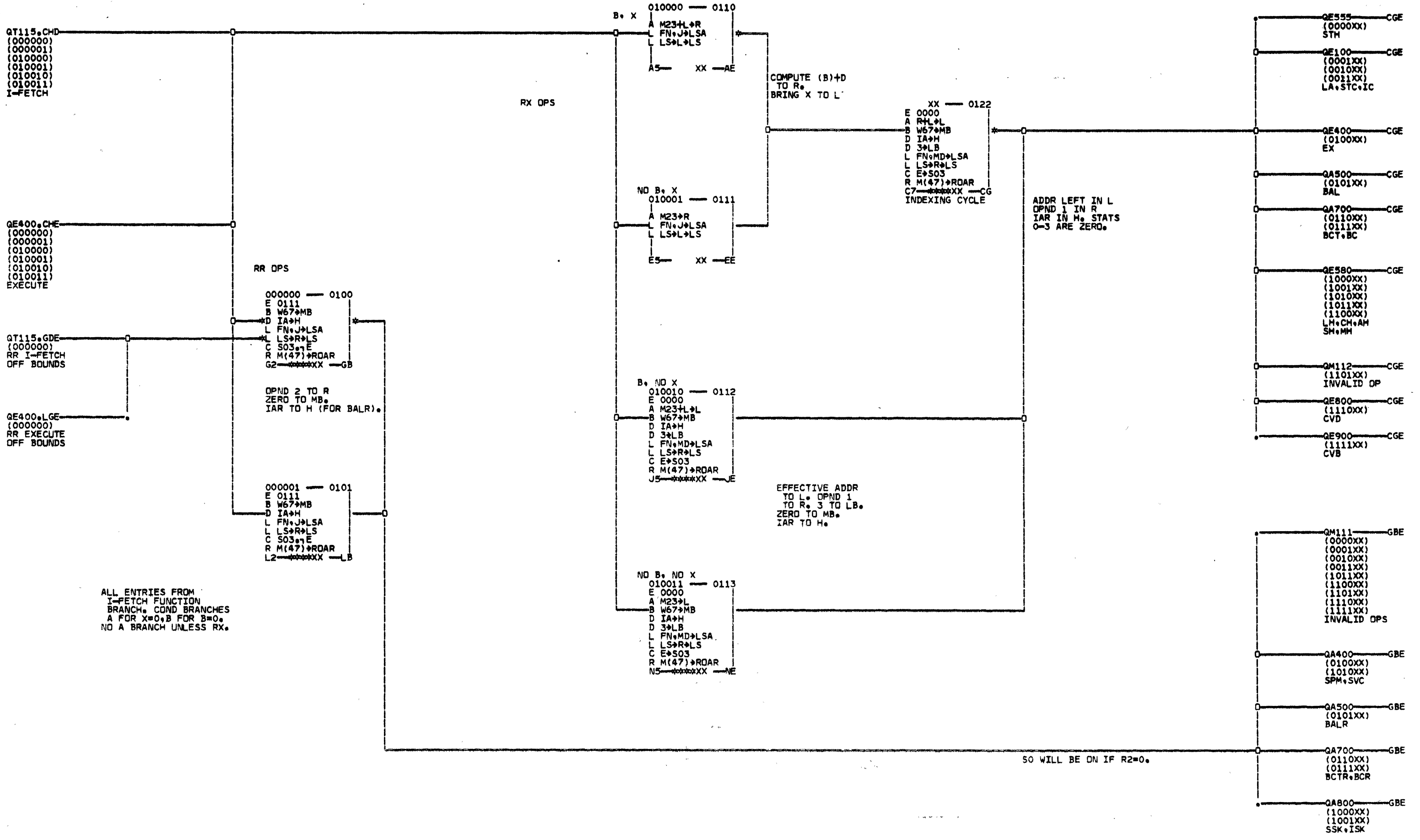


PAGE NO. ST	TITLE	PART NO	EC NO.	FEATURE B/M OR B/MS
** LOGIC TYPE	SYSTEM DIAGRAMS		0	
QA110	RR-RX CTRL 0-4 2ND LEVEL	V000	5364910	255449
QA111	RR-RX FIXED POINT 2ND LEVEL	V000	5364716	255449
QA400	SVC AND SPM RR CTRL OP	V000	5364717	255449
QA500	BRANCH AND LINK RR RX	V000	5364718	255449
QA700	BRANCH ON CGND COUNT BC	V000	5364719	255449
QA800	STR PROT INSTRS SS ISK	V000	5364720	257389
QB100	FXD PT RR LOADS LPR LNR	V000	5364721	255449
QB400	RR RX AND OR EXCL OR	V000	5364722	255449
QB500	COMPARES ALGEBRAIC LOGIC	V000	5364723	255449
QB730	ADD SUBTRACT RR RX	V000	5364724	255449
QB750	LOGICAL ADD SUBTRACT RR	V000	5364725	255449
QB800	FIXED POINT MULTIPLY HA	V000	5364726	255449
QB801	FIXED POINT MULT POST LOOP	V000	5364727	255449
QB900	FIXED POINT DIV PRE LOOP CO	V000	5364728	255449
QB901	FIXED POINT DIV DR D DIVIDE	V000	5364729	255449
QB902	FIXED POINT DIVIDE	V000	5364730	255785
QC030	INV MOVE OP DECODE E8	VC03	5379138	282229 .W. 5379137
QC031	DATA ADD DECO	VC03	5379139	282229 .W. 5379137
QC032	DATA HANDLING	VC03	5379140	282229 .W. 5379137
QE100	RX CTRL LOAD ADDR INS ST	V000	5364731	255449
QE400	EXECUTE INSTRUCTION	VC03	5379141	282229 .W. 5379137
QE555	HALF WORD STORE STH	V000	5364733	255449
QE580	HALF WORD OPS CCMCN SET UP	V000	5364734	255449
QE800	CONY TO DEC FROM BINARY	V000	5364735	255449
QE802	CONY TO DEC FROM BINARY	V000	5364736	255449
QE900	CONY TO BINARY FROM DEC	V000	5364737	255449
QE901	CONY TO BINARY FROM DEC	V000	5364738	255449
QF100	FXD PNT RX LOAD STORE	V000	5364739	255449
QG010	FL PT OPS DECO 2ND LVL	V000	5449051	258430
QG100	LNG SHRT FL PNT LDS REF	V000	5449052	258430
QG200	FLTNG PNT HALVE RR LONG	V000	5449053	258545
QG300	FLTNG PNT LD STR RR RX FCR	V000	5449054	258430
QG400	FLTNG PNT ADD SUB CCMP RR	V000	5469501	258430
QG401	FLTNG PNT ADD SUB CCMP EXP	V000	5449055	258430
QG402	FLTNG PNT ADD SUB CCMP CAL	V000	5449056	258430
QG403	FLTNG PNT ADD SUB COMP PRE	V000	5449057	258430
QG404	FLTNG PNT ADD SUB COMP PRE	V000	5449058	258430
QG405	FLTNG PNT ADD SUB COMP PRE	V000	5449059	258430
QG406	FLTNG PNT ADD SUB COMP	V000	5449060	258545
QG407	LONG FL ADD FRACT TST C	V000	5449061	258430
QG408	LONG FLTNG ADD SUBT CCMP O	V000	5449062	258545
QG409	FLTNG PNT ADD SUBT O TST CH	V000	5449063	258545
QG416	FLTNG PNT ADD SUBT COMP	V000	5449064	258430
QG420	FLTNG PNT ADD SUBT	V000	5449065	258545
QG500	FLTNG PNT DIV SHRT RR RX PR	V000	5449066	258545
QG501	FLTNG PNT DIV EXPONENT SUBT	V000	5449067	258430
QG502	FLTNG PNT DIV LOOP FOR FR	V000	5449068	258430
QG503	FLTNG PNT DIV POST NORMAL	V000	5449069	258545
QG700	FLTNG PNT MULT RR RX FORM	V000	5364759	255449
QG701	FLTNG PNT MULT MULTIPLY L	V000	5364760	255449
QG702	FLTNG PNT MULT POST NCRM O	V000	5449070	258545
QG800	FLTNG PNT LONG MULTIPLY DIVI	V000	5449071	258430
QG801	FLTNG PNT LONG MULTIPLY DIVI	V000	5449072	258430
QG802	FLTNG PNT LONG MULTIPLY DIVI	V000	5449073	258430
QG803	FLTNG PNT LONG MULTIPLY LDR	V000	5449074	258430
QG804	FLTNG PNT LONG MULTIPLY MDR	V000	5449075	258430
QG805	FLTNG PNT LONG MULT MDR MD	V000	5449076	258430
QG900	FLTNG PNT LONG DIVIDE DDR DD	V000	5449077	258430
QG902	FLTNG PNT LONG DIVIDE DDR DD	V000	5449078	258430
QJ011	RS SI 2ND LVL I FETCH	V000	5364769	255449
QJ080	SHIFT RIGHT LOGICAL SRL	V000	5364770	255449
QJ090	SHIFT LEFT LOGICAL SLL	V000	5364771	255449
QJ100	SHIFT RIGHT ALGEBRAIC S	V000	5364772	255449
QJ110	SHIFT LEFT ALGEBRAIC SL	V000	5364773	255449
QJ120	SHIFT RIGHT DOUBLE LOGIC	V000	5364774	255449
QJ130	SHIFT LEFT DOUBLE LOGIC	V000	5364775	255449
QJ140	SHIFT RIGHT DOUBLE ALGEB	V000	5364776	255449
QJ150	SHIFT LEFT DOUBLE ALGEB	V000	5364777	255449
QJ200	LOAD PSW LPSW SET SYST	V000	5364778	255449
QJ600	BRANCH ON INDEX BXH BXL	V000	5364780	255449
QK222	SI MOVE LOGICAL OPS MVI	V000	5364781	255449
QK300	TEST AND SET	V000	5364909	255449
QK555	SI COMP CLI TST UNDER M	V000	5364782	255449
QK666	LOAD STORE MULTIPLE LM	V000	5364783	255449
QK700	START I/O HALT I/O TST I/O	V000	5364784	255449
QK701	DUMMY START I/O	V000	5364785	259855
QK702	TIME OUT CHECK LOOP IB TS	V000	5364786	259855
QK703	MPX CHAN INTERRUPT HAND	V000	5364787	255449
QK704	MPX CHAN INTERRUPT HAND	V000	5364788	255449
QK705	LOAD 64 CPU MODE	V000	5364789	255449
QK800	IPL	V000	5364790	255449
QK801	IPL STATUS ANALYSIS	V000	5364791	255449

Dec 22

TOTAL PAGES THIS VOLUME 82

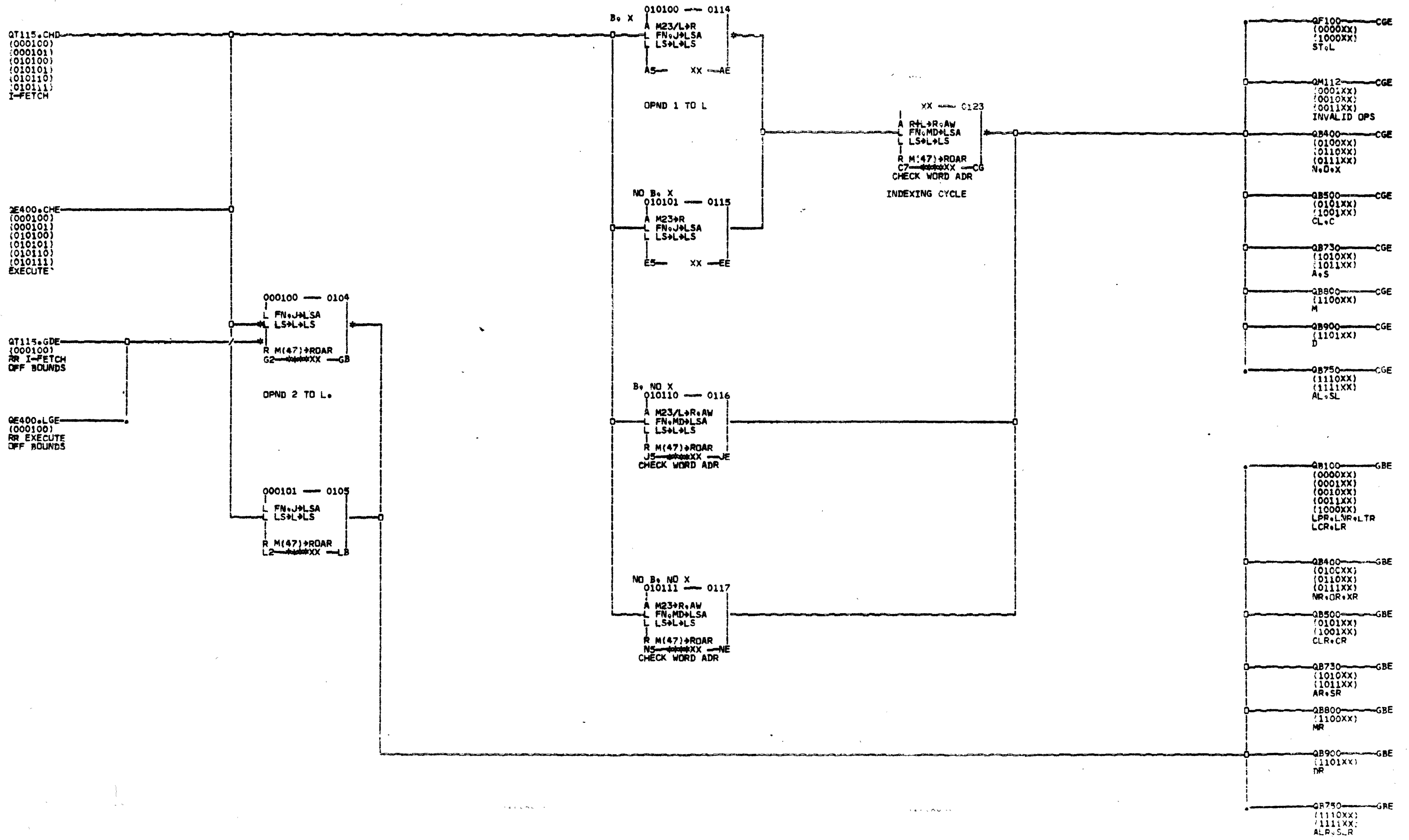
Dec 72



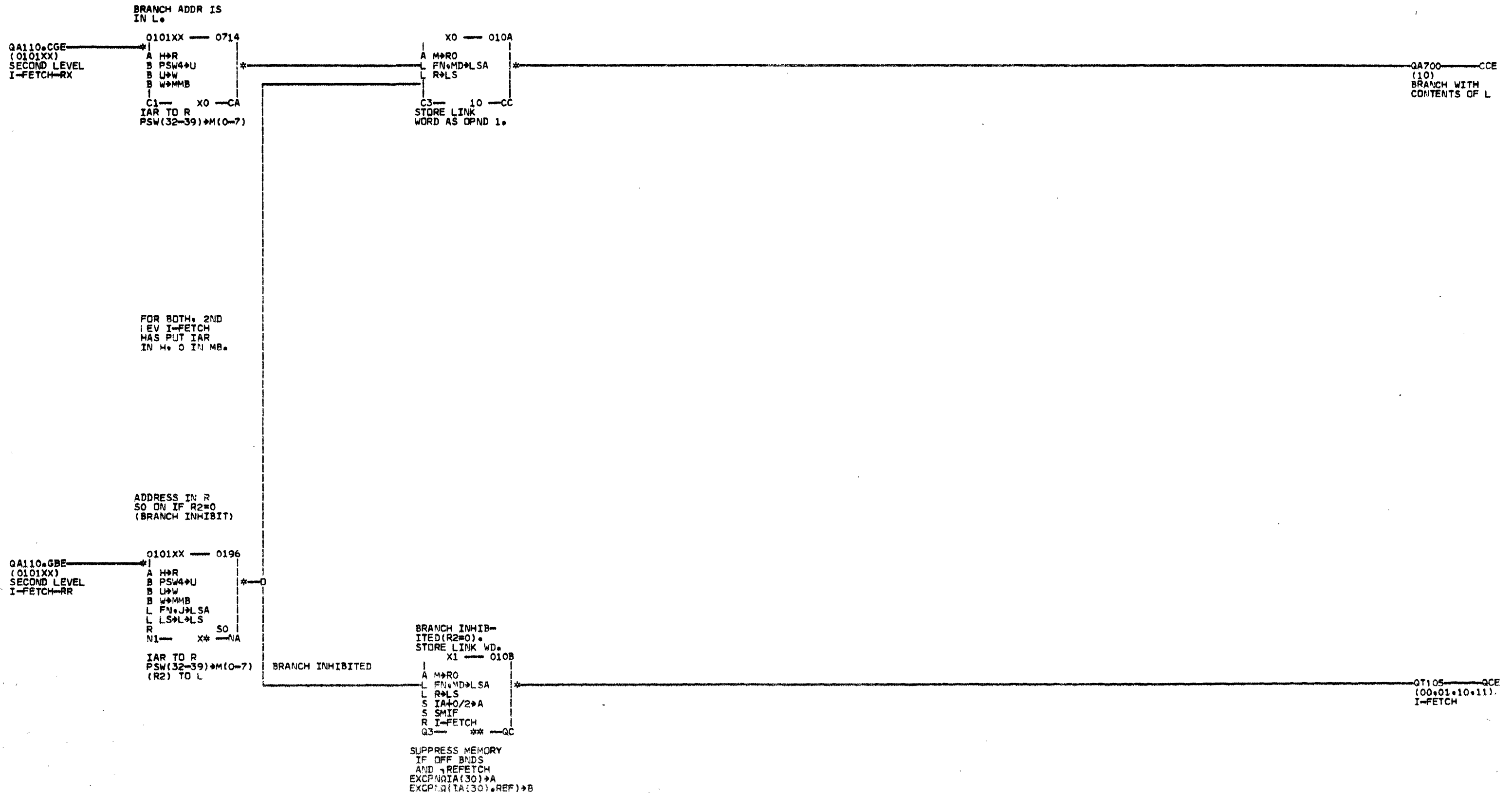
ALL ENTRIES FROM
 I-FETCH FUNCTION
 BRANCH, COND BRANCHES
 A FOR X=0, B FOR B=0.
 NO A BRANCH UNLESS RX.

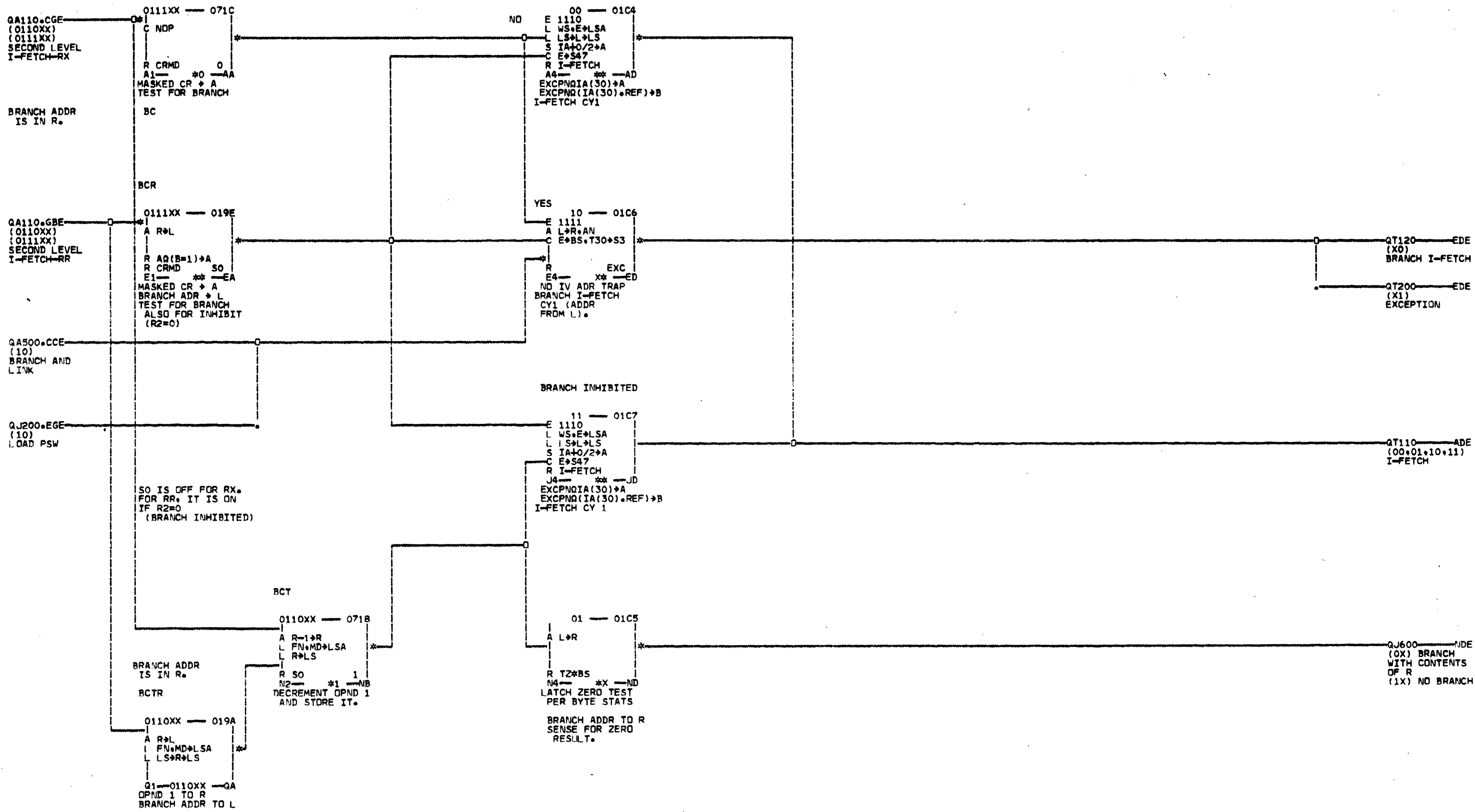
SO WILL BE ON IF R2=0.

QA110



Q
A
1
1
1

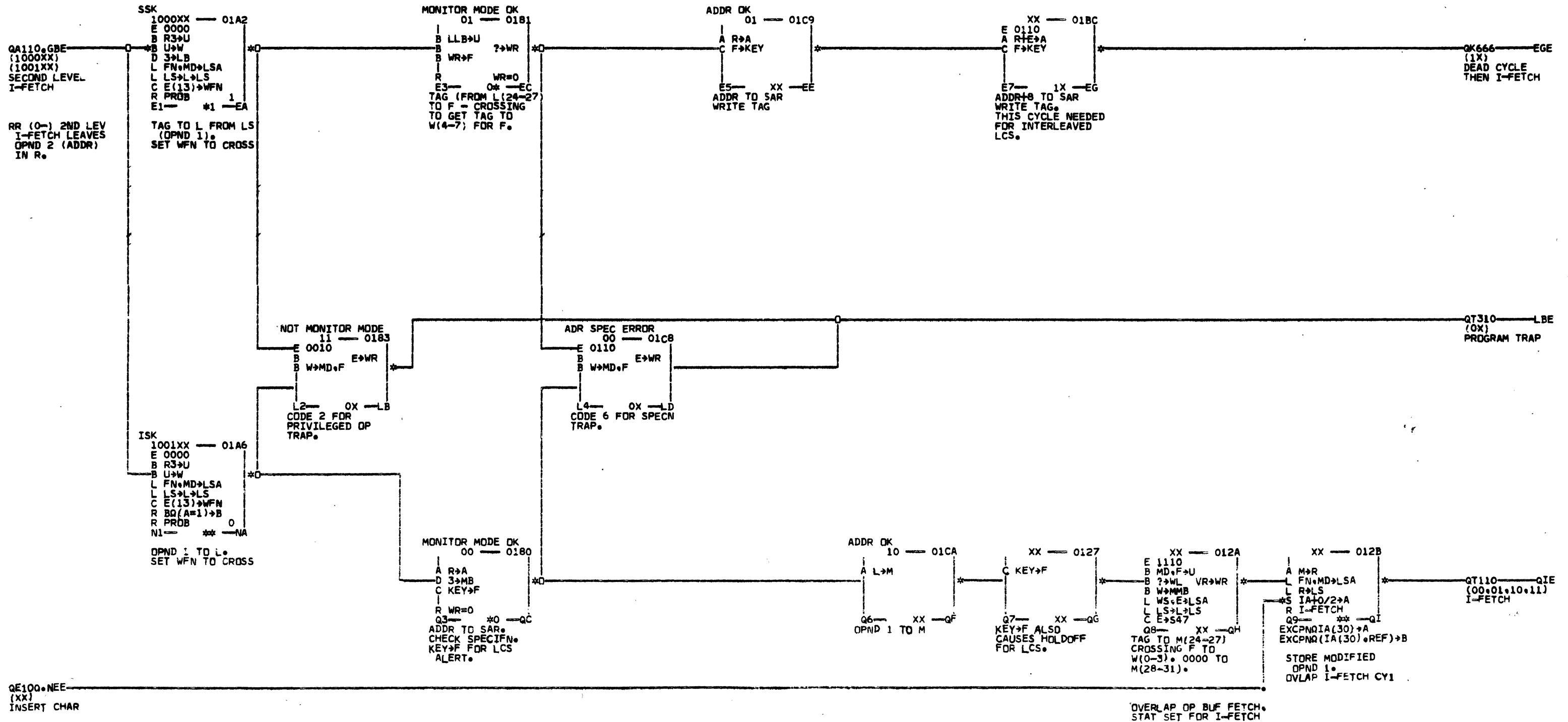




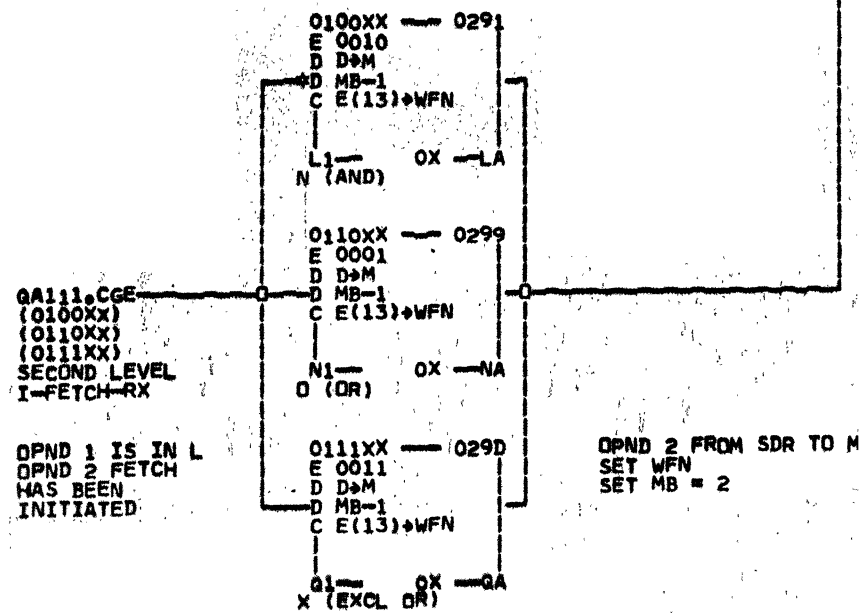
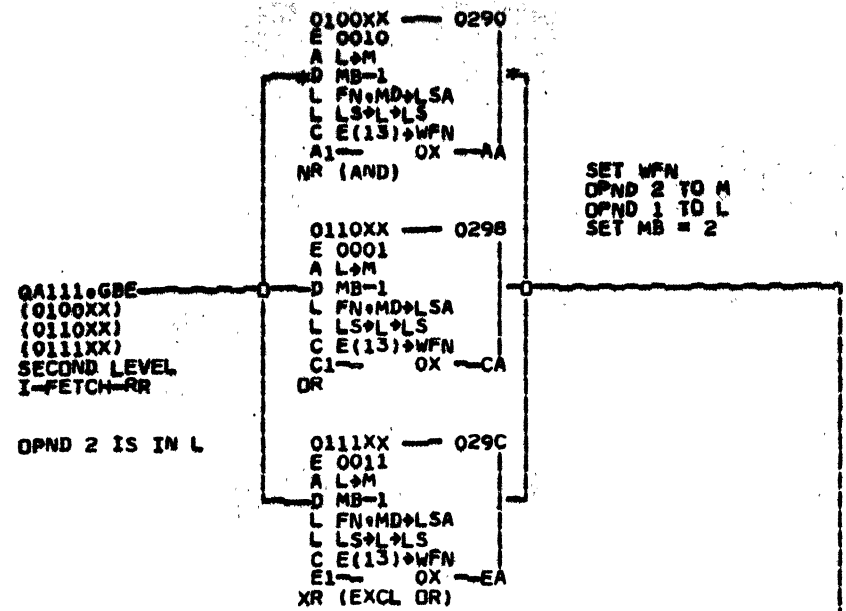
QA700

THESE INSTRS WORK WITH
 OPND 1 IN THE FORM
 XX XX XX T2 WHERE
 X - UNDISTURBED
 T - 4-BIT TAG
 Z - 0 FOR INSERT KEY
 - UNUSED FOR SSK.

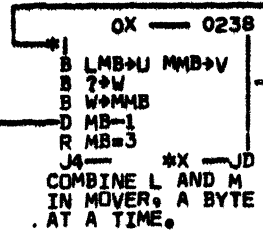
OPND 2 IS ANY ADDR.
 IN THE BLOCK (2048 BYTES)
 WHOSE TAG IS CONCERNED.
 IT MUST HAVE 4 LO ZERO
 BITS TO AVOID SPECN TRAP.



