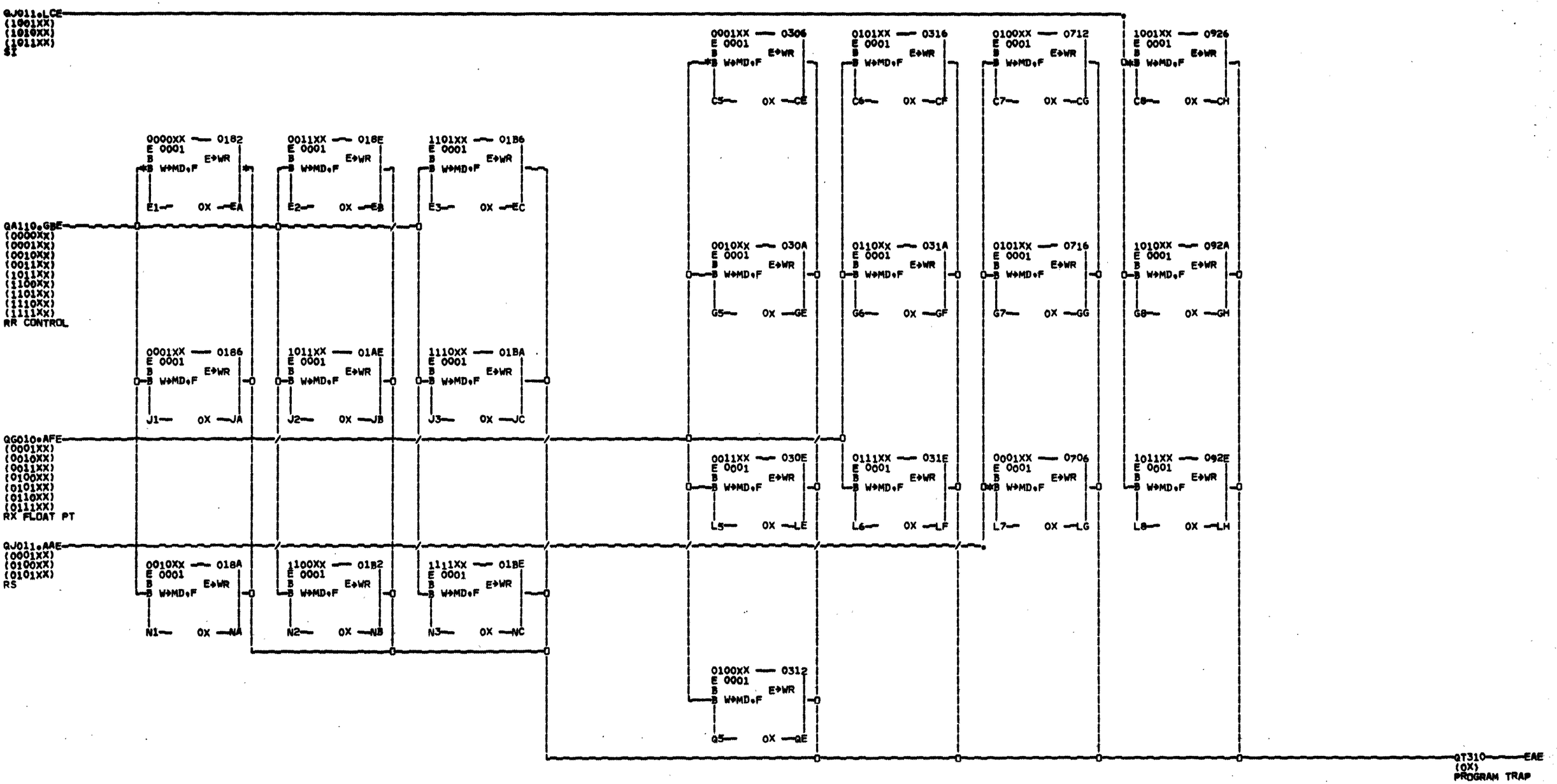
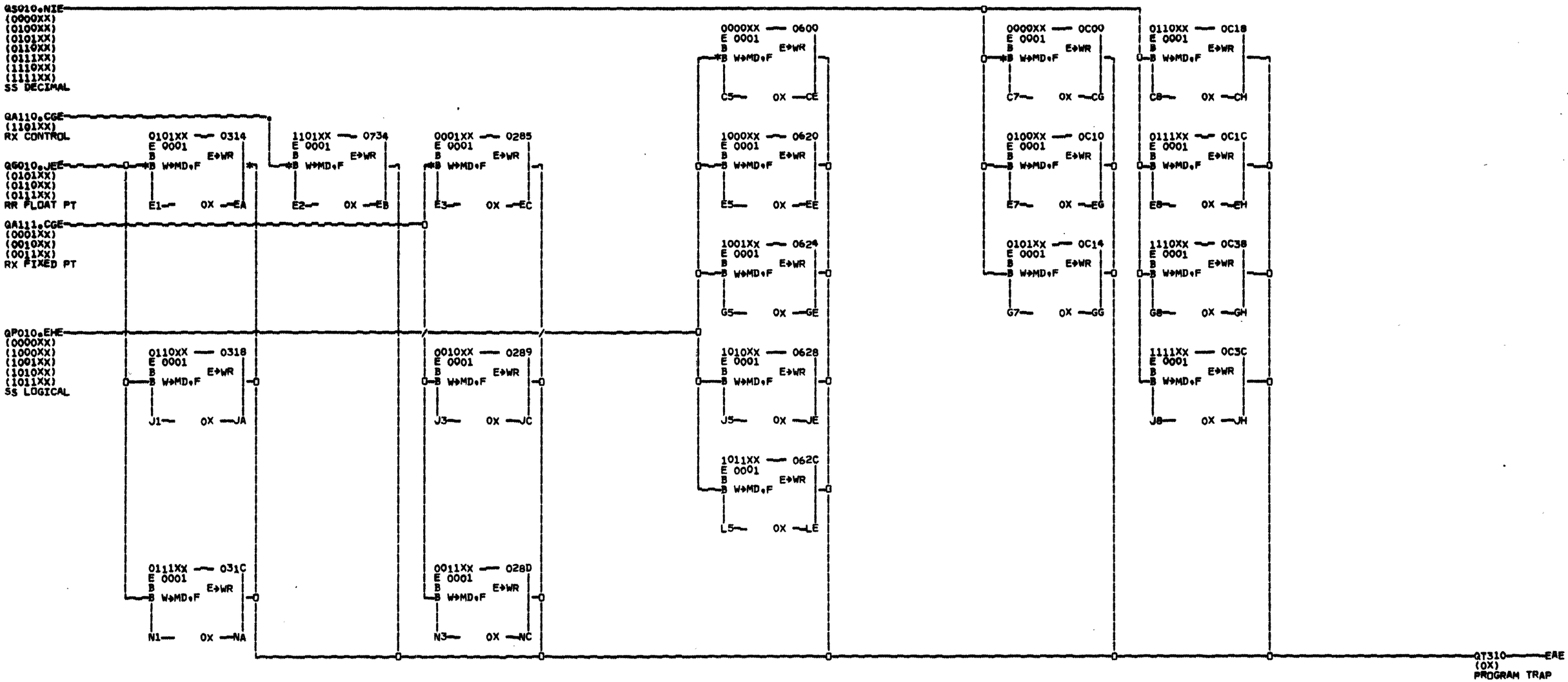
TOTAL PAGES THIS VOLUME 42

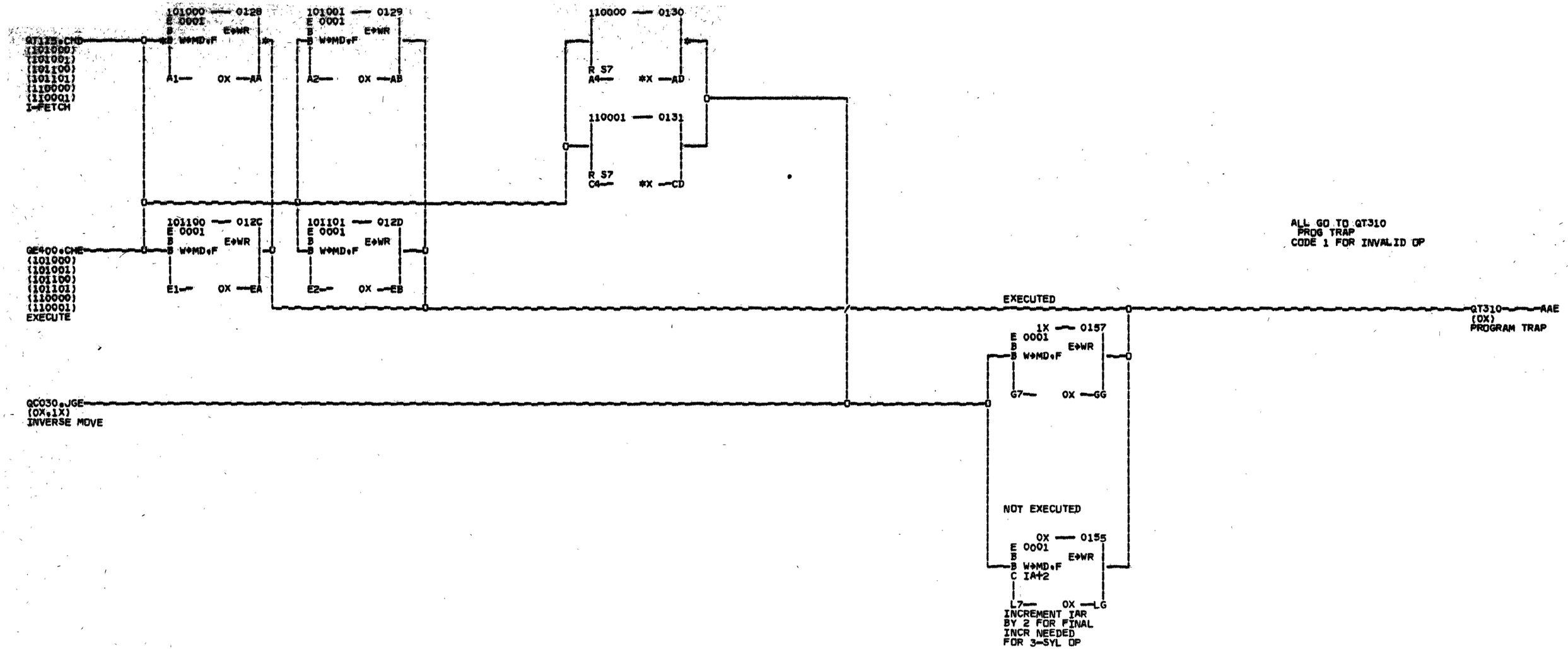
Dec 72


QJ011.LCE
(1001XX)
(1010XX)
(1011XX)
$1
0000XX — 0182
E 0001
B
B W→MD.F
E→WR
E1— 0X —EA
0011XX — 018E
E 0001
B
B W→MD.F
E→WR
E2— 0X —EB
1101XX — 01B6
E 0001
B
B W→MD.F
E→WR
E3— 0X —EC
QA110.GBE
(0000XX)
(0001XX)
(0010XX)
(0011XX)
(1011XX)
(1100XX)
(1101XX)
(1110XX)
(1111XX)
RR CONTROL
0001XX — 0186
E 0001
B
B W→MD.F
E→WR
J1— 0X —JA
1011XX — 01AE
E 0001
B
B W→MD.F
E→WR
J2— 0X —JB
1110XX — 01BA
E 0001
B
B W→MD.F
E→WR
J3— 0X —JC
QG010.AFE
(0001XX)
(0010XX)
(0011XX)
(0100XX)
(0101XX)
(0110XX)
(0111XX)
RX FLOAT PT
QJ011.AAE
(0001XX)
(0100XX)
(0101XX)
RS
0010XX — 018A
E 0001
B
B W→MD.F
E→WR
N1— 0X —NA
1100XX — 01B2
E 0001
B
B W→MD.F
E→WR
N2— 0X —NB
1111XX — 01BE
E 0001
B
B W→MD.F
E→WR
N3— 0X —NC
0001XX — 0306
E 0001
B
B W→MD.F
E→WR
C5— 0X —CE
0101XX — 0316
E 0001
B
B W→MD.F
E→WR
C6— 0X —CF
0100XX — 0712
E 0001
B
B W→MD.F
E→WR
C7— 0X —CG
1001XX — 0926
E 0001
B
B W→MD.F
E→WR
C8— 0X —CH
0010XX — 030A
E 0001
B
B W→MD.F
E→WR
G5— 0X —GE
0110XX — 031A
E 0001
B
B W→MD.F
E→WR
G6— 0X —GF
0101XX — 0716
E 0001
B
B W→MD.F
E→WR
G7— 0X —GG
1010XX — 092A
E 0001
B
B W→MD.F
E→WR
G8— 0X —GH
0011XX — 030E
E 0001
B
B W→MD.F
E→WR
L5— 0X —LE
0111XX — 031E
E 0001
B
B W→MD.F
E→WR
L6— 0X —LF
0001XX — 0706
E 0001
B
B W→MD.F
E→WR
L7— 0X —LG
1011XX — 092E
E 0001
B
B W→MD.F
E→WR
L8— 0X —LH
0100XX — 0312
E 0001
B
B W→MD.F
E→WR
Q5— 0X —QE
QT310 —EAE
(0X)
PROGRAM TRAP



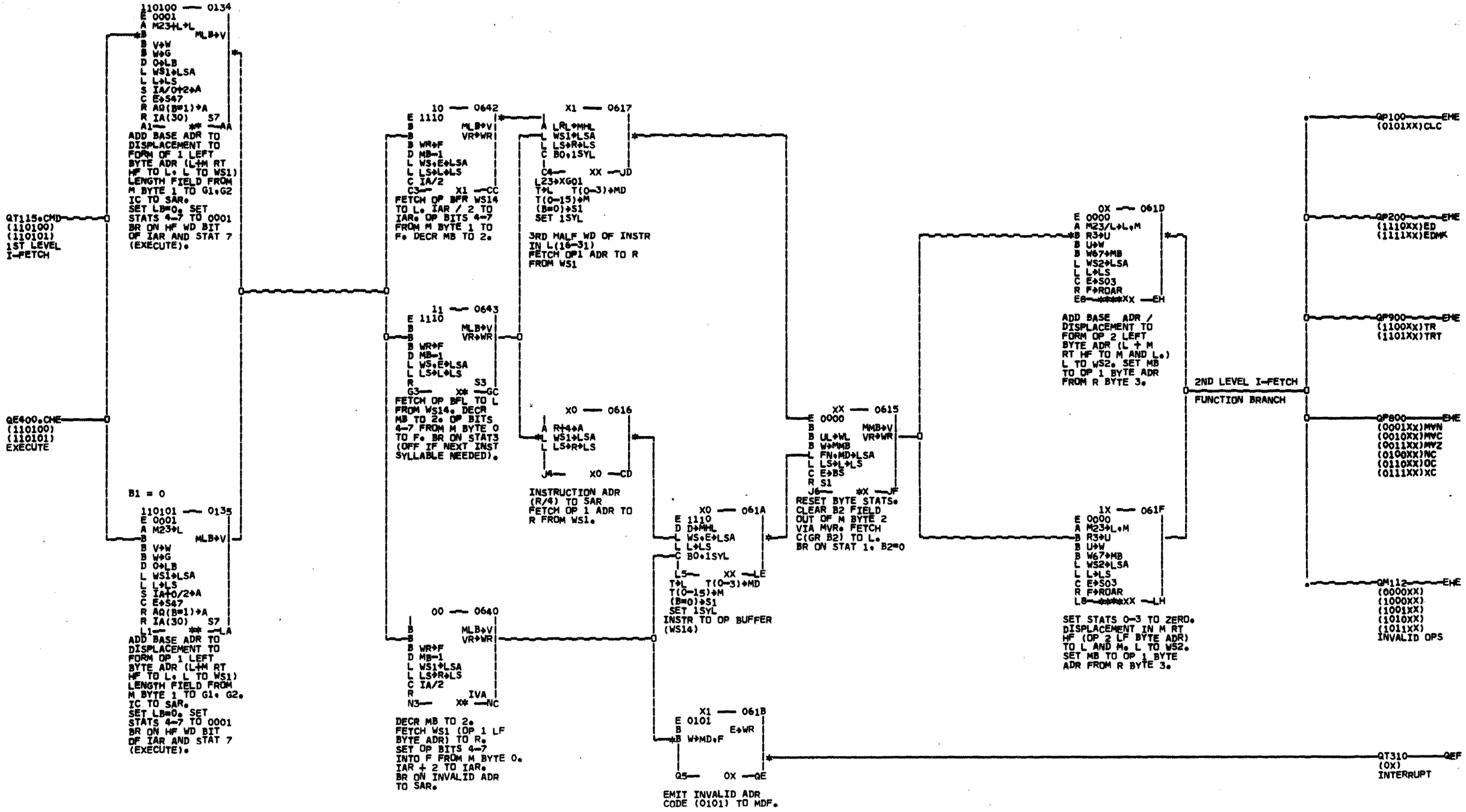
QM112

255449	09.20.65	MACH	C2050	DATE	10.01.65	SHEET	1 QM112
		NAME		LOG	902	VERSION	
		MODE	MANUAL				
		P.N.	5364920		INVALID OPS.		
		IBM CORP.	SDD				