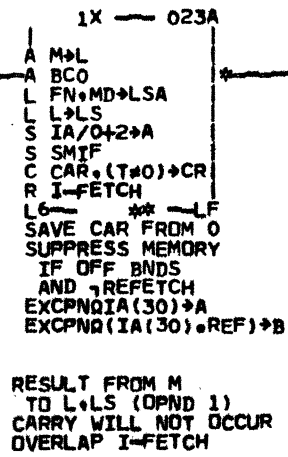
Q
A
B
O
O



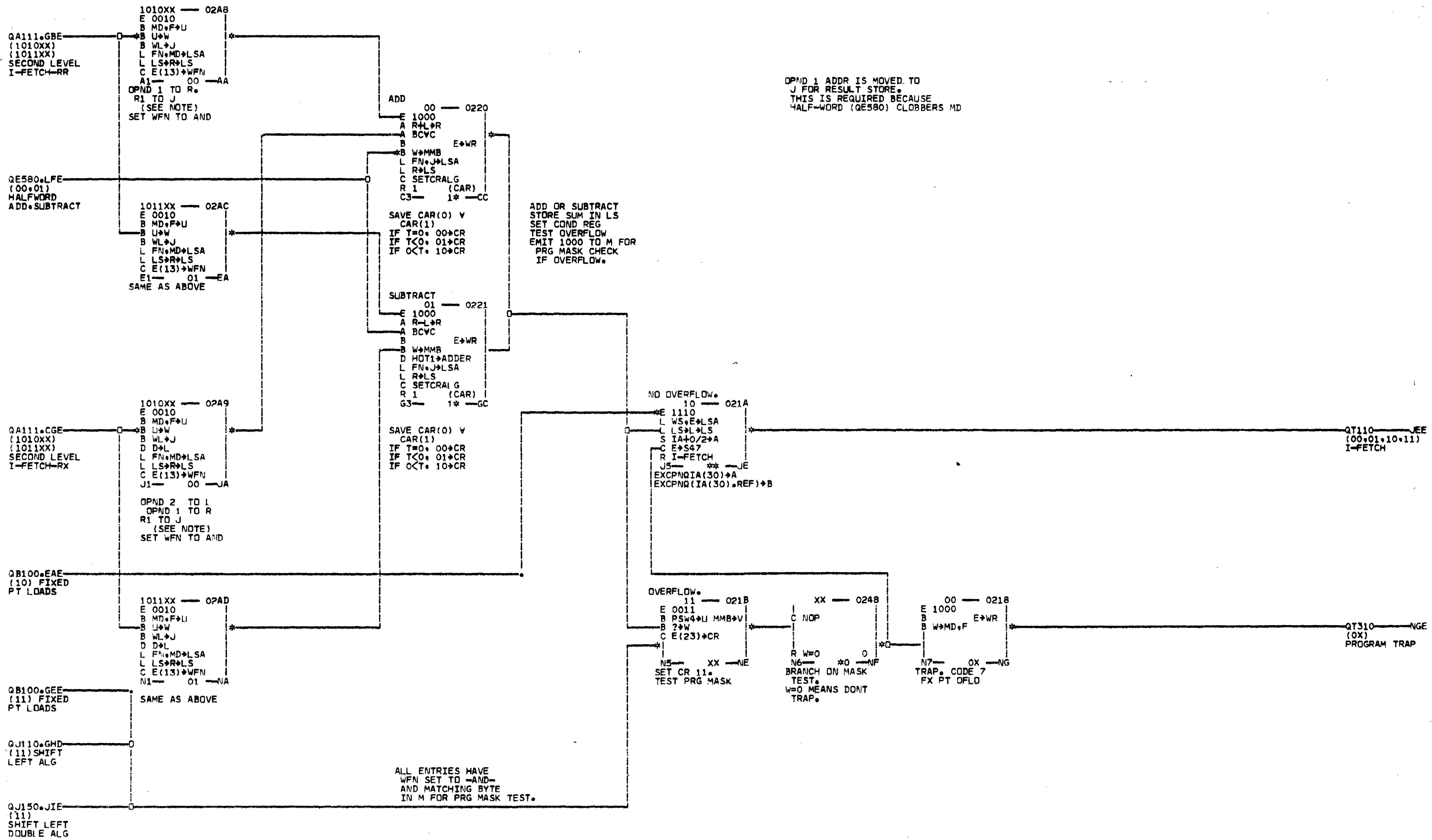
MB=2+1+0, LOOP 3 TIMES.

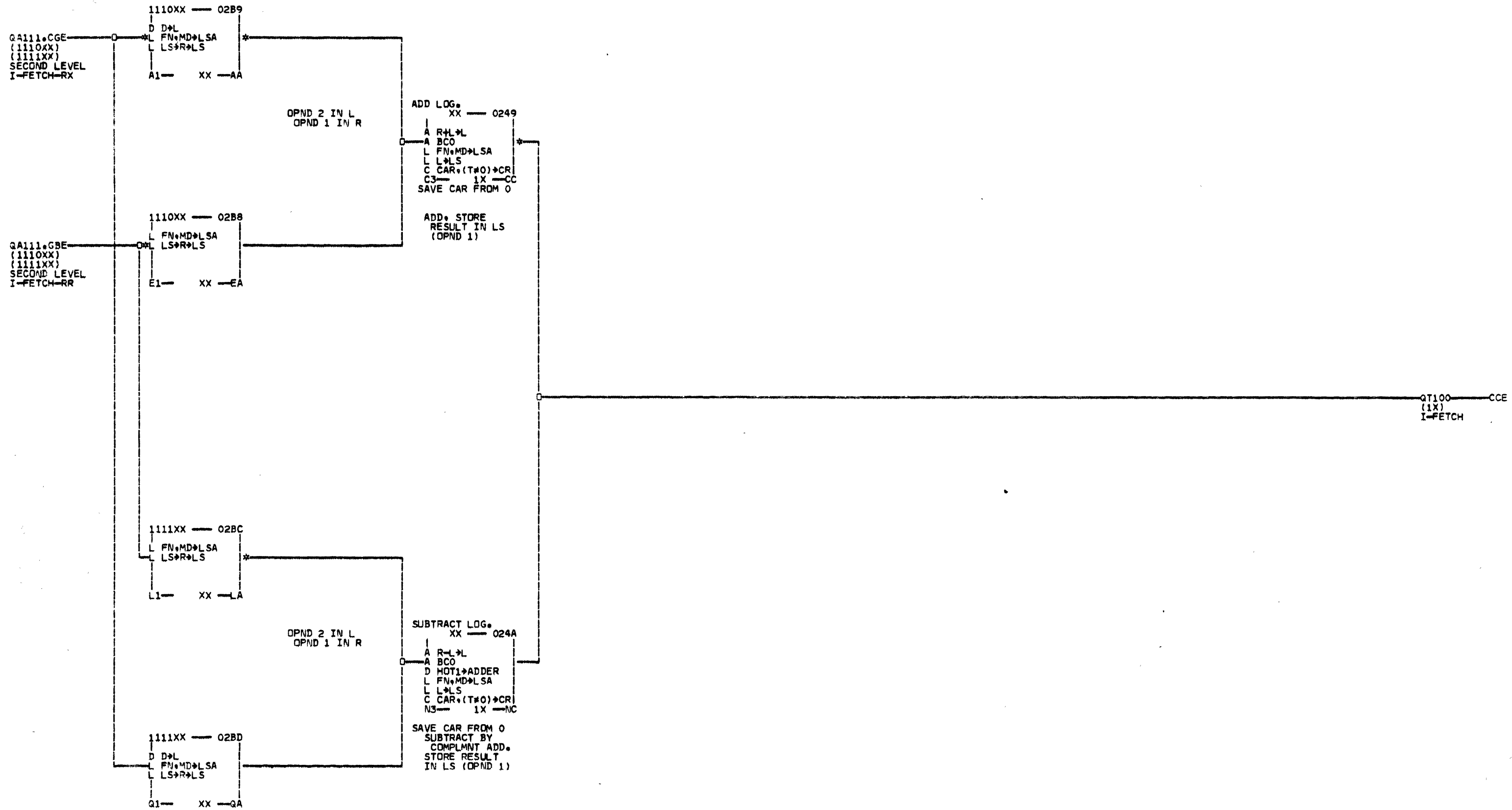


4TH TIME,
LOOP FINISHED



QT105 LFE
(00,01,10,11)
I-FETCH





QB750

254760
255449

12/21/64
09/20/65

MACH
NAME
MODE
P.N.
IBM CORP.

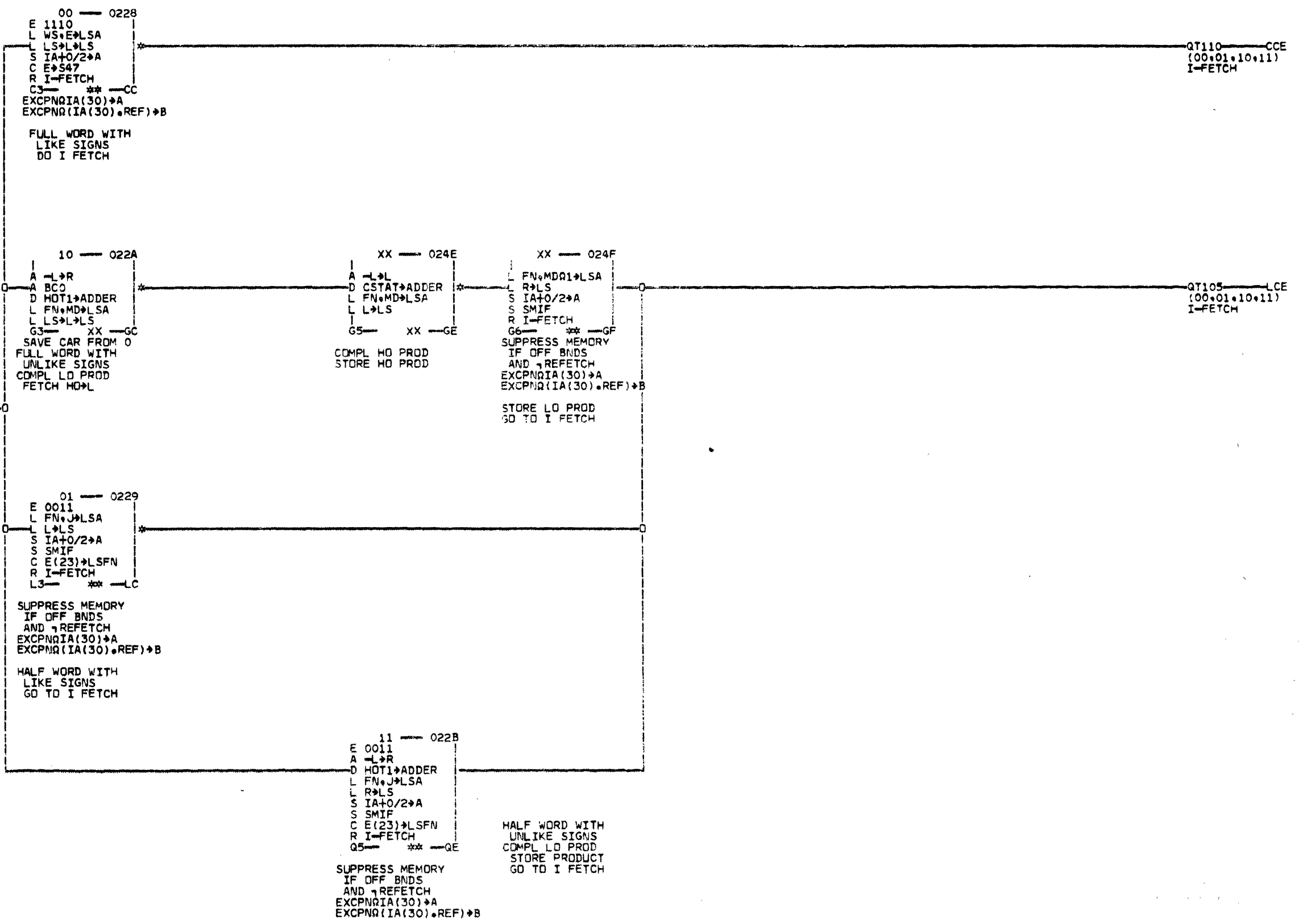
C2050
MANUAL
5364725
SDD

DATE 10/01/65
LOG 902

SHEET 1
VERSION

QB750

LOGICAL ADD/SUBTRACT RR/RX
(AL.SL ALR.SLR).



QG701.AFE
(10)
MULTIPLY LOOP

QB900.QDE
(X0,X1)
PRE-LOOP
SETUP

* ILLEGAL WD SIZE

X1 — 0297
E 1001 E→WR
B W→MD,F
C2 — 0X — CB

* IRPT CODE TO F
GO TO IRPT ROUTINE

* COMPL ADD 1

X0 — 0296
A R-L→R F→SL1→F
D HOT1→ADDER
R (Z00)
G2 — X* — GB

* DVD=DVR TO R,SL1

* TRUE ADD 2

X1 — 029B
A R+L→R F→SL1→F
R (Z00)
G3 — X* — GC

* DVD+DVR TO R,SL1

* TRUE ADD 3

X1 — 029F
A R+L→R F→SL1→F
R (Z00)
G4 — X* — GD

* DVD+DVR TO R,SL1

* TRUE ADD 4

X1 — 02A3
A R+L→R F→SL1→F
R G1=0 (Z00)
G5 — ** — GE

* DVD+DVR TO R,SL1

* TRUE ADD 1

XX — 0255
A R+L→R F→SL1→F
D G1-1
R (Z00)
L2 — X* — LB

* DVD+DVR TO R,SL1
REDUCE LOOP COUNT

* COMPL ADD 2

X0 — 029A
A R-L→R F→SL1→F
D HOT1→ADDER
R (Z00)
L3 — X* — LC

* DVD=DVR TO R,SL1

* COMPL ADD 3

X0 — 029E
A R-L→R F→SL1→F
D HOT1→ADDER
R (Z00)
L4 — X* — LD

* DVD=DVR TO R,SL1

* COMPL ADD 4

X0 — 02A2
A R-L→R F→SL1→F
D HOT1→ADDER
R G1=0 (Z00)
L5 — ** — LE

* DVD=DVR TO R,SL1

* LOOP FINISHED
CORRECT REMAINDER

11 — 022F
A R→R F→SR1→Q
C6 — XX — CF

* RMDR TO R SR1(IN-
SERT 1 TO SIGN TO
RECOVER SPILL)

* LOOP FINISHED
NO RMDR CORRECTION

10 — 022E
A R→L SR1→Q
L FN→MD→LSA
L LS→R→L→LS
R RSGNSI
G6 — X* — GF
RMDR TO L SR1
AND TO LS PER MD
(ASSUMING A
POSITIVE DVD)

*SAVE ORIGINAL
REGISTER DATA
IN R

* REPEAT LOOP WITH
NEXT ADD TRUE

01 — 022D
A M→M F→SL4→F
L6 — XX — LF

*BRING NEXT DIGIT
OF LD DVD INTO
F
TAKE THE QUOT
DIGIT IN INVERT-
ED FORM INTO M

* REPEAT LOOP WITH
NEXT ADD COMPL

00 — 022C
A M→M F→SL4→F
D G1-1
G6 — X0 — QF

BRING NEXT DIGIT
OF LD DVD INTO
F
TAKE THE QUOT
DIGIT IN INVERT-
ED FORM INTO M
REDUCE LOOP COUNT

QT310 — CBE
(OX)
PROGRAM TRAP

QB902 — GFE
(X0,X1)
POST-LOOP
ROUTINE

XX — 0256
A R+L→L
L FN→MD→LSA
L LS→R→L→LS
R RSGNSI
CB — ** — CH

* RMDR+DVR TO L
AND LS PER MD
(STORE ASSUMING A
POS DVD)

*SAVE ORIGINAL
REGISTER DATA
IN R

254760
255449

12/21/64
09/20/65

MACH
NAME
MODE
P.N.
IBM CORP.

C2050
MANUAL
5364729
SDD

DATE 10/01/65
LOG 902

SHEET
VERSION

1 QB901

FIXED POINT DIVIDE (DR,D)
DIVIDE LOOP

QB901.GFE
(X0,X1)
DIVIDE LOOP

* DVD WAS NEGATIVE

X1 — 02A7
A —L→L
D HOT1→ADDER
L FN,MD→LSA
L L→LS
C2 — X0 — CB
* RMDR (COMPL) TO R
AND LS PER MD

* DVD WAS POSITIVE

X0 — 02A6
A M→L+M
A —L→R
L FN,MDR1→LSA
L L→LS
G3 — XX — GC
QUOT TO L+M (SL4)
(BRINGS LAST
DIGIT IN FROM F)
*SAVE ORIGINAL DATA
IN WS1

XX — J257
A —L→R
L FN,MDR1→LSA
L L→LS
R VSGNS T(0)
G5 — ** — GE
LSGNWRSGN→A
* (ASSUME SIGNS ARE
EVEN)
INVERT QUOT TO R
AND TO LS
*SAVE ORIGINAL DATA
IN L

*RANGE HAS BEEN
EXCEEDED

00 — 0230
L FN,MDR1→LSA
L L→LS
C6 — XX — CF
* RESTORE ORIGINAL
LD DVD IN GEN REG.

XX — 02C4
L WS1→LSA
LS→R→LS
C7 — XX — CG
* FETCH HO DVD TO R

XX — 02C5
E 1001 E→WR
B W→MD,F
L FN,MD→LSA
L R→LS
C8 — 0X — CH
* RESTORE HO DVD
IN GEN REG
SET IRPT CODE

QT310 — CHE
(0X)
PROGRAM TRAP

* RANGE
EXCEEDED

*SIGNS UNLIKE
TEST RANGE

10 — 0232
A M→T
A BCVC
D HOT1→ADDER
R 0 (CAR)
G7 — ** — GG
SAVE CAR(0) ↓
CAR(1)
*+1 TO T FOR A
ZERO TEST

*RESULTS IN RANGE

*SIGNS ALIKE RESULTS
WITHIN RANGE

01 — 0231
E 1110
L WS→LSA
L LS→L→LS
S IA+0/2→A
C E→S47
R I→FETCH
L7 — ** — LG
EXCPNQIA(30)→A
EXCPNQ(IA(30),REF)→B
*DIVIDE IS COMPLETE
DO I→FETCH

QT110 — LGE
(00,01,10,11)
I→FETCH

*SIGNS UNLIKE RESULTS
WITHIN RANGE

11 — 0233
A M→L
D HOT1→ADDER
L FN,MDR1→LSA
L L→LS
S IA+0/2→A
S SMIF
R I→FETCH
Q7 — ** — QG
TWO'S COMPL OF QUOT
TO L, LS

QT105 — QGE
(00,01,10,11)
I→FETCH

SUPPRESS MEMORY
IF OFF BNDS
AND →REFETCH
EXCPNQIA(30)→A
EXCPNQ(IA(30),REF)→B

081001

QT115.CMD
 (111000)
 (111001)
 FIRST LEVEL
 I-FETCH

111000 C03 0138
 E 0011
 A M23+L
 B V+W
 B W+G
 L WS1+LSA
 L L+LS
 C E(13)+WFN
 A1 XX AA

ADD BASE ADR TO
 DISPLACEMENT TO
 FORM FIRST LEFT
 BYTE ADR (OP1)
 AND PUT TO WS1.
 LENGTH FIELD FROM
 M BYTE 1 TO G1.G2.
 SET WFN TO XOR.

B1=0

QE400.CHE
 (111000)
 (111001)
 EXECUTE

111001 C03 0139
 E 0011
 A M23+L
 B V+W
 B W+G
 L WS1+LSA
 L L+LS
 C E(13)+WFN
 L1 XX LA

FIRST LEFT BYTE
 ADR (OP1) TO L
 AND WS1.
 LENGTH FIELD FROM
 M BYTE 1 TO G1.G2.
 SET WFN TO XOR.

XX C03 000A
 E 1000
 A L+M
 B W+MD.F
 D O+LB
 G3 XX GC

B TO F REG
 OP1 ADR TO M
 SET LB=0

XX C03 000E
 B MD.F+U
 B 7+W
 G4 XX GD

EXCLUSIVE OR
 BETWEEN OP
 CODE AND B.

XX C03 0016
 R
 G5 WR=0
 XX GE

BRANCH ON XOR
 RESULT BETWEEN
 B AND SECOND
 HALF OF OP CODE

OP CODE E8
 X1 C03 0DBB
 E 0001
 S IA+O/2+A
 C E+S47
 R AP(B=1)+A
 R IA(30) S7
 C7 *X CG

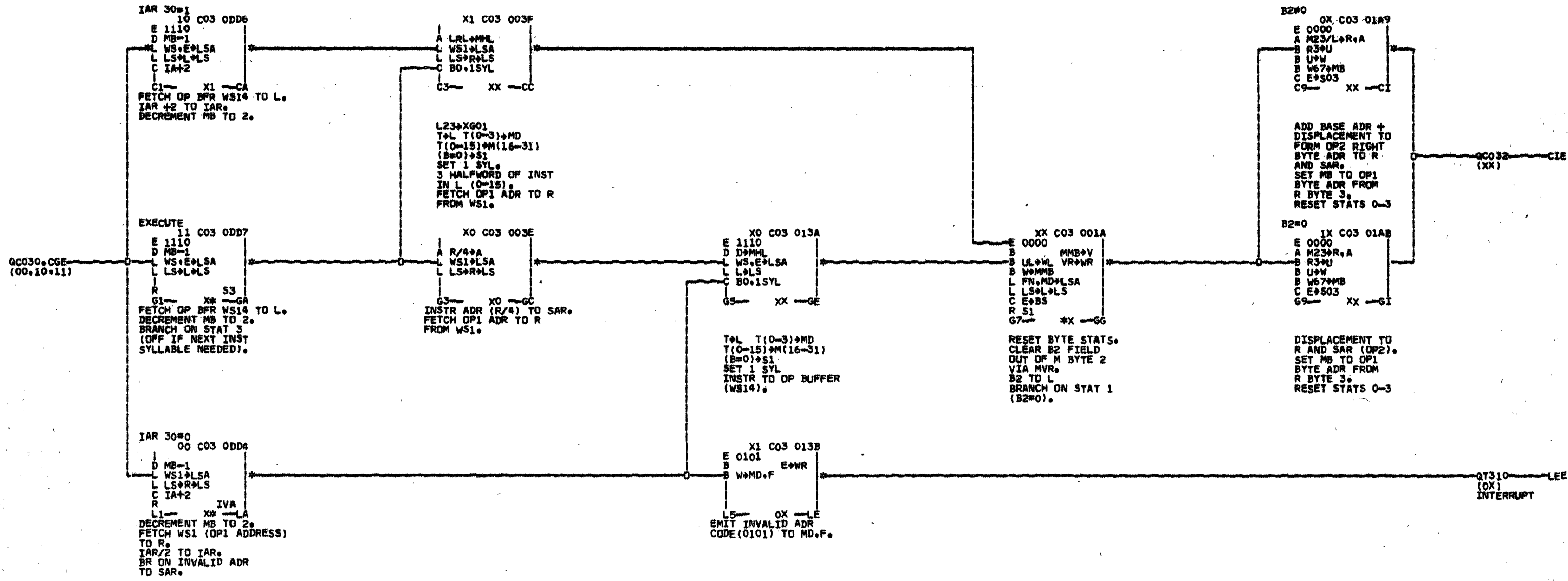
IC TO SAR. SET
 STATS 4-7 TO 0001.
 BRANCH ON HP-WD
 BIT OF IAR AND
 STAT 7 (EXECUTE).

QC031 CGE
 (00.10.11)

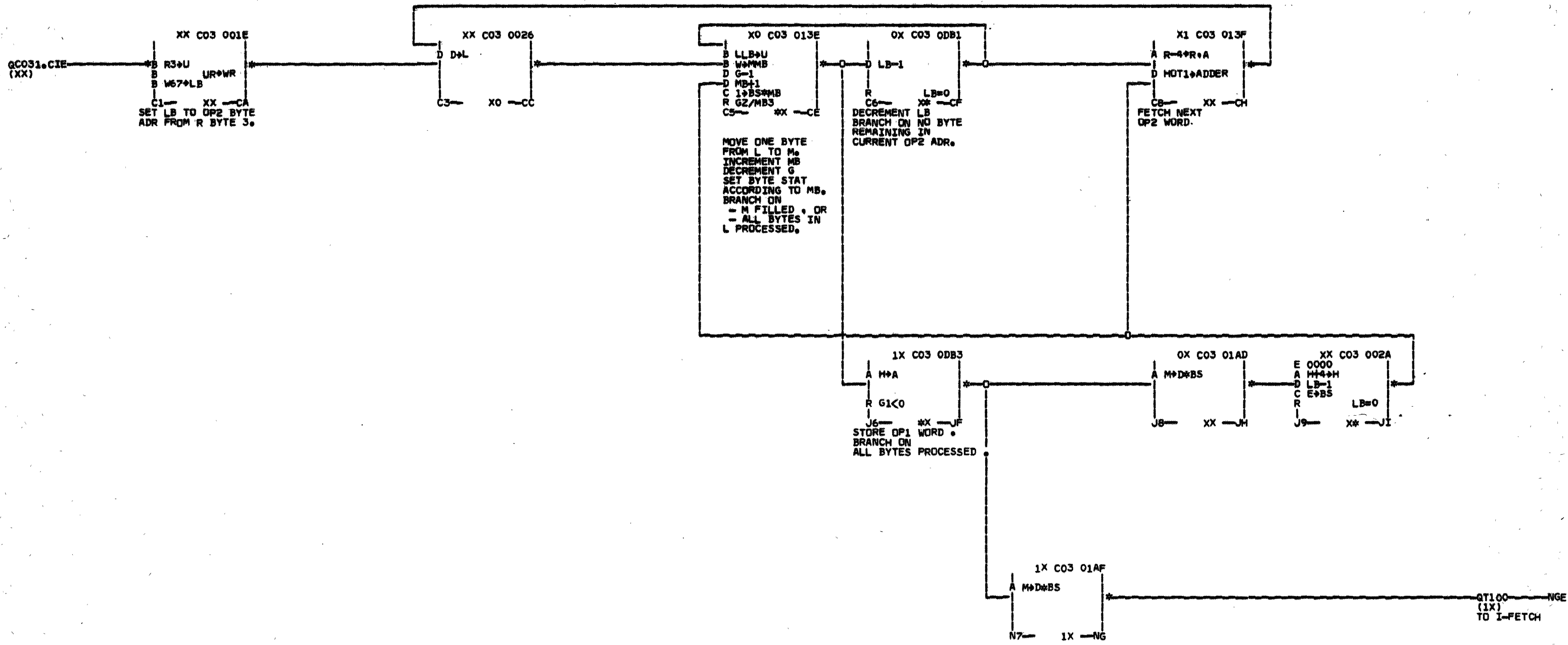
OP CODE # E8
 X0 C03 0DBA
 R S7
 J7 *X JG
 EMIT INVALID OP
 CODE (0001) TO MDF.

QN111 JGE
 (0X.1X)
 INVALID GROUPS

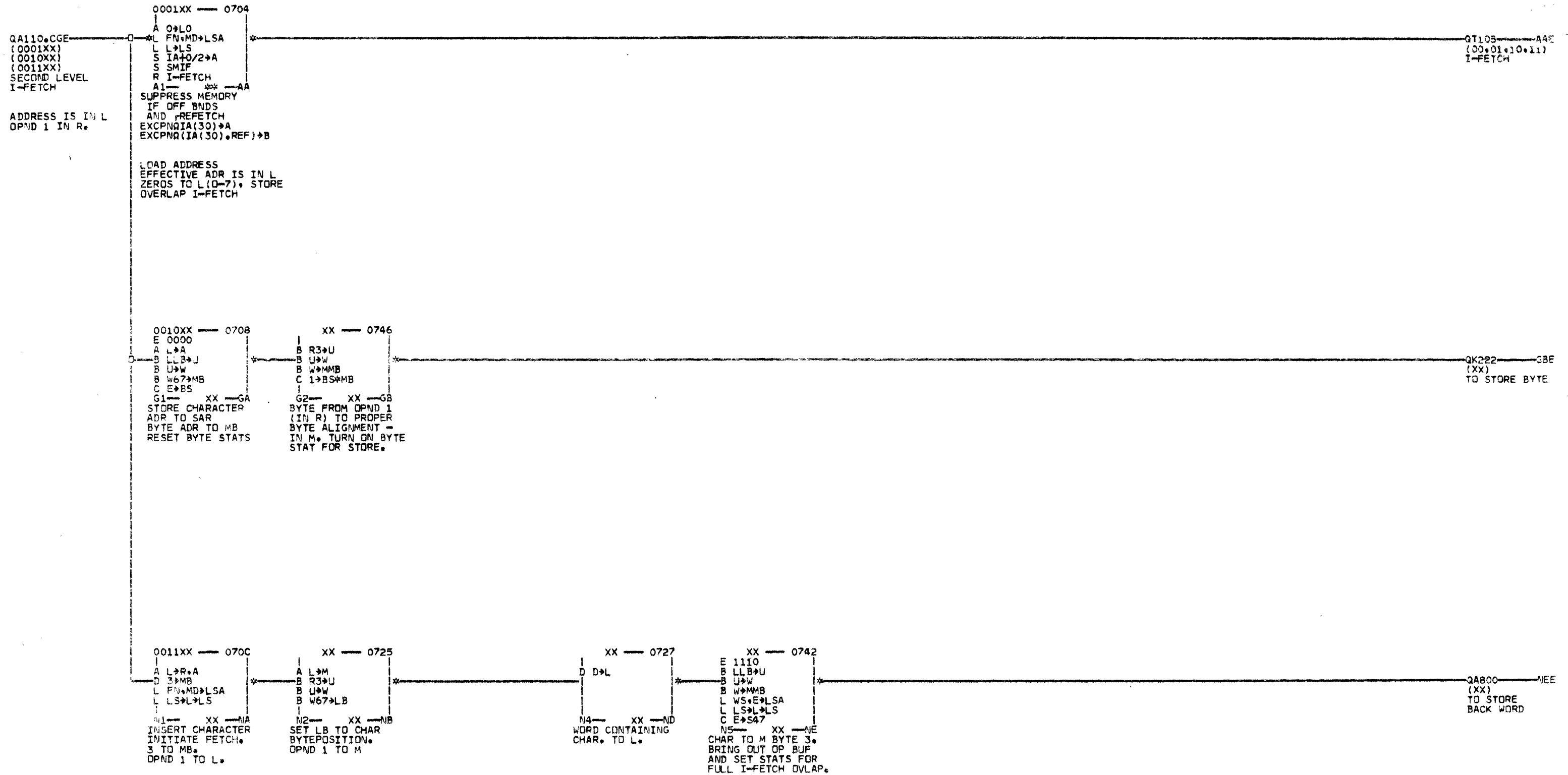
04000



WOOD



NU000



00100

254760
255449

12/21/64
09/20/65

MACH
NAME
MODE
P.N.
IBM CORP.