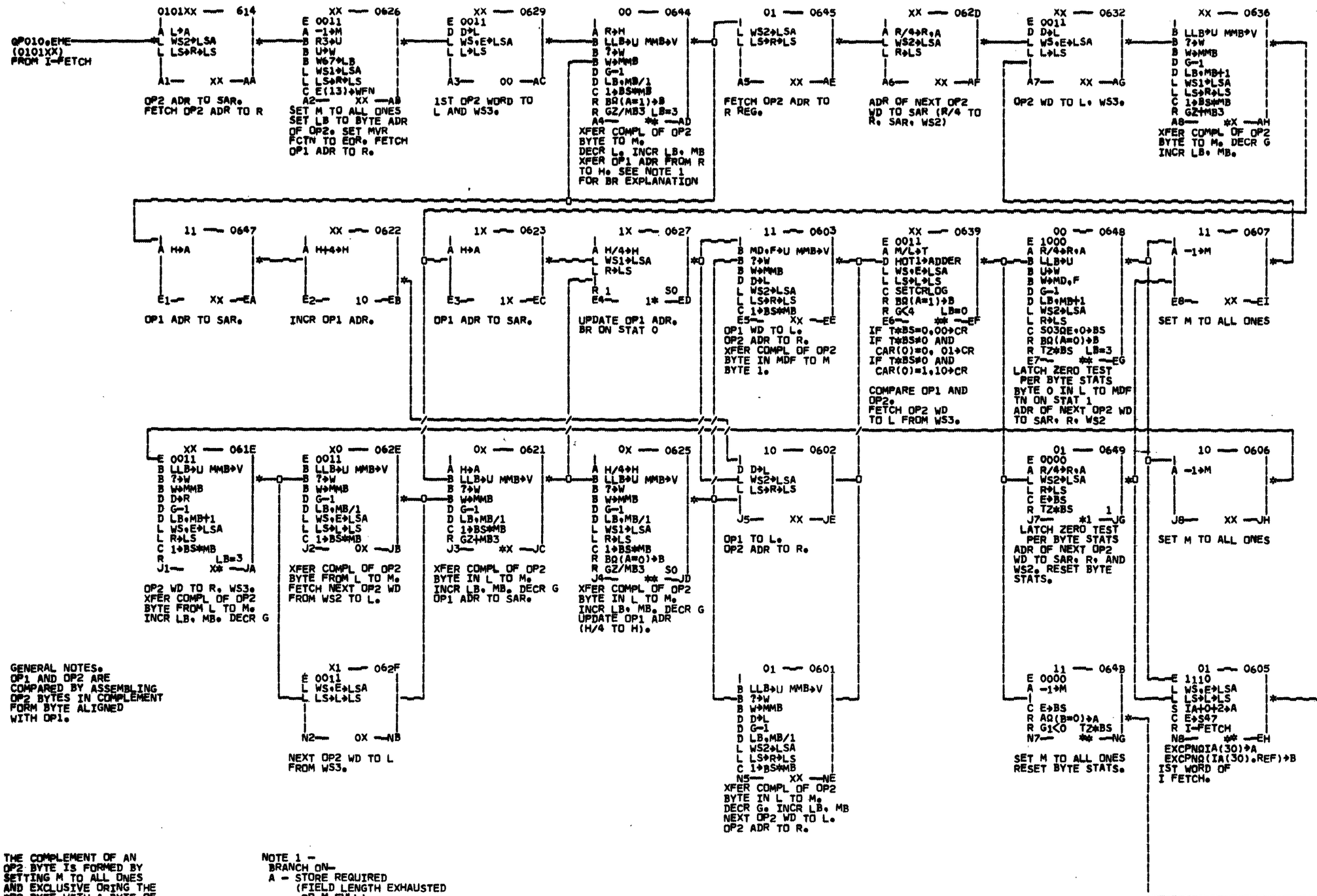


282229	11+17/67	MACH	C2050	DATE 01+30/68	SHEET 1 QN111
		NAME		LOG 627	VERSION C03
		MODE	MANUAL	INVALID OP GROUPS-INVERSE MOVE	
		P•N•	5379142	(A- B- C- E-)•	
		IBM CORP•	SDD		

QN111

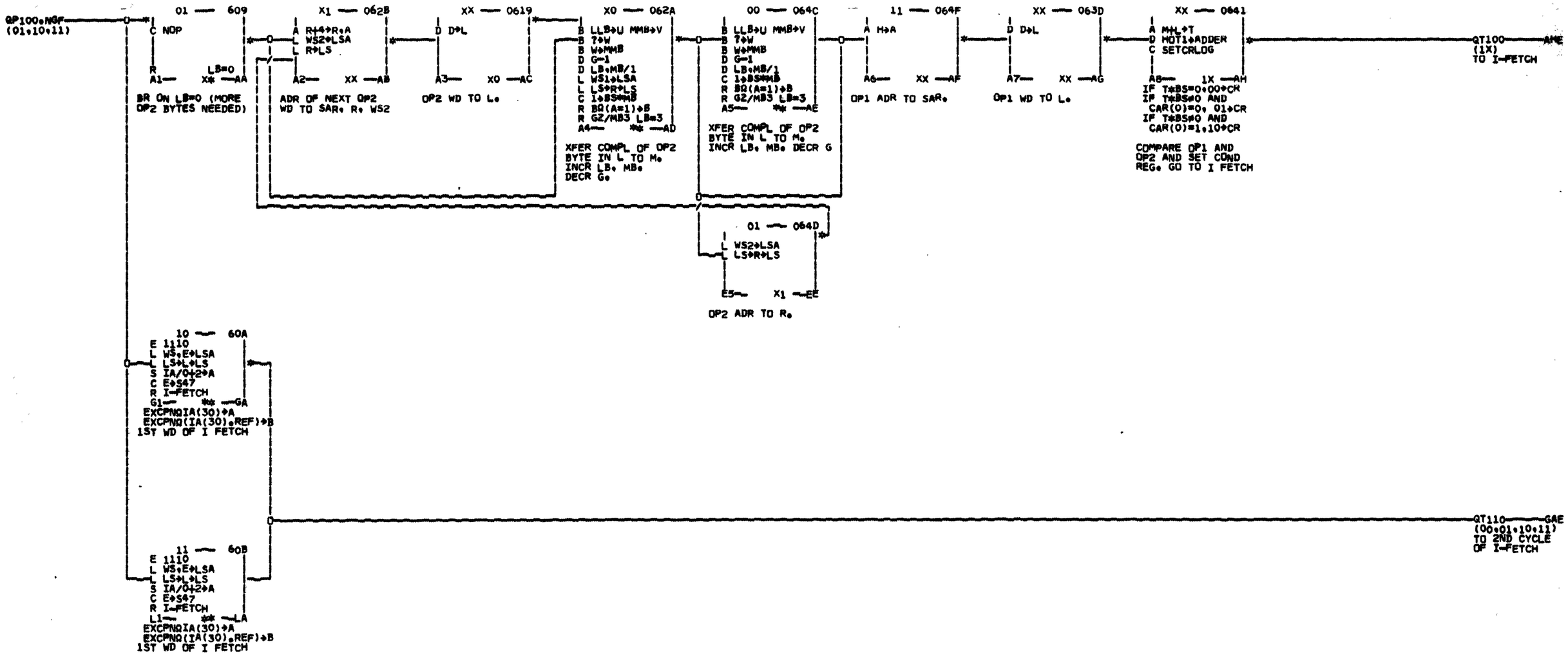


254760	12+21/64	MACH	C2050	DATE 10+01/65	SHEET 1 QP010
255449	09+20/65	NAME		LOG 902	VERSION
		MODE	MANUAL	2ND LEVEL I-FETCH FOR VFL CHARACTER OPS	
		P.N. IBM CORP.	5364794 SDD		

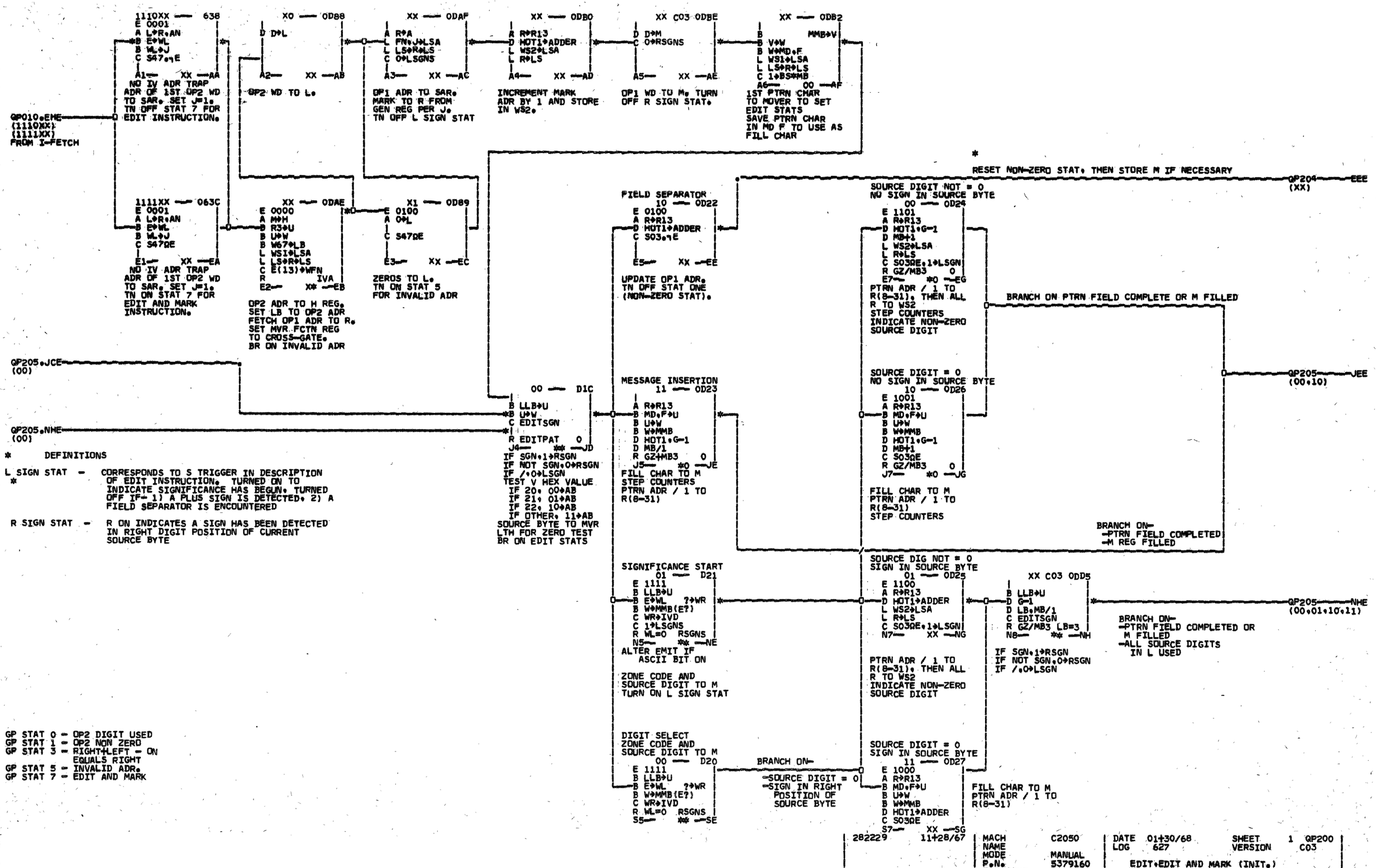


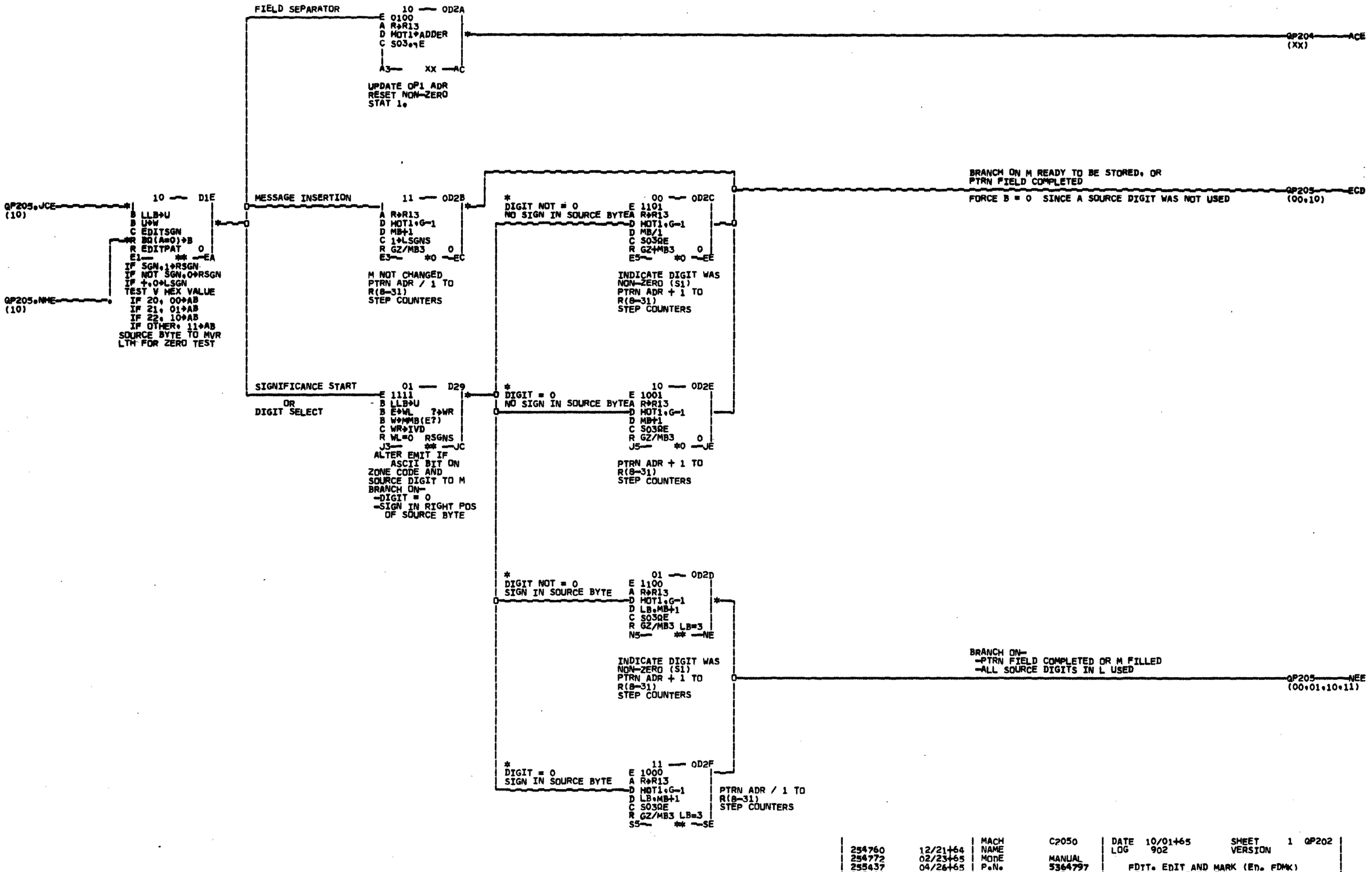
QP100

254760	12/21+64	MACH	C2050	DATE 10/01+65	SHEET 1 QP100
254772	02/23+65	NAME		LOG 902	VERSION
255449	09/20+65	MODE	MANUAL	COMPARE LOGICAL (SS)	
		P.N.	5364795		
		IBM CORP.	SDD		



254760	12-21-64	MACH	C2050	DATE	10-01-65	SHEET	1 QP102
254772	02-23-65	NAME		LOG	902	VERSION	
255449	09-20-65	MODE	MANUAL				
		P.N.	5364904		COMPARE LOGICAL(SS)		
		IBM CORP.	SDD				


1110XX — 638
E 0001
A L→R•AN
B E→WL
B WL→J
C S47•¬E
A1— XX —AA
NO IV ADR TRAP
ADR OF 1ST OP2 WD
TO SAR• SET J=1•
TN OFF STAT 7 FOR
EDIT INSTRUCTION•
QP010•EME
(1110XX)
(1111XX)
FROM I-FETCH
1111XX — 063C
E 0001
A L→R•AN
B E→WL
B WL→J
C S47ΩE
E1— XX —EA
NO IV ADR TRAP
ADR OF 1ST OP2 WD
TO SAR• SET J=1•
TN ON STAT 7 FOR
EDIT AND MARK
INSTRUCTION•
XO — OD88
D D→L
A2— XX —AB
OP2 WD TO L•
XX — ODAE
E 0000
A M→M
B R3→U
B U→W
B W67→LB
L WSI→LSA
L LS→R→LS
C E(13)→WFN
R IVA
E2— X* —EB
OP2 ADR TO H REG•
SET LB TO OP2 ADR
FETCH OP1 ADR TO R•
SET MVR FCTN REG
TO CROSS-GATE•
BR ON INVALID ADR
XX — ODAF
A R→A
L FN•J→LSA
L LS→R→LS
C 0→LSGNS
A3— XX —AC
OP1 ADR TO SAR•
MARK TO R FROM
GEN REG PER J•
TN OFF L SIGN STAT
XX — ODB0
A R→R13
D MDT1→ADDER
L WS2→LSA
L R→LS
A4— XX —AD
INCREMENT MARK
ADR BY 1 AND STORE
IN WS2•
XX C03 ODBE
D D→M
C 0→RSGNS
A5— XX —AE
OP1 WD TO M• TURN
OFF R SIGN STAT•
XX — ODB2
B MMB→V
B V→W
B W→MD•F
L WS1→LSA
L LS→R→LS
C 1→BS*MB
A6— 00 —AF
1ST PTRN CHAR
TO MOVER TO SET
EDIT STATS
SAVE PTRN CHAR
IN MD F TO USE AS
FILL CHAR
X1 — OD89
E 0100
A 0→L
C S47ΩE
E3— XX —EC
ZEROS TO L•
TN ON STAT 5
FOR INVALID ADR
QP205•JCE
(00)
QP205•NHE
(00)
* DEFINITIONS
L SIGN STAT — CORRESPONDS TO S TRIGGER IN DESCRIPTION
OF EDIT INSTRUCTION• TURNED ON TO
INDICATE SIGNIFICANCE HAS BEGUN• TURNED
OFF IF- 1) A PLUS SIGN IS DETECTED• 2) A
FIELD SEPARATOR IS ENCOUNTERED
R SIGN STAT — R ON INDICATES A SIGN HAS BEEN DETECTED
IN RIGHT DIGIT POSITION OF CURRENT
SOURCE BYTE
GP STAT 0 = OP2 DIGIT USED
GP STAT 1 = OP2 NON ZERO
GP STAT 3 = RIGHT+LEFT — ON
EQUALS RIGHT
GP STAT 5 = INVALID ADR•
GP STAT 7 = EDIT AND MARK
00 — D1C
B LLB→U
B U→W
C EDITSGN
R EDITPAT 0
J4— ** —JD
IF SGN•1→RSGN
IF NOT SGN•0→RSGN
IF /•0→LSGN
TEST V HEX VALUE
IF 20• 00→AB
IF 21• 01→AB
IF 22• 10→AB
IF OTHER• 11→AB
SOURCE BYTE TO MVR
LTH FOR ZERO TEST
BR ON EDIT STATS
FIELD SEPARATOR
10 — OD22
E 0100
A R→R13
D MDT1→ADDER
C S03•¬E
E5— XX —EE
UPDATE OP1 ADR•
TN OFF STAT ONE
(NON-ZERO STAT)•
MESSAGE INSERTION
11 — OD23
A R→R13
B MD•F→U
B U→W
B W→MMB
D MDT1•G-1
D MB/1
R GZ+MB3 0
J5— *0 —JE
FILL CHAR TO M
STEP COUNTERS
PTRN ADR / 1 TO
R(8-31)
SIGNIFICANCE START
01 — D21
E 1111
B LLB→U
B E→WL ?→WR
B W→MMB(E7)
C WR→IVD
C 1→LSGNS
R WL=0 RSGNS
N5— ** —NE
ALTER EMIT IF
ASCII BIT ON
ZONE CODE AND
SOURCE DIGIT TO M
TURN ON L SIGN STAT
DIGIT SELECT
ZONE CODE AND
SOURCE DIGIT TO M
00 — D20
E 1111
B LLB→U
B E→WL ?→WR
B W→MMB(E7)
C WR→IVD
R WL=0 RSGNS
S5— ** —SE
BRANCH ON—
—SOURCE DIGIT = 0
—SIGN IN RIGHT
POSITION OF
SOURCE BYTE
* RESET NON-ZERO STAT• THEN STORE M IF NECESSARY
QP204 — EEE
(XX)
SOURCE DIGIT NOT = 0
NO SIGN IN SOURCE BYTE
00 — OD24
E 1101
A R→R13
D MDT1•G-1
D MB+1
L WS2→LSA
L R→LS
C S03ΩE•1→LSGN
R GZ/MB3 0
E7— *0 —EG
PTRN ADR / 1 TO
R(8-31)• THEN ALL
R TO WS2
STEP COUNTERS
INDICATE NON-ZERO
SOURCE DIGIT
BRANCH ON PTRN FIELD COMPLETE OR M FILLED
QP205 — JEE
(00•10)
SOURCE DIGIT = 0
NO SIGN IN SOURCE BYTE
10 — OD26
E 1001
A R→R13
B MD•F→U
B U→W
B W→MMB
D MDT1•G-1
D MB+1
C S03ΩE
R GZ/MB3 0
J7— *0 —JG
FILL CHAR TO M
PTRN ADR / 1 TO
R(8-31)
STEP COUNTERS
BRANCH ON—
—PTRN FIELD COMPLETED
—M REG FILLED
SOURCE DIG NOT = 0
SIGN IN SOURCE BYTE
01 — OD25
E 1100
A R→R13
D MDT1→ADDER
L WS2→LSA
L R→LS
C S03ΩE•1→LSGN
N7— XX —NG
PTRN ADR / 1 TO
R(8-31)• THEN ALL
R TO WS2
INDICATE NON-ZERO
SOURCE DIGIT
XX C03 ODD5
B LLB→U
D G-1
D LB•MB/1
C EDITSGN
R GZ/MB3 LB=3
N8— ** —NH
IF SGN•1→RSGN
IF NOT SGN•0→RSGN
IF /•0→LSGN
QP205 — NHE
(00•01•10•11)
BRANCH ON—
—PTRN FIELD COMPLETED OR
M FILLED
—ALL SOURCE DIGITS
IN L USED
SOURCE DIGIT = 0
SIGN IN SOURCE BYTE
11 — OD27
E 1000
A R→R13
B MD•F→U
B U→W
B W→MMB
D MDT1→ADDER
C S03ΩE
S7— XX —SG
FILL CHAR TO M
PTRN ADR / 1 TO
R(8-31)
282229 11+28/67
MACH C2050
NAME
MODE MANUAL
P•N• 5379160
IBM CORP• SDD
DATE 01+30/68 SHEET 1 QP200
LOG 627 VERSION C03
EDIT•EDIT AND MARK (INIT•)
LEFT DIGIT• S OFF• INV• MOVE
QP200



		MACH	C2050	DATE 10/01+65	SHEET 1 QP202
254760	12/21+64	NAME		LOG 902	VERSION
254772	02/23+65	MODE	MANUAL		
255437	04/26+65	P•N•	5364797	EDIT• EDIT AND MARK (En• EDMK)	
255449	09/20+65	IBM CORP•	SDD	LEFT SOURCE DIGIT• S ON	

FIELD SEPARATOR — 10 — 0D32

E 1001
A R→R13
D MOT1*ADDER
C E→S03

A3— XX —AC

PTRN ADR+1 TO R(0-31).
TN ON STAT0
(DIGIT USED).
RESET STAT1
(NON-ZERO).
TN ON STAT3
(R/L TO RIGHT).

TURN OFF NON-ZERO STAT. THEN STORE M IF NECESSARY

QP204 (XX) —ACD

MESSAGE INSERTION — 11 — 0D33

E 0001
A R→R13
B MD.F→U
B U→W
B W→MMB
D MOT1.G-1
D MB+1
C S03QE
R GZ/MB3 0
E3— *0 —EC

FILL CHAR TO M
PTRN ADR / 1 TO R(0-31)
STEP COUNTERS

BR ON PTRN FIELD COMPLETED OR M FILLED

QP205 (00.10) —ECD

QP205.JCE (01) — 01 — 0D1D

E 0001
B LLB→U
B U→W
C S03.7E

R EDITPAT 0
J1— *= —JA

TEST V HEX VALUE
IF 20. 00→AB
IF 21. 01→AB
IF 22. 10→AB
IF OTHER. 11→AB

SOURCE BYTE TO MVR
FOR ZERO TEST
BR ON EDIT STATS

SIGNIFICANCE START — 01 — 0D31

E 1111
B LLB→U
B E→WL UR→WR
B W→MMB(E?)
C 1→LSGNS

R WR=0
J3— *X —JC

ALTER EMIT IF
ASCII BIT ON
ZONE CODE AND SOURCE
DIGIT TO M
TURN ON L SIGN STAT
BR ON DIGIT = 0

DIGIT NOT = 0 — 0X — 0D8C

E 0100
A R→R13
D MOT1.G-1
D LB.MB/1
L WS2→LSA
L R→LS
C S03QE.1→LSGN
R GZ/MB3 LB=3
J5— *= —JE

TURN ON NON-ZERO STAT
PTRN ADR / 1 TO
R(8-31). THEN R
TO WS2
STEP COUNTERS

DIGIT SELECT — 00 — 0D30

E 1111
B LLB→U
B E→WL UR→WR
B W→MMB(E?)

R WR=0
N3— *X —NC

ALTER EMIT IF
ASCII BIT ON
ZONE CODE AND SOURCE
DIGIT TO M
BR ON DIGIT = 0

* DIGIT = 0 — 1X — 0D8E

A R→R13
B MD.F→U
B U→W
B W→MMB
D MOT1.G-1
D LB.MB/1
R GZ/MB3 LB=3
N5— *= —NE

FILL CHAR TO M
PTRN ADR / 1 TO
R(8-31)
STEP COUNTER

BRANCH ON—
—PTRN FIELD COMPLETED OR M FILLED
—NEXT SOURCE WORD REQUIRED

QP205 (00.01.10.11) —JEE

QP203

254760	12/21+64	MACH	C2050	DATE 10/01+65	SHEET 1 QP203
255449	09/20+65	NAME		LOG 902	VERSION
		MODE	MANUAL	EDIT. EDIT AND MARK (En. EDMk)	
		P.N.	5364798	RIGHT SOURCE DIGIT. S OFF	
		IBM CORP.	SND		

QP200•EEE (XX)

QP202•ACE (XX)

QP203•ACD (XX)

* FIELD SEPARATOR

10 — 0D36
E 1001
A R→R13
D HOT1→ADDER
C E→S03
A3— XX —AC

PTRN ADR+1
TO R(0-31).
TN ON STAT0
(DIGIT USED).
TN OFF STAT1
(NON-ZERO).
TN ON STAT3
(R+L TO RIGHT).

XX — 0D84
B MD•F→U
B U→W
B W→MMB
*D G-1
D MB+1
C 0→L SGNS
R GZ/MB3 0
A5— *0 —AE

FILL CHAR TO M
TN OFF L SIGN STAT

BR ON PTRN FIELD COMPLETED OR M READY TO STORED — QP205 (00•10) —AED

MESSAGE INSERTION

11 — 0D37
E 0001
A R→R13
D HOT1•G-1
D MB+1
C S03QE
R GZ/MB3 0
E3— *0 —EC

NO CHANGE TO PTRN
PTRN ADR / 1 TO R(8-31)
STEP COUNTERS

QP205•JCE (11)

11 — 0D1F
*E 0001
B LLB→U
B U→W
C S03→E
R BQ(A=0)→B
R EDITPAT 0
J1— ** —JA

TEST V HEX VALUE
IF 20• 00→AB
IF 21• 01→AB
IF 22• 10→AB
IF OTHER• 11→AB

SOURCE BYTE TO MVR
FOR ZERO TESTS
BR ON EDIT STATS

SIGNIFICANCE START
OR
DIGIT SELECT

01 — 0D35
E 1111
B LLB→U
B E→WL UR→WR
B W→MMB(E?)
R WR=0
J3— *X —JC

ALTER EMIT IF
ASCII BIT ON
ZONE CODE AND SOURCE
DIGIT TO M
BR ON DIGIT = 0

DIGIT NOT = 0

0X — 0D8D
E 0100
A R→R13
D HOT1•G-1
D LB•MB/1
C S03QE
R GZ/MB3 LB=3
J5— ** —JE

INDICATE DIGIT WAS
NON-ZERO
PTRN ADR / 1 TO R(8-31)
STEP COUNTERS

DIGIT = 0

1X — 0D8F
A R→R13
D HOT1•G-1
D LB•MB/1
R GZ/MB3 LB=3
N5— ** —NE

PTRN ADR / 1 TO R(8-31)
STEP COUNTERS

BRANCH ON—
—PTRN FIELD COMPLETED OR M FILLED
—NEXT SOURCE WORD REQUIRED
— QP205 (00•01•10•11) —JEE

QP204

254760	12+21+64	MACH	C2050	DATE 10+01+65	SHEET 1 QP204
255449	09+20+65	NAME		LOG 902	VERSION
		MODE	MANUAL	EDIT• EDIT AND MARK (ED• EDMK)	
		P•N•	5364799	RIGHT SOURCE DIGIT• S ON	
		IBM CORP•	SDD		

QP200.JEE (00.10)

QP202.ECD (00.10)

QP203.ECD (00.10)

QP204.AED (00.10)

QP200.NHE (00.01.10.11)

QP202.NEE (00.01.10.11)

QP203.JEE (00.01.10.11)

QP204.JEE (00.01.10.11)

STORE REQUIRED

10 — 0D3E
E 1000
A R-1→A
C S47◦qE

E1— XX —EA

PTRN ADR-1 TO SAR
TURN OFF SOURCE WORD
REQUIRED STAT

STORE REQUIRED AND
SOURCE FETCH RQD

11 — 0D3F
E 1000
A R-1→A
C S47QE

J1— XX —JA

PTRN ADR-1 TO SAR
TURN ON SOURCE
REQUIRED STAT

XX — 0DB7
A M→D*BS

R G1<0
E3— *X —EC

COMPLETED BYTES
TO SDR
BR ON PTRN FIELD
COMPLETED

00 — 0D3C
B MMB→V
C 1→BS*MB

R LSGNS S3
J3— ** —JC

PATTERN BYTE
TO MOVER.
BR ON L SIGN
STAT AND R-L STAT

SOURCE FETCH RQD

PTRN FIELD NOT
COMPLETED - FETCH
NEXT PTR WORD

0X — 0D90
E 0000
A R→A
C E→BS

E4— XX —ED

NEXT PTRN WD
ADR TO SAR

XX — 0DB8
D D→M

R 0
E5— 0* S4 —EE

NEXT PTRN WD TO M
BR ON NEXT SOURCE
WORD REQUIRED

01 — 0D3D
L WS1→LSA
L R→LS

N3— XX —NC

TEMPORARILY
STORE ADR.
IN R TO WS1.

XX — 0DB5
E 1000
A H+4→R.AN

R AQ(B=0)→A
R SO S5
N4— ** —ND

NO IV ADR TRAP
ADR OF NEXT
OP2 WD TO SAR
BR ON DIGIT
USED STAT AND
INVALID ADR STAT

11 — 0D3B
E 0101 E→WR
B W→MD.F

Q5— 0X —SE

GO TO INVALID
ADR INTERRUPT

01 — 0D39
A H/4→H
L WS1→LSA
L LS→R→LS

R IVA
L6— X* —NF

UPDATE OP2 ADR
IN H.
FETCH OP1 ADR
TO R.
BR ON INVALID
OP2 ADR.

10 — 0D3A
E 1000
A H+4→H
L WS1→LSA
L LS→R→LS
C $03.◦qE

R IVA
Q6— X* —SF

LIKE NF
ALSO TURN OFF
DIGIT USED STAT 0.

X0 — 0D8A
D D→L

L7— XX —NG

NEXT SOURCE WD TO L

X1 — 0D8B
E 0100
A 0→L

C S47QE

Q7— XX —SG

TN ON INVALID
OP2 ADR STAT 5.
ZEROS TO L.

PROCESS NEXT PTRN
CHAR

XX — 0DB6
B MMB→V
C 1→BS*MB

R LSGNS 0
N8— *0 —NH

PTRN CHAR TO MVR
TO SET EDIT STATS
BR ON L SIGN
FORCE B=0 AS NEXT
SOURCE DIGIT MUST BE IN
THE LEFT DIGIT POS OF
THE NEXT BYTE

QP206 —ECE
(1X)
TO END ROUTINE

NEXT SOURCE DIGIT ON LEFT SIDE OF BYTE. L SIGN OFF — QP200 —JCE (00)

NEXT DIGIT ON RIGHT L SIGN OFF — QP202 —JCE (10)

NEXT DIGIT ON LEFT L SIGN ON — QP203 —JCE (01)

NEXT DIGIT ON RGHT L SIGN ON — QP204 —JCE (11)

QP200 —NHE (00)

QP202 —NHE (10)

QT310 —SEF (0X)
TO INTERRUPT
ROUTINE

QP205

254760	12-21/64	MACH	C2050	DATE 10-01/65	SHEET 1 QP205
254772	02-23/65	NAME		LOG 902	VERSION
255449	09-20/65	MODE	MANUAL	EDIT. EDIT AND MARK (ED. EDMK)	
		P.N.	5364800	STORE.FETCH PTRN.SOURCE	
		IBM CORP.	SDD		

QP205.ECE (1X)

1X — OD92
L WS2→LSA
L LS→R→LS
R AQ(B=0)→A
R SO S5
A1— ** —AA

FETCH MARK ADR TO R FROM WS2. BR ON DIGIT USED STAT 0 AND INVALID ADR STAT 5.

SOURCE WAS GTR THAN ZERO

01 — OD45
E 0010
C E(23)→CR
R S7
E2— X* —EB

2 TO COND REG
BR ON EDIT AND MARK

EDIT

X0 — OD94
E 1110
L WS.E→LSA
L LS→L→LS
S IA/0+2→A
C E→S47
R I-FETCH
E4— ** —ED
EXCPNQIA(30)→A
EXCPNQ(IA(30).REF)→B
DO NOT CHANGE GR1
PREPARE FOR SPECIAL ENTRY TO I-FETCH

QT110 — EDD (00.01.10.11) TO 2ND CYCLE OF I-FETCH

SOURCE WAS ZERO

01 — OD41
L FN.J→LSA
L LS→L→LS
R AQ(B=0)→A
R LSGNS S1
J1— ** —JA

GEN REG PER J TO L REG. BR ON L SIGN STAT AND NON-ZERO STAT.

10 — OD46
E 0000
C E(23)→CR
R S7
J2— X* —JB

0 TO COND REG
BR ON EDIT AND MARK

EDIT AND MARK

X1 — OD95
A R-1→L13
L FN.J→LSA
L L→LS
J4— 1X —JD

REDUCE R(8-31) BY 1 AND STORE RESULT TOGETHER WITH ORIGINAL GR1(0-7) IN GR1

QT100 — JDD (1X) TO I-FETCH

SOURCE WAS LESS THAN ZERO

10 — OD42
L FN.J→LSA
L LS→L→LS
R AQ(B=0)→A
R LSGNS S1
N1— ** —NA

LIKE ABOVE.

11 — OD47
E 0001
C E(23)→CR
R S7
N2— X* —NB

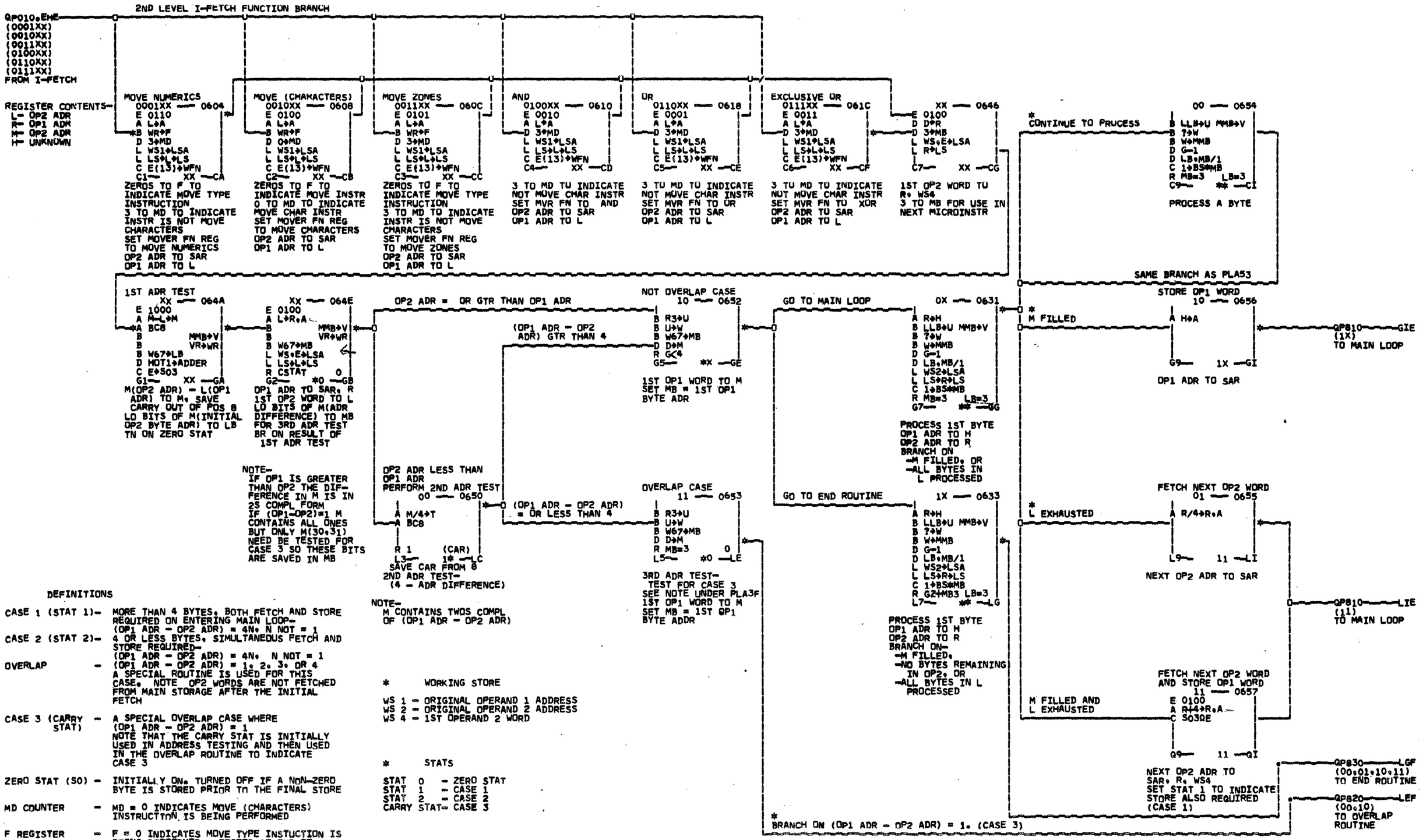
1 TO COND REG
BR ON EDIT AND MARK

GO TO INVALID ADR INTERRUPT

11 — OD43
E 0101
B E→WR
B W→MD.F
S1— 0X —SA

QT310 — SAD (0X) TO INTERRUPT ROUTINE

254760	12+21/64	MACH	C2050	DATE 10+01/65	SHEET 1 QP206
254772	02+23/65	NAME		LOG 902	VERSION
255449	09+20/65	MODE	MANUAL	EDIT. EDIT AND MARK (ED. EDMK)	
		P.N.	5364801	SET COND REG. GEN REG 1	
		IBM CORP.	SDD		



DEFINITIONS

CASE 1 (STAT 1)- MORE THAN 4 BYTES. BOTH FETCH AND STORE REQUIRED ON ENTERING MAIN LOOP- (OP1 ADR - OP2 ADR) = 4N. N NOT = 1