C2050
MANUAL
5364731
SDD

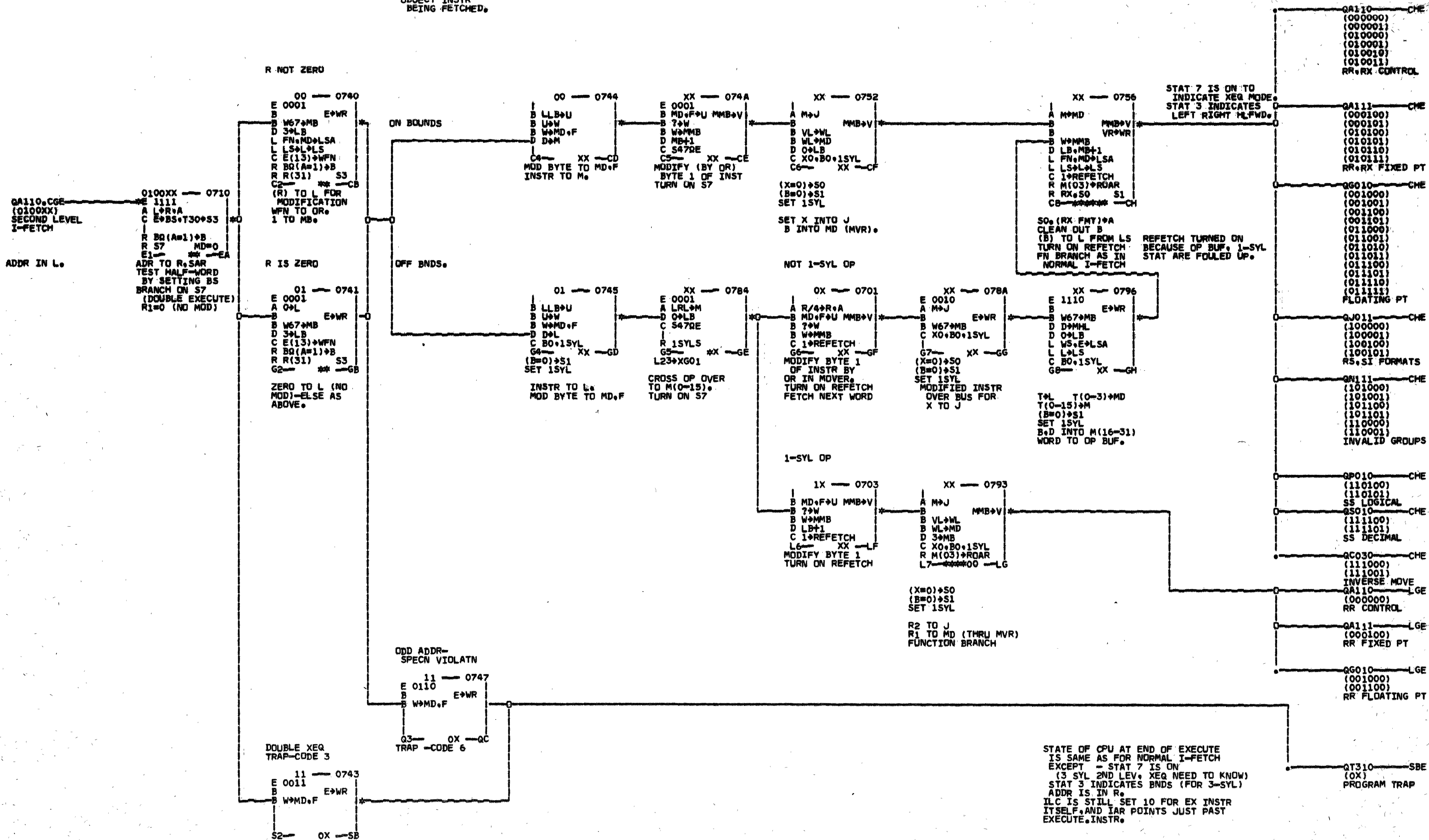
DATE 10/01/65
LOG 902

SHEET
VERSION

1 GE100

RX CONTROL-LOAD ADDRESS.
INSERT/STORE CHARACTER.

OBJECT INSTR
BEING FETCHED.



000000

QA110.CGE
(0000XX)
SECOND LEVEL
I-FETCH

0000XX 0700
E 1100
A L→R
L FN,MD→LSA
L LS→L→LS
C E→BS,T30→S3
E1 XX EA
ADDR TO R. SET
S3 ON BIT 30
FOR LEFT RIGHT
INDICATOR.
OPND TO L.
SET BS FOR LEFT.

XX 0729
A R→A
R BQ(A=1)→B
R R(31) S3
E3 ** EC
ADDR TO SAR.
TEST R31 FOR
LD ADDR BIT-
ILLEGAL.
TEST S3 FOR
LEFT-RIGHT.

LEFT 00 0748
A LRL→D*BS
C5 1X CE
L23→XG01
CROSS L(16-31)
TO LEFT - SDR.
BS PREV SET 1100.

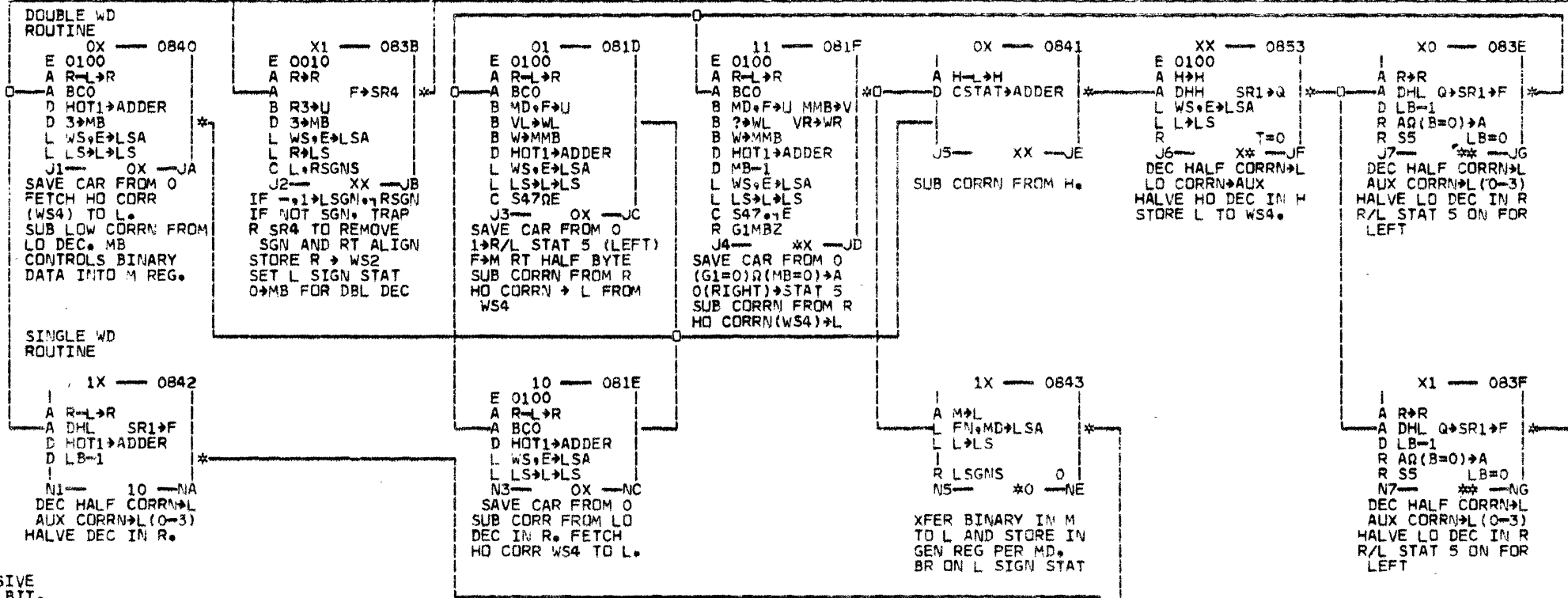
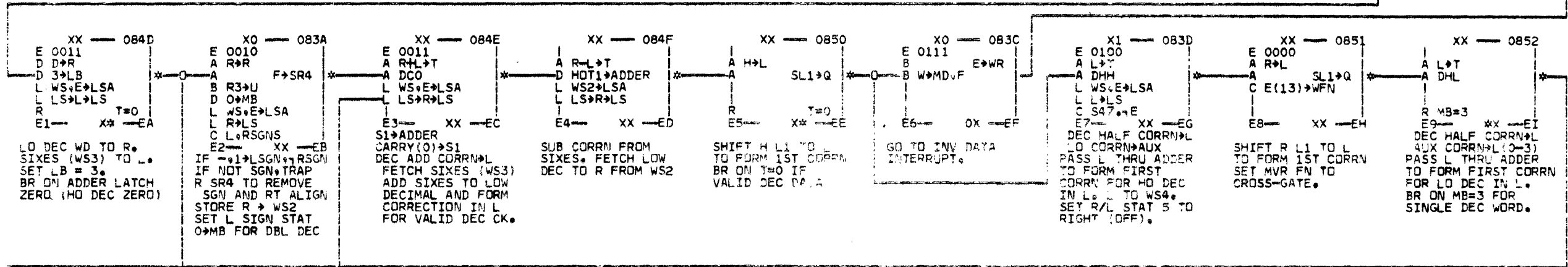
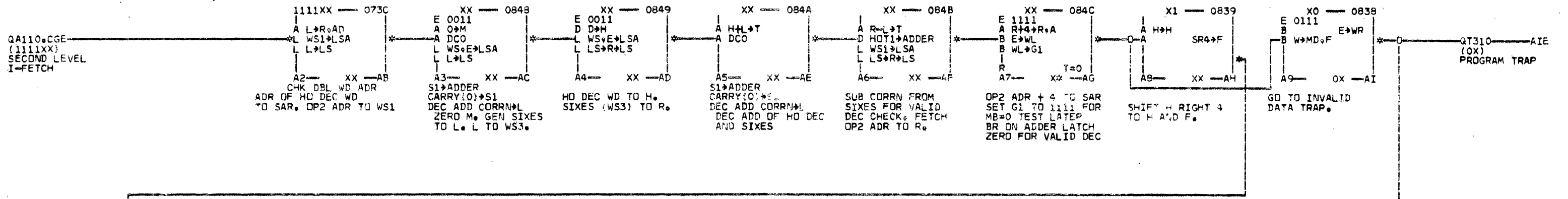
RIGHT 01 0749
E 0011
A L→M
C E→BS
G5 XX GE
SET BS FOR
RIGHT.
OPND TO M.

WRONG SPECN. 11 074B
E 0110
B W→MD→F E→WR
G5 0X GE
PROG TRAP
CODE 6
(SPECN)

QT100 CEE
(1X)
I-FETCH

QT222 GEE
(XX)
TO STORE
HALF WORD

QT310 QEE
(OX)
PROGRAM TRAP



HO DEC WD IN H.
LO DEC WD IN R.
CONVERT BY SUCCESSIVE
SHIFTS RIGHT BY 1 BIT.
SPILL TO F. GEN CORR
TO L.

IF INITIAL HO 7 DEC
DIGITS EQUAL ZERO OR
WHEN CONTENTS OF H
GOES TO ZERO ENTER
SINGLE ROUTINE.

QE901 NGE
(01,10,11)

QE901 NEE
(00,10)

QE901 NAF
(10)

QA111 CGE
 (0000XX)
 (1000XX)
 SECOND LEVEL
 I-FETCH

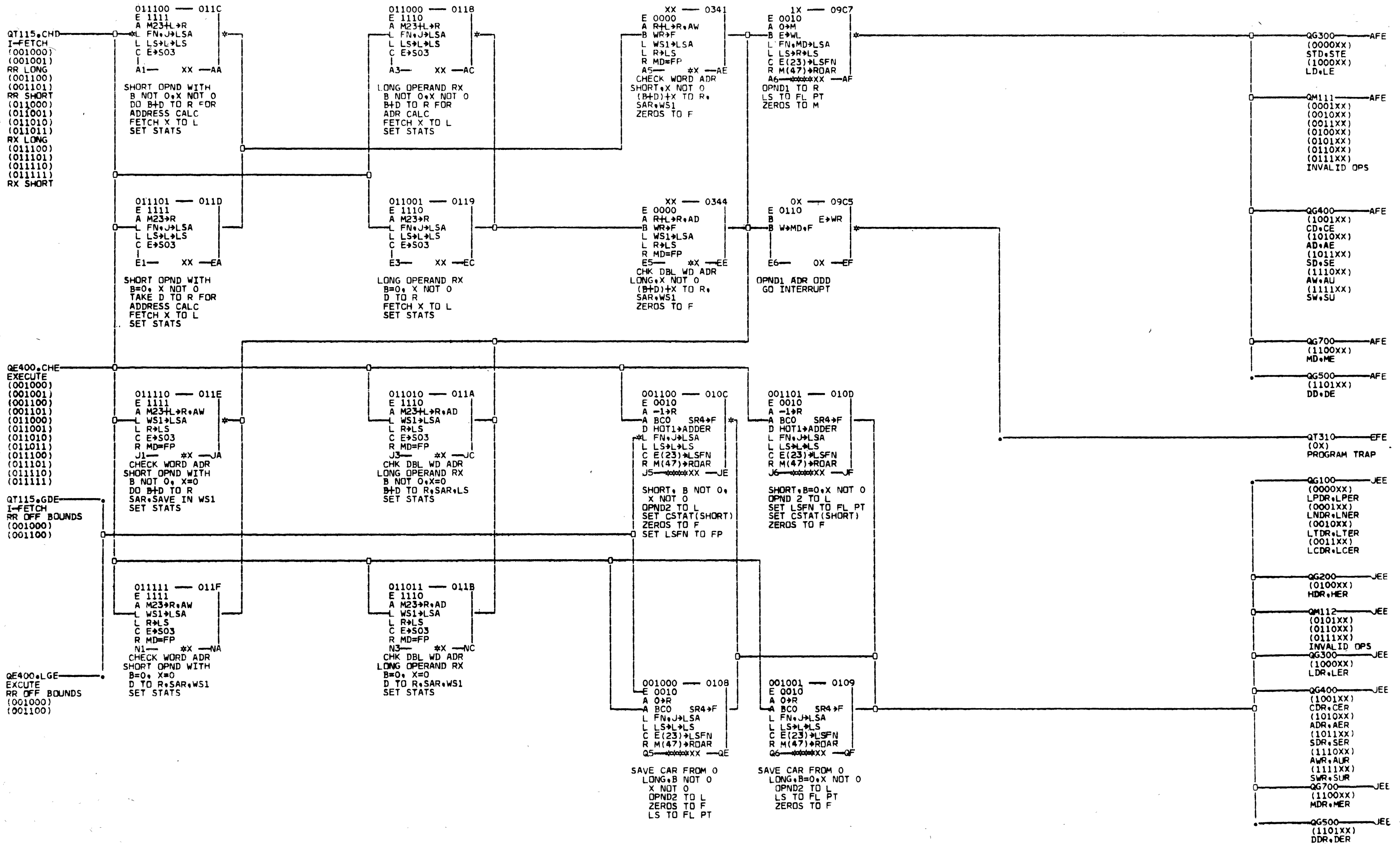
OPND 1 IS IN L.
 STORAGE CYCLE
 (OPND 2) HAS
 BEEN INITIATED.

1000XX — 02A1
 D D→L
 L FN→MD→LSA
 L L→LS
 A1 — 1X — AA
 LOAD
 OPND 2 FROM
 SDR TO L→LS
 (OPND 1 ADDR).

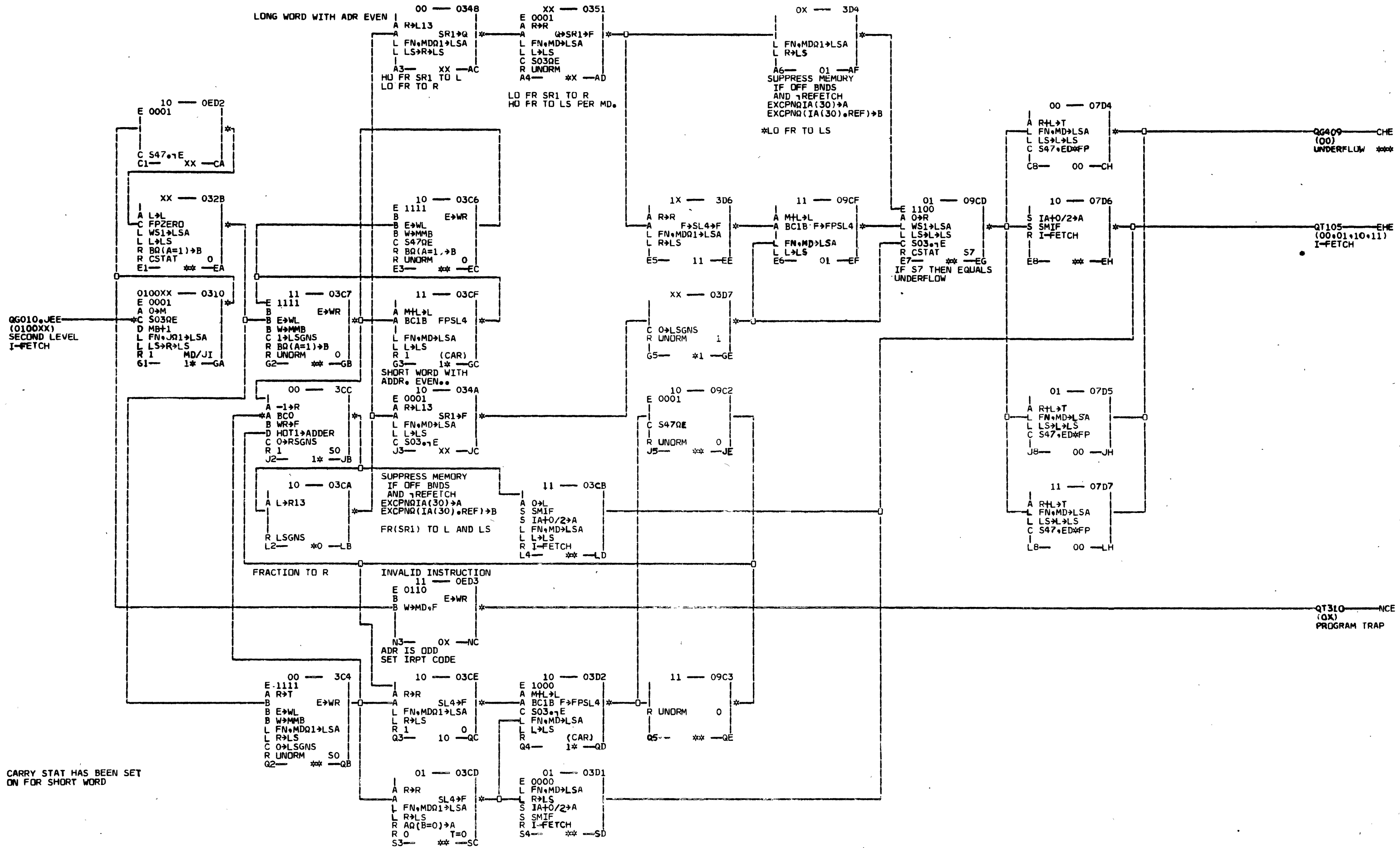
0000XX — 02B1
 A L→D
 J1 — 1X — JA
 STORE
 OPND 1 TO SDR

QT100 — AAE
 (1X)
 I-FETCH

00000



0-000



QG010, JEE
(0100XX)
SECOND LEVEL
I-FETCH

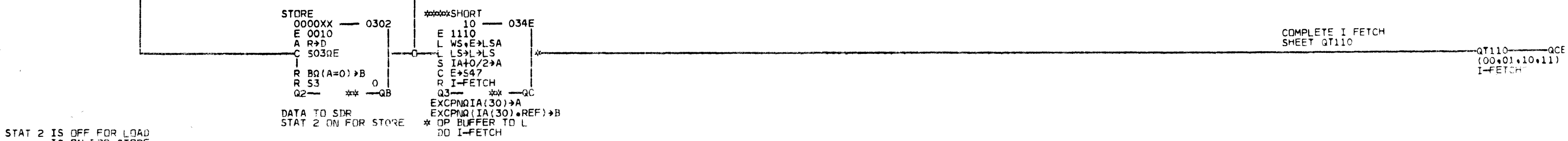
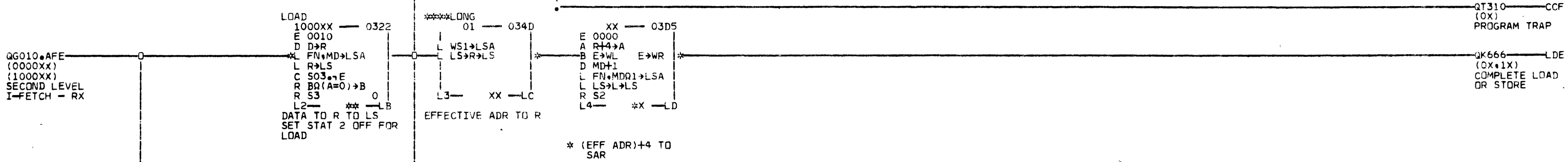
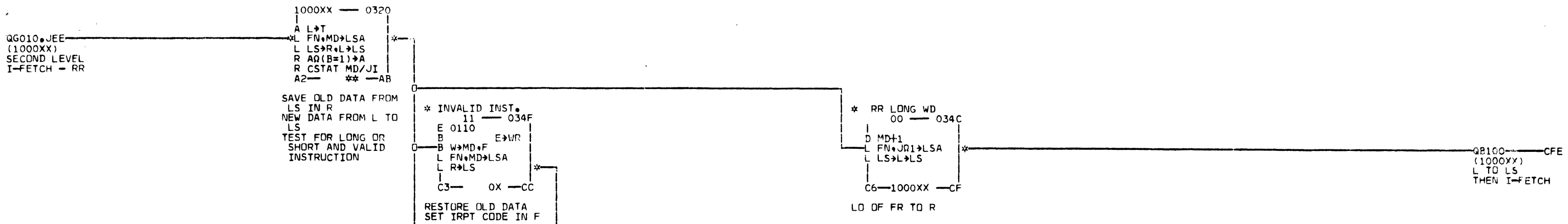
CARRY STAT HAS BEEN SET
ON FOR SHORT WORD