CASE 2 (STAT 2)- 4 OR LESS BYTES. SIMULTANEOUS FETCH AND STORE REQUIRED- (OP1 ADR - OP2 ADR) = 4N. N NOT = 1

OVERLAP - (OP1 ADR - OP2 ADR) = 1, 2, 3, OR 4
A SPECIAL ROUTINE IS USED FOR THIS CASE. NOTE OP2 WORDS ARE NOT FETCHED FROM MAIN STORAGE AFTER THE INITIAL FETCH

CASE 3 (CARRY STAT) - A SPECIAL OVERLAP CASE WHERE (OP1 ADR - OP2 ADR) = 1
NOTE THAT THE CARRY STAT IS INITIALLY USED IN ADDRESS TESTING AND THEN USED IN THE OVERLAP ROUTINE TO INDICATE CASE 3

ZERO STAT (S0) - INITIALLY ON. TURNED OFF IF A NON-ZERO BYTE IS STORED PRIOR TO THE FINAL STORE

MD COUNTER - MD = 0 INDICATES MOVE (CHARACTERS) INSTRUCTION IS BEING PERFORMED

F REGISTER - F = 0 INDICATES MOVE TYPE INSTUCTION IS BEING PERFORMED. TESTED AT END OF PROGRAM TO DETERMINE WHETHER CONDITION CODE SHOULD BE SET

* WORKING STORE

WS 1 = ORIGINAL OPERAND 1 ADDRESS
WS 2 = ORIGINAL OPERAND 2 ADDRESS
WS 4 = 1ST OPERAND 2 WORD

* STATS

STAT 0 = ZERO STAT
STAT 1 = CASE 1
STAT 2 = CASE 2
CARRY STAT= CASE 3

QP800

254760	12/21/64	MACH	C2050	DATE 10/01/65	SHEET 1 QP800
255449	09/20/65	NAME		LOG 902	VERSION
		MODE	MANUAL	LOGICAL AND MOVE INSTR (SS)	
		P.N.	5364802	ADR TESTS. INITIALIZATION	
		IBM CORP.	SDD		

REGISTERS—
L= OP2
R= OP2 ADR
M= OP1
H= OP1 ADR

NOT MOVE INSTR
FETCH NEXT OP1 WORD
10 — 060E
A M+4→A
R C2— TZ*BS
X* —CB
NEXT OP1 ADR TO SAR
TEST FOR NON-ZERO
BYTE STORED IN
OP1 FIELD

X0 — 063A
E 1000
D D→M
C S03→E
R S2 1
A4— *1 —AD
NEXT OP1 WD TO M
TN ON S0 TO INDICATE
NON-ZERO BYTE STORED
BR ON OP2 FETCH
RQD (CASE 2)

GO TO END ROUTINE
11 — 0683
R G1<0 LB=0
E3— ** —EC

X1 — 063B
D D→M
R S2 1
E4— *1 —ED
NEXT OP1 WD TO M
BR ON OP2 FETCH
RQD (CASE 2)

BRANCH ON
—ALL BYTES PROCESSED
—LAST BYTE IN L PROCESSED
QP830——ECE
(00•01•10•11)
TO END ROUTINE

BRANCH ON MOVE (CHAR) INSTR

NOTE — OP1 FETCH NOT REQUIRED FOR MOVE INSTR

QP830•QDE
(01•10•11)
FROM END
ROUTINE

BRANCH ON
A—OP2 FETCH REQD
OR NOT MOVE
B—MOVE (CHAR)

01 — 060D
E 1000
A H+4→H
B LLB→U MMB→V
B ?→W
B W→MMB
D G←1
D LB•MB+1
C E→BS
R BQ(A=1)→B
R G<4 LB=3
J2— ** —JB
PROCESS A BYTE
OP1 ADR / 4 TO H
BRANCH ON
—LAST BYTE OF L•
—4 OR LESS BYTES
REMAINING IN
OP2 FIELD

00 — 0680
A R+4→R•A
B LLB→U MMB→V
B ?→W
B W→MMB
D G←1
D LB•MB/1
C 1→BS*MB
R AQ(B=1)→A
R 0 LB=3
J3— ** —JC
PROCESS A BYTE
NEXT OP2 ADR
TO SAR• R
BR ON LAST
BYTE OF L

00 — 0684
B LLB→U MMB→V
B ?→W
B W→MMB
D G←1
D LB•MB/1
C 1→BS*MB
R 1 LB=3
J4— 1* —JD
PROCESS A BYTE
BR ON LAST
BYTE OF L

10 — 0686
B LLB→U MMB→V
B ?→W
B W→MMB
D D→L
D G←1
D LB•MB/1
C 1→BS*MB
J5— 11 —JE
PROCESS A BYTE
NEXT OP2 WORD
TO L

00 — 0688
B LLB→U MMB→V
B ?→W
B W→MMB
D G←1
D LB•MB/1
C 1→BS*MB
R MB=3 1
J6— *1 —JF
PROCESS A BYTE
BR ON M FILLED

01 — 0689
A H→A
B LLB→U MMB→V
B ?→W
B W→MMB
D G←1
D LB•MB/1
C 1→BS*MB
R MB=3
J7— *X —JG
PROCESS A BYTE
OP1 ADR TO SAR
BR ON M FILLED

0X — 0635
B LLB→U MMB→V
B ?→W
B W→MMB
D G←1
D LB•MB+1
C 1→BS*MB
J8— 1X —JH
PROCESS A BYTE

FETCH NEXT OP2 WD
01 — 0681
A R+4→R•A
N3— 11 —NC
OP2 ADR / 4 TO SAR• R

11 — 0687
D D→L
R BQ(A=1)→B
R S1 0
N5— ** —NE
NEXT OP2 WORD TO L
BRANCH ON CASE 1
(CASE 1 BR TAKEN
ON FIRST PASS IF
STORE REQD)

STORE OP1 BYTES
11 — 068B
A H→A
N7— 1X —NG
OP1 ADR TO SAR

1X — 0637
A M→D*BS
R AQ(B=0)→A
R 0 MD=0
N9— ** —NI
STORE COMPLETED
OP1 BYTES

CASE 2
FETCH NEXT OP2 WD
11 — 060F
E 0000
A H+4→H
C E→BS
Q2— 01 —QB
OP1 ADR / 4 TO H

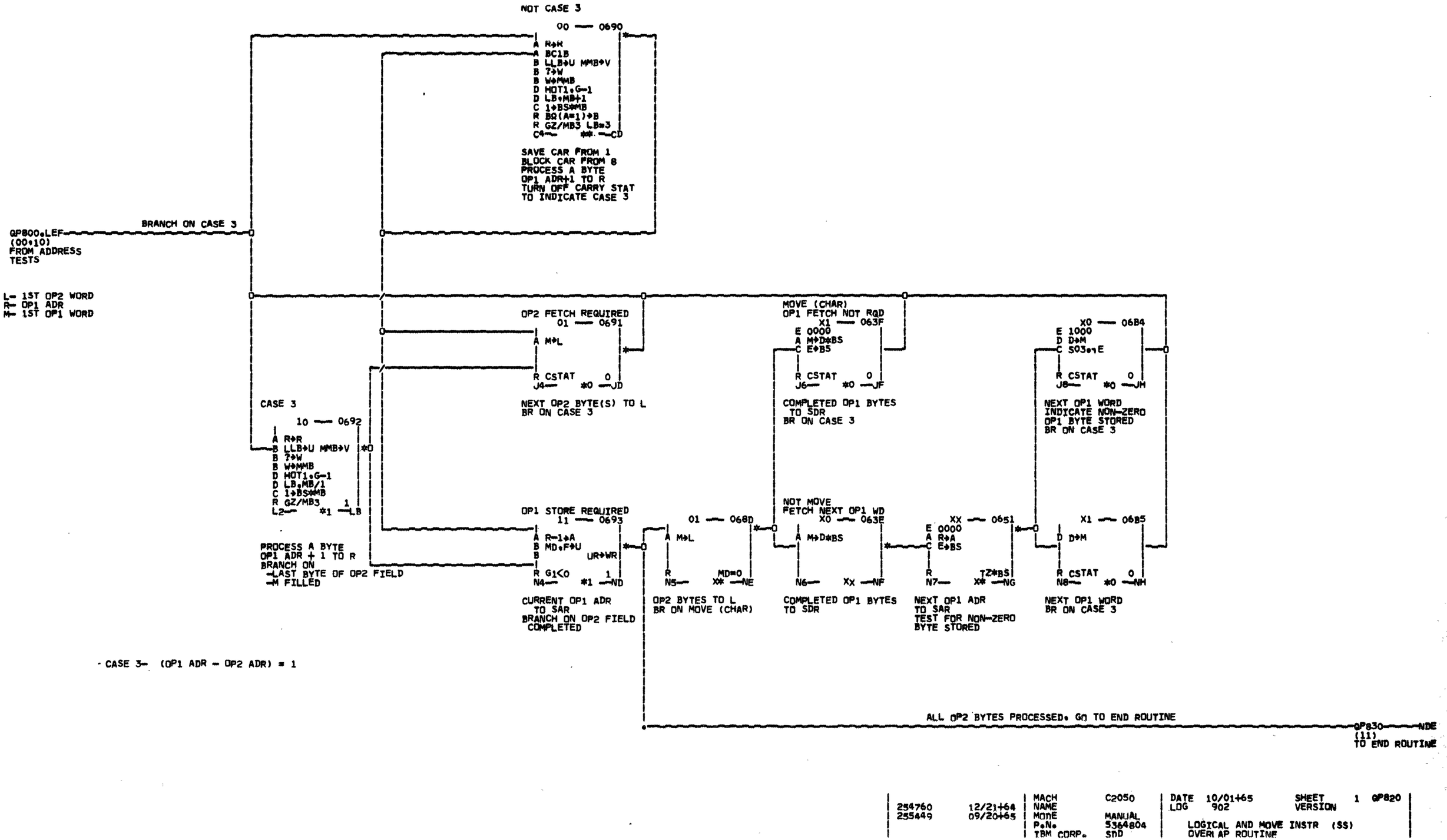
QP830——QBE
(01)
TO END ROUTINE

QP800•LIE
(11)

QP800•GIE
(1X)

QP810

254760	12+21+64	MACH	C2050	DATE 10+01+65	SHEET 1 QP810
255449	09+20+65	NAME		LOG 902	VERSION
		MODE	MANUAL	LOGICAL AND MOVE INSTR (SS)	
		P.N.	5364803	MAIN LOOP	
		IBM CORP.	SDD		


NOT CASE 3
00 — 0690
A R→R
A BC1B
B LLB→U MMB→V
B ?→W
B W→MMB
D HOT1◦G=1
D LB◦MB+1
C 1→BS*MB
R BΩ(A=1)→B
R GZ/MB3 LB=3
C4 — ** — CD
SAVE CAR FROM 1
BLOCK CAR FROM B
PROCESS A BYTE
OP1 ADR+1 TO R
TURN OFF CARRY STAT
TO INDICATE CASE 3
BRANCH ON CASE 3
QP800◦LEF
(00◦10)
FROM ADDRESS
TESTS
L= 1ST OP2 WORD
R= OP1 ADR
M= 1ST OP1 WORD
OP2 FETCH REQUIRED
01 — 0691
A M→L
R CSTAT 0
J4 — *0 — JD
NEXT OP2 BYTE(S) TO L
BR ON CASE 3
MOVE (CHAR)
OP1 FETCH NOT RQD
X1 — 063F
E 0000
A M→D*BS
C E→B5
R CSTAT 0
J6 — *0 — JF
COMPLETED OP1 BYTES
TO SDR
BR ON CASE 3
X0 — 06B4
E 1000
D D→M
C S03◦1E
R CSTAT 0
J8 — *0 — JH
NEXT OP1 WORD
INDICATE NON-ZERO
OP1 BYTE STORED
BR ON CASE 3
CASE 3
10 — 0692
A R→R
B LLB→U MMB→V *0
B ?→W
B W→MMB
D HOT1◦G=1
D LB◦MB/1
C 1→BS*MB
R GZ/MB3 1
L2 — *1 — LB
PROCESS A BYTE
OP1 ADR + 1 TO R
BRANCH ON
-LAST BYTE OF OP2 FIELD
-M FILLED
OP1 STORE REQUIRED
11 — 0693
A R-1→A
B MD◦F→U
B UR→WR
R G1<0 1
N4 — *1 — ND
CURRENT OP1 ADR
TO SAR
BRANCH ON OP2 FIELD
COMPLETED
01 — 068D
A M→L
R MD=0
N5 — X* — NE
OP2 BYTES TO L
BR ON MOVE (CHAR)
NOT MOVE
FETCH NEXT OP1 WD
X0 — 063E
A M→D*BS
N6 — XX — NF
COMPLETED OP1 BYTES
TO SDR
XX — 0651
E 0000
A R→A
C E→BS
R TZ*BS
N7 — X* — NG
NEXT OP1 ADR
TO SAR
TEST FOR NON-ZERO
BYTE STORED
X1 — 06B5
D D→M
R CSTAT 0
N8 — *0 — NH
NEXT OP1 WORD
BR ON CASE 3
ALL OP2 BYTES PROCESSED◦ GO TO END ROUTINE
QP830 — NDE
(11)
TO END ROUTINE
- CASE 3- (OP1 ADR - OP2 ADR) = 1

REGISTERS-
L- OP2
R- OP2 ADR
M- OP1
H- OP1 ADR

00 — 0694
B LLB→U MMB→V
B T→W
B W*MMB
D G-1
D LB•MB/1
C 1→BS*MB
R GZ/MB3 LB=3
C2— ** —CB

PROCESS A BYTE
BRANCH ON-
-LAST BYTE OR
M FILLED
-LAST BYTE OF L

QP800•LGF
(00•01•10•11)
FROM LENGTH
TEST

FETCH NEXT OP2 WORD
01 — 0695
A R/4→R•A

G2— XX —GB

OP2 ADR + 4 TO SAR• R

QP810•QBE
(01)
FROM MAIN LOOP
CASE 2

XX — 0682
D D→L

G4— 00 —GD

NEXT OP2 WORD

QP810•ECE
(00•01•10•11)
FROM MAIN LOOP

QP820•NDE
(11)
FROM OVERLAP
ROUTINE

STORE COMPLETED BYTES
10 — 0696
A H→A
B MD•F→U
B UR→WR
R BQ(A=1)→B
R G1<0 0
L2— ** —LB

OP1 ADR TO SAR
F TO MVR LTH TO
TEST FOR LOG INSTR
BRANCH OUT IF ALL
BYTES HAVE BEEN
PROCESSED

CASE 2
STORE COMPLETED BYTES
11 — 0697
E 0010
A H→A
B MD•F→U
B UR→WR
C S030E
R BQ(A=1)→B
R G1<0 0
Q2— ** —QB

OP1 ADR TO SAR
F TO MVR LTH TO
TEST FOR LOG INSTR
TN ON S2 FOR CASE 2
BRANCH OUT IF ALL
BYTES PROCESSED

FIELD COMPLETED
11 — 068F
A M→D*BS

R AQ(B=1)→A
R S0 WR=0
L4— ** —LD

STORE COMPLETED BYTES
BRANCH ON
-NO ZERO BYTES STORED
PRIOR TO THIS STORE
-LOGICAL INSTR

FIELD NOT COMPLETED
00 — 068C
A M→D*BS

R AQ(B=0)→A
R S2 MD=0
Q4— ** —QD

COMPLETED OP1 BYTES
TO SDR

NOT LOGICAL INSTR
DONT CHANGE COND CODE
11 — 069B
C NOP

C6— 1X —CF

PREVIOUS OP1 BYTES
WERE ALL ZEROS
TEST BYTES NOW BEING STORED
10 — 069A

R 0 TZ*BS
L6— 0* —LF

AT LEAST ONE NON-
ZERO BYTE STORED
IN OP1 FIELD
00 — 0698
E 0001
C E(23)→CR

G8— 1X —GH

01 TO PSW(34•35)

ALL ZEROS STORED
IN OP1 FIELD
01 — 0699
E 0000
C E(23)→CR

L8— 1X —LH

00 TO PSW(34•35)

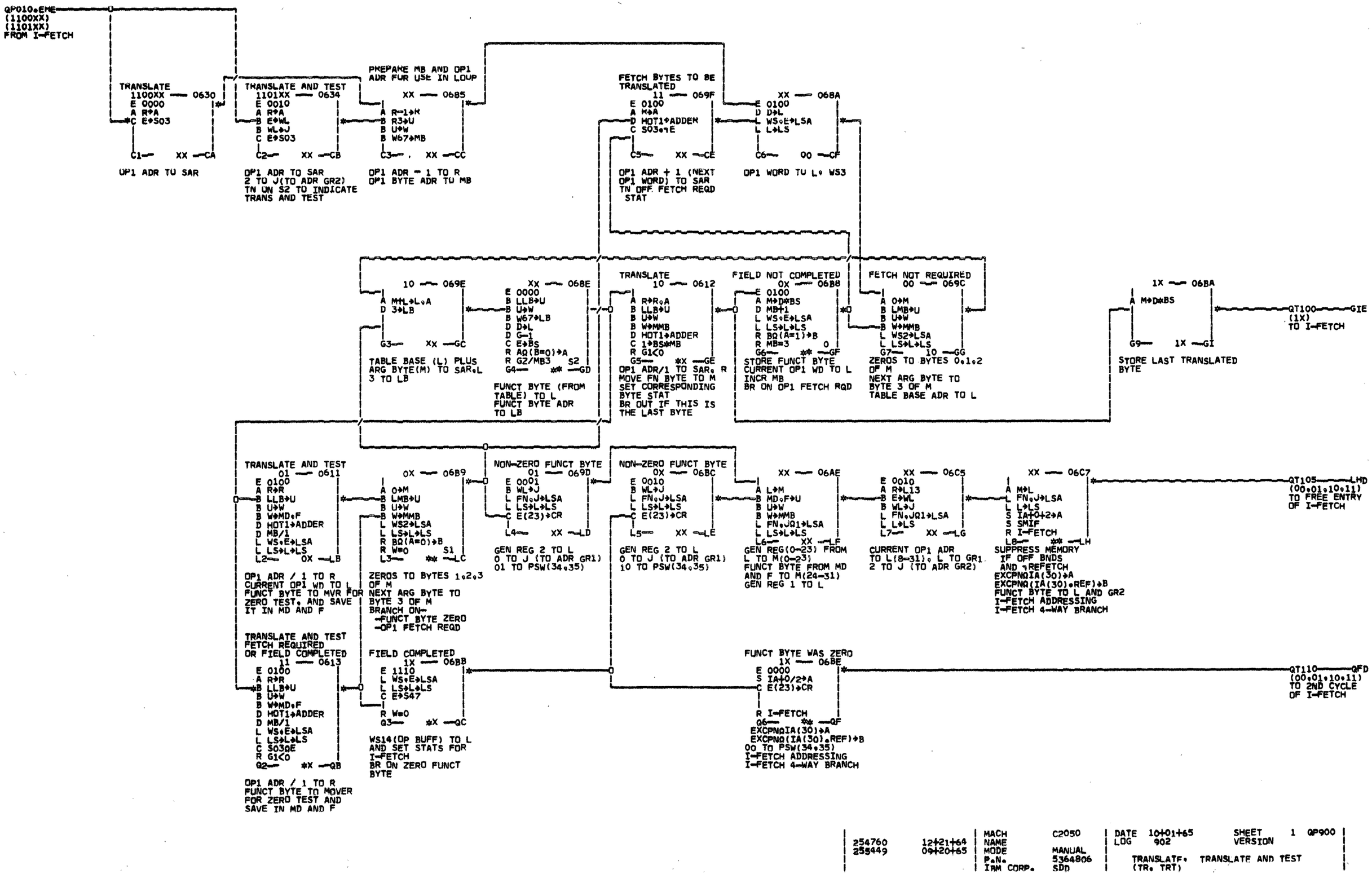
QT100 — LME
(1X)
TO I-FETCH

BRANCH ON-
-CASE 2
-MOVE INSTR

QP810 — QDE
(01•10•11)

QP830

254760	12/21+64	MACH	C2050	DATE 10/01+65	SHEET 1 QP830
255449	09/20+65	NAME		LOG 902	VERSION
		MODE	MANUAL	LOGICAL AND MOVE INSTR (SS)	
		P.N.	5364805	END ROUTINE	
		IBM CORP.	SDD		



254760	12+21+64	MACH	C2050	DATE 10+01+65	SHEET 1 QP900
255449	09+20+65	NAME		LOG 902	VERSION
		MODE	MANUAL	TRANSLATE, TRANSLATE AND TEST	
		P◦N◦	5364806	(TR, TRT)	
		IBM CORP◦	SDD		

QP900

QT115•CMD
(111100)
(111101)
1ST LEVEL
I-FETCH

111100 — 013C
E 1001
A M23+L→L
*B MLB→V *
B V→W
B W→MMB
D 0→LB
L WS•E→LSA
L L→LS
S IA/0+2→A
R AQ(B=1)→A
R IA(30) S7
A1— ** —AA

ADD BASE ADR TO
DISPLACEMENT (L/M
RT HALF TO L• WS9
LENGTH FIELD L1L2
TO M BYTE 3•
SET LB=0•
IC TO SAR•

QE400•CHE
(111100)
(111101)
EXECUTE

111101 — 013D
E 1001
A M23→L
B MLB→V
B V→W
B W→MMB
D 0→LB
L WS•E→LSA
L L→LS
S IA/0+2→A
R AQ(B=1)→A
R IA(30) S7
J1— ** —EA

BASE ADR=0•
XFER DISPLACEMENT
FROM M RT HAFT TO
L• L TO WS9
LENGTH FIELD L1L2
TO M BYTE 3•
SET LB=0•
IC TO SAR•

00 — 0C40
E 1011
A M→R
B MLB→V *
B VR→WR
B WR→F
D MB-1
L WS•E→LSA
L R→LS
C IA/2
R IVA
G2— X* —GB

LENGTH FIELD FROM
M TO R• R TO WS11
INCR IAR BY 2•
DECR MB TO 1•
OP BITS 4-7 FROM
M BYTE 0 TO F•
BR ON INVALID ADR

ON BOUNDS•

X1 — 0C0B
E 1110
A M→R
*B MLB→V *
B VR→WR
B WR→F
D MB-1
L WS•E→LSA
L LS→L→LS
L3— XX —LC

M CONTAINING LENGTH
FIELD IN BYTE 3 TO
R REG• FETCH OP
BFR TO L• OP BITS
4-7 FROM M BYTE 0
TO F• DECR MB TO 2•

OFF BOUNDS

11 — 0C43
R S3
Q2— X* —QB

BR ON 1ST SYLLABLE
ON BOUNDS•

XO — 0C0A
A R+4→A
Q3— XX —QC

ADR OF NEXT
INSTRUCTION WD TO
SAR•

10 — 0C42
E 1110
A M→R
B MLB→V *
B VR→WR
B WR→F
D MB-1
L WS•E→LSA
L LS→L→LS
C IA/2
C4— XX —CD

M CONTAINING LENGTH
FIELD IN BYTE 3 TO
R REG• FETCH OP
BFR TO L• OP BITS
4-7 FROM M BYTE 0
TO F• DECR MB TO 2
INCR IAR BY 2•

XX — 0C19
E 1011
A L→L
B R3→U
B U→W
B W→G
L WS•E→LSA
L R→LS
C BO•1SYL
C5— XX —CE

L23→XG01
T→L T(0-3)→MD
T(0-15)→M
(B=0)→S1
SET 1SYL

B2D2 TO M(16-31)
B2 TO MD
L1L2 TO G1G2
SAVE L1L2 IN WS11

XX — 0C0D
E 1011
A M→R
B MLB→V *
B VR→WR
B WR→F
D MB-1
L WS•E→LSA
L R→LS
N4— XO —ND

LENGTH (BYTE 3) FROM M TO R TO WS11
OP BITS 4-7 FROM M (BYTE 0) TO F
MB DOWN TO 2

XO — 0C0E
E 1110
B R3→U
B U→W
B W→G
D D→MHL
L WS•E→LSA
L L→LS
C BO•1SYL
G5— XX —GE

T→L T(0-3)→MD
T(0-15)→M
(B=0)→S1
SET 1SYL
NEXT INST WD TO L•
L TO OP BFR (WS14)
LF SYLLABLE (B2D2)
TO M RIGHT HALF•
B2 TO MD• SET G1G2
TO L1L2 IN R BYTE 3

X1 — 0C0F
E 0101
B E→WR
B W→MD•F

L5— 0X —LE

INVALID ADDRESS
GO TO INTERRUPT•

XX — 0C1D
E 0000
B MMB→V
B UL→WL VR→WR *
B W→MMB
L FN•MD→LSA
L LS→L→LS
C E(13)→WFN
R S1
A7— X* —AG

SET MVR FCTN REG
TO 000 (CROSS-GATE)
CLEAR B2 OUT OF
M BYTE 2• FETCH
CONTENTS OF B2 GR
TO L• BR ON B2=0•

X0 — 0C12
E 1010
A M23/L→L
B R3→U *
B U→W
B WL→MD
D 3→MB
L WS•E→LSA
L L→LS
C O→RSGNS
A8— XX —AH

ADD BASE ADR TO
DISPLACEMENT (M RT
HALF /L TO L)•
L TO WS10•
SET MD TO L1 IN
R BYTE 3• SET LB=3
TN OFF R SIGN STAT

BASE ADR=0•

X1 — 0C13
E 1010
A M23→L
B R3→U
B U→W
B WL→MD
D 3→MB
L WS•E→LSA
L L→LS
C O→RSGNS
E8— XX —EH

XFER DISPLACEMENT
IN M RT HALF TO L
L TO WS10•
SET MD TO L1 IN
R BYTE 3• SET LB= 3
TN OFF R SIGN STAT

XX — 0C22
E 0000
A 0→M
B R3→U *
B VL→WL UR→WR
B W→MMB
C E→BS
E9— XX —EI

XFER L2 IN BYTE 3
OF R TO RT HALF OF
BYTE 3 IN M AND
ZERO HIGH ORDER M•
RESET BYTE STATS•

XX — 0C25
E 1001
A M/L→R
B R3→U *
B VL→WL ?→WR
B W→MMB
L WS•E→LSA
L LS→L→LS
J7— XX —JG

ADD L2 TO LEFTMOST
BYTE ADR OF OP2
(L+M TO R)• FETCH
LEFTMOST BYTE ADR
OF OP1 TO L (WS9)•
XFER L1 IN BYTE 3
OF R TO RT HALF OF
BYTE 3 IN M•

XX — 0C11
E 0000
A M/L→L•M
B R3→U *
B U→W
B W67→LB
L WS2→LSA
L R→LS
C E→S03
J8— XX —JH

ADD L1 TO LEFTMOST
BYTE ADR OF OP1
(L+M TO L• M)•
STORE R TO WS2
(RIGHTMOST BYTE
ADR OF OP2)•
SET LB TO OP2 RT
BYTE ADR IN R
BYTE 3•

XX — 0C09
A R→A
N8— XX —NH

OP2 RIGHTMOST BYTE
ADR TO SAR•

XX — 0C16
E 0000
A M→H
B MMB→V *
B VR→WR
B W67→MB
L WS1→LSA
L L→LS
C E→S47
R F→ROAR
N9— **XX —NI

SET MB TO RIGHTMOST
BYTE ADR OF OP19
IN M BYTE 3• XFER
M TO H• XFER OP1
RIGHTMOST BYTE ADR
FROM H TO L•

QS110 — NIE
(1010XX) AP
(1011XX) SP

QS200 — NIE
(1001XX) CP

QS400 — NIE
(1100XX) MP
(1101XX) DP

QS500 — NIE
(0010XX) PACK

QS502 — NIE
(0001XX) MVO

QS600 — NIE
(0011XX) UNPK

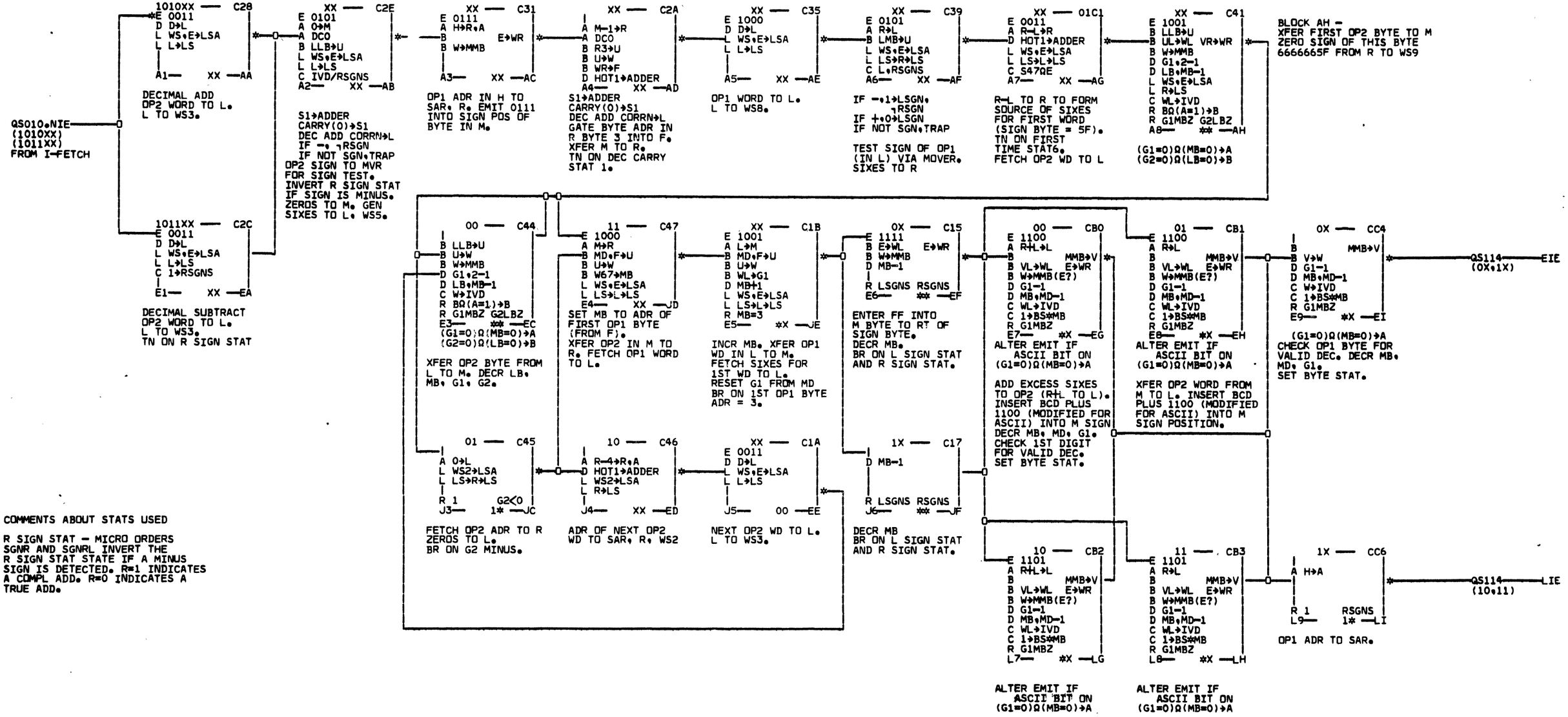
QS705 — NIE
(1000XX) ZAP

QM112 — NIE
(0000XX)
(0100XX)
(0101XX)
(0110XX)
(0111XX)
(1110XX)
(1111XX)
INVALID OPS

QT310 — LEE
(0X)
TO INTERRUPT
ROUTINE

254760	12+21/64	MACH	C2050	DATE	10+01/65	SHEET	1 QS010
254770	02+01/65	NAME		LOG	902	VERSION	
255449	09+20/65	MODE	MANUAL				
		P.N.	5364807		DECIMAL ARITHMETIC		
		IBM CORP.	SDn		SECOND LEVEL I FETCH		

QS10



COMMENTS ABOUT STATS USED

R SIGN STAT — MICRO ORDERS SGNR AND SGNRL INVERT THE R SIGN STAT STATE IF A MINUS SIGN IS DETECTED. R=1 INDICATES A COMPL ADD. R=0 INDICATES A TRUE ADD.

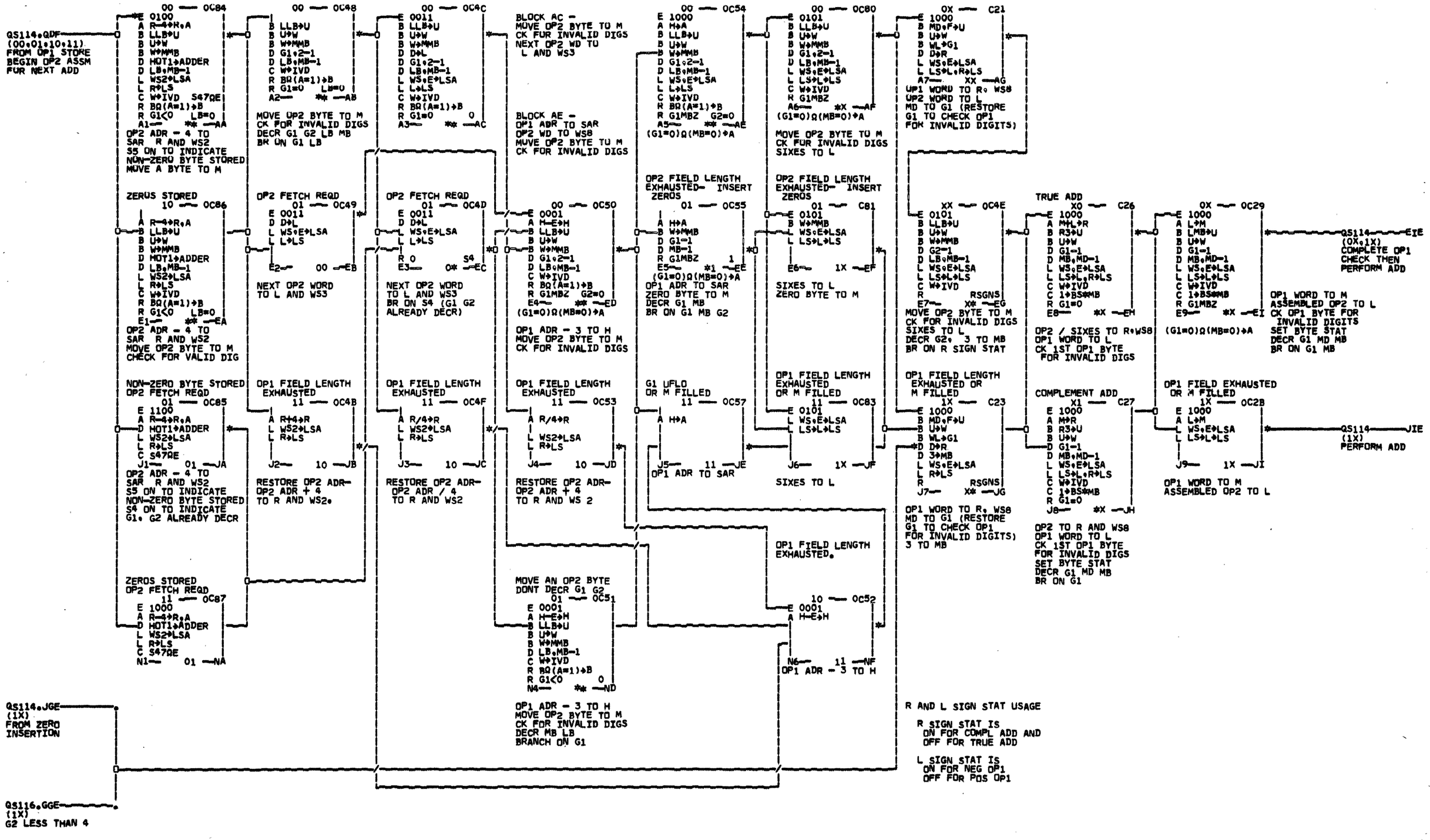
STAT 0 = OVERFLOW OF OP1 FIELD.
STAT 1 = DECIMAL CARRY.
STAT 2 = RECOMPLEMENT.
STAT 4 = G1, G2 DECREMENTED.
STAT 5 = NON-ZERO RESULTS.
STAT 6 = FIRST WORD.

COMMENTS ABOUT WORKING STORE LOCATIONS USED.