QG409 CHE
(00)
UNDERFLOW ***

QT105 EME
(00,01,10,11)
I-FETCH

QT310 NCE
(0X)
PROGRAM TRAP

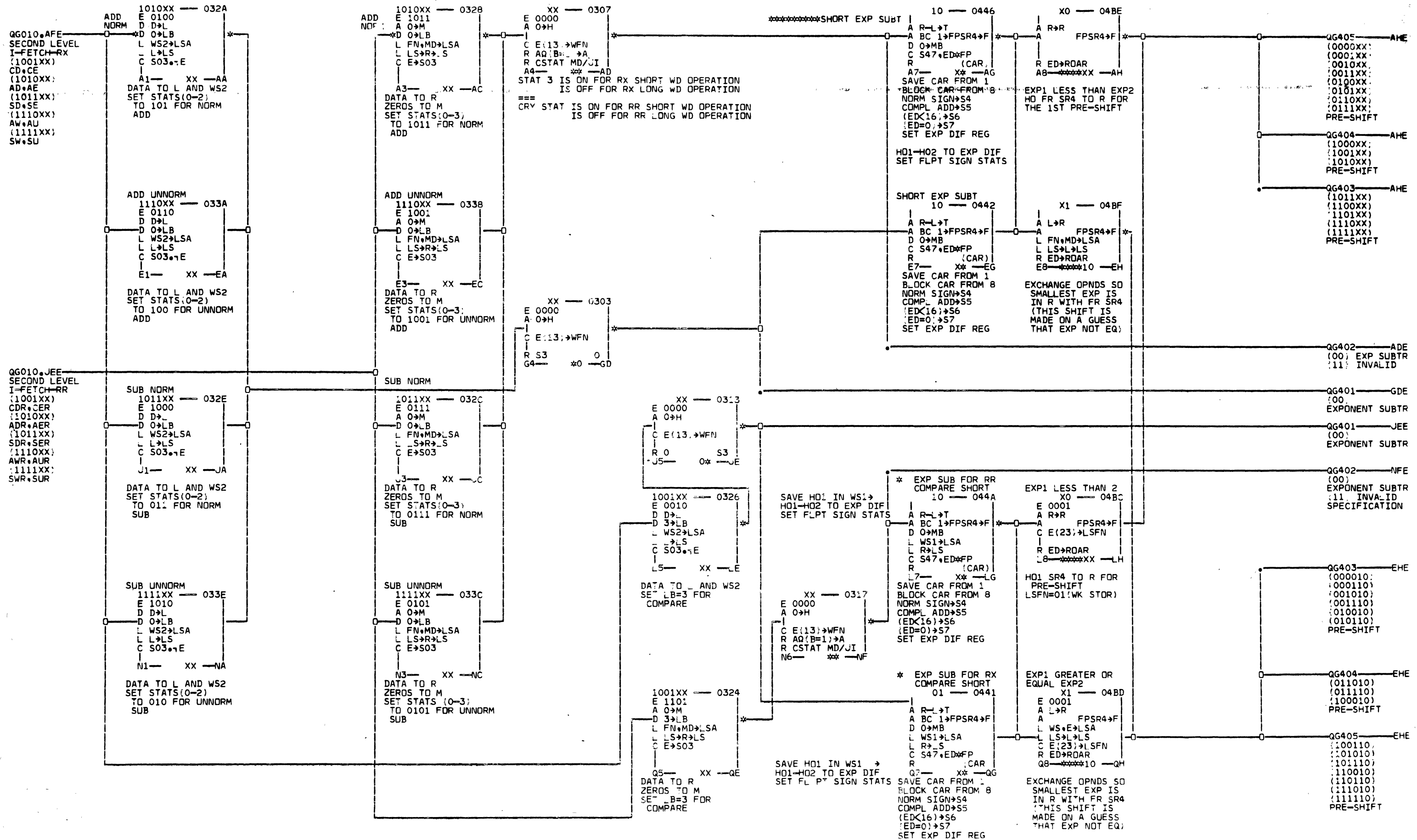
00000



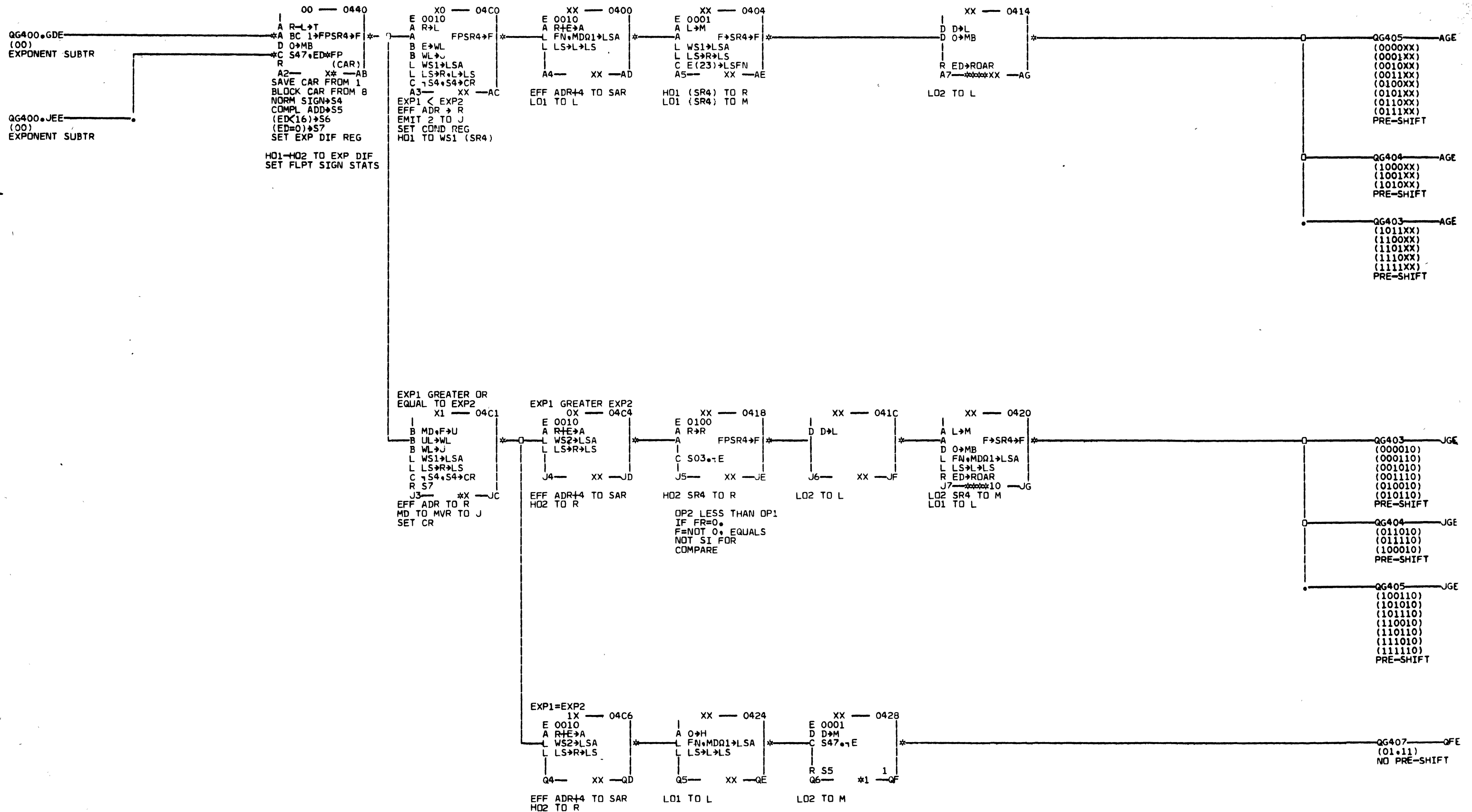
STAT 2 IS OFF FOR LOAD
IS ON FOR STORE

STAT 3 IS ON FOR SHORT WORD
IS OFF FOR LONG WORD

00000

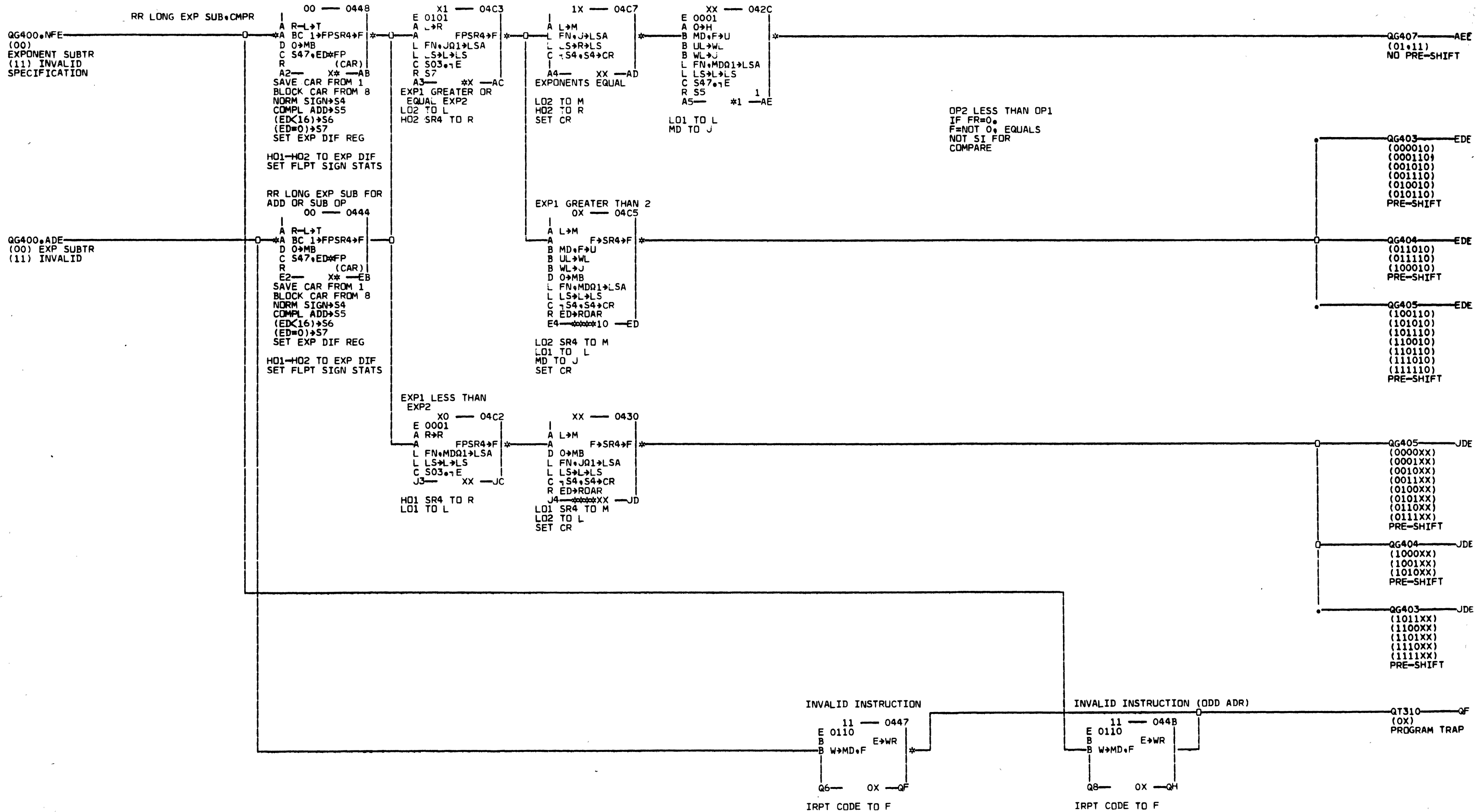


00000



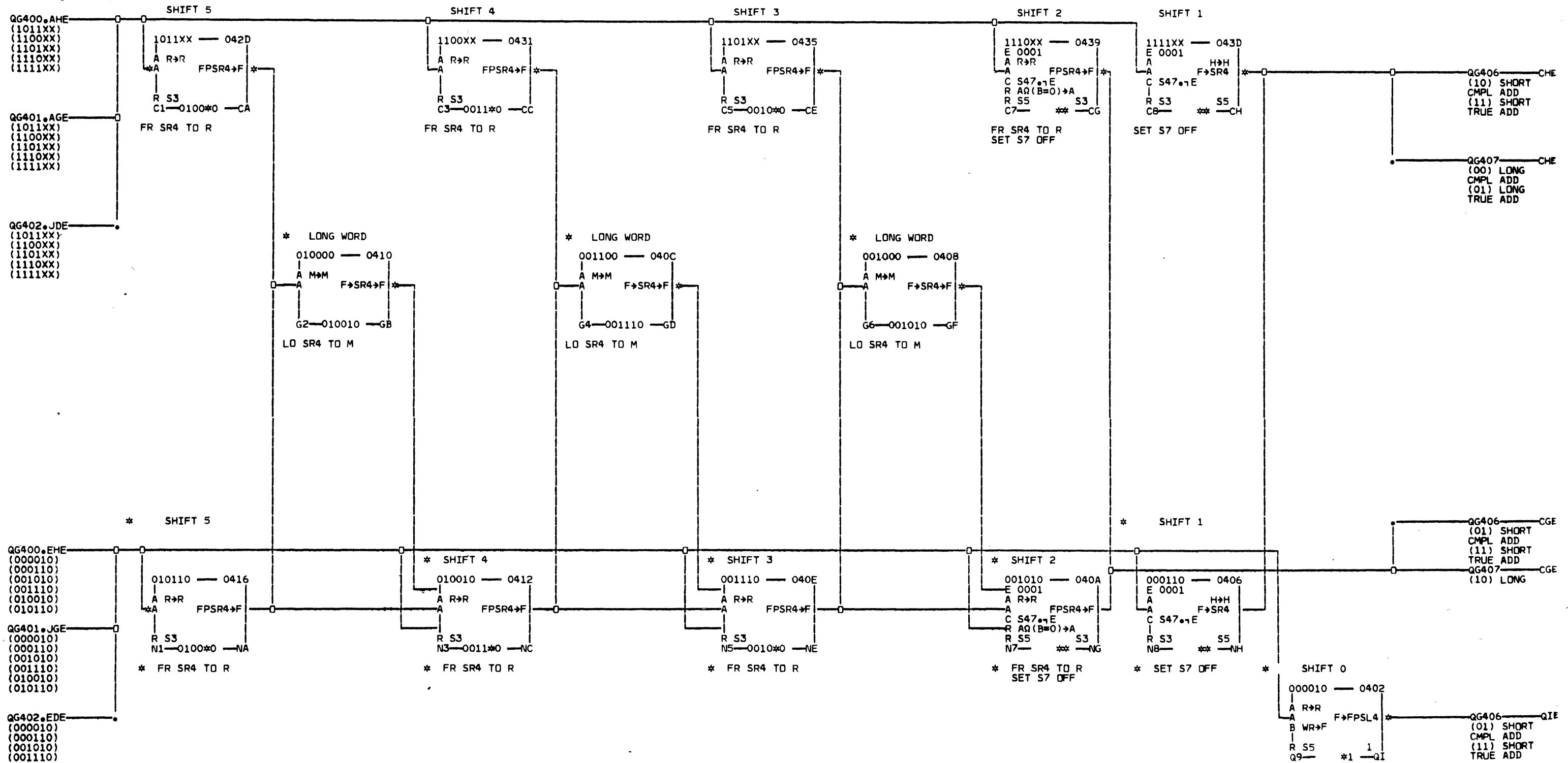
STAT 7 IS ON WHEN EXPONENTS ARE EQUAL

10260



STAT 7 IS ON WHEN EXPONENTS ARE EQUAL

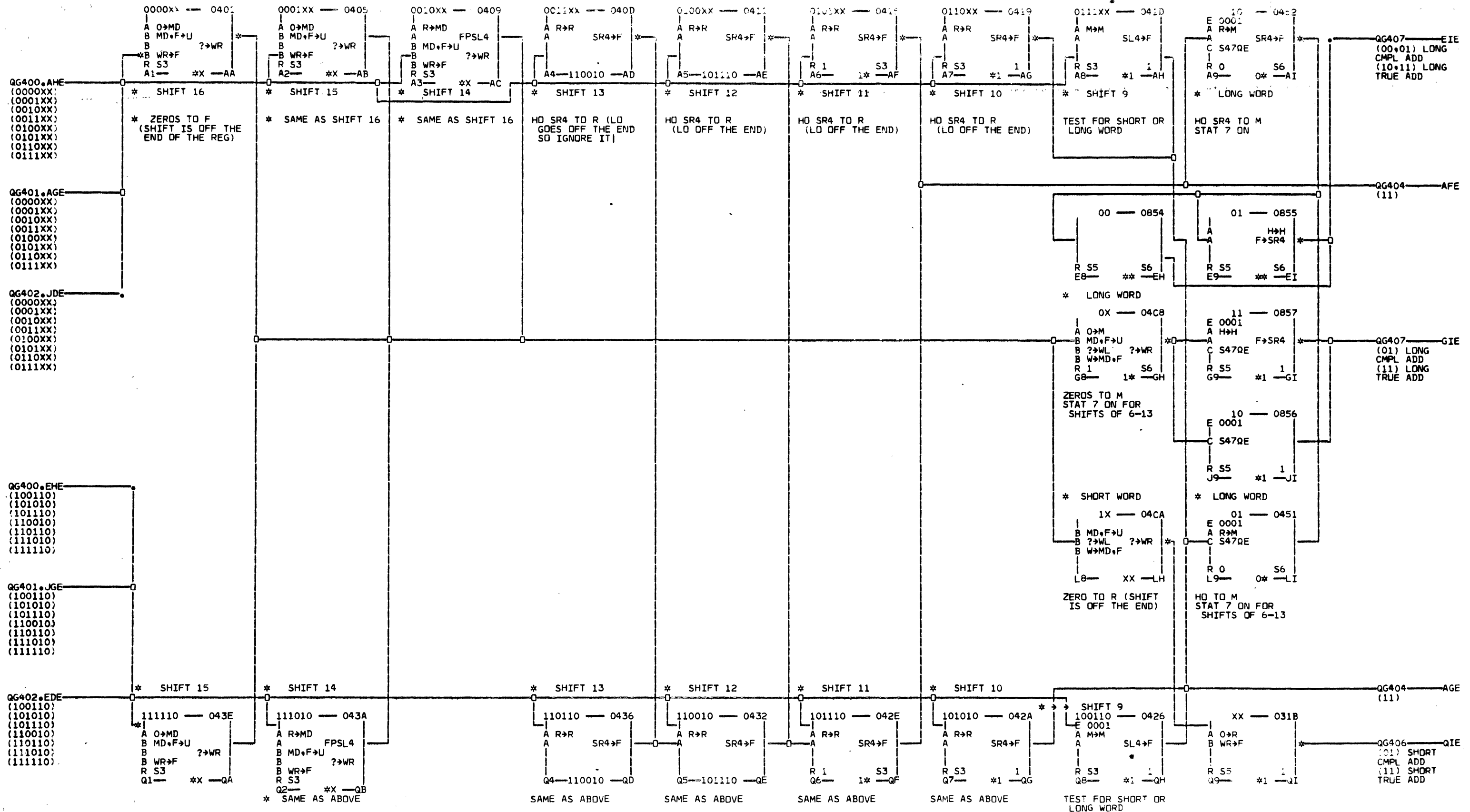
NO. 66



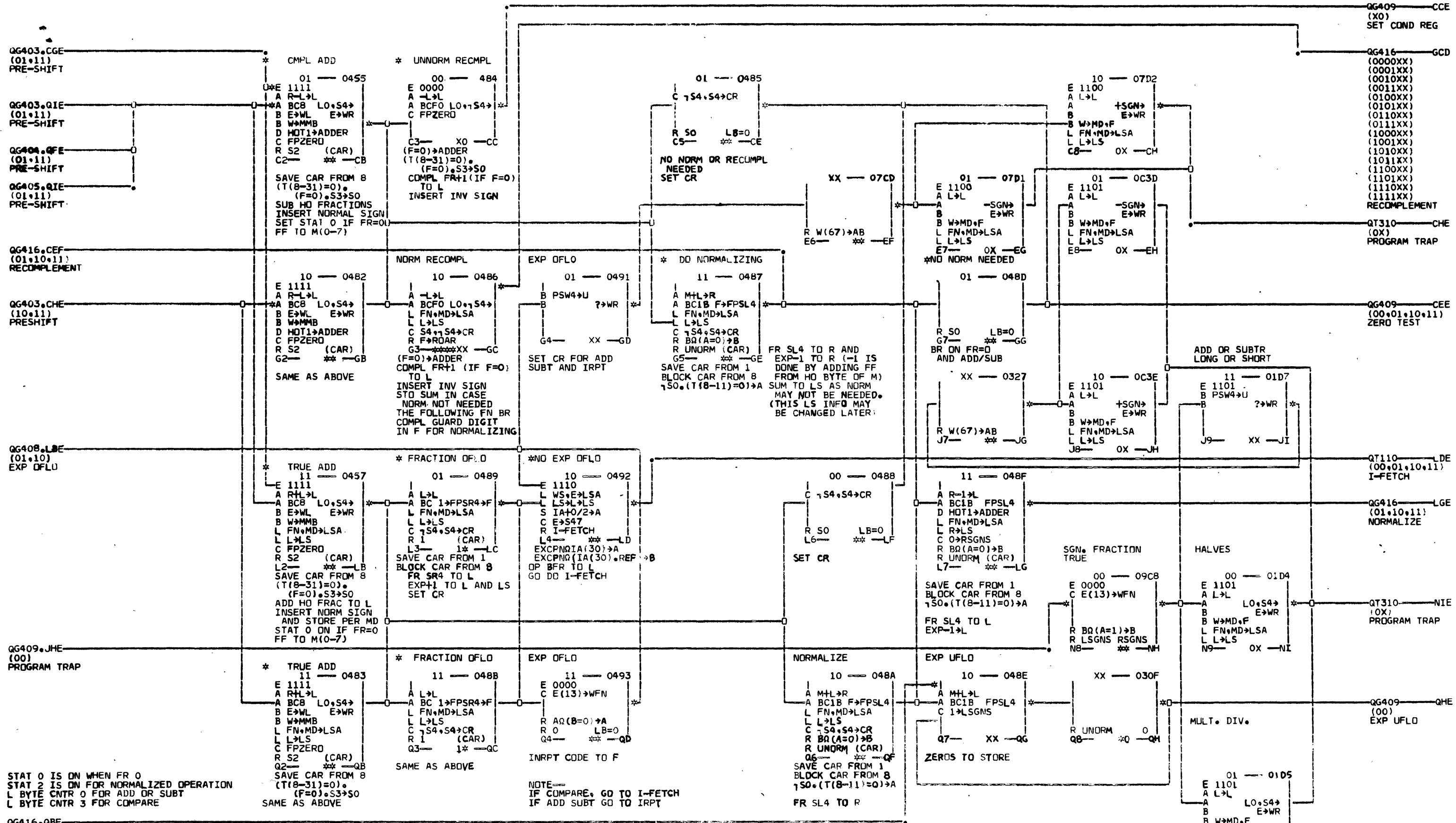
STAT 3 IS ON FOR SHORT WORD
 **
 STAT 5 IS ON FOR TRUE ADD
 **
 STAT 7 IS ON FOR AN EXP DIFF OF 6-13

* FR SL4 TO R
 (CORRECTS FOR THE FIRST SR4
 THAT WAS MADE BY GUESS)

00403
 3



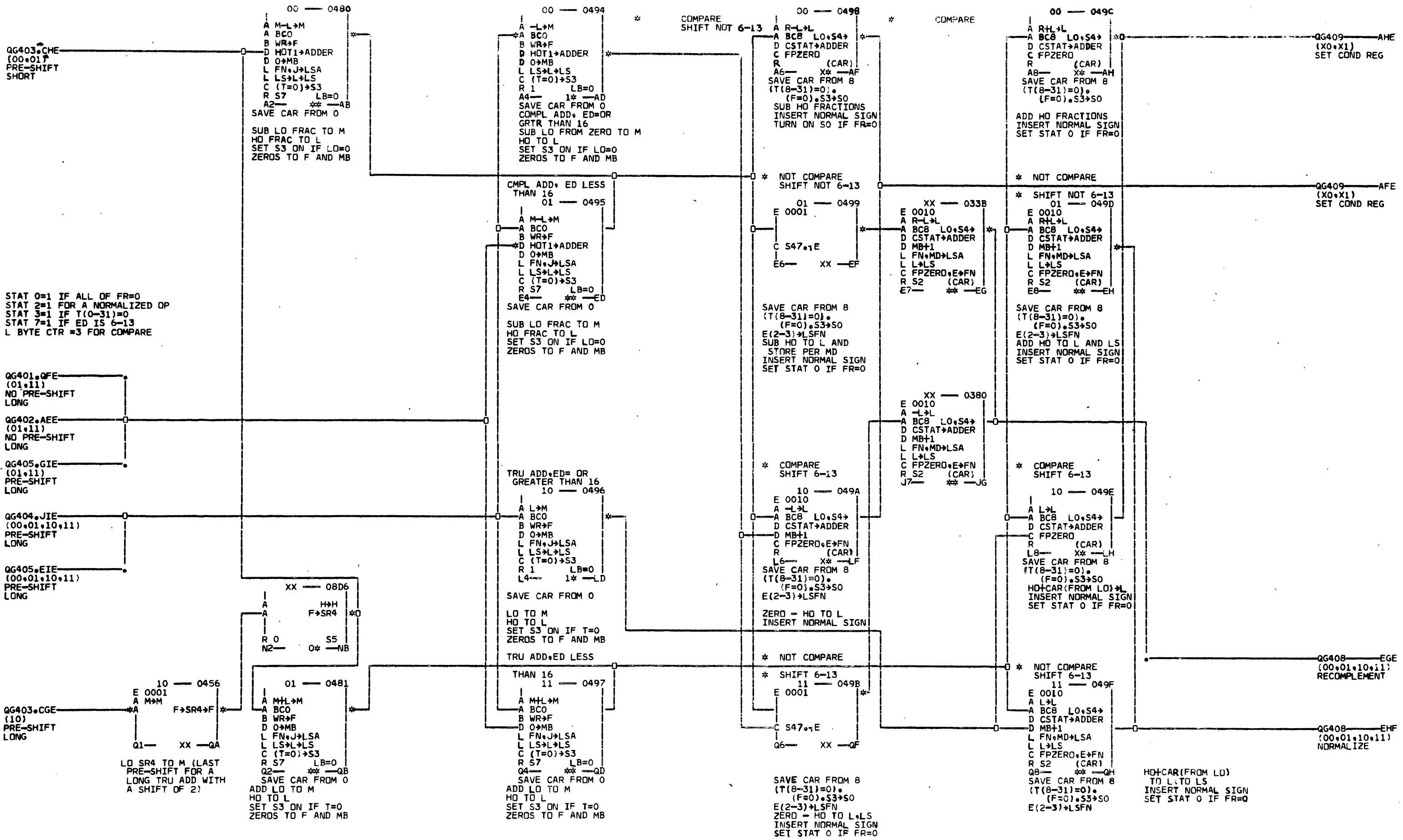
STAT 3 ON FOR SHORT WORD
STAT 5 ON FOR TRUE ADD
STAT 6 ON FOR EXP DIF LESS THAN 16
STAT 7 ON FOR SHIFT OF 6-13



STAT 0 IS ON WHEN FR 0
 STAT 2 IS ON FOR NORMALIZED OPERATION
 L BYTE CNTR 0 FOR ADD OR SUBT
 L BYTE CNTR 3 FOR COMPARE

QG416.QBF
 (10)
 EXP UNDERFLOW

6
 4
 0
 6



STAT 0=1 IF ALL OF FR=0
 STAT 2=1 FOR A NORMALIZED OP
 STAT 3=1 IF T(0-31)=0
 STAT 7=1 IF ED IS 6-13
 L BYTE CTR =3 FOR COMPARE

QG401.QFE (01.11) NO PRE-SHIFT LONG
 QG402.AEE (01.11) NO PRE-SHIFT LONG
 QG405.GIE (01.11) PRE-SHIFT LONG
 QG404.JIE (00.01.10.11) PRE-SHIFT LONG
 QG405.EIE (00.01.10.11) PRE-SHIFT LONG

QG403.CGE (10) PRE-SHIFT LONG
 LD SR4 TO M (LAST PRE-SHIFT FOR A LONG TRU ADD WITH A SHIFT OF 2)

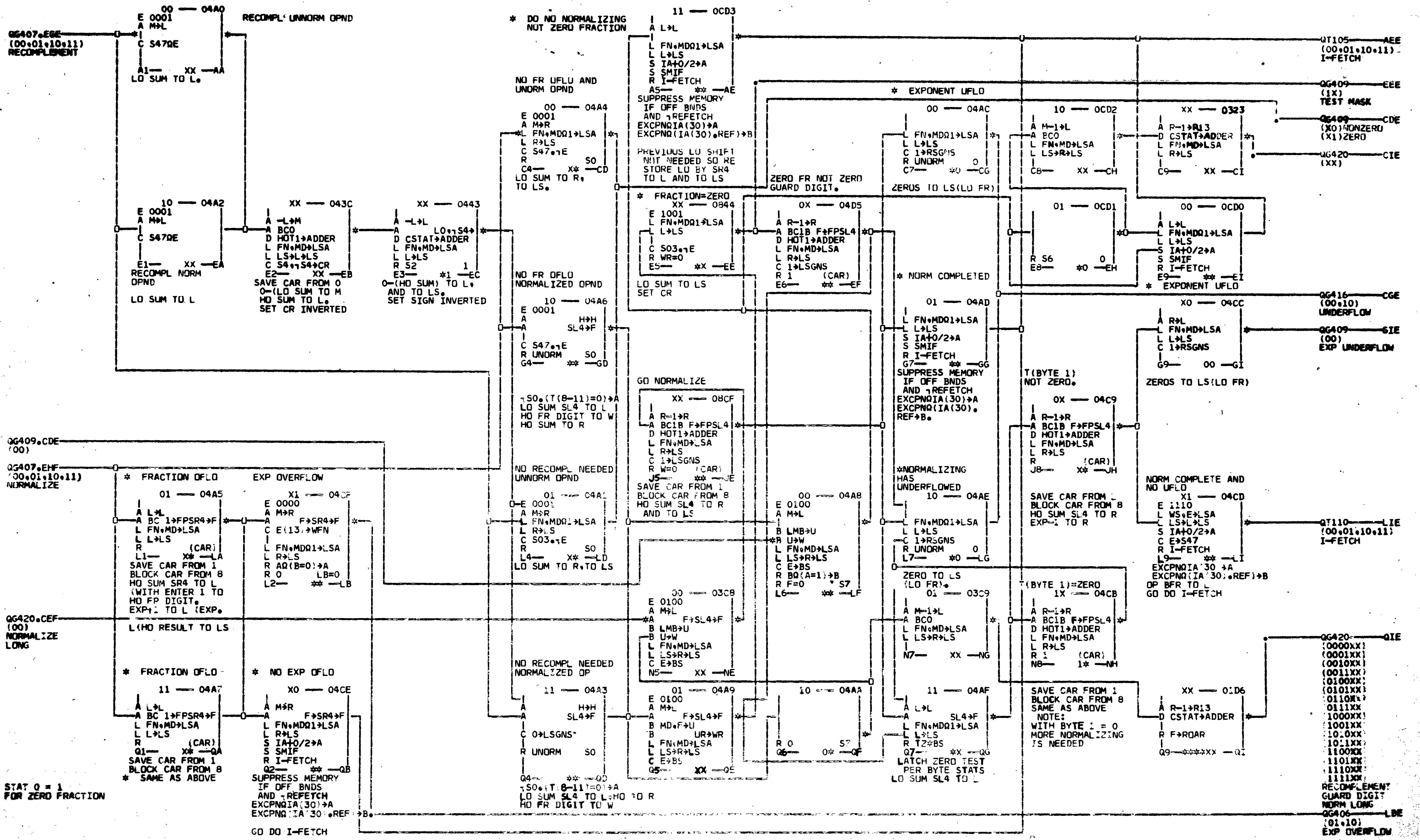
QG409.AHE (X0.X1) SET COND REG

QG409.AFE (X0.X1) SET COND REG

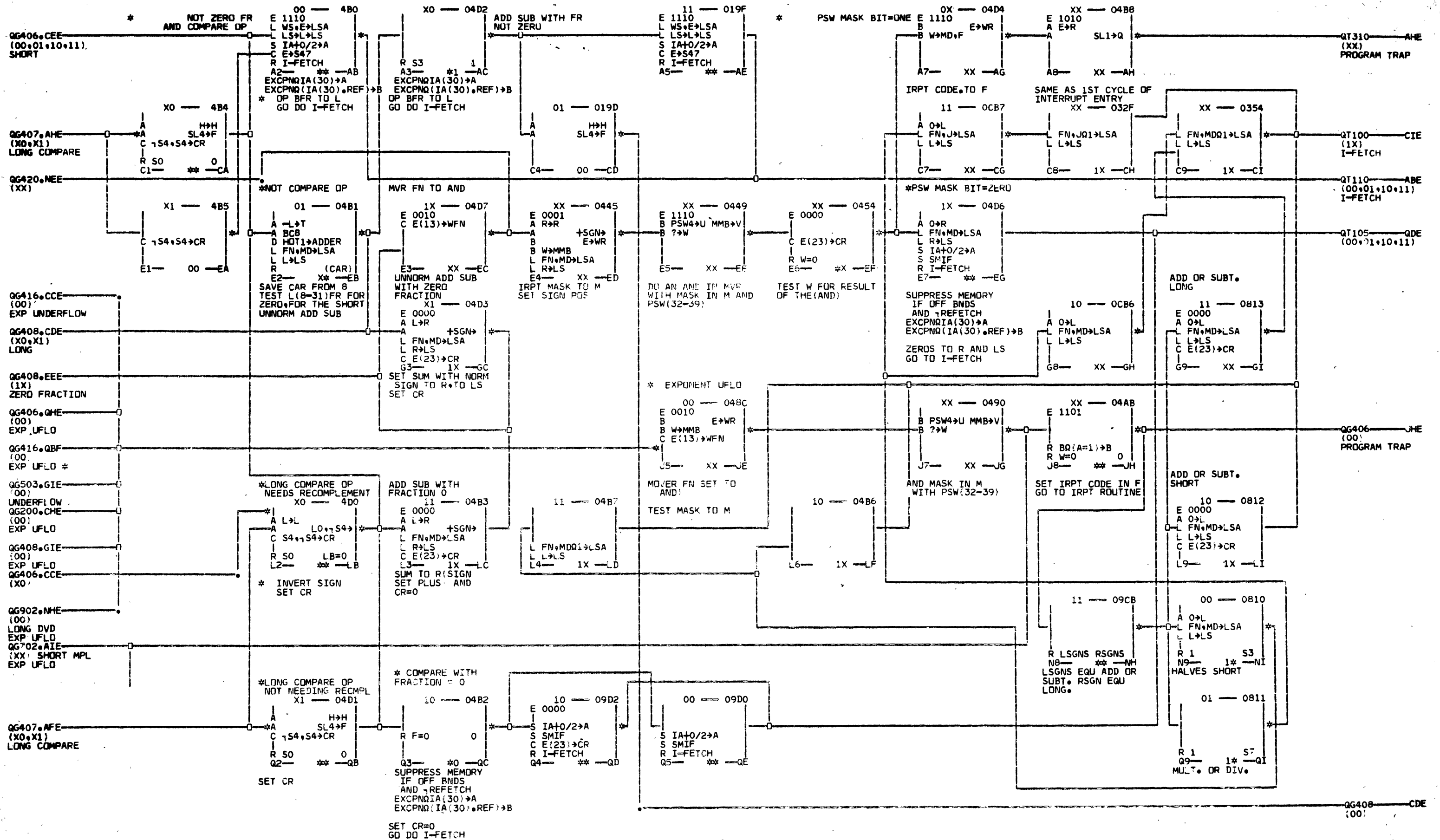
QG408.EGE (00.01.10.11) RECOMPLEMENT

QG408.EHF (00.01.10.11) NORMALIZE

HD-CAR (FROM LD) TO L; TO LS
 INSERT NORMAL SIGN
 SET STAT 0 IF FR=0



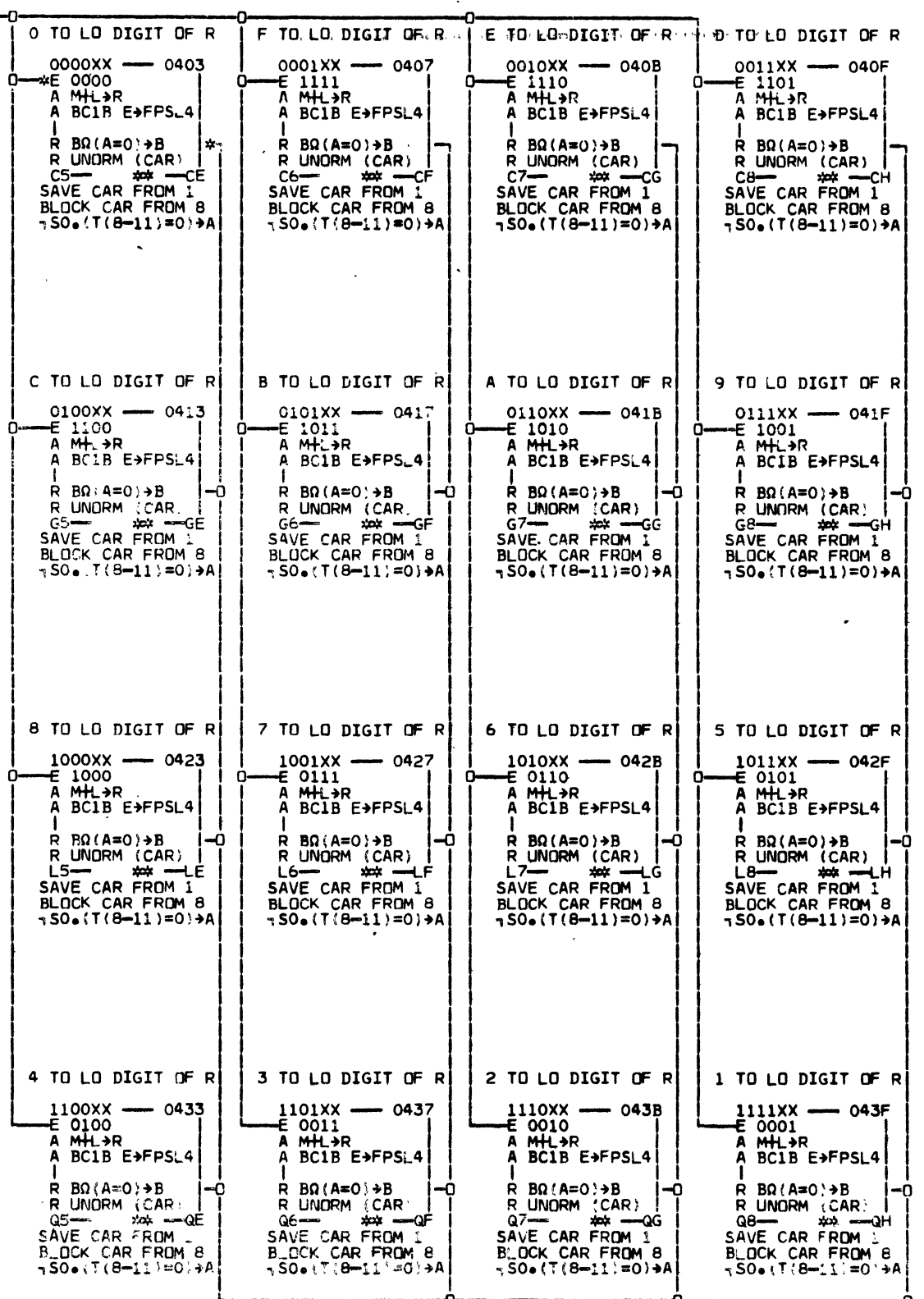
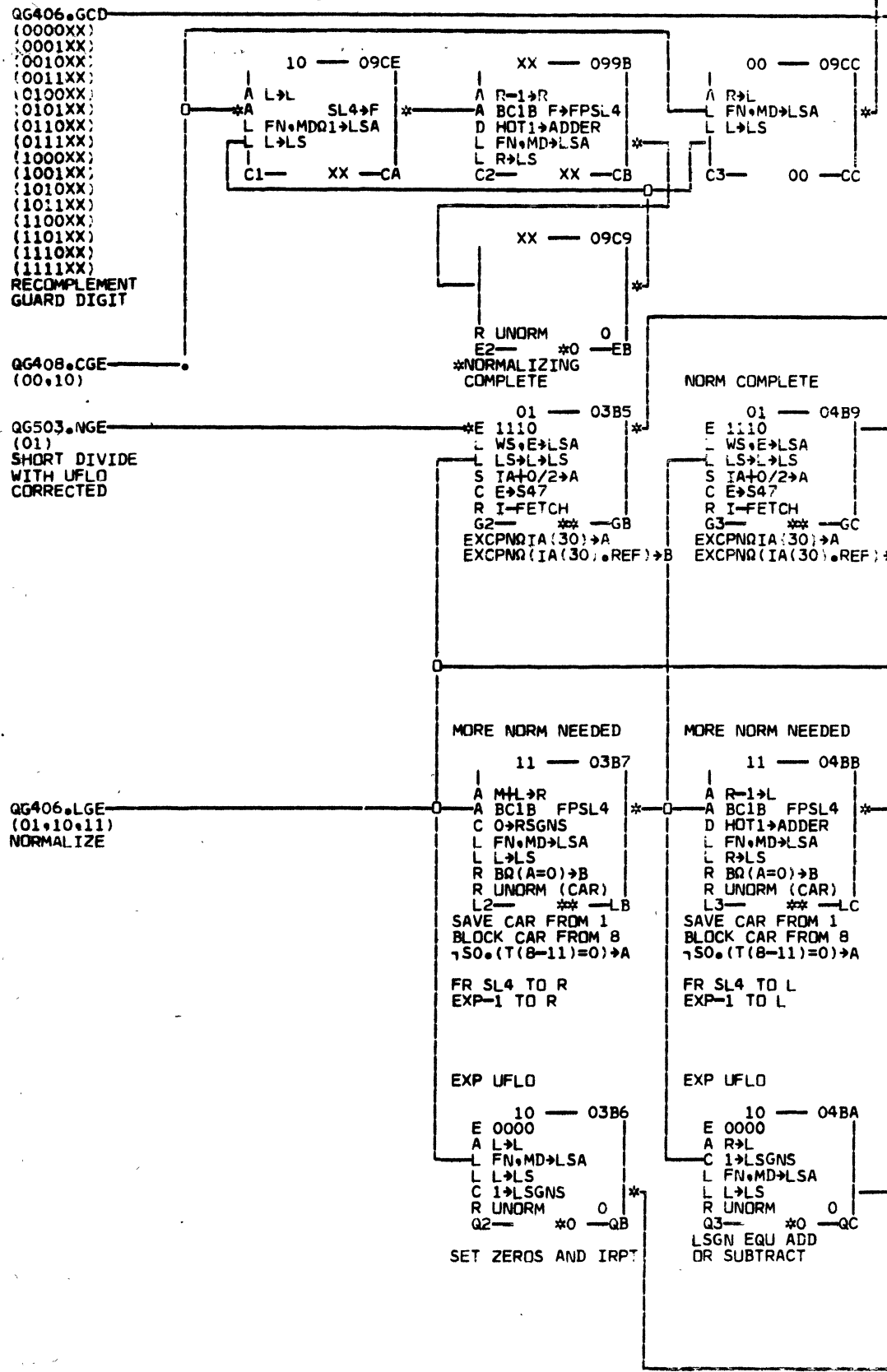
STAT 0 = 1
FOR ZERO FRACTION