WS3 = OP2 WORD
WS5 = 66666666
WS8 = OP1 WORD OR ASSEMBLED OP2 WORD
WS9 = SOURCE OF SIXES WITH 5F IN SIGN BYTE.
WS11 = OP FIELD L1L2 IN BYTES 1 AND 3

QS110

258068	10/20/67	MACH	2050FP	DATE 12/29/67	SHEET 1 QS110
258430	12/21/67	NAME		LOG 662	VERSION
		MODE	MANUAL	DECIMAL ADD, AP, SUBTRACT, AP	
		P.N.	5469499	FIRST WORD ASSEMBLY.	
		IBM CORP.	SDD		



QS112

254765	01/13/65	MACH	C2050	DATE 03/28/66	SHEET 1 QS112
254772	02/23/65	NAME		LOG 695	VERSION
255449	09/20/65	MODE	MANUAL	DEC ADD. SUBTRACT. OP2 ASSEMB	
256498	03/25/66	P.N.	5364809	BEGIN OP1 VALIDITY CHECK.	
		IBM CORP.	SND		

QS112•EIE (0X•1X)

0X — C2D
A H→A
B MMB→V
B V→W
D G1−1
D MB•MD−1
C W→IVD
C 1→BS✻MB
R G1MBZ RSGNS
A1— ✻✻ —AA

QS110•EIE (0X•1X)

(G1=0)Ω(MB=0)→A

OP1 ADR TO SAR
OP1 BYTE TO MOVER
FOR INVALID DIG CK

OP 1 FIELD EXHAUSTED OR M FILLED•

1X — C2F
A H→A

QS112•JIE (1X)

R 1 RSGNS
E1— 1✻ —EA

OP1 ADR TO SAR
BR ON R SIGN STAT

00 — C88
A M+L→M
A DCO
B MMB→V
B V→W
D G1−1
D MB•MD−1
C W→IVD
C 1→BS✻MB
R AQ(B=1)→A
R G1=0 S6
A2— ✻✻ —AB

S1→ADDER
CARRY(0)→S1
DEC ADD CORRN→L

OP2(L) + OP1(M) TO M AND GEN DEC CORRECTION IN L•
OP1 BYTE TO MOVER FOR DEC DIGIT CK•
ORIG OP1 ADR TO R
DECR G1 MD MB•
BR ON G1=0 AND FIRST WD STAT 6•

COMPLEMENT ADD

01 — C89
A M−L→M
A DCO
B MMB→V
B V→W
D G1−1
D MB•MD−1
C W→IVD
C 1→BS✻MB
R AQ(B=1)→A
R G1=0 S6
G2— ✻✻ —GB

S1→ADDER
CARRY(0)→S1
DEC ADD CORRN→L

OP1(M) − OP2(L)
ORIG OP1 ADR TO R
BR ON G1=0 AND R FIRST WD STAT 6•

11 — C8F
E 0010
A M−L→D✻BS
B MMB→V
B VL→WL UR→WR
B W→MMB
D HOT1→ADDER
C S47•┐E
R G1<0
A3— ✻X —AC

FIRST RESULT WORD

CORRECT RESULTS (M−L TO SDR)•
TN OFF 1ST TIME STAT 6•
OP2 WD (WS3) TO L

10 — C8E
E 1011
A M−L→D✻BS
B MD•F→U
B U→W
B W67→MB
D HOT1→ADDER
L WS•E→LSA
L LS→R→LS
G3— XX —GC

CORRECT RESULTS (M−L) AND STORE•
SET MB TO ADR OF FIRST OP1 BYTE
FETCH LENGTH FIELD L1L2 TO R•

COMPLEMENT ADD

11 — C8B
A M−L→M
A DCBS
B MD•F→U
B U→W
B W67→MB
R AQ(B=1)→A
R G1<0 S6
L3— ✻✻ —LC

S1→ADDER
CARRY✻BS→S1
DEC ADD CORRN→L

OP1(M)−OP2(L) TO M
DECIMAL CORR TO L
ORIG OP1 ADR TO R
BR ON G1 NEGATIVE AND FIRST WORD STAT 6•

QS110•LIE (10•11)

TRUE ADD

10 — C8A
A M+L→M
A DCBS
B MD•F→U
B U→W
B W67→MB
R AQ(B=1)→A
R G1<0 S6
Q1— ✻✻ —LA

S1→ADDER
CARRY✻BS→S1
DEC ADD CORRN→L

OP2(L) + OP1(M)
ORIG OP1 ADR TO R
BR ON G1 NEGATIVE AND FIRST WORD STAT 6

00 — C8C
A M−L→D✻BS
D HOT1→ADDER
D 3→MB
L WS2→LSA
L LS→R→LS
R G2<=LB G2<0
L4— ✻✻ —LD

BRANCH IGNORES G2 SIGN

CORRECTED RESULTS (M−L) TO SAR•
OP2 ADR TO R•
BR ON G2 EQ OR LESS THAN LB AND G2 NEGATIVE•

00 — C90
E 0011
A H−1→H
D G1•2−1
L WS•E→LSA
L LS→L→LS
C S03ΩE•0→BS
R TZ✻BS LB=3
Q4— ✻✻ —QD

LATCH ZERO TEST PER BYTE STATS

PARTIALLY UPDATE OP1 ADR (H−1 TO H)
OP2 WD (WS3) TO L
RESET BYTE STATS

G2 NEGATIVE

01 — C91
E 0000
A H−4→H
D HOT1→ADDER
C E→BS
R TZ✻BS
L5— X✻ —LE

UPDATE OP1 ADR TO NEXT WD (H−4 TO H)
RESET BYTE STATS
BR ON RESULTS = 0

G2 LESS THAN OR EQ TO LB

10 — C92
E 0011
A H−4→H
D HOT1→ADDER
L WS•E→LSA
L LS→L→LS
C S03ΩE•0→BS
R TZ✻BS
Q5— X✻ —QE

UPDATE OP1 ADR TO NEXT WD (H−4 TO H)
OP2 WD (WS3) TO L
BR ON RESULTS = 0

1X — CC7
E 1011
A M−L→T
D HOT1→ADDER
L WS•E→LSA
L LS→R→LS
A6— XX —AF

OP1 EXHAUSTED
RECREATED SUM FOR ZERO CHECK
LENGTH (L1L2) TO R

0X — CC5
A M−L→T
D HOT1→ADDER
D 3→MB
L WS2→LSA
L LS→R→LS
R G2<=LB G2<0
E6— ✻✻ —EF

BRANCH IGNORES G2 SIGN

RECREATE SUM FOR ZERO CHECK
OP2 ADR TO R

QS118 (XX) —AFE

X0 — C32
E 0100
A H→A
C S47ΩE
J6— XX —JF

OP1 ADR TO SAR
TN ON NON−ZERO STAT 5•

XX — 01C3
E 0101
A 0→M
L WS•E→LSA
L LS→L→LS
J7— 1X —JG

ZEROS TO M FOR EXTENDED OP2 FIELD
FETCH SIXES (WS5) TO L•

QS112 (1X) —JGE

X1 — C33
A H→A
N6— XX —NF

OP1 ADR TO SAR

QS116 (X0•X1) —QEE

QS112 (00•01•10•11) —QDF

258068	10/20/67	MACH	2050FP	DATE 12/29/67	SHEET 1 QS114
258430	12/21/67	NAME		LOG 662	VERSION
		MODE	MANUAL	DECIMAL ADD, AP, SUBTRACT, SP ADD, SUBTRACT INNER LOOP	
		P•No•	5469500		
		IBM CORP•	SDD		

QS114

QS114•QEE
(XO•X1)
FROM MAIN LOOP

NON-ZERO DIG STORED
XO — C3A
E 0100
A 0→M
C S47QE
R 0 LB=3
C3— 0✻ —CC

ZEROS TO M
TN ON NON-ZERO STAT
BR ON LB (FETCH RQD)

ZEROS STORED
X1 — C3B
A 0→M
R 0 LB=3
G3— 0✻ —GC

ZEROS TO M
BR ON LB EQ 3
(FETCH REQD)

FETCH NEXT OP2 WORD
01 — C95
A R-4→R•A
D HOT1→ADDER
L WS2→LSA
L R→LS
G4— XX —GD

OP2 ADR - 4
TO SAR• R• WS2

XX — C8D
E 0011
D D→L
L WS•E→LSA
L L→LS
G5— 00 —GE

LAST OP2 WORD TO L

00 — C94
B LLB→U
B U→W
B W→MMB
D G1•2-1
D LB•MB-1
C W→IVD
C 1→BS✻MB
R AQ(B=1)→A
R G1=0 G2=0
C7— ✻✻ —CG

OP2 BYTE TO M
CK FOR VALID DIGITS
SET BYTE STAT
DECR G1 G2 LB MB
BR ON G1 G2

OP2 FIELD EXHAUSTED
10 — C96
E 0101
A H→A
L WS•E→LSA
L LS→L→LS
G7— 1X —GG

OP1 ADR TO SAR
SIXES TO L

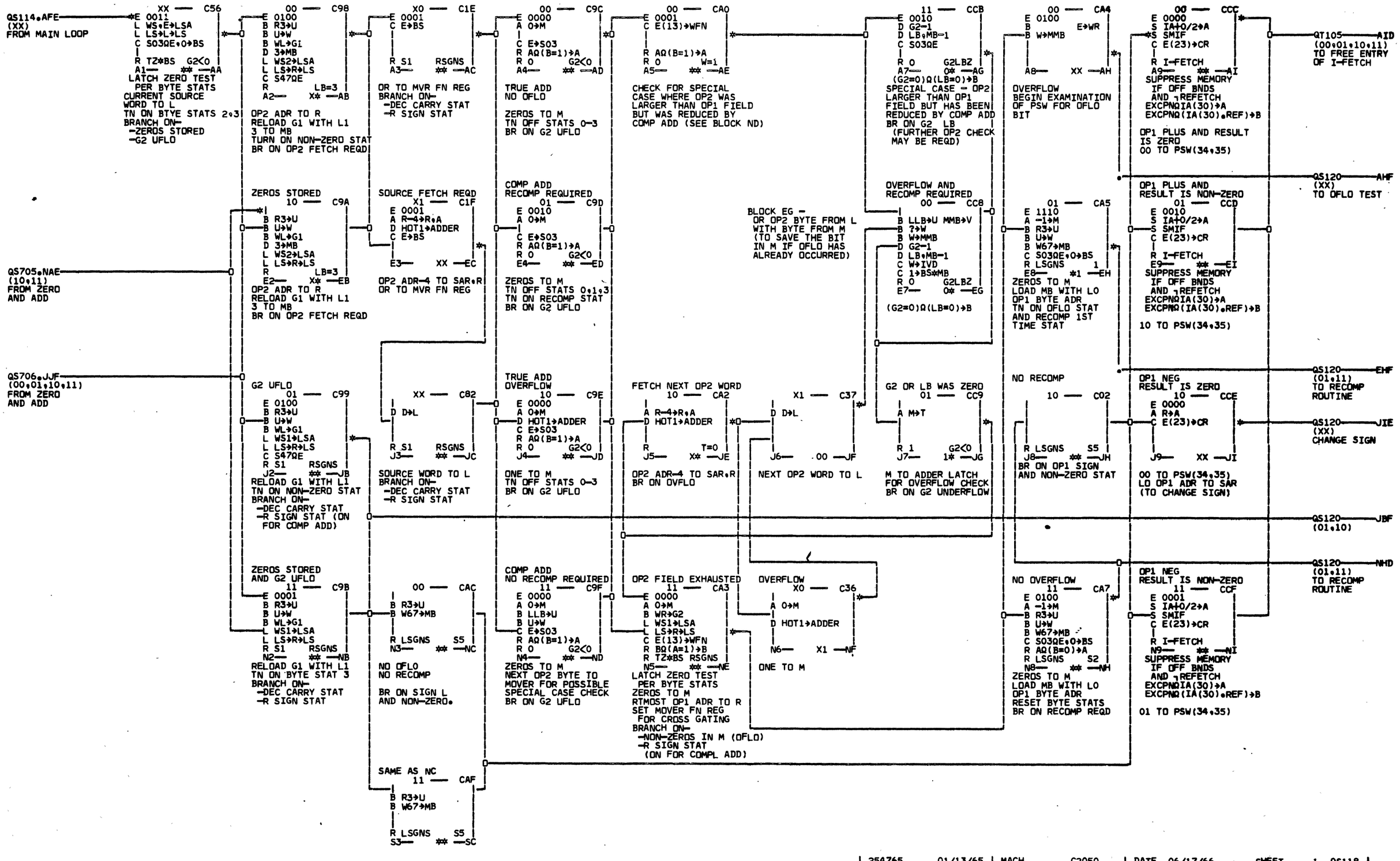
OP1 FIELD EXHAUSTED
(OP2 MAY ALSO BE
EXHAUSTED)
11 — 0C97
E 0101
A H→A
L WS•E→LSA
L LS→L→LS
L7— 1X —LG

OP1 ADR TO SAR
SIXES TO L

QS112 —GGE
(1X)
CHECK OP1 FOR
VALID DECIMAL
DIGITS

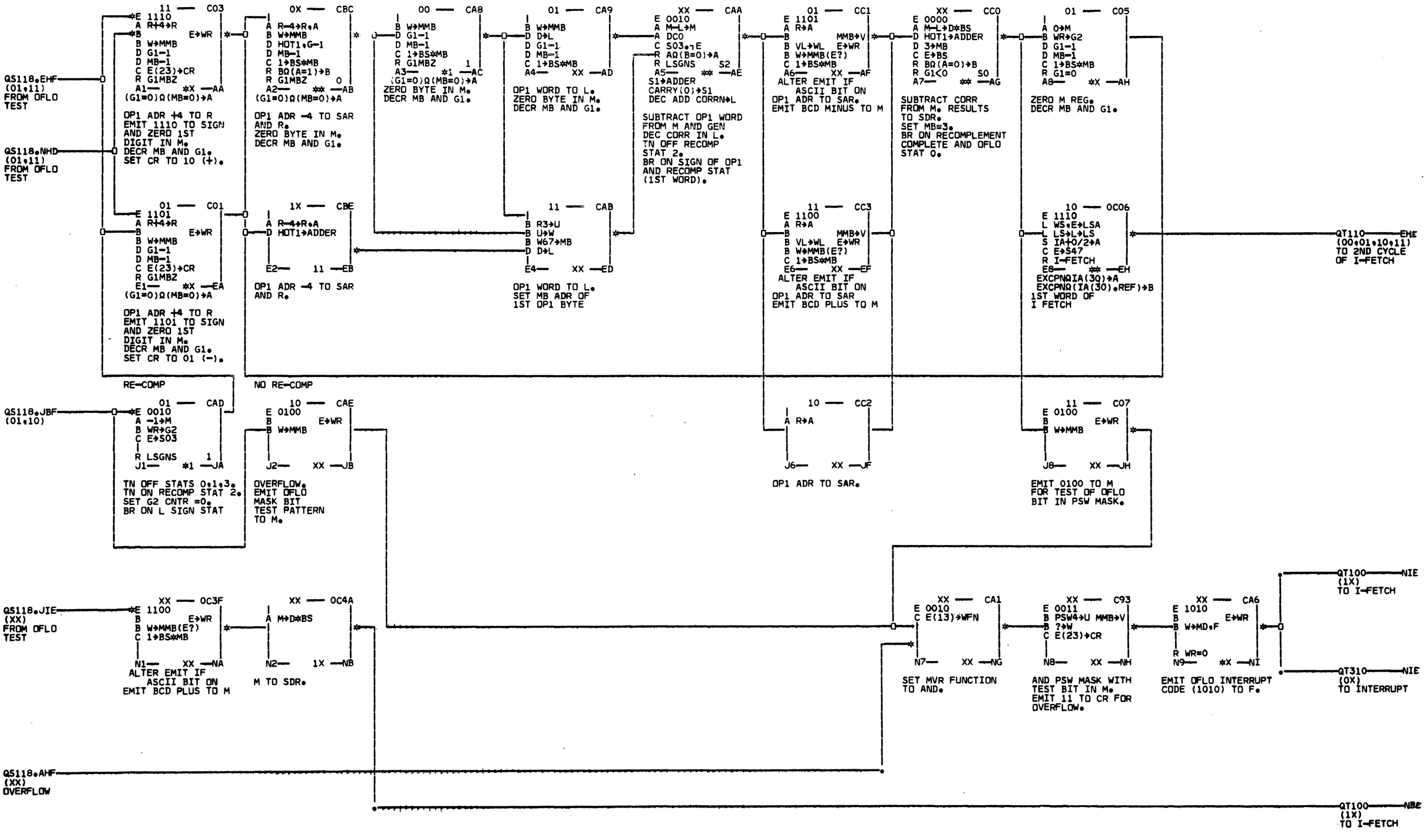
254765	01+13/65	MACH	C2050	DATE 10+01/65	SHEET 1 QS116
254769	01+13/65	NAME		LOG 902	VERSION
255449	09+20/65	MODE	MANUAL	DECIMAL ADD• AP• SUBTRACT• SP	
		P•N•	5364811	G2 COUNTER LESS THAN FOUR	
		IBM CORP•	SDD		

QS116



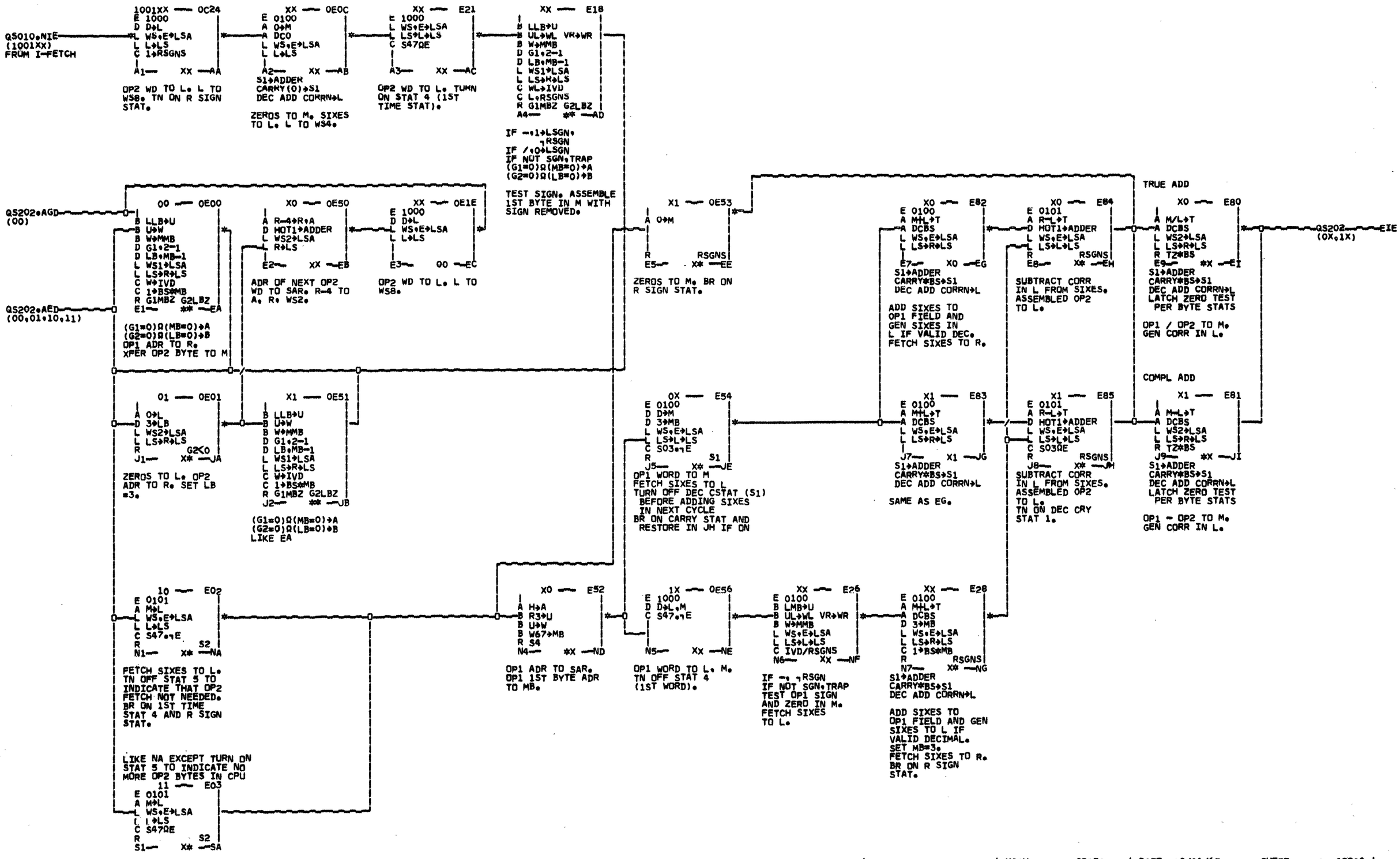
254765	01/13/65	MACH	C2050	DATE 06/17/66	SHEET 1 QS118
254772	02/23/65	NAME		LOG 631	VERSION
255449	09/20/65	MODE	MANUAL		
256498	03/25/66	P•N•	5364812	DECIMAL ADD, AP, SUBTRACT, SP	
256907	06/10/66	IBM CORP•	SDD	G1 CTR LTH ZERO, CHECK OFLO	

QS118



QS120

258060	09/29/67	MACH	2050FP	DATE 12/29/67	SHEET 1 QS120
258430	12/21/67	NAME		LOG 662	VERSION
		MODE	MANUAL	DECIMAL ADD• AP• SUBTRACT• SP	
		P•N•	5449079	RECOMPLEMENT AND END ROUTINE	
		IBM CORP•	SDD		



254765	01/13/65	MACH	C2050	DATE 10/01/65	SHEET 1 QS200
255449	09/20/65	NAME		LOG 902	VERSION
		MODE	MANUAL	COMPARE DECIMAL (CP)	
		P•N•	5364814	COMPARISON OF 1ST WD OPND 1	
		IBM CORP•	SND		

QS200

QS200•EIE
(0X•1X)

1X — E57
E 1000
A H+4→H
D HOT1→ADDER
L WS•E→LSA
L LS•L→LS
R G1<0 TZ*BS
E1— ** —EA

FETCH OP2 WORD TO L•
DECR OP1 ADR
(H-4 TO H)•

00 — E04
E 1000
A 0→M
C S03QE
R AQ(B=1)→A
R S5 G2<0
A2— ** —AB

RESET BYTE STATS
BR ON EQUAL COMP

10 — 0E0A
E 0000
A R+4→R•A
D HOT1→ADDER
L WS2→LSA
L R→LS
C E→BS
A3— XX —AF

ADR OF NEXT OP2
WD TO SAR• R+4 TO
R• A• WS2•

XX — 0E34
E 1000
D D→L
L WS•E→LSA
L L→LS
A4— 00 —AG

OP2 WD TO L•
L TO WS8•

QS200 — AGD
(00)

QS200 — AED
(00•01•10•11)

01 — E05
A 0→M
R AQ(B=1)→A
R S5 G2<0
E2— ** —EB

ZEROS TO M•
BR ON Q STAT5
AND G2 EXHAUSTED•

00 — E08
E 0001
B LLB→U
B U→W
B W→MMB
D G1•2-1
D LB•MB-1
L WS1→LSA
L LS→R→LS
C E→BS
R G1MBZ G2LBZ
E3— ** —AE

(G1=0)Ω(MB=0)→A
(G2=0)Ω(LB=0)→B
OP1 ADR TO R•
XFER OP2 BYTE TO
M• DECR G1• G2•
LB• MB•

0X — 0E88
B LLB→U
B U→W
B W→MMB
D G1•2-1
D LB•MB-1
L WS1→LSA
L LS→R→LS
C 1→BS*MB
R G1MBZ G2LBZ
E4— ** —SB

(G1=0)Ω(MB=0)→A
(G2=0)Ω(LB=0)→B
OP1 ADR TO R•
XFER OP2 BYTE TO M

00 — ED4
R BQ(A=0)→B
R S1 RSGNS
C6— ** —CE

01 — ED5
E 0000
S IA+0/2→A
S SMIF
C E(23)→CR
R I-FETCH
E6— ** —JE

SUPPRESS MEMORY
IF OFF BNDS
AND ¬ REFETCH
EXCPNΩIA(30)→A
EXCPNΩ(IA(30)
REF)→B

SET COND REG TO
00 FOR EG COMP•

11 — E13
E 0001
S IA+0/2→A
S SMIF
C E(23)→CR
R I-FETCH
C6— ** —SG

SUPPRESS MEMORY
IF OF BNDS
AND ¬ REFETCH
EXCPNΩIA(30)→A
EXCPNΩ(IA(30)•
REF)→B

BLOCK SG —
OP2 GTR THAN OP1
SET COND REG TO 01

QT105 — SGE
(00•01•10•11)
TO FREE ENTRY
OF I-FETCH

00 — E10
E 0001
S IA+0/2→A
S SMIF
C E(23)→CR
R I-FETCH
G8— ** —JF

SUPPRESS MEMORY
IF OFF BNDS
AND ¬REFETCH
EXCPNΩIA(30)→A
EXCPNΩ(IA(30)•REF)→B

SET COND REG TO
01 FOR OP1 LESS
THAN OP2•

10 — E06
E 1010
A 0→M
C S03QE
R AQ(B=1)→A
R S5 G2<0
J2— ** —EC

ZEROS TO M•
TN ON NON-ZERO
STAT 0 AND
END STAT 2•
BR ON Q STAT 5
AND G2 EXHAUSTED•

11 — 0E0B
E 0000
A 0→L
C E→BS
R G1<0
J3— *X —NA

ZEROS TO L•

1X — 0E8A
E 0010
C E(23)→CR
R S0 RSGNS
J4— ** —NB

SET COND REG TO
10 FOR OP1 GTR
THAN OP2•

10 — ED6
R LSGNS 0
J6— *0 —NC

BR ON L SIGN STAT
(OP1 SIGN)•

10 — 0E12
E 1110
L WS•E→LSA
L LS→L→LS
S IA+0/2→A
C E→S47
R I-FETCH
L8— ** —NG

EXCPNΩIA(30)→A
EXCPNΩ(IA(30)•
REF(→B

LIKE NF•

QT110 — NGE
(00•01•10•11)
TO 2ND CYCLE
OF I-FETCH

11 — E07
E 0010
A 0→M
C S03QE
R AQ(B=1)→A
R S5 G2<0
N2— ** —ED

ZEROS TO M•
TN ON END
STAT 2•
BR ON Q STAT 5
AND G2 EXHAUSTED•

11 — ED7
R LSGNS S1
N6— ** —SC

BR ON L SIGN STAT
AND DEC CRY STAT 1•

01 — 0E11
E 1110
L WS•E→LSA
L LS→L→LS
S IA+0/2→A
C E→S47
R I-FETCH
Q8— ** —NF

EXCPNΩIA(30)→A
EXCPNΩ(IA(30)•REF)→B
FIRST WORD OF
I FETCH•

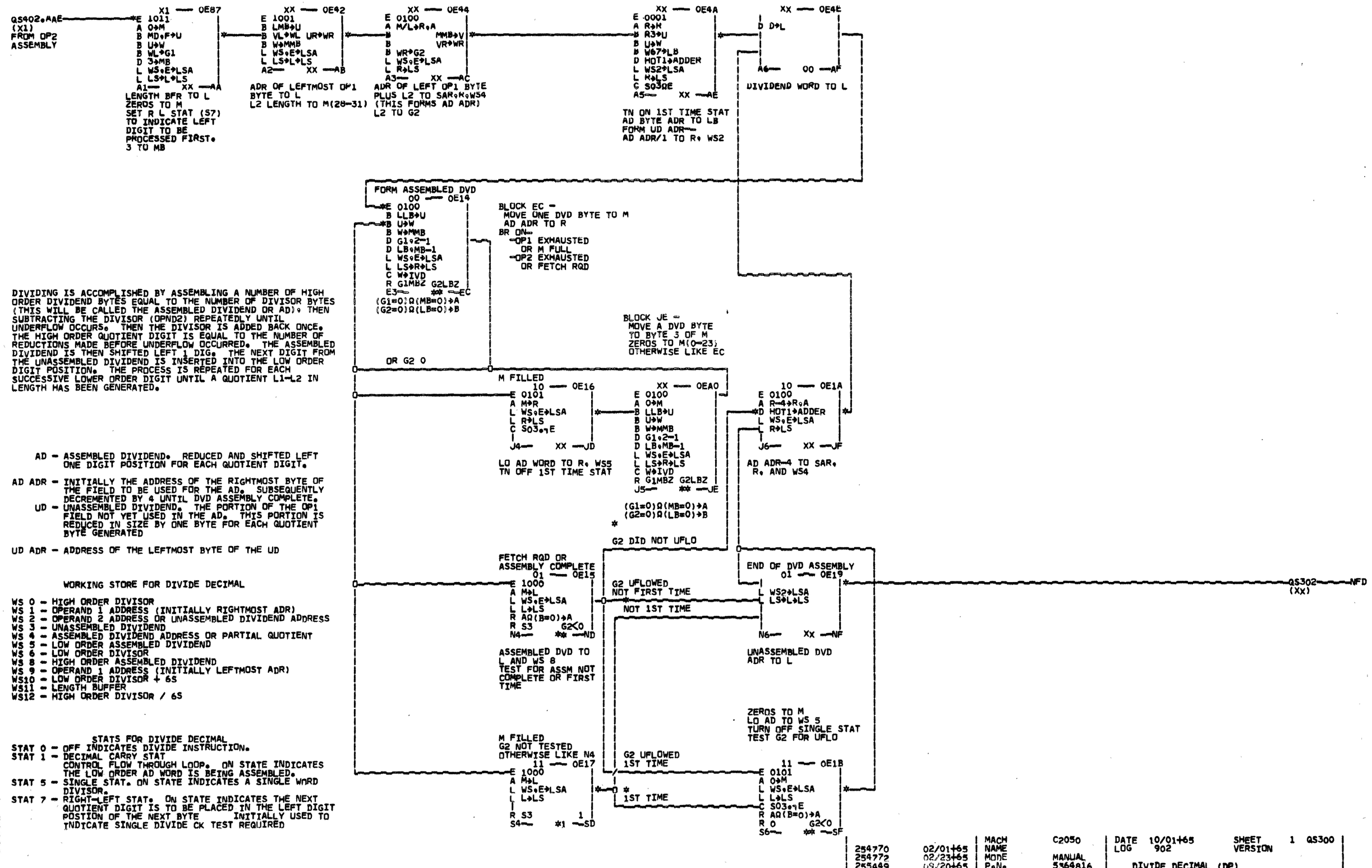
0X — E55
E 0111
B E→WR
B W→MD•F
Q1— 0X —JA

GO TO INVALID
DATA TRAP

QT310 — JAE
(0X) TO
INTERRUPT
ROUTINE

254760	12/21/64	MACH	C2050	DATE 11/09/66	SHEET 1 QS202
254765	01/13/65	NAME		LOG 642	VERSION
255449	09/20/65	MODE	MANUAL	COMPARE DECIMAL (CP)	
257390	11/10/66	P•N•	5364815	COMPARISON OF 1ST WD OPND 1	
		IBM CORP•	SDD		

QSNN



DIVIDING IS ACCOMPLISHED BY ASSEMBLING A NUMBER OF HIGH ORDER DIVIDEND BYTES EQUAL TO THE NUMBER OF DIVISOR BYTES (THIS WILL BE CALLED THE ASSEMBLED DIVIDEND OR AD)• THEN SUBTRACTING THE DIVISOR (OPND2) REPEATEDLY UNTIL UNDERFLOW OCCURS• THEN THE DIVISOR IS ADDED BACK ONCE• THE HIGH ORDER QUOTIENT DIGIT IS EQUAL TO THE NUMBER OF REDUCTIONS MADE BEFORE UNDERFLOW OCCURRED• THE ASSEMBLED DIVIDEND IS THEN SHIFTED LEFT 1 DIG• THE NEXT DIGIT FROM THE UNASSEMBLED DIVIDEND IS INSERTED INTO THE LOW ORDER DIGIT POSITION• THE PROCESS IS REPEATED FOR EACH SUCCESSIVE LOWER ORDER DIGIT UNTIL A QUOTIENT L1-L2 IN LENGTH HAS BEEN GENERATED•

AD — ASSEMBLED DIVIDEND• REDUCED AND SHIFTED LEFT ONE DIGIT POSITION FOR EACH QUOTIENT DIGIT•

AD ADR — INITIALLY THE ADDRESS OF THE RIGHTMOST BYTE OF THE FIELD TO BE USED FOR THE AD• SUBSEQUENTLY DECREMENTED BY 4 UNTIL DVD ASSEMBLY COMPLETE•

UD — UNASSEMBLED DIVIDEND• THE PORTION OF THE OP1 FIELD NOT YET USED IN THE AD• THIS PORTION IS REDUCED IN SIZE BY ONE BYTE FOR EACH QUOTIENT BYTE GENERATED

UD ADR — ADDRESS OF THE LEFTMOST BYTE OF THE UD

WORKING STORE FOR DIVIDE DECIMAL

WS 0 — HIGH ORDER DIVISOR
WS 1 — OPERAND 1 ADDRESS (INITIALLY RIGHTMOST ADR)
WS 2 — OPERAND 2 ADDRESS OR UNASSEMBLED DIVIDEND ADDRESS
WS 3 — UNASSEMBLED DIVIDEND
WS 4 — ASSEMBLED DIVIDEND ADDRESS OR PARTIAL QUOTIENT
WS 5 — LOW ORDER ASSEMBLED DIVIDEND
WS 6 — LOW ORDER DIVISOR
WS 8 — HIGH ORDER ASSEMBLED DIVIDEND
WS 9 — OPERAND 1 ADDRESS (INITIALLY LEFTMOST ADR)
WS10 — LOW ORDER DIVISOR + 6S
WS11 — LENGTH BUFFER
WS12 — HIGH ORDER DIVISOR / 6S

STATS FOR DIVIDE DECIMAL

STAT 0 — OFF INDICATES DIVIDE INSTRUCTION•
STAT 1 — DECIMAL CARRY STAT CONTROL FLOW THROUGH LOOP• ON STATE INDICATES THE LOW ORDER AD WORD IS BEING ASSEMBLED•
STAT 5 — SINGLE STAT• ON STATE INDICATES A SINGLE WORD DIVISOR•
STAT 7 — RIGHT-LEFT STAT• ON STATE INDICATES THE NEXT QUOTIENT DIGIT IS TO BE PLACED IN THE LEFT DIGIT POSTION OF THE NEXT BYTE INITIALLY USED TO INDICATE SINGLE DIVIDE CK TEST REQUIRED

254770	02/01+65	MACH	C2050	DATE	10/01+65	SHEET	1 QS300
254772	02/23+65	NAME		LOG	902	VERSION	
255449	09/20+65	MODE	MANUAL				
		P•N•	5364816		DIVIDE DECIMAL (DP)		
		IBM CORP•	SND		DIVIDEND ASSEMBLY		

QS300

QS300.NFD (XX) FROM DIVIDEND ASSEMBLY

XX — 0EA8
E 0000
A L→A
L WS2→LSA
L LS→R→LS
C E(13)→WFN
A1— XX —AA
UD ADR TO SAR
SET MVR FN REG TO CROSS

XX — 0EA9
E 1001
B R3→U
B U→W
B W67→LB
L WS•E→LSA
L LS→R→LS
A2— XX —AB
OP1 ADR TO R
1ST UP BYTE ADR TO LB

XX — 0EAA
E 0011
B R3→U
B U→W
B W67→MB
D D→L
L WS•E→LSA
L L→LS
A3— XX —AC
UD WORD TO L, WS3 LEFTMOST OP1 BYTE ADR TO MB

XX — 0EA5
E 0110
B LLB→U
B ?→WR
B WR→F
L WS•E→LSA
L LS→L→LS
C WR→IVD
C 1→BS#MB
A7— XX —AG
LO DVR TO L
1ST UD BYTE TO F
SET A BYTE STAT FOR FIRST STORE

XX — 0EA6
E 0101
L WS•E→LSA
L LS→R→LS
A8— XX —AH
LO AD TO R

XX — 0EA7
E 0000
A R→L→T
A BCO
D HOT1→ADDER
L WS•E→LSA
L LS→L→LS
R S7
A9— #X —AI
SAVE CAR FROM 0
DIVIDE CK TEST (LO AD WD) − (LO DVR WD) TO LATCH
SAVE CARRY OUT OF ADDER(0)
WU DVR TO L
BR ON SINGLE STAT

QS304 (0X) TO DOUBLE WORD ROUTINE — AIE

QS306.AEE (01) FROM QUOTIENT STORE

SINGLE WD DVR
1X — 0E92
A R→R
A F→SL4→F
R CSTAT 1
E1— #1 —EA
SHIFT AD WD LEFT 4
ENTER 1ST UD DIGIT FROM F TO LO DIG POS
SPILL HO DIG TO F
TAKE RESULTING WD TO R
BR ON RESULT OF DIV CK TEST

01 — 0E2D
E 0110
B WL→MD
L WS•E→LSA
L LS→L→LS
E2— 11 —EB
LO DVR TO L
0 TO MD

REDUCTION CYCLE
11 — 0E2B
A R→L→R
A DCO
D HOT1→ADDER
E3— XX —EC
S1→ADDER
CARRY(0)→S1
DEC ADD CORRN→L
AD − DVR TO R
CORR FACTOR TO L
SAVE CARRY OUT OF ADDER(0) IN S1

DECIMAL CORRECTION
XX — 0EA3
E 0110
A R→L→R
D HOT1→ADDER
D MD+1
L WS•E→LSA
L LS→L→LS
R S1 1
E4— #1 —ED
(AD−DVR)−CORR TO R
INCR MD
DVR TO L
BR ON S1
(CONTINUE IN LOOP UNTIL UFLO OCCURS)

UFLO OCCURRED
01 — 0E29
E 1010
L WS•E→LSA
L LS→L→LS
E5— XX —EE
LO DVR / 6S TO L

XX — 0EA4
A R/L→R
A DCO
B WR→G2
D MD−1
R S7
E6— #X —EF
S1→ADDER
CARRY(0)→S1
DEC ADD CORRN→L
REDUCED AD + (DVR + 6S) TO R
DEC CORR TO L
DECR MD
ZEROS TO MD
BR ON R#L STAT

RIGHT QUOT DIGIT
0X — 0E89
E 0011
A R→L→R
B MD.F→U MMB→V
B VL→WL ?→WR
B W→MMB
D HOT1.G−1
D MB/1
L WS•E→LSA
L LS→L→LS
C S47QE
R MB=3 1
E7— #1 —EG
AD−CORR FACTOR TO R
UD WD TO L
QUOT DIGIT TO M
DECR G1 INCR MB
SET R#L TO LEFT
BR ON M FILLED

M NOT FILLED
01 — 0E1D
B LLB→U
B WR→F ?→WR
C WR→IVD
C 1→BS#MB
E8— 00 —EH
LEFT DIGIT FROM UNASSM DVD TO F
INTERRUPT IF THIS DIG IS NOT VALID DEC
SET BYTE STAT FOR NEXT QUOT BYTE

00 — 0E20
A R→R
A F→SL4→F
B WL→MD
L WS•E→LSA
L LS→L→LS
E9— 11 —EI
SHIFT AD LEFT
ENTER F INTO LO POS
SPILL HO DIG TO F (SHOULD BE ZERO)
DVR TO L
ZEROS TO MD
GO TO REDUCTION LOOP

QS306.EGF (11) TO QUOTIENT STORE

SINGLE WORD LOOP

QS306.JFF (11) FROM UNASSM DIV FETCH

QS304.ABE (01.11) FROM DOUBLE DIVIDE CHECK TEST

DIVIDE CHECK
11 — 0E2F
E 1011
B E→WR
B W→MD.F
J2— 0X —JB
DIV CK CODE TO MD.F

* REDUCTION LOOP

LEFT QUOT DIGIT
1X — 0E8B
E 0011
A R→L→R
B MD.F→U
B UL→WL VR→WR
B W→MMB
D HOT1→ADDER
L WS•E→LSA
L LS→L→LS
C S47•1E
R BQ(A=1)→B
R G1=0 0
J7— ## —JG
AD − CORR TO R
UD WORD TO L
QUOT DIGIT TO M
SET R#L TO RIGHT
BR ON QUOT COMPLETE