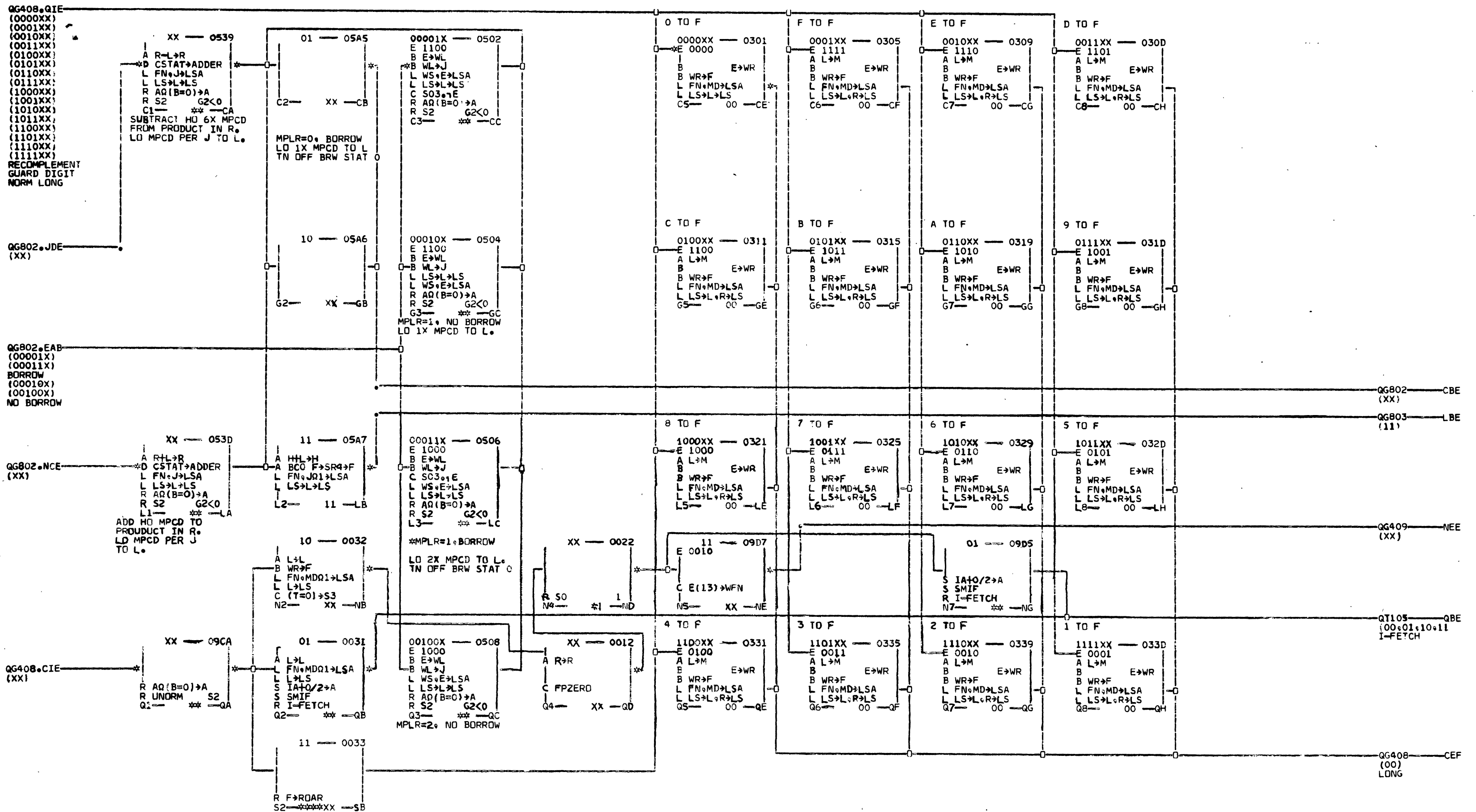
STAT 0 ON FOR FR=0

SET CR=0
GD DD I-FETCH

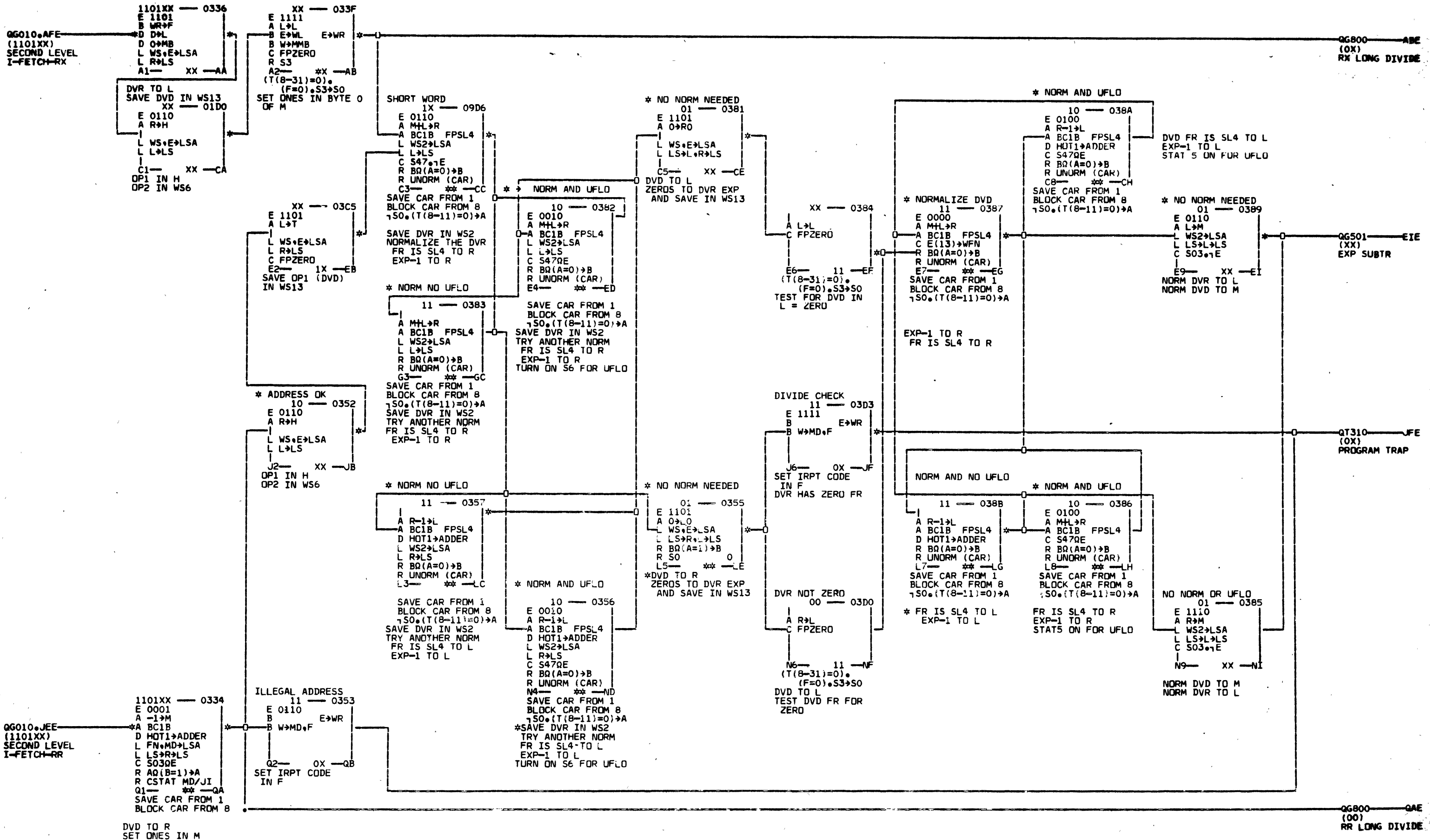
QG408 CDE (00)



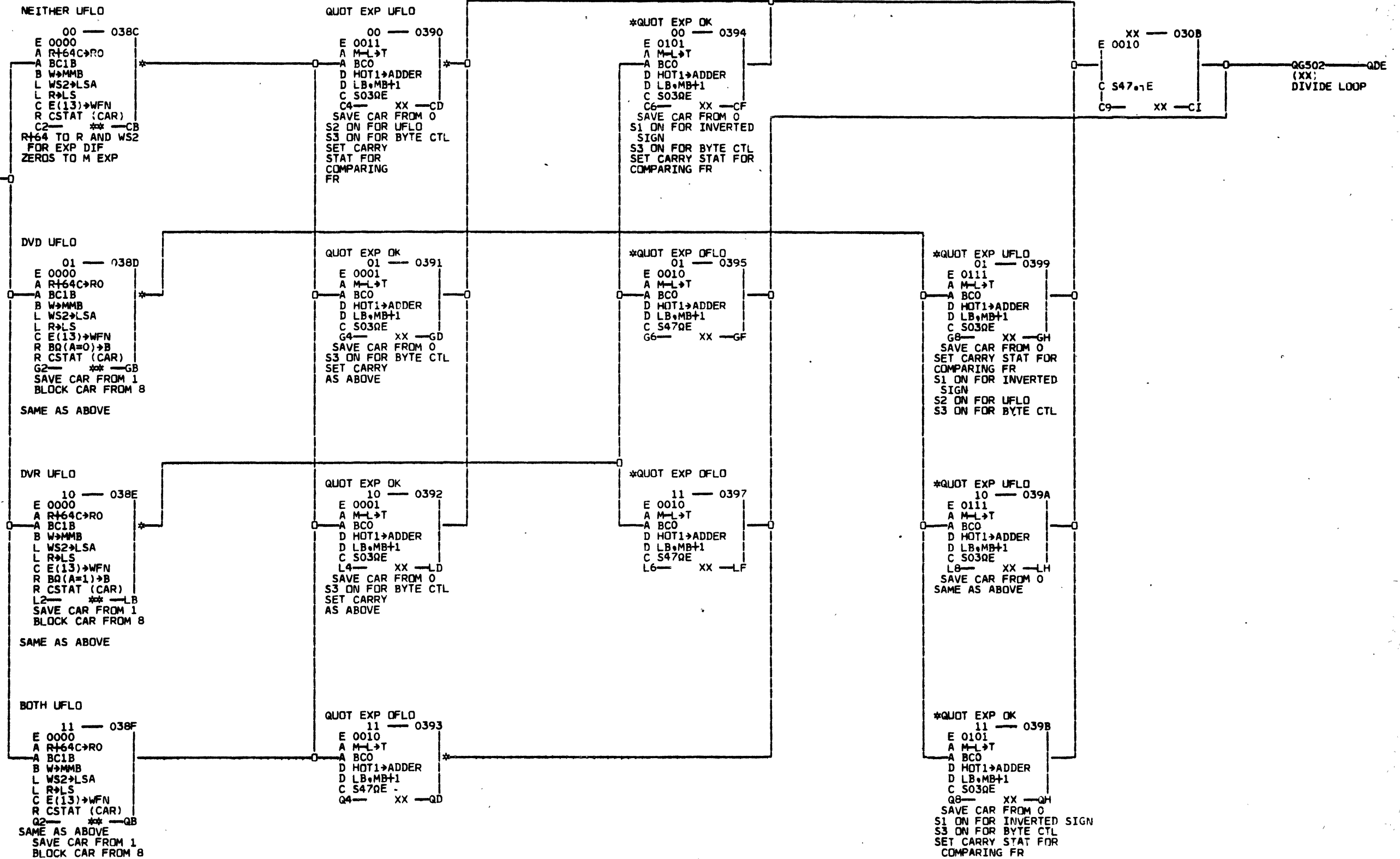
INSTRUCTION WORDS TO THE LEFT IN COLS. 5 THRU 8 ARE A 16 WAY FN BR FROM F. EACH IS THE SAME EXCEPT FOR THE EMIT FIELD. EACH WILL DO A NORMALIZE STEP WITH THE COMPL OF THE GUARD DIGIT BEING SHIFTED INTO THE FR IN R



ON 200

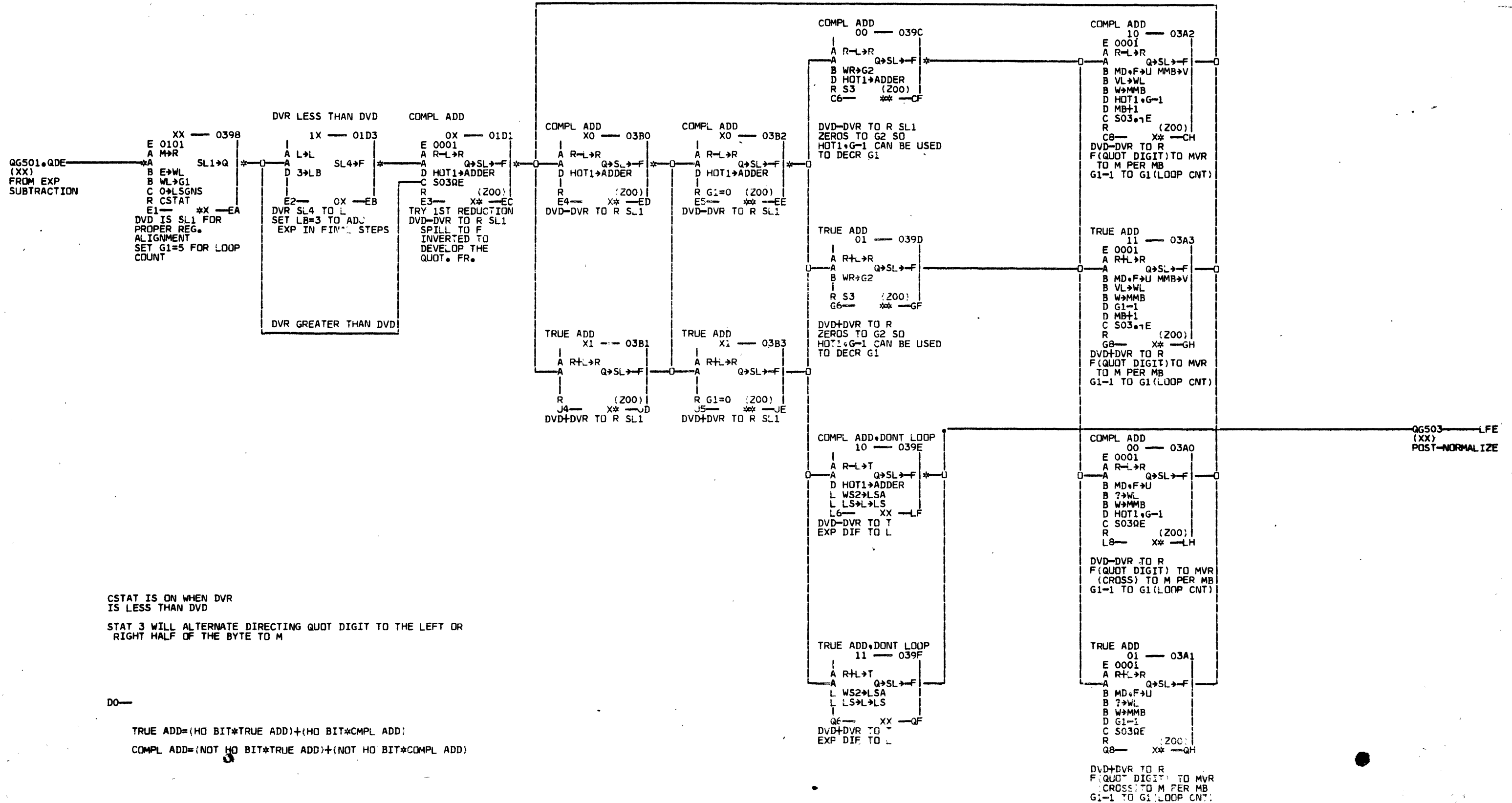


XX — 0396
 E 1101
 A M-L→RO
 A BC 1→FPSR4→F
 D O→LB→MB
 L WS→LSA
 L LS→LS
 C S47→ED→FP
 R S6
 E1 — ** —EA
 SAVE CAR FROM 1
 BLOCK CAR FROM 8
 NORM SIGN→S4
 COMPL ADD→S5
 (ED=16)→S6
 (ED=0)→S7
 SET EXP DIF REG
 M(DVD EXP)→L(DVR
 EXP) TO R
 DVR(WITH ZERO EXP)
 TO L



STAT 5 IS ON FOR DVD UNDERFLOW
 STAT 6 IS ON FOR DVR UNDERFLOW

THIS LOOP RETURNS TO
DEVELOP EACH DIGIT OF THE QUOT UNTIL G1 GOES NEGATIVE



CSTAT IS ON WHEN DVR
IS LESS THAN DVD

STAT 3 WILL ALTERNATE DIRECTING QUOT DIGIT TO THE LEFT OR
RIGHT HALF OF THE BYTE TO M

DO—

TRUE ADD=(HO BIT*TRUE ADD)+(HO BIT*COMPL ADD)
COMPL ADD=(NOT HO BIT*TRUE ADD)+(NOT HO BIT*COMPL ADD)

00000

QG502 LFE (XX) FROM DIVIDE LOOP

XX 03AB
 E 0001
 A M>L13
 C S47.1E
 R S2 S1
 G1 ** GA
 QUOT FR TO L. THE SL4 BRINGS LAST DIGIT FROM F

NO SIGN INVERSION 00 03A4

A L>L
 A LO.S4>
 L FN.MD>LSA
 L L>LS
 R AQ(B=1)>A
 R S0 LB=3
 C3 ** CC
 QUOT TO LS PER MD (MAY GET CHANGED LATER)

INVERT SIGN 01 03A5

A L>L
 A LO.S4>
 L FN.MD>LSA
 L L>LS
 R AQ(B=1)>A
 C S47QE
 R S0 LB=3
 G3 ** GC
 CHANGE SIGN OF QUOT AND STORE PER MD

INVERT SIGN 10 03A6

E 1100
 A L>L
 A LO.S4>
 L FN.MD>LSA
 L L>LS
 C S03.1E
 R S0 LB=3
 L3 ** LC
 CHANGE SIGN OF QUOT AND STORE PER MD

NO SIGN INVERSION 11 03A7

E 1100
 A L>L
 A LO.S4>
 L FN.MD>LSA
 L L>LS
 C S03.1E
 R S0 LB=3
 Q3 ** QC
 QUOT TO LS PER MD (MAY GET CHANGED LATER)

00 03AC
 E 1100
 C S03.1E
 R BQ(A=1)>B
 R S6 0
 AS ** AE
 EXCPNQIA(30)>A
 EXCPNQ(IA(30).REF)>B
 FR NOT ZERO EXP OK
 DO THE I-FETCH ROUTINE

INCR EXPONENT 11 03AF

A L>LO
 A BC 1>FPSR4>F
 L FN.MD>LSA
 L L>LS
 R 0 (CAR)
 ES 0* EE
 SAVE CAR FROM 1
 BLOCK CAR FROM 8
 EXP+1 TO L
 STORE QUOT PER MD

ZERO FRACTION 10 03AE

A 0>L
 L FN.MD>LSA
 L L>LS
 S IA+0/2>A
 S SMIF
 R I-FETCH
 J5 ** JE

SUPPRESS MEMORY IF OFF BNDS AND REFETCH EXCPNQIA(30)>A EXCPNQ(IA(30).REF)>B STORE ZERO QUOT PER MD

00 07D0
 E 1110
 B WS.E>LSA
 S IA+0/2>A
 C E>S47
 L LS>LS
 R I-FETCH
 A6 ** AF

OVERFLOW 01 03AD

E 1100
 A L>L
 L FN.MD>LSA
 L L>LS
 C S03.1E
 E6 11 EF

UNDERFLOW 00 03A8

E 0110
 L WS.E>LSA
 L LS>LS
 C 0>RSGNS
 J7 XX JG
 SET ZEROS TO LS PER MD
 SET MOVER FN FOR AND

INCR EXPONENT 01 03A9

A L>LO
 A BC 1>FPSR4>F
 L FN.MD>LSA
 L L>LS
 R 0 (CAR)
 N7 0* NG
 SAVE CAR FROM 1
 BLOCK CAR FROM 8
 EXP+1 TO L AND
 STORE PER MD

DVD FR WAS ZERO 10 03AA

A 0>L
 L FN.MD>LSA
 L L>LS
 S IA+0/2>A
 S SMIF
 R I-FETCH
 S7 ** SG

11 07D3
 E 0110
 L WS.E>LSA
 L LS>LS
 AB XX AH

GO LOOK AT MASK FOR INTERRUPT

UNDERFLOW 00 03B4

E 0110
 L WS.E>LSA
 L LS>LS
 C 0>RSGNS
 J8 XX JH
 SAME AS AT LEFT

XX 0137
 E 1100
 A L>L
 A LO.S4>
 B W>MD>F
 L FN.MD>LSA
 L L>LS
 A9 0X AI

XX 0136

A H>L>T
 A BC 1>FPSR4>F
 L FN.MD>LSA
 L LS>LS
 C S47.ED>FP
 C9 XX CI

XX 01D2

A H>L>T
 A BC 1>FPSR4>F
 L FN.MD>LSA
 L LS>LS
 C S47.ED>FP
 G9 00 GI

*UFLD CORRECTED GO TO I-FETCH

SET ZEROS TO LS PER MD SUPPRESS MEMORY IF OFF BNDS AND REFETCH EXCPNQIA(30)>A EXCPNQ(IA(30).REF)>B

QT310 AIE (OX) PROGRAM TRAP

QT110 AFE (00.01.10.11) I-FETCH

QG409 GIE (00) TEST MASK

QG416 NGE (01) I-FETCH

QT105 JEE (00.01.10.11) I-FETCH

STAT 0 IS ON FOR ZERO DVD
 STAT 1 IS ON FOR UNDERFLOW
 LB=3 WHEN EXPONENT MUST BE INCREASED

1100XX — 0332
 E 0000
 D D+M
 L FN+MD+LSA
 L LS+L+LS
 C E+S03
 R S3
 A2 — *X — AB
 OPND 2 TO M
 OPND 1 TO L

QG010.AFE
 (1100XX)
 SECOND LEVEL
 I-FETCH-RX

EXPONENT ADD 1X — 0503
 A M+L+RO
 A BC1B
 B W+MMB
 L WS1+LSA
 L L+LS
 C S47+ED+FP
 AS — XX — AE
 SAVE CAR FROM 1
 BLOCK CAR FROM 8
 NORM SIGN+S4
 COMPL ADD+S5
 (ED<16)+S6
 (ED=0)+S7
 SET EXP DIF REG
 EXP1+EXP2 TO R
 ZEROS TO M(EXP)
 SET FLPT SGN STATS
 L(OPND1) TO WS1

00 — 0544
 E 0100
 A L+R
 A L0+S4+
 B LLB+U UR+WR
 D G2-1
 L WS1+LSA
 L R+LS
 C S03+1E
 A7 — XX — CG
 EXPONENT UFLO
 SAME AS BELOW

QG800 — ABE
 (OX)
 RX LONG MPY

QG800 — QBE
 (OO)
 RR LONG MPY

EXCESS 64 CORRECT

XX — 0541
 E 0110
 A R+64C+LO
 A BC1B
 B E+W
 B WL+G1
 D 3+LB
 L WS1+LSA
 L LS+R+LS
 C S03+1E
 R CSTAT (CAR)
 G6 — ** — GF
 SAVE CAR FROM 1
 BLOCK CAR FROM 8
 EXP DIF -64 TO L
 G1=6 FOR LOOP CNT
 OPND1 TO R
 SET LB=3

NO EXP OFLO OR UFLO

01 — 0545
 E 0100
 A L+R
 A L0+S4+
 B LLB+U UR+WR
 L WS1+LSA
 L R+LS
 C S03+1E
 G7 — XX — GG
 SET NORM SIGN TO
 EXP SUM AND STORE
 IN WS1
 STAT 1 OFF FOR NO
 OFLO
 LO MPLR DIGIT TO W
 (FOR FUTURE 0 TEST)

QG701 — GGE
 (XX)
 MULTIPLY LOOP

* EXPONENT ADD

10 — 0542
 A M+L+RO
 A BC1B
 B W+MMB
 L WS1+LSA
 L L+LS
 C S47+ED+FP
 L3 — XX — LC
 EXP1+EXP2 TO R
 ZEROS TO M(EXP)
 SET FP STATS
 L(OPND1) TO WS1

SAVE CAR FROM 1
 BLOCK CAR FROM 8
 NORM SIGN+S4
 COMPL ADD+S5
 (ED<16)+S6
 (ED=0)+S7
 SET EXP DIF REG

INVALID ADDRESS

11 — 0543
 E 0110
 B W+MD+F E+WR
 Q3 — OX — QC
 IRPT CODE TO F
 GO INTERRUPT

NO EXP OFLO OR UFLO

10 — 0546
 E 0100
 A L+R
 A L0+S4+
 B LLB+U UR+WR
 L WS1+LSA
 L R+LS
 C S03+1E
 L7 — XX — LG
 SAME AS ABOVE

EXPONENT OFLO

11 — 0547
 I L+R
 A L0+S4+
 B LLB+U UR+WR
 L WS1+LSA
 L R+LS
 G7 — XX — GG
 SET INV SIGN TO
 EXP SUM AND STORE
 IN WS1
 LO MPLR DIGIT TO W

LOADING AFTER THIS PAGE

STAT 1 IS ON FOR EXP OVERFLOW
 STAT 2 IS ON FOR FLOATING POINT
 STAT 4 IS ON FOR NORMAL SIGN
 STAT 5 IS ON FOR TRUE ADD
 STAT 6 IS ON FOR EXP DIFF LESS THAN 16

1100XX — 0330
 E 0000
 A L+M
 D O+MB
 L FN+MD+LSA
 L LS+L+LS
 C E+S03
 R AQ(B=1)+A
 R CSTAT MD/JI
 Q2 — ** — QB
 OPND 2 TO M
 OPND 1 TO L
 SET STAT 3 OFF

QG010.JEE
 (1100XX)
 SECOND LEVEL
 I-FETCH-RR

QT310 — QCE
 (OX)
 PROGRAM TRAP

254760
 255449

12/21/64
 09/20/65

MACH
 NAME
 MODE
 P.N.
 IBM CORP.