00 — 0E24
B LLB→U
B U→W
B WR→F
D LB/1
C WR→IVD
R AQ(B=1)→A
R 0 LB=3
J8— ## —JH
QUOT NOT COMPLETE
RIGHT DIGIT FROM UD TO F
INTERRUPT IF THIS DIGIT IS A SIGN
DECR LB
BR ON FETCH RQD

QS306 (11) TO UNASSM DIVIDEND FETCH — JHD

* INTERRUPT FOR DIVIDE CHECK

QT310 (0X) TO INTERRUPT ROUTINE — JBE

QS304.NAE (11) FROM DOUBLE WORD LOOP

BEGIN FINAL QUOT STORE
11 — 0E27
E 1011
A R→L
A SL4→F
B LLB→U
L WS•E→LSA
L LS→R→LS
C L•RSGNS
N1— XX —NA
IF −•1→LSGN•RSGN
IF +•0→LSGN
IF NOT SGN.TRAP
SIGN TO MVR FOR TEST
REMAINING AD LEFT 4 TO L
LENGTH BFR TO R

XX — 0EA1
E 1001
B R3→U
B U→W
B WR→G2
L WS•E→LSA
L LS→R→LS
R RSGNS
N2— X# —NB
OP 1 ADR TO R
L2 TO G2
BR ON RESULT OF SIGN TESTS

SIGNS ALIKE
XO — 0E8C
E 1100
A R→A
B MMB→V
B VL→WL E→WR
B W→MMB(E?)
D G1−1
L WS1→LSA
L LS→R→LS
N3— XX —NC
ALTER EMIT IF ASCII BIT ON
PLUS SIGN TO QUOT
DECR G1
FINAL QUOT ADR TO SAR
RTMOST OP1 ADR TO R

MINUS SIGN TO QUOT
OTHERWISE LIKE N3
X1 — 0E8D
E 1101
A R→A
B MMB→V
B VL→WL E→WR
B W→MMB(E?)
D G1−1
L WS1→LSA
L LS→R→LS
S3— XX —SC

XX — 0EA2
E 0000
A M→D#BS
B R3→U
B U→W
B W67→MB
D 3→LB
L WS1→LSA
L LS→R→LS
C E→BS
R LSGNS
N5— #X —SE
LO QUOT BYTES TO SDR
RTMOST OP1 ADR TO R
BYTE ADR OF RTMOST OP1 BYTE TO MB
3 TO MB
RESET BYTE STATS

BRANCH ON DIVIDEND SIGN

PLUS TO REMAINDER
0X — 0E91
E 1100
B LLB→U
B UL→WL E→WR
B W→MMB(E?)
D G2−1
D LB•MB−1
C 1→BS#MB
R G1MBZ G2=0
N7— ## —NG
ALTER EMIT IF ASCII BIT ON
(G1=0)Ω(MB=0)→A

PLUS SIGN AND LO DIGIT TO RMDR (IN M)
SET CORRESPONDING BYTE STAT
DECR LB MB G2
BRANCH ON−
A−M FILLED
B−RMDR COMPLETE

1X — 0E93
E 1101
B LLB→U
B UL→WL E→WR
B W→MMB(E?)
D G2−1
D LB•MB−1
C 1→BS#MB
R G1MBZ G2=0
Q8— ## —SG
ALTER EMIT IF ASCII BIT ON
(G1=0)Ω(MB=0)→A

MINUS SIGN TO RMDR
OTHERWISE LIKE N7

QS308 (00.01.10.11) STORE REMAINDER — NGE

254772	02/23+65	MACH	C2050	DATE 10/01+65	SHEET 1 QS302
255449	09/20+65	NAME		LOG 902	VERSION
		MODE	MANUAL	DIVDE DEC (DP) SIGN INSERTION	
		P.N.	5364817	SINGLE WORD DIVISOR ROUTINE	
		IBM CORP.	SND		

QS302

QS302•AIE
(0X)
FROM SINGLE WORD DIVIDE CHECK TEST

0X — 0E90
A M→L→T
A BCO
D CSTAT→ADDER
A1— XX —AA
SAVE CAR FROM 0
CONTINUE DIV CK TEST (HO AD) - (HO DVR) TO T CARRY INTO ADDER(31) FROM LO SUBTRACTION
SAVE CARRY OUT OF ADDER(0)

XX — 0EB1
E 0001
A R→R
A F→SL4→F
C S47ØE
R BR(A=1)→B
R CSTAT 55
A2— *X —AB
SHIFT LO AD WD LEFT 4 ENTER ZERO INTO LO DIGIT POS SPILL HO DIGIT TO F TAKE RESULT TO R
BRANCH ON—
A—CARRY STAT
B—HO DVR IS ZERO

00 — 0E2C
E 0110
A M→H
A F→SL4→F
B WL→MD
L WS•E→LSA
L LS→L→LS
A3— 1X —AC
DBL WD DIVIDE
NOT DIVIDE CHECK
SHIFT HO AD WD LEFT 4
ENTER HO DIGIT FROM LO AD WORD SPILL
HO ZERO TO F
LO DVR TO L
ZEROS TO MD

1X — 0E96
A R→L→R
A DCO
D HOT1→ADDER
A5— XX —AE
S1→ADDER
CARRY(0)→S1
DEC ADD CORRN→L
(LO AD) - (LO DVR) TO H
DEC CORR TO L
SAVE CARRY IN S1

XX — 0EB0
E 0000
A R→L→R
D HOT1→ADDER
L WS•E→LSA
L LS→L→LS
A6— XX —AF
(LO AD - LO DVR) - CORR FACTOR TO R
HO DVR TO L

XX — 0EAB
A H→L→H
A DCO
A7— XX —AG
S1→ADDER
CARRY(0)→S1
DEC ADD CORRN→L
(HO AD) - (HO DVR) TO H
ENTER S1 AS CARRY INTO ADDER(31)
SAVE CARRY OUT OF ADDER(0) IN S1
DEC CORR TO L

XX — 0EAC
E 0110
A H→L→H
D HOT1→ADDER
D MD/1
L WS•E→LSA
L LS→L→LS
R S1
A8— *X —AH
(HO AD - HO DVR) - CORR FACTOR TO H
LO DVR TO L
INCR MD
BRANCH OUT ON UFLO

ADD DVR BACK TO AD ONCE / REDUCTION LOOP

UFLO OCCURRED

0X — 0E94
E 1010
L WS•E→LSA
L LS→L→LS
E1— XX —EA
(LO DVR / 6S) TO L

XX — 0EAD
A R→L→R
A DCO
D MD-1
E2— XX —EB
S1→ADDER
CARRY(0)→S1
DEC ADD CORRN→L
(LO AD) + (LO DVR / 6S) TO R

XX — 0EAE
E 1100
A R→L→R
D HOT1→ADDER
L WS•E→LSA
L LS→L→LS
E3— XX —EC
PARTIAL SUM MINUS CORR FACTOR TO R
(HO DVR + 6S) TO L

XX — 0EAF
A M/L→H
A DCO
R S7
E4— *X —ED
S1→ADDER
CARRY(0)→S1
DEC ADD CORRN→L
(HO AD) + (HO DVR / 6S) TO H
BRANCH ON R*L STAT

QS306•AEE
(00)
FROM QUOTIENT STORE

RIGHT QUOT DIGIT

0X — 0E95
E 0011
A H→L→H
B MD•F→U MMB→V
B VL→WL ?→WR
B W→MMB
D HOT1→ADDER
D MB/1
L WS•E→LSA
L LS→L→LS
C S47ØE
R BR(A=1)→B
R MB=3 0
J1— ** —JA
PART SUM - CORR TO H
UD TO L INCR MB
QUOT DIGIT TO M
SET R/L STAT TO LEFT
BR ON M FILLED

M NOT FILLED

00 — 0E1C
B LLB→U
B ?→WR
B WR→F
D G1-1
C WR→IVD
C 1→BS*MB
J2— 10 —JB
NEXT UNASSEMBLED DIVIDEND DIGIT TO F
CK FOR VALID DEC DIG
SET BYTE STAT FOR NEXT QUOT BYTE
DECR G1

LEFT QUOT DIGIT

1X — 0E97
E 0011
A H→L→H
B MD•F→U
B UL→WL VR→WR
B W→MMB
D HOT1→ADDER
L WS•E→LSA
L LS→L→LS
C S47•E
R G1=0 1
N1— *1 —NA
(HO PARTIAL SUM) - (DEC CORR) TO H
QUOT DIGIT TO M
SET R*L STAT TO RT
BR ON QUOT COMPLETE

QUOT NOT COMPLETE

01 — 0E25
B LLB→U
B U→W
B WR→F
D LB/1
C WR→IVD
R 1 LB=3
N2— 1* —NB
NEXT UNASSEMBLED DIV DIGIT TO F CHECK FOR INVALID DEC DIGIT
INCR LB
BR ON OP1 (UNASSM DIV) FETCH REQD

10 — 0E22
E 0110
A R→R
A F→SL4→F
B WL→MD
L WS•E→LSA
L LS→L→LS
J4— 10 —JD
SHIFT LO AD LEFT 4
ENTER UD DIGIT FROM F INTO LO AD
POSITION SPILL HO DIGIT TO F LO DVR TO L ZEROS TO MD

10 — 0E2A
A H→H
A F→SL4→F
J5— 1X —JE
SHIFT HO AD LEFT 4
ENTER SPILLED DIGIT (FROM LO AD) INTO LO DIGIT POSTION
SPILL HO ZERO TO F
RETURN TO REDUCTION LOOP

DIVIDE CHECK OR HIGH ORDER DIVISOR IS ZERO → QS302 (01•11) —ABE

QS306 (11) TO QUOTIENT STORE —JAF

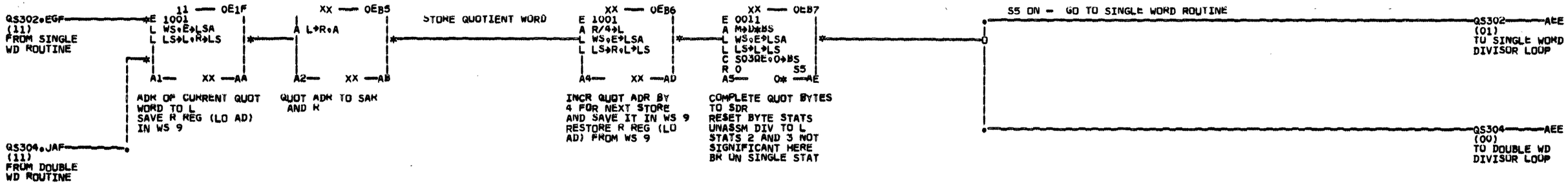
QS306 (11) TO UNASSM DIVIDEND FETCH —NBE

QS302 (11) TO LAST QUOTIENT STORE —NAE

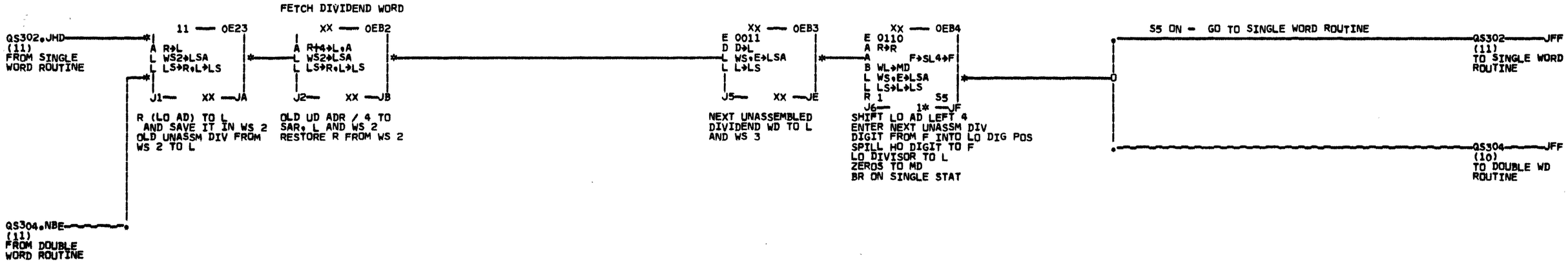
QS306•JFF
(10)
FROM UNASSM DIV FETCH

254772	02/23•65	MACH	C2050	DATE 10/01•65	SHEET 1 QS304
255449	09/20•65	NAME		LOG 902	VERSION
		MODE	MANUAL	DIVIDE DECIMAL (DP)	
		P.N.	5364818	DOUBLE WORD DIVISOR ROUTINE	
		IBM CORP.	SDD		

4 0 1 5 6 8

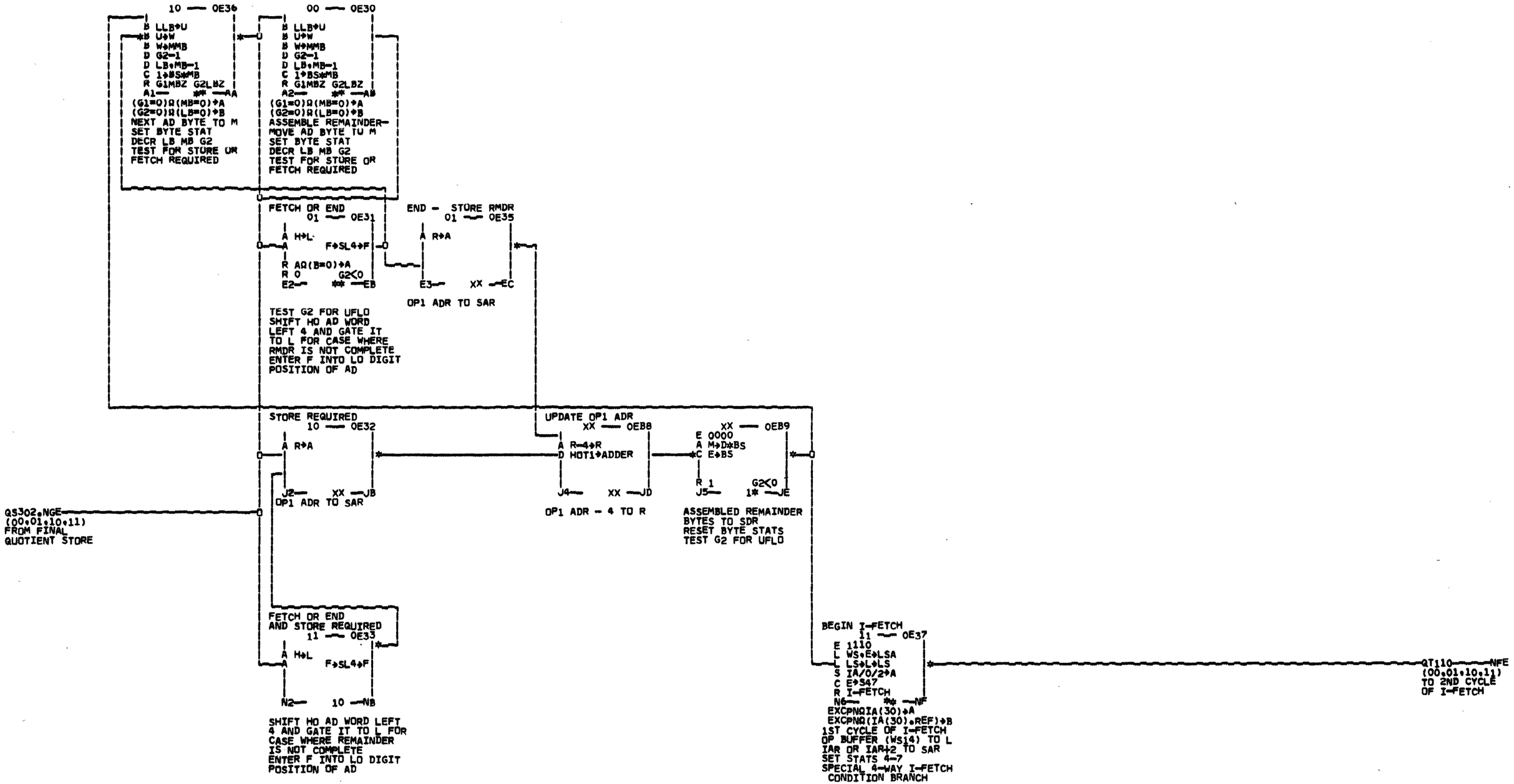


FETCH DIVIDEND WORD



QS306

254772	02/23/65	MACH	C2050	DATE 10/01/65	SHEET 1 QS306
255449	09/20/65	NAME		LOG 902	VERSION
		MODE	MANUAL	DIVINE DECIMAL (DP)	
		P.N.	5364819	QUOT STORE, DIVIDEND FETCH	
		IBM CORP.	SDD		



Q5308

254772	02+23/65	MACH	C2050	DATE 10+01/65	SHEET 1 Q5308
255449	09+20/65	NAME		LOG 902	VERSION
		MODE	MANUAL	DIVIDE DECIMAL (DP)	
		P•N•	5364820	REMAINDER ASSEMBLY	
		IBM CORP•	SDD		

QS010•NIE
(1100XX)
(1101XX)
FROM I-FETCH

1100XX — C30
E 0101
B WR•F
D D•L
L WS•E•LSA
L L•LS
C E•S47
A1— XX —AA

MULTIPLY DECIMAL
LO MPCN (OP2) WD
TO L• WS5
TN ON SINGLE STAT
AND SINGLE DIVIDE
CHECK TEST STAT

XX — 0D18
E 0100
A 0•M
A DCO
D 3•MB
L WS•E•LSA
L L•LS
A2— XX —AB

S1•ADDER
CARRY(0)•S1
DEC ADD CORRN•L
SIXES TO L• WS 4
TN OFF DEC CARRY STAT

XX — 0D28
E 0101
L WS•E•LSA
L LS•L•LS
A3— XX —AC

LO MPCN WD TO L

XX — 0D05
B LLB•U
B UL•WL VR•WR
B W•MMB
D G1•2=1
D LB•MB-1
C WL•IVD
C L•RSGNS
R AQ(B=1)•A
R G1MBZ G2LBZ
A4— ** —AD
IF -•1•LSGN•
¬RSGN
IF +•0•LSGN
IF NOT SGN•TRAP
(G1=0)Ω(MB=0)•A
(G2=0)Ω(LB=0)•B

00 — 0D04
B LLB•U
B U•W
B W•MMB
D G1•2=1
D LB•MB-1
C W•IVD
R AQ(B=1)•A
R G1MBZ G2LBZ
A5— ** —AE
(G1=0)Ω(MB=0)•A
(G2=0)Ω(LB=0)•B

1X — 0D4A
E 0110
B E•WR
B W•MD•F
A6— 0X —AF

M FILLED TWICE AND
G2 HAS NOT UFLOWED
THUS G2 WAS GTR
THAN 7 — A SPEC ERROR
LOAD CODE FOR SPEC
ERROR INTO MD F

BLOCK AD —
BEGIN ASSEMBLY
1ST OP2 BYTE TO MVR
LEFT DIGIT TO M
TEST SIGN
BR ON FETCH
OR STORE REQUIRED

1101XX — C34
E 0101
B WR•F
D D•L
L WS•E•LSA
L L•LS
C E•S47
E1— XX —EA

DIVIDE DECIMAL
LO DVR (OP2) WD
TO L•WS5
TN ON SINGLE STAT AND
SINGLE DIV CK TEST STAT

XX — 0D16
E 1000
A 0•M
D 3•MB
C 503QE
E2— XX