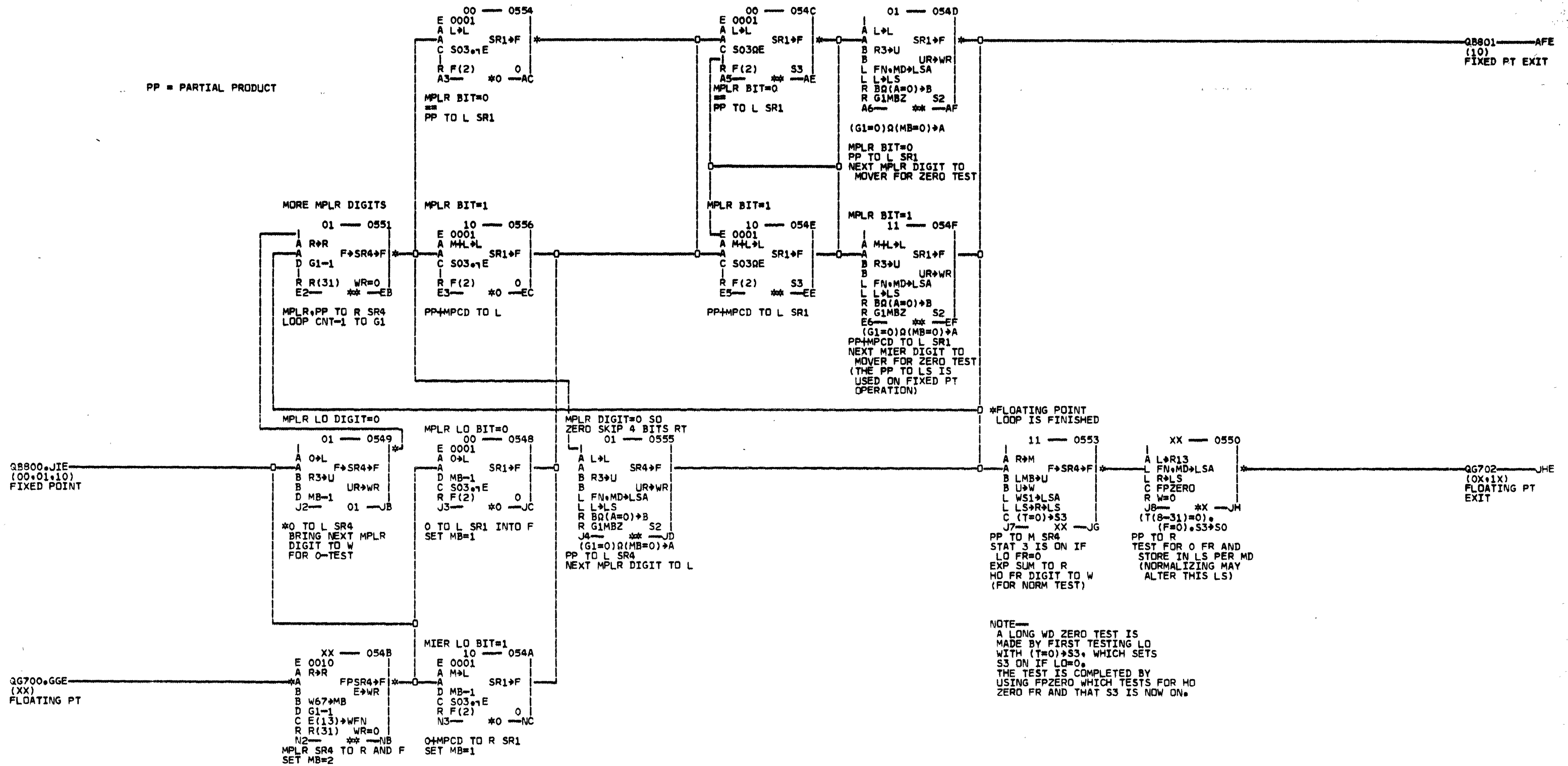
C2050
 MANUAL
 5364759
 SDD

DATE 10/01/65
 LOG 902

SHEET 1
 VERSION
 06700

FLOATING POINT MULTIPLY RR
 AND RX FORMAT LONG/SHORT

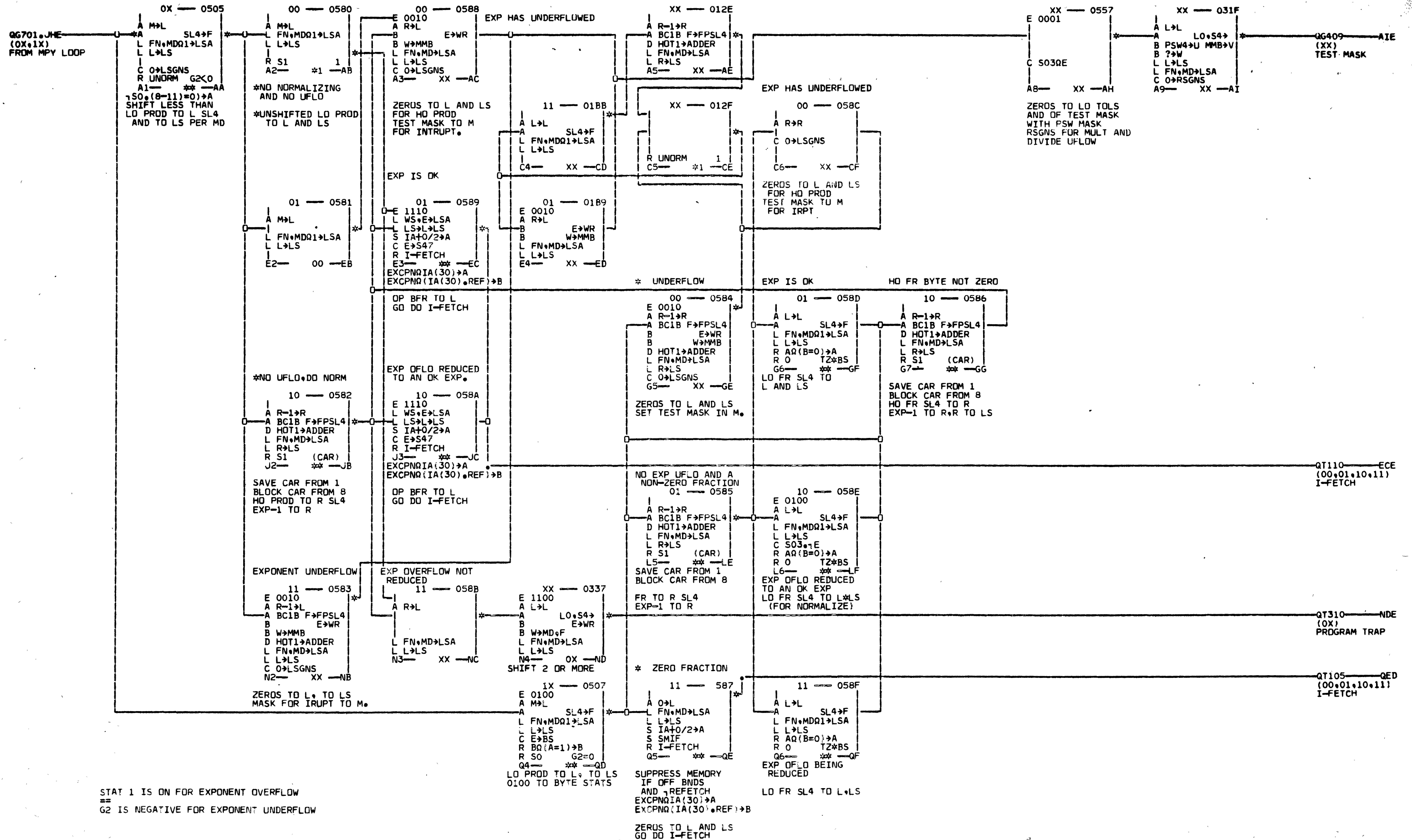
PP = PARTIAL PRODUCT



NOTE—
A LONG WD ZERO TEST IS
MADE BY FIRST TESTING LO
WITH (T=0)S3, WHICH SETS
S3 ON IF LO=0.
THE TEST IS COMPLETED BY
USING FPZERO WHICH TESTS FOR HO
ZERO FR AND THAT S3 IS NOW ON.

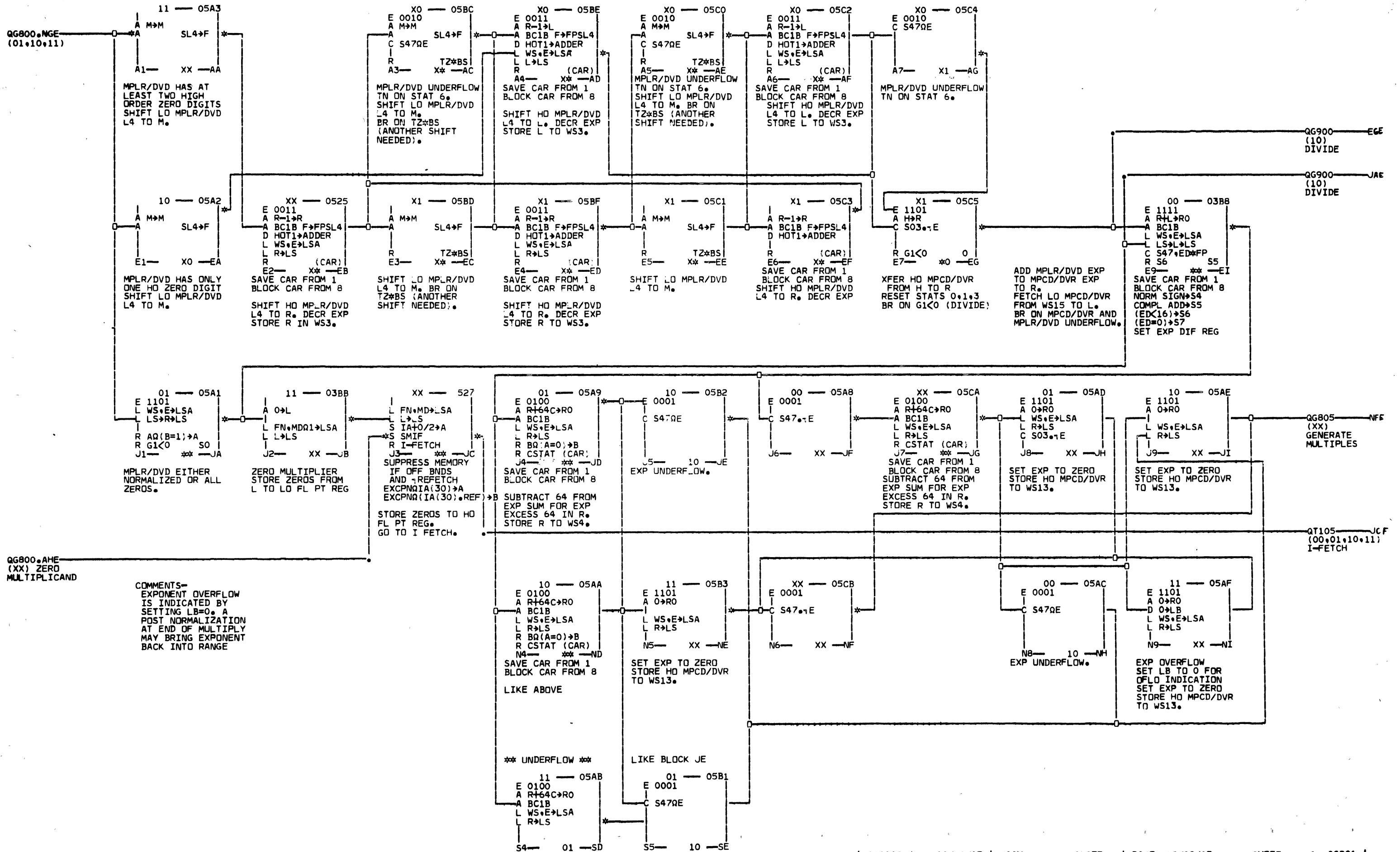
STAT 2 IS ON FOR FLOATING POINT
R(31) IS A TEST FOR A MPLR BIT
F(2) IS A TEST FOR A MPLR BIT
G1MBZ TESTS FOR G1=0, WHICH IS END OF MPY LOOP

05701



STAT 1 IS ON FOR EXPONENT OVERFLOW
 == G2 IS NEGATIVE FOR EXPONENT UNDERFLOW

NO 760



MPLR/DVD HAS AT LEAST TWO HIGH ORDER ZERO DIGITS SHIFT LO MPLR/DVD L4 TO M.

MPLR/DVD UNDERFLOW TN ON STAT 6. SHIFT LO MPLR/DVD L4 TO M. BR ON TZ*BS (ANOTHER SHIFT NEEDED).

SAVE CAR FROM 1 BLOCK CAR FROM 8. SHIFT HO MPLR/DVD L4 TO L. DECR EXP STORE L TO WS3.

MPLR/DVD UNDERFLOW TN ON STAT 6. SHIFT LO MPLR/DVD L4 TO M. BR ON TZ*BS (ANOTHER SHIFT NEEDED).

SAVE CAR FROM 1 BLOCK CAR FROM 8. SHIFT HO MPLR/DVD L4 TO L. DECR EXP STORE L TO WS3.

MPLR/DVD UNDERFLOW TN ON STAT 6.

MPLR/DVD HAS ONLY ONE HO ZERO DIGIT SHIFT LO MPLR/DVD L4 TO M.

SAVE CAR FROM 1 BLOCK CAR FROM 8. SHIFT HO MPLR/DVD L4 TO R. DECR EXP STORE R IN WS3.

SHIFT LO MPLR/DVD L4 TO M. BR ON TZ*BS (ANOTHER SHIFT NEEDED).

SAVE CAR FROM 1 BLOCK CAR FROM 8. SHIFT HO MPLR/DVD L4 TO R. DECR EXP STORE R TO WS3.

SHIFT LO MPLR/DVD L4 TO M.

SAVE CAR FROM 1 BLOCK CAR FROM 8. SHIFT HO MPLR/DVD L4 TO R. DECR EXP

XFER HO MPCD/DVR FROM H TO R. RESET STATS 0,1,3 BR ON G1<0 (DIVIDE)

ADD MPLR/DVD EXP TO MPCD/DVR EXP TO R. FETCH LO MPCD/DVR FROM WS15 TO L. BR ON MPCD/DVR AND MPLR/DVD UNDERFLOW.

SAVE CAR FROM 1 BLOCK CAR FROM 8. NORM SIGN S4. COMPL ADD S5 (ED<16) S6 (ED=0) S7. SET EXP DIF REG

MPLR/DVD EITHER NORMALIZED OR ALL ZEROS.

ZERO MULTIPLIER STORE ZEROS FROM L TO LO FL PT REG

SUPPRESS MEMORY IF OFF BNDS AND REFETCH EXCPN QIA(30) A EXCPN QIA(30) REF. STORE ZEROS TO HO FL PT REG. GO TO I FETCH.

SAVE CAR FROM 1 BLOCK CAR FROM 8. SUBTRACT 64 FROM EXP SUM FOR EXP EXCESS 64 IN R. STORE R TO WS4.

EXP UNDERFLOW.

SAVE CAR FROM 1 BLOCK CAR FROM 8. SUBTRACT 64 FROM EXP SUM FOR EXP EXCESS 64 IN R. STORE R TO WS4.

SET EXP TO ZERO STORE HO MPCD/DVR TO WS13.

SET EXP TO ZERO STORE HO MPCD/DVR TO WS13.

SET EXP TO ZERO STORE HO MPCD/DVR TO WS13.

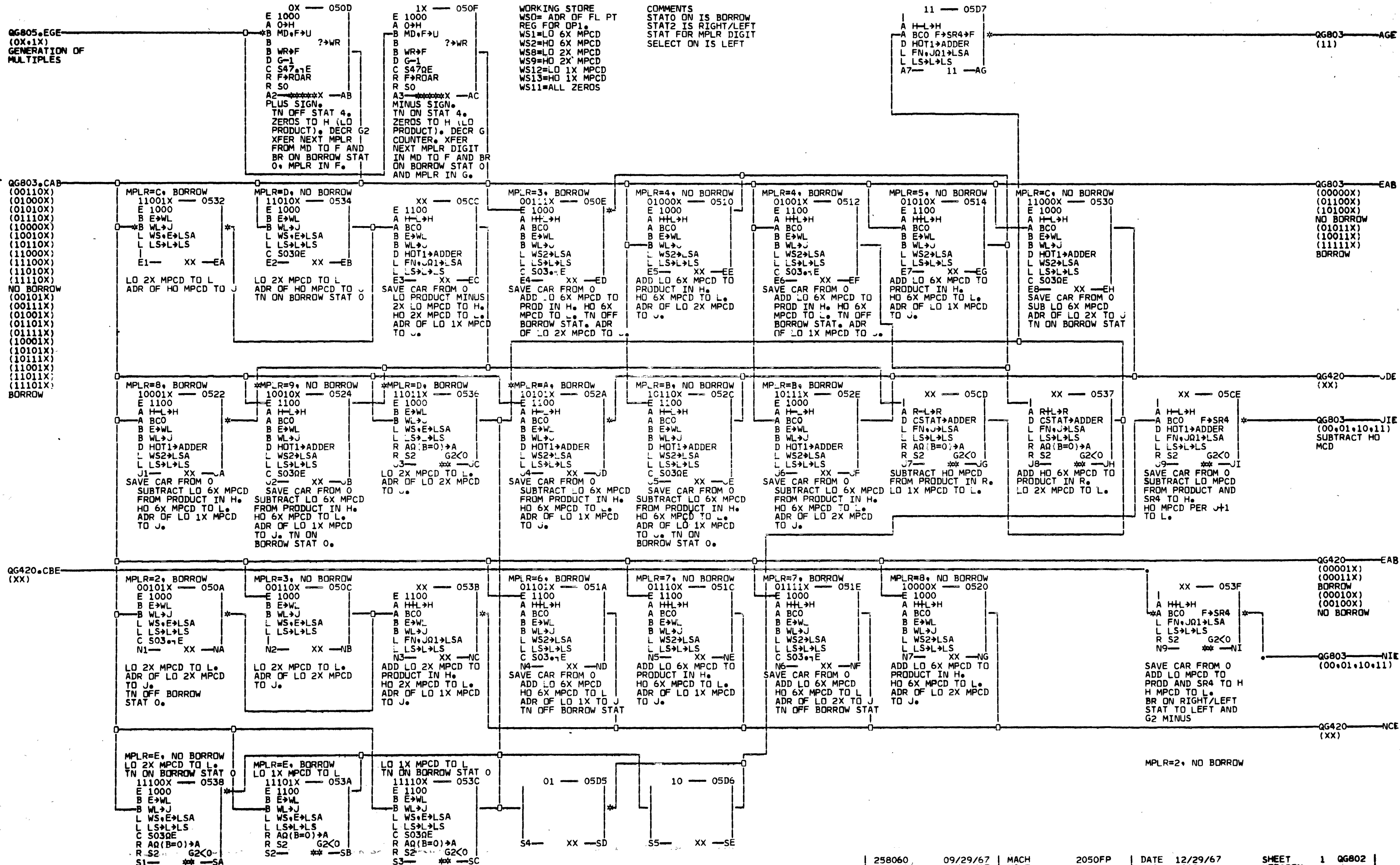
COMMENTS- EXPONENT OVERFLOW IS INDICATED BY SETTING LB=0. A POST NORMALIZATION AT END OF MULTIPLY MAY BRING EXPONENT BACK INTO RANGE

** UNDERFLOW **

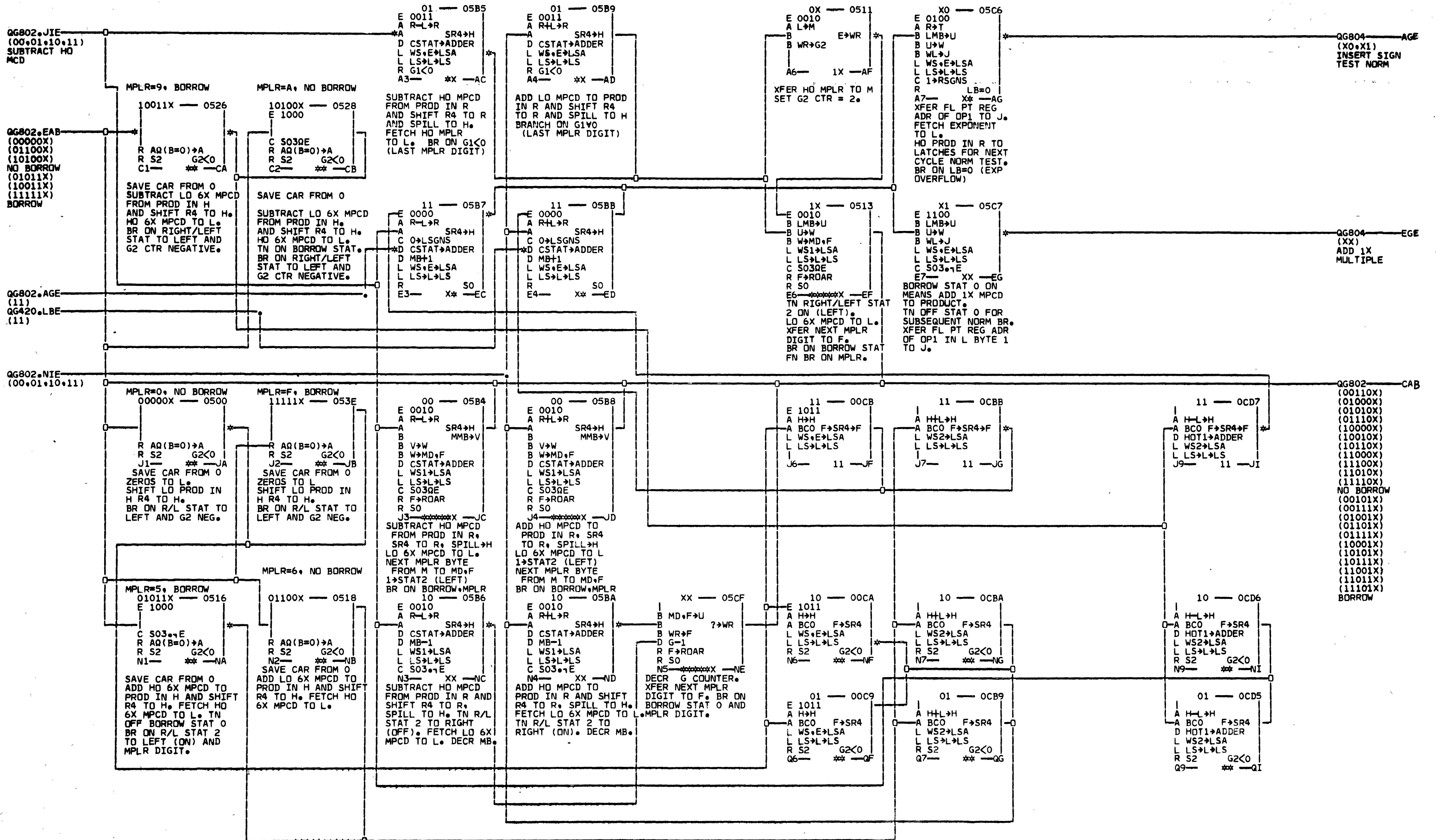
LIKE BLOCK JE

EXP OVERFLOW SET LB TO 0 FOR OFLO INDICATION SET EXP TO ZERO STORE HO MPCD/DVR TO WS13.

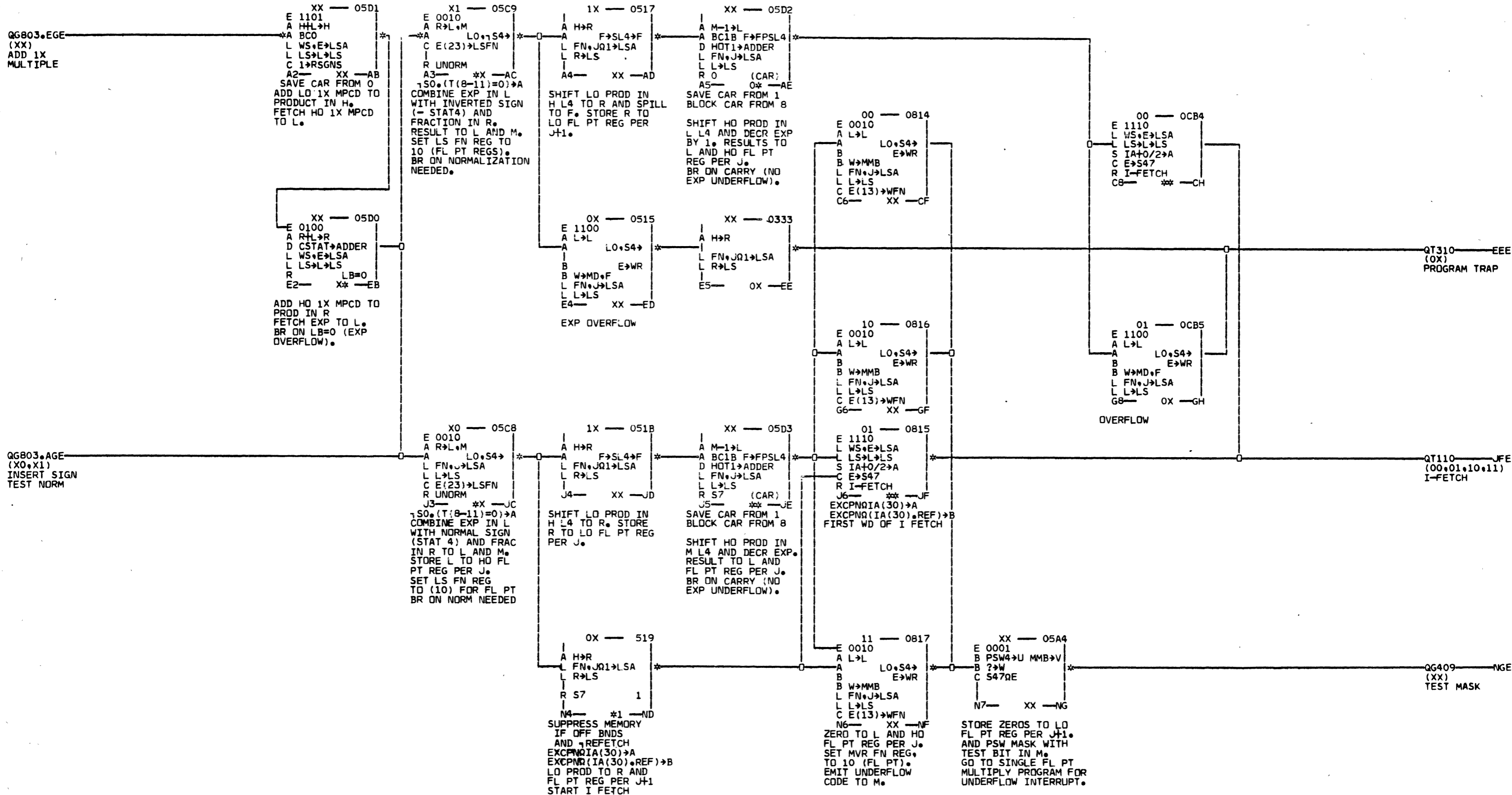
100800



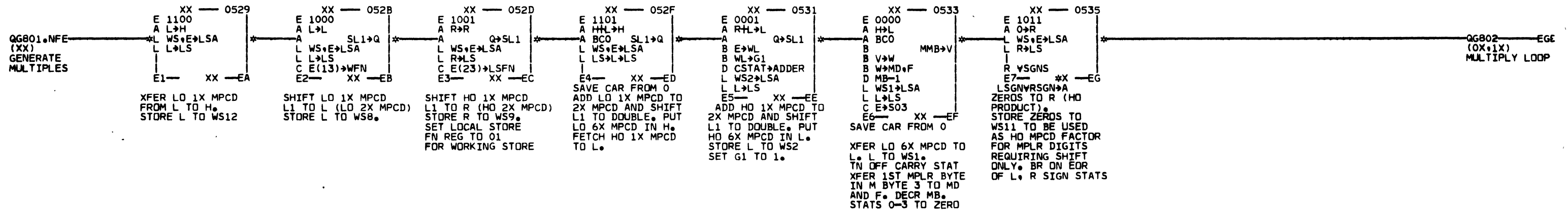
20600



00000

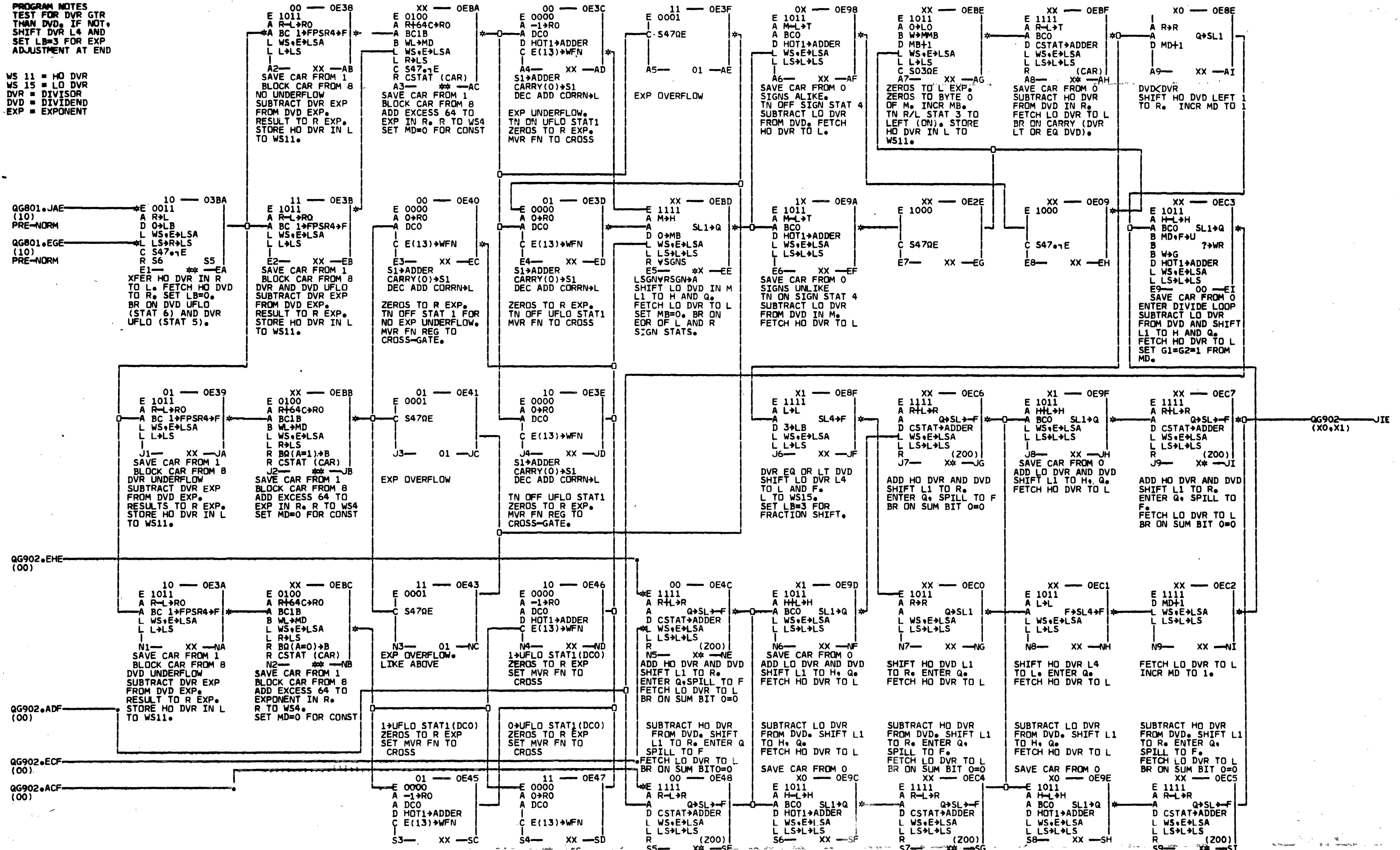


40860

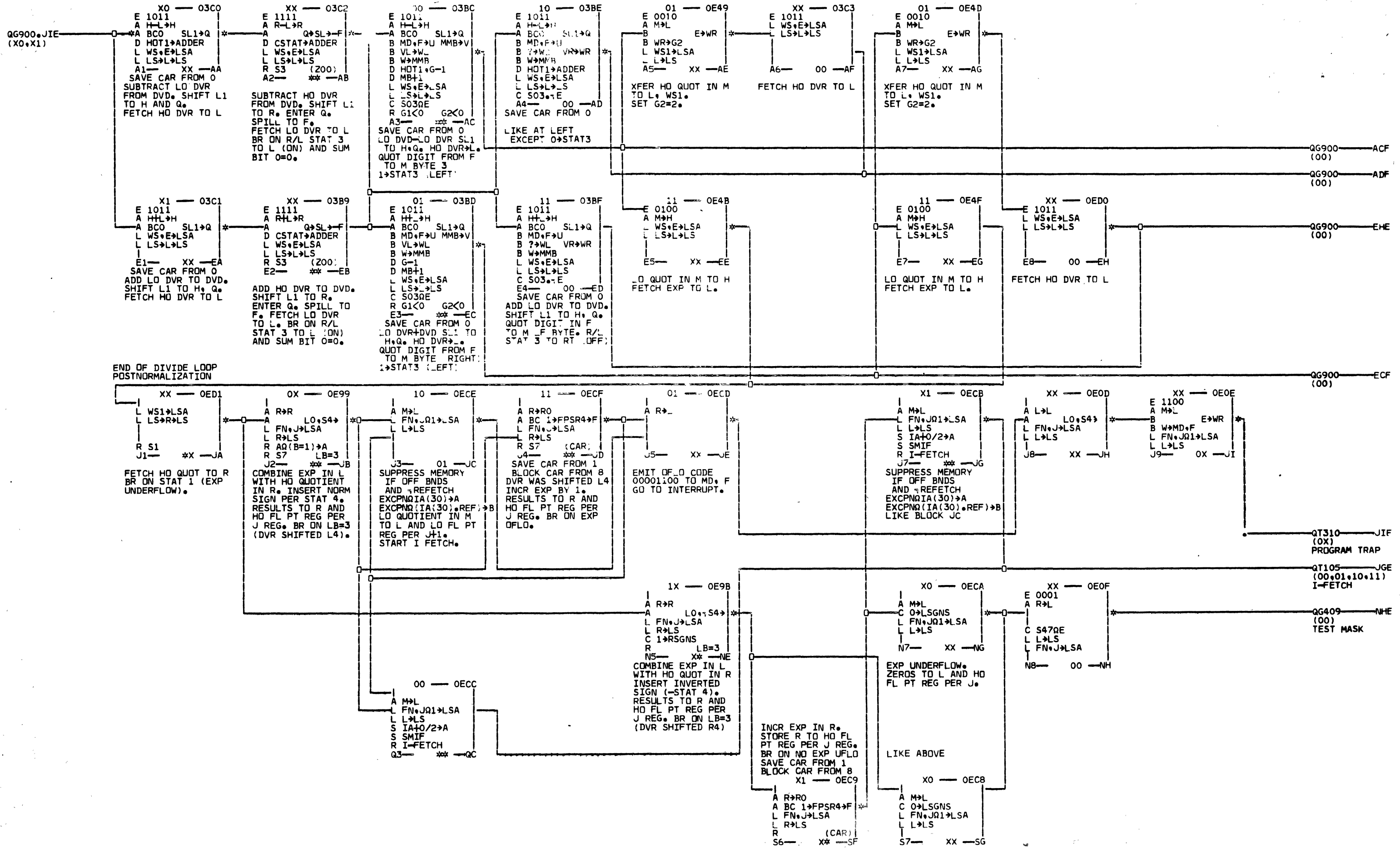


PROGRAM NOTES
 TEST FOR DVR GTR
 THAN DVD. IF NOT,
 SHIFT DVR L4 AND
 SET LB=3 FOR EXP
 ADJUSTMENT AT END

WS 11 = HO DVR
 WS 15 = LO DVR
 DVR = DIVISOR
 DVD = DIVIDEND
 EXP = EXPONENT



QG900



END OF DIVIDE LOOP
POSTNORMALIZATION

NO. 660

100000 — 0120
 A M23←R
 B WR←F
 L FN←MD←LSA
 L LS←L←LS
 R M(47)←RDAR
 A1 ←XXXXX ←AA

B NOT 0.
 B←D TO R.
 OPND 1 TO L
 CLEAR F

QT115.CHD
 (100000)
 (100001)
 (100100)
 (100101)
 FIRST LEVEL
 I-FETCH

100001 — 0121
 A M23←R
 B WR←F
 L FN←MD←LSA
 L LS←L←LS
 R M(47)←RDAR
 E1 ←XXXXX ←EA

B = 0
 D TO R (ADDR)
 OPND 1 TO L
 CLEAR F

100100 — 0124
 E 0010
 A M23←R
 L WS1←LSA
 L R←LS
 C E←S03
 R M(47)←RDAR
 L3 ←XXXXX ←LC
 B←D TO R

100101 — 0125
 E 0010
 A M23←R
 L WS1←LSA
 L R←LS
 C E←S03
 R M(47)←RDAR
 N3 ←XXXXX ←NC
 D TO R.
 ADDR TO LS (WS1)
 SET STATS (FOR
 IO OPS).

QJ200 — AAE
 (0000XX)SSM
 (0010XX)LPSW

QM111 — AAE
 (0001XX)
 (0100XX)
 (0101XX)
 INVALID OP

QY110 — AAE
 (0011XX)
 DIAGNOSE

QJ600 — AAE
 (0110XX)BXH
 (0111XX)BKLE

QJ080 — AAE
 (1000XX)SRL

QJ090 — AAE
 (1001XX)SLL

QJ100 — AAE
 (1010XX)SRA

QJ110 — AAE
 (1011XX)SLA

QJ120 — AAE
 (1100XX)SRDL

QJ130 — AAE
 (1101XX)SLDL

QJ140 — AAE
 (1110XX)SRDA

QJ150 — AAE
 (1111XX)SLDA

QK666 — LCE
 (0000XX)STM
 (1000XX)LM

QK555 — LCE
 (0001XX)TM
 (0101XX)CLI

QK222 — LCE
 (0010XX)MVI
 (0100XX)INI
 (0110XX)OI
 (0111XX)XI

QK300 — LCE
 (0011XX)TS

QM111 — LCE
 (1001XX)
 (1010XX)
 (1011XX)
 INVALID OPS

QK700 — LCE
 (1100XX)SID
 (1101XX)TID
 (1110XX)HID
 (1111XX)TCH

PHOLD

255096
 255443
 255449

03/02/65
 07/20/65
 09/20/65

MACH
 NAME
 MODE
 P.N.
 IBM CORP.

C2050
 MANUAL
 5364769
 SDD

DATE 10/01/65
 LOG 902

SHEET
 VERSION

1 QJ011

RS-SI 2ND LEVEL I-FETCH.
 (8-19-)

QJ011.AAE
(1000XX)
SECOND LEVEL
I-FETCH

1000XX 0722
A R→R
A SR1→Q
B R3→U
B U→W
B W67→LB
D O→MB
E1 XX EA
N(1) TO LB
N RIGHT 1.

XX 07AE
A R→R
A SR1→Q
D LB-1
R E2
E2 XX EB
LB=0
N RIGHT 1.
TEST FOR
N(1)=0.

NO 1-BIT SHIFTS
X1 07B1
A L→R
B R3→U
B VL→WL
B W→G
C3 01 CC

SET N(4) INTO G

N(1) NOT ZERO
X0 07B0
A L→R
A SR1→Q
B R3→U
B VL→WL
B W→G
D LB-1
R O LB=0
J3 O* JC

FIRST 1-BIT SH.
SET N(4).

1-BIT SHIFT LOOP

00 07AC
A R→R
A SR1→Q
D LB-1
R O LB=0
JA O* JD

DPND NOW IN R

4-BIT SHIFT LOOP

01 07AD
E 1110
A R→R
A SR4→F
D G-1
L WS→E→LSA
L LS→L→LS
C E→S47
R GZ/MB3 1
E7 *1 EG

OP BUF TO L
SET STATS FOR
I-FETCH OVLAP.
SHIFT 4.

OVERSHIFT ONE
4-BITTER.

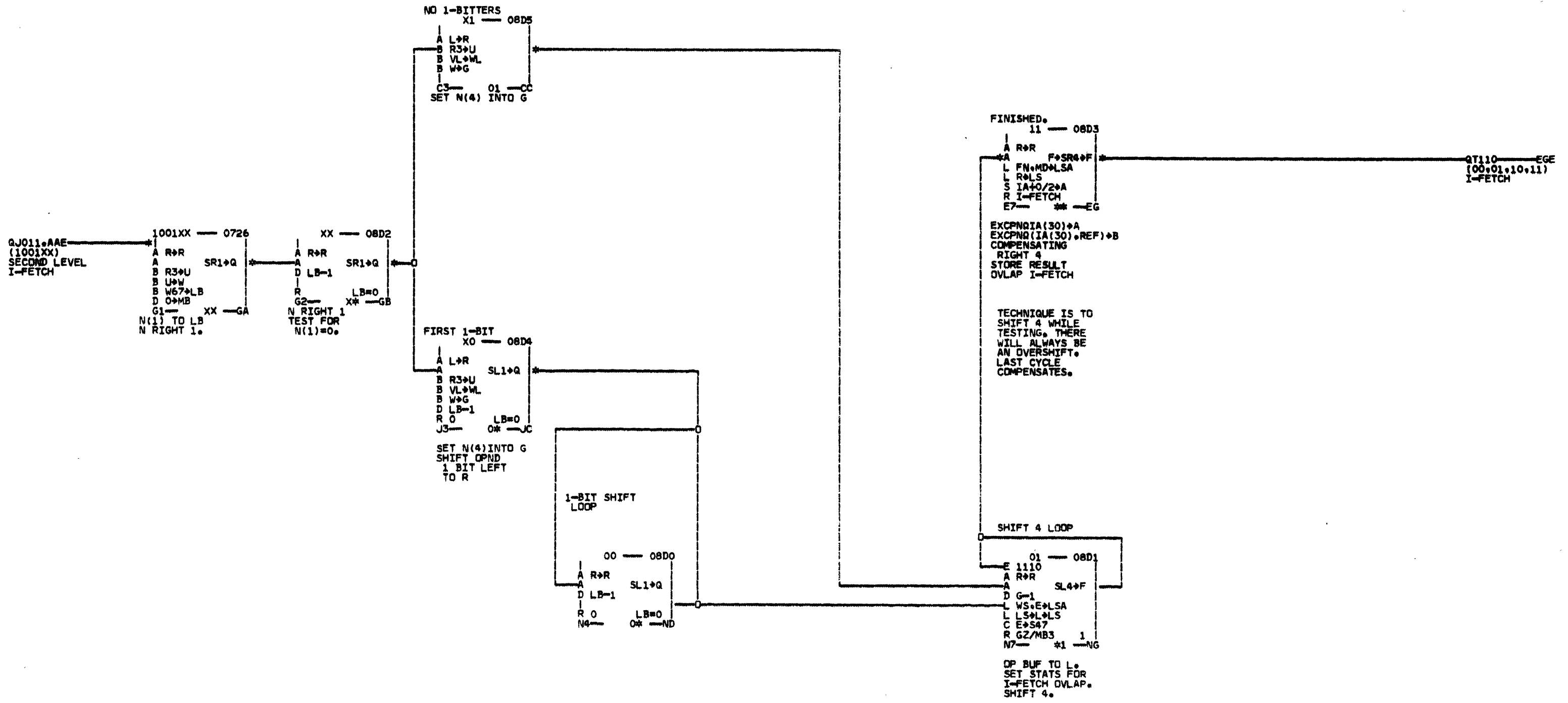
NO MORE 4-BITS
11 07AF
A R→R
A F→SL4→F
L FN→MD→LSA
L R→LS
S IA+O/2→A
R I-FETCH
J7 *JG

EXCPNDIA(30)→A
EXCPND(IA(30).REF)→B
SHIFT LEFT 4
TO COMPENSATE
FOR OVERSHIFT
STORE RESULT
OVLAP I-FETCH

QT110 JGE
(00.01.10.11)
I-FETCH

GZ/MB3 BRANCHES WHEN G=0
(N(4)=0) SINCE MB WAS SET TO ZERO
AND DOES NOT ENTER INTO BRANCH

ALL SHIFTS ARE DONE BY
0-3 1-BIT SHIFTS FOLLOWED
BY 0-15 4-BIT SHIFTS.
1-BIT SHIFTS ARE COUNTED
IN LB - 4-BITERS IN G2.
N(4) IS OBTAINED BY
SHIFTING N (IN R) 2 BITS
RIGHT, THEN THRU MVR TO G2.



SINCE MB WAS SET TO 0,
G2/MB3 BRANCHES ONLY ON
G=0 (IE N(4)=0).

09010

F→SR1→Q SPILLS TO Q.
 ENTERS FROM F
 BUT DOES NOT
 SHIFT F.

QJ011.AAE
 (1010XX)
 SECOND LEVEL
 I-FETCH

1010XX — 072A
 A R→R
 B R3→U
 B U→W
 B W67→LB
 C L(0)→LSGNS
 R L(0)
 E1 — *X — EA
 SET N(1) TO LB
 N RIGHT 1.

OPND NEGATIVE
 1X — 0BAA
 E 1111
 A R→R
 B SR1→Q
 B WR→F
 D E→WR
 D LB=1
 R LB=0
 C3 — *X — CC

N RIGHT 1. SET
 ONES FOR FILL

OPND POSITIVE
 0X — 0BA8
 A R→R
 B SR1→Q
 B WR→F
 D E→WR
 D LB=1
 R LB=0
 J3 — *X — CC

FIRST 1-BIT SHIFT
 X0 — 0BA6

A L→L
 B R3→U
 B U→W
 B WR→G2
 D LB=1
 R LB=0
 E4 — *X — ED

RIGHT 1.
 N(4) TO G2.

NO 1-BIT
 SHIFTS
 X1 — 0BA7

B R3→U
 B U→W
 B WR→G2
 L4 — X1 — LD
 N(4) TO G2.

NO MORE 1-BIT
 SHIFTS. STORE
 RES. TEST N(4).
 X1 — 0BA5

A L→T
 D G2=1
 L FN→MD→LSA
 L L→LS
 C SETCRALG
 R AR(B=1)→A
 R O G2=0
 L5 — ** — LE
 IF T=0: 00→CR
 IF T<0: 01→CR
 IF O<T: 10→CR

STORE RESULT. SET
 CR IN CASE N(4)
 IS ZERO.
 TEST N(4).

RIGHT 1 LOOP
 X0 — 0BA4

A L→L
 B R3→U
 B U→W
 B WR→G2
 D LB=1
 R LB=0
 Q4 — *X — QD
 RIGHT 1.

RIGHT 4 LOOP

00 — 0B80
 A L→L
 B R3→U
 B U→W
 B WR→G2
 D G2=1
 L FN→MD→LSA
 L L→LS
 C SETCRALG
 R AR(B=1)→A
 R O G2=0
 J7 — ** — JG
 IF T=0: 00→CR
 IF T<0: 01→CR
 IF O<T: 10→CR

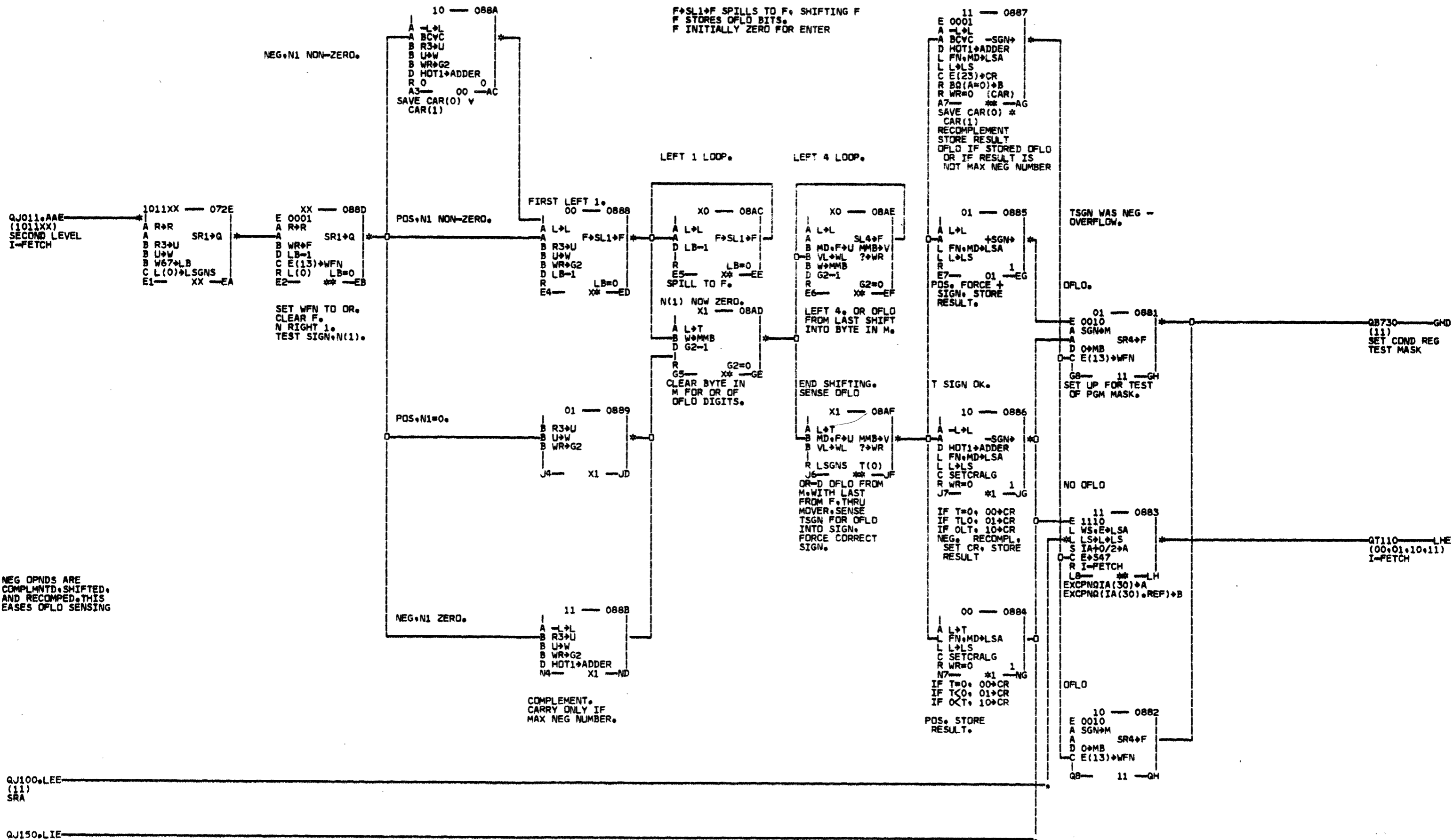
FINISHED

PROCEDURE HERE IS
 TO TEST BEFORE
 SHIFTING 4. AND
 TO STORE RESULT
 AFTER EACH SHIFT.

ALGEBRAIC RIGHT SHIFTS
 PUT 0000 OR 1111 INTO
 F. THEN ENTER FROM F
 TO ENTER THE SAME BIT
 AS THE SIGN BIT OF THE
 OPERAND.

QJ110 — LEE
 (11)
 I-FETCH

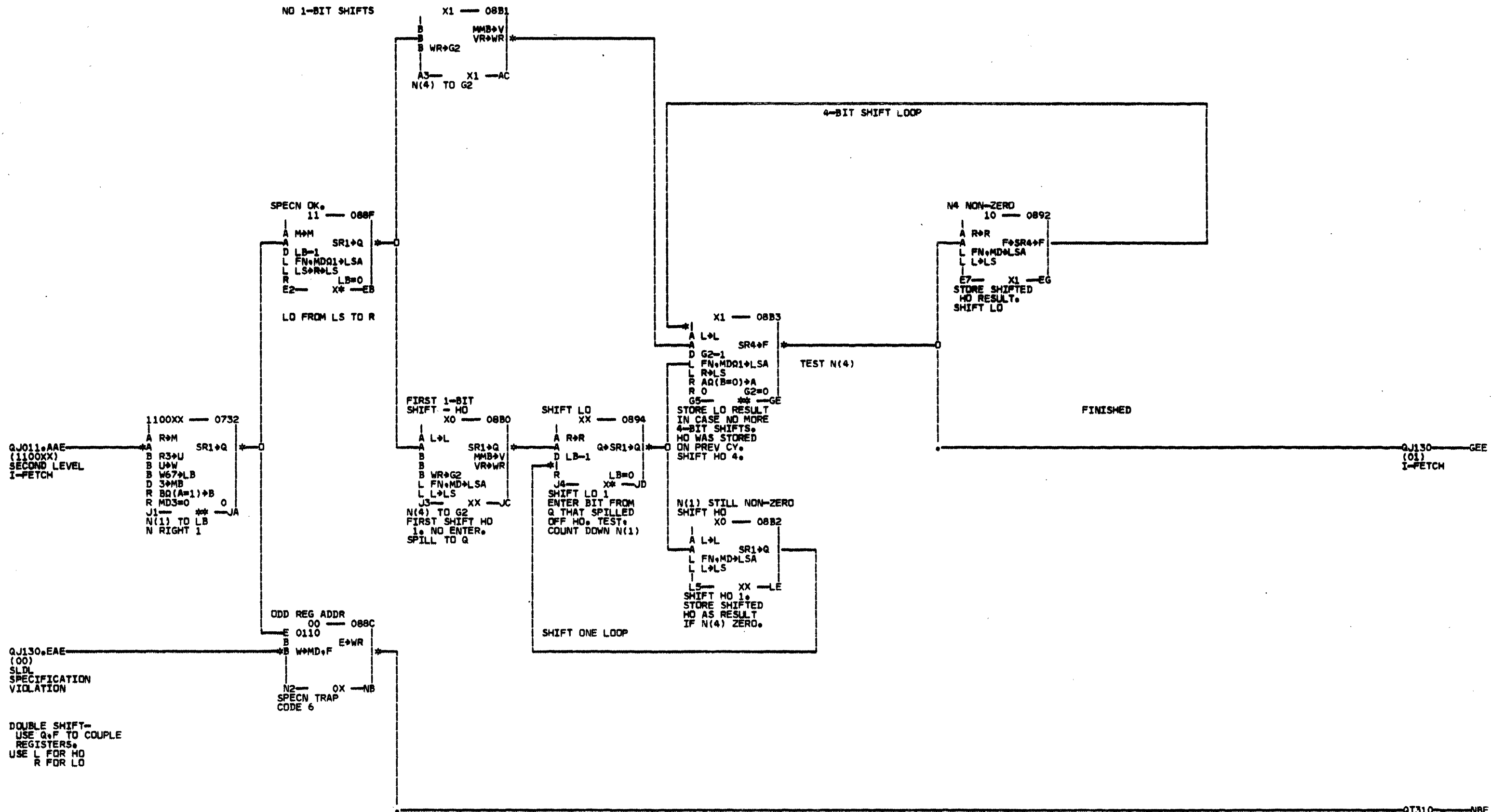
00100



NEG OPNDs ARE COMPLEMENTED, SHIFTED, AND RECOMPED. THIS EASES OFLO SENSING

QJ100.LEE (11) SRA

QJ150.LIE (01,11) SLDA OVERFLOW TEST



DOUBLE SHIFT-
 USE Q,F TO COUPLE
 REGISTERS.
 USE L FOR HO
 R FOR LO

ON-C-D

254760
 255449

12/21/64
 09/20/65

MACH
 NAME
 MODE
 P.No.
 IBM CORP.

C2050
 MANUAL
 5364774
 SDD

DATE 10/01/65
 LOG 902

SHEET
 VERSION

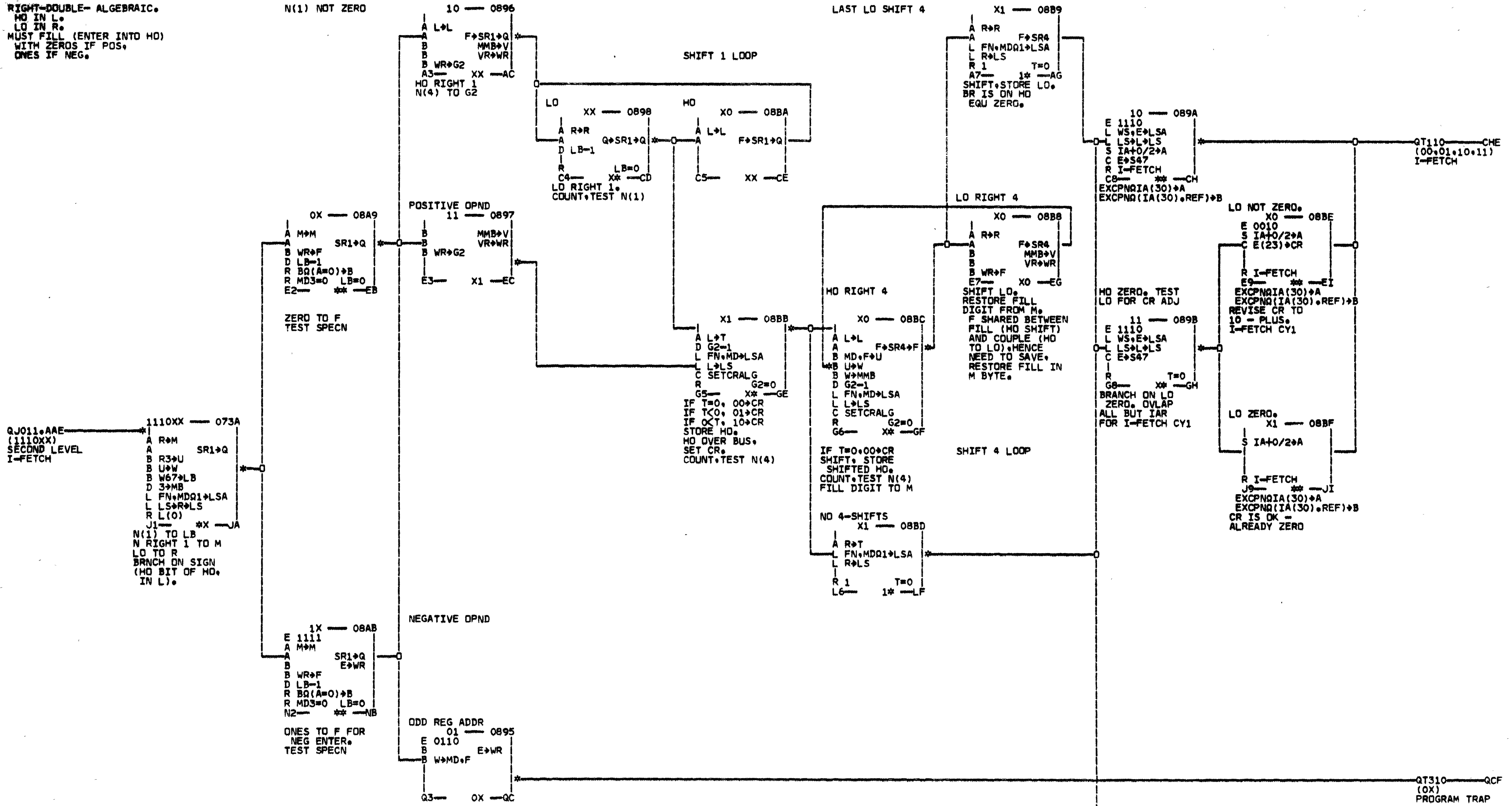
1 QJ120

SHIFT RIGHT DOUBLE LOGICAL
 (SRDL)

RIGHT-DOUBLE- ALGEBRAIC.
 HD IN L.
 LD IN R.
 MUST FILL (ENTER INTO HD)
 WITH ZEROS IF POS.
 ONES IF NEG.

N(1) NOT ZERO

LAST LD SHIFT 4



QJ150.EHE
 (10.11)
 SLDL
 SET COND REG

QJ011.LCE
(0010XX)
(0100XX)
(0110XX)
(0111XX)
SECOND LEVEL
I-FETCH

ADDR IS IN R.
I OPND IN M BYTE 1

LB IS 1.

0100XX — 0912
E 0010
A R→R,A
B R3→U
B U→W
B W67→MB
C E(13)→WFN
A1 — XX — AA

0110XX — 091A
E 0001
A R→R,A
B R3→U
B U→W
B W67→MB
C E(13)→WFN
E1 — XX — EA

0111XX — 091E
E 0011
A R→R,A
B R3→U
B U→W
B W67→MB
C E(13)→WFN
J1 — XX — JA

SET WFN FOR OP.
SET MB FROM ADDR
START MEM CY
RESET BYTE STATS

QE100.GBE
(XX)
STORE CHAR

QE555.GEE
(XX)
STORE HALFWORD

QJ400.NFE
(XX)
READ DIRECT

MVI

0010XX — 090A
E 0000
A R→A
B R3→U
B U→W
B W67→MB
C E→BS
Q1 — XX — QA
SET MB FROM ADDR
RESET BYTE STATS
START STORE CYCLE

XX — 091C
B V→W
B W→MMB
C 1→BS*MB
Q2 — XX — QB
BYTE FROM M(1) TO
M(X). TURN ON BS
FOR X BYTE FOR
STORE

XX — 0920
A M→D*BS
Q3 — 1X — QC
M TO SDR.
STORE PER BYTE
STATS.

XX — 0924
E 0000
D D→L
C E→BS
E3 — XX — EC
STORAGE OPND TO L
RESET BYTE STATS

XX — 0928
A R→A
B LMB→U MLB→V
B 7→W
B W→MMB
C 1→BS*MB
E5 — XX — EE
LOGIC OP ON M
BYTE 1 AND L
BYTE X. RESULT
TO M BYTE X.
TURN ON BYTE STAT
FOR BYTE X.
START STORE CY

XX — 092C
A M→D*BS
R W=0
E6 — *X — EF
M TO SDR —
STORE PER BS.
BRANCH ON ZERO
RESULT

NONZERO

OX — 091D
E 0001
C E(23)→CR
C7 — 1X — CG

ZERO

1X — 091F
E 0000
C E(23)→CR
G7 — 1X — GG

QT100 — QCE
(1X)
I-FETCH

NNNNKQ

254760
255449

12/21/64
09/20/65

MACH
NAME
MODE
P.N.
IBM CORP.

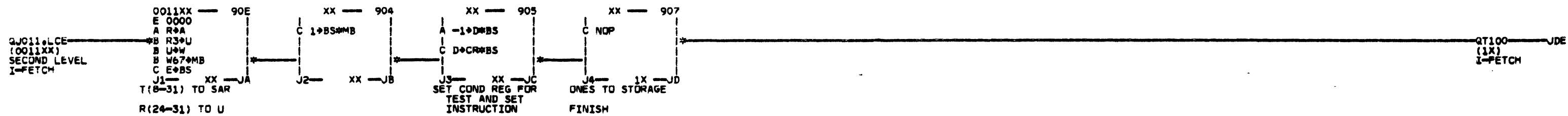
C2050
MANUAL
5364781
SDD

DATE 10/01/65
LOG 902

SHEET
VERSION

1 GK222

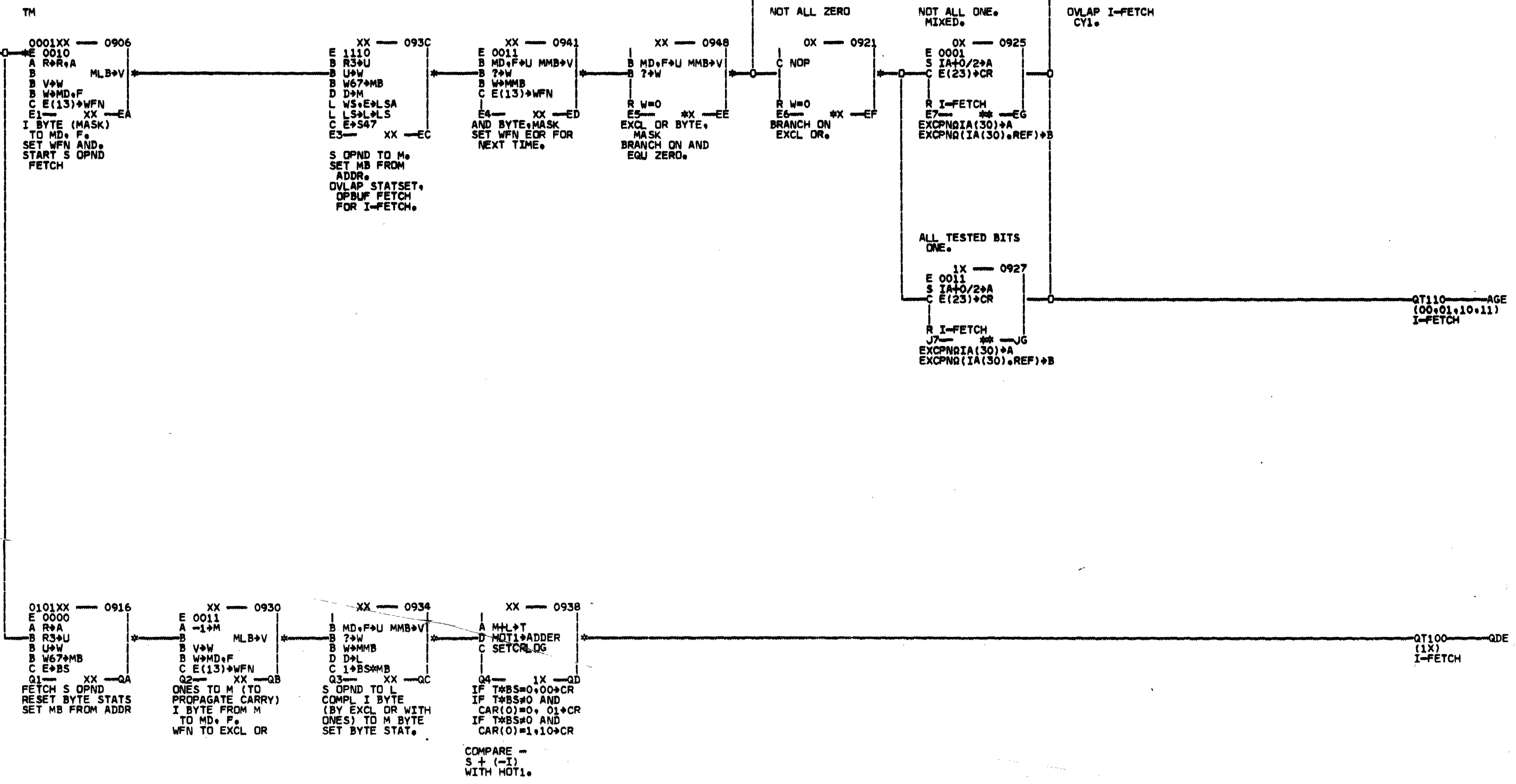
SI MOVE AND LOGICAL OPS
MVI NI OI XI.



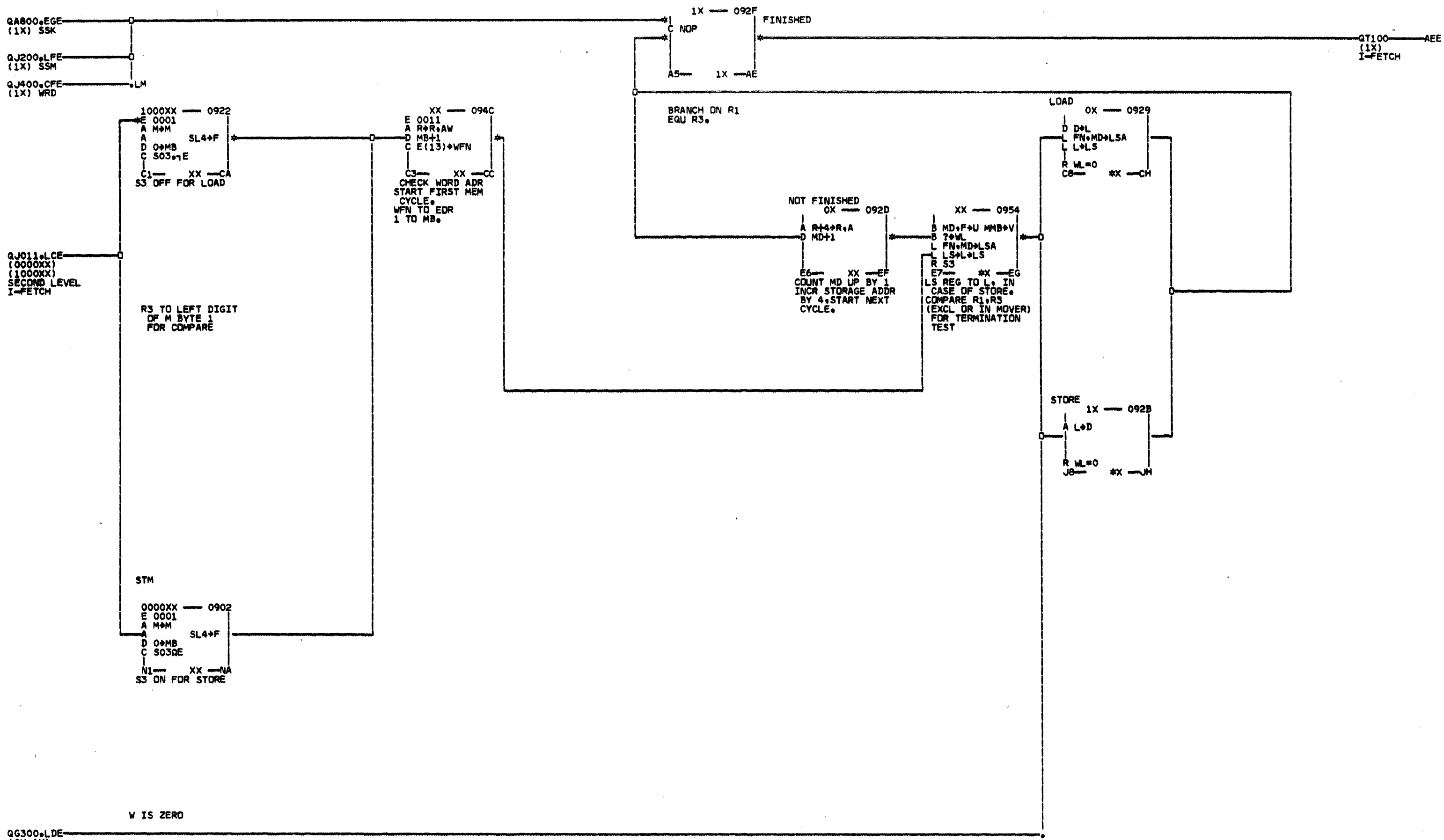
00000

QJ011.LCE
(0001XX)
(0101XX)
SECOND LEVEL
I-FETCH

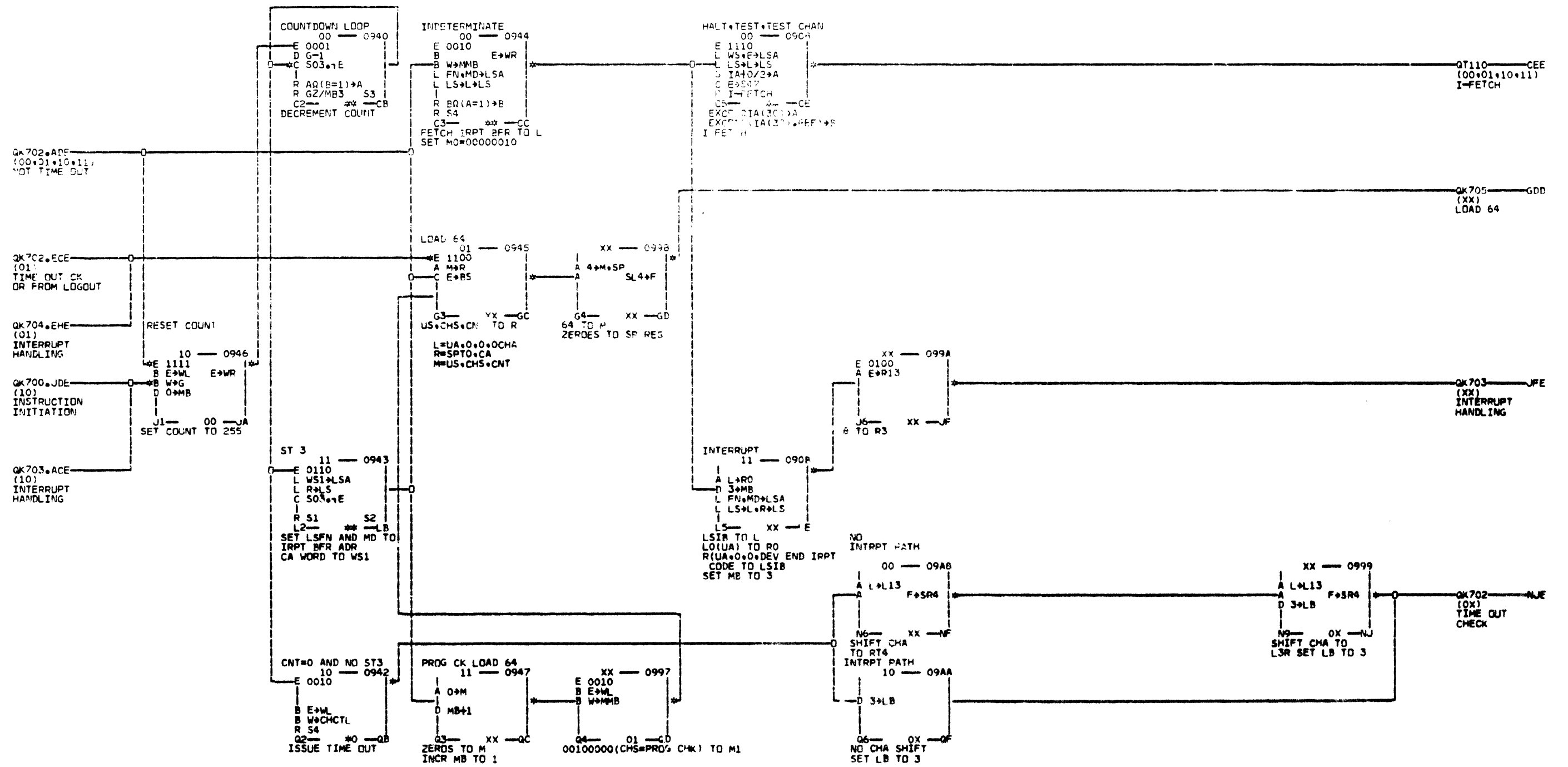
ADDR IS IN R
T IN BYTE 1 OF M
LB IS 1.



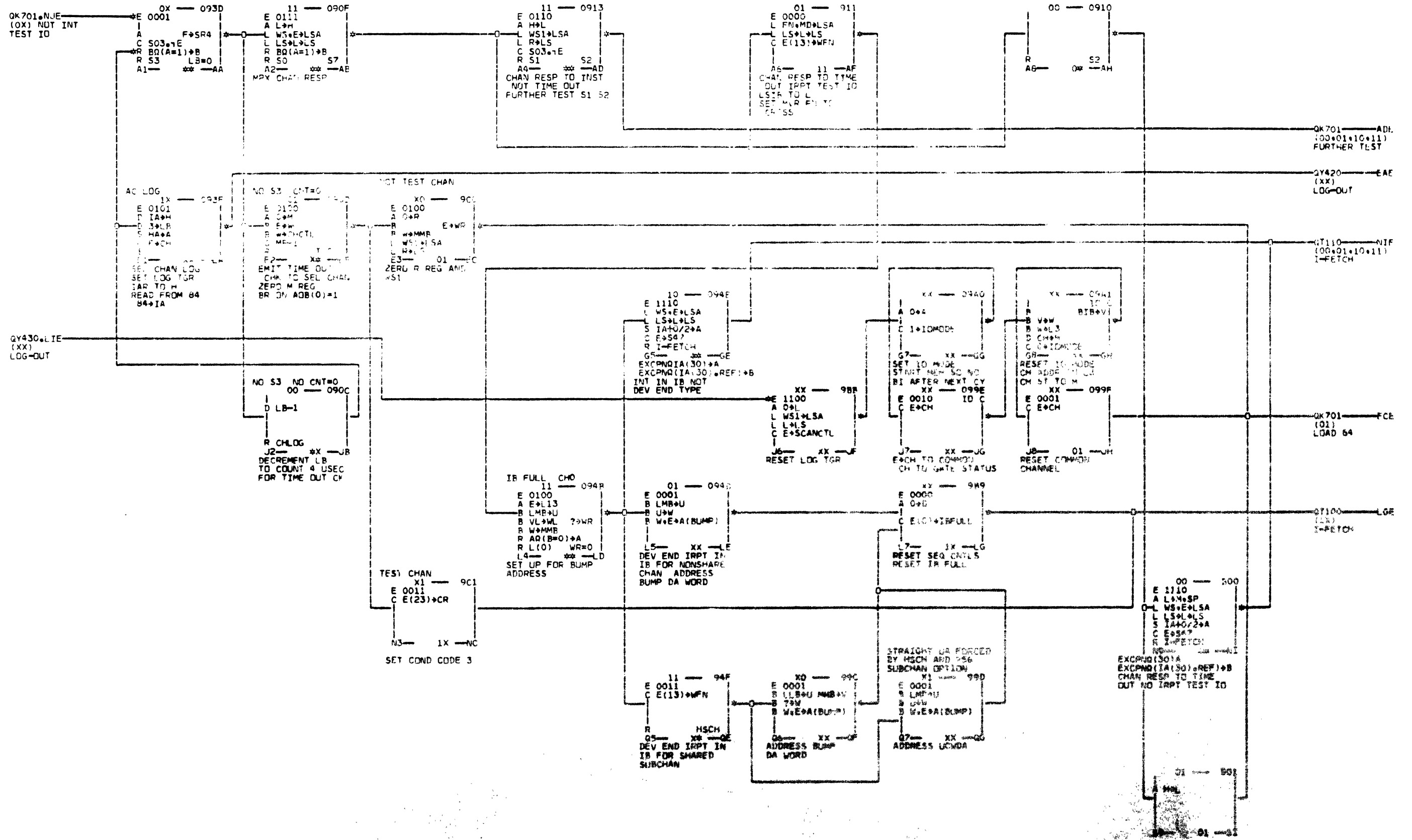
58000



QK666

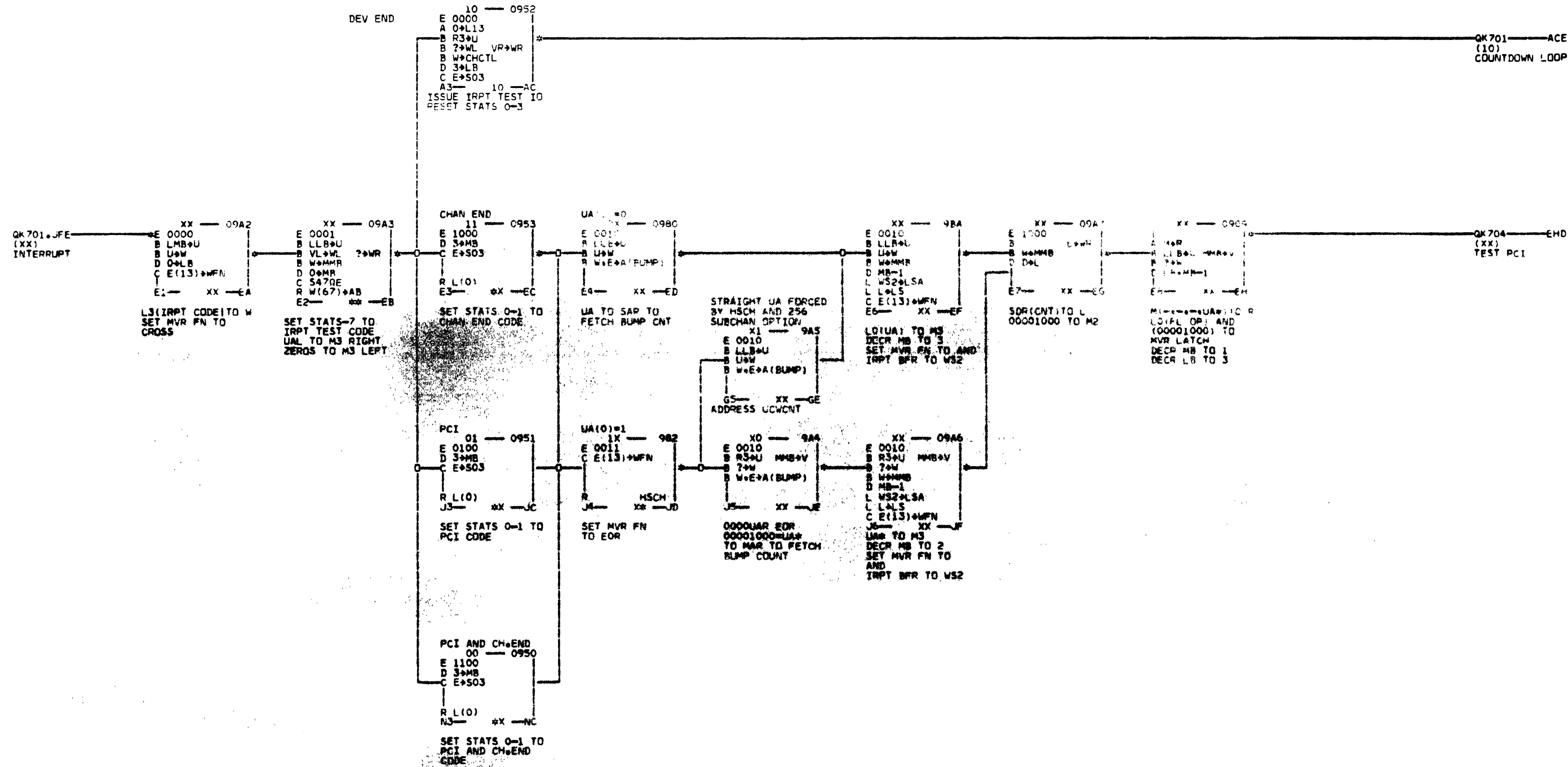


QK701

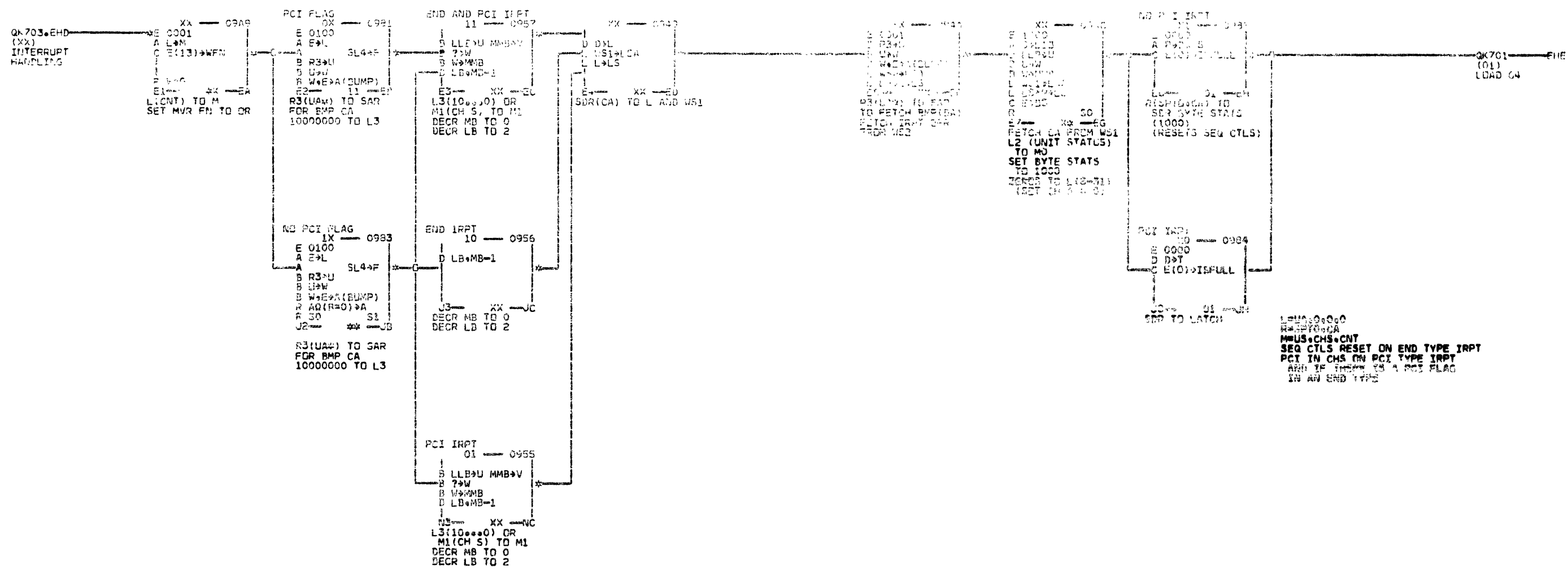


NOV 70

255440	06/10/65	254760	12/21/64	MACH	2080PP	DATE 02/01/71	SHEET	1	QK702
255449	09/20/65	254765	01/13/65	NAME	MANUAL	LOG 050	VERSION		
255399	03/01/68	254770	01/18/65	MODE	5304786				
255825	10/15/69	255033	02/24/65	P.N.	500	TIME OUT CHECK LOOP			
260526	07/01/70	255066	03/02/65	IBM CORP.		IB TEST ON START IO			



4070



L=00000000
 R=00000000
 M=00000000
 SEQ CTLs RESET ON END TYPE IRPT
 PCI IN CHS ON PCI TYPE IRPT
 AND IF THERE IS A PCI FLAG
 IN AN END TYPE

00000

QK701.GDD
(XX)
LOAD 64

XX 09AD
A M+4→A

J1 XX JA
M(00001000000)
+4 TO SAR
(6B TO SAR)

XX 09AE
E 0111 A R→M
L WS→E→LSA
L LS→R→LS
R BQ(A=1)→B
R S4 S5
J2 ** JB

FETCH STO PROT
TO R
R(US,CHS,CNT)
TO M

—START HALT—
00 0914
E 0001 A M→D→BS
C E(23)→CR

E3 XX EC
M(US,CHS,CNT)
TO SDR PER
BYTE STATS (1100)
SET COND REG TO 01

—TEST—
01 0915
E 0001 A M→D
D 0→LB
C E(23)→CR

J3 XX JC
M(US,CHS,CNT)
TO SDR
SET COND REG=01

—INTERRUPT—
11 0917
A M→D
D 0→LB

N3 XX NC
M(US,CHS,CNT)
TO SDR

XX 09AF
E 0000 A 4→A
E→FPSL4

B LLB→U
B U→W
B W→MD→F
D 3→MB
L WS1→LSA
L LS→R→LS
J5 XX JE
64 TO SAR
FETCH CA TO R
LO(UA) TO MD→F

XX 09B1
E 0111 A R→D
D LB,MB-1
L WS→E→LSA
L LS→R→LS

J6 XX JF
R(CA) TO SDR
FETCH STO PROT
TO R
DECR MB TO 2
DECR LB TO 3

XX 09B0
A R→M
R S4
J7 X* JG
R(STO PROT) TO M

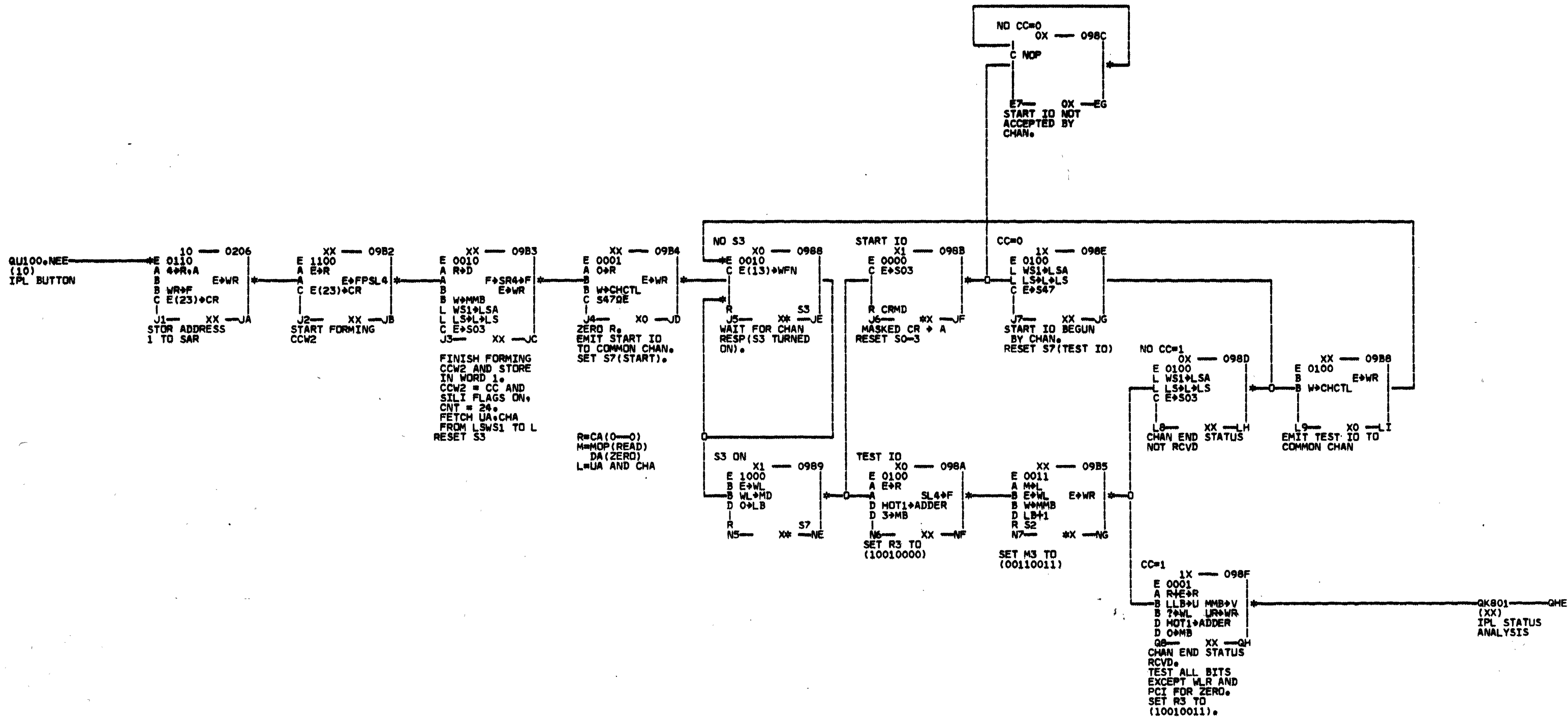
START-TEST-HALT
X0 0986
E 1110 A R→M,SP
L WS→E→LSA
L LS→L→LS
S IA+0/2→A
C E→S47
R I→FETCH
J8 ** JH
EXCPNQIA(30)→A
EXCPNQ(IA(30).REF)→B
I FETCH
RESTORE STG PROTECT

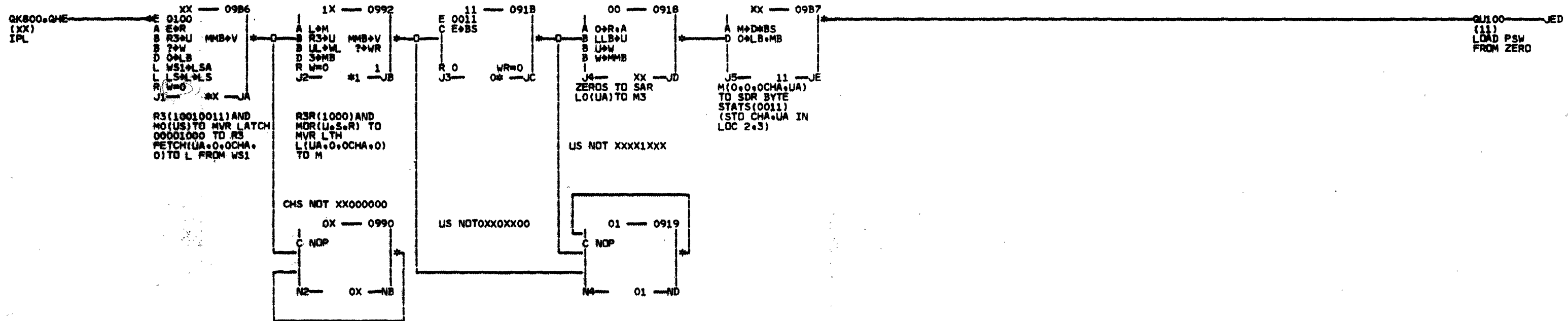
—INTERRUPT—
X1 0987
E 1110 A E→R→A
A SL1→Q
B LLB→U
B U→W
B W→MB
D 3→MB
N8 XX NH
56 TO SAR AND R
L3(CA) TO M2

QT110 JHD
(00,01,10,11)
I→FETCH

QT310 NMD
(XX)
STORE PSW

00000





4-00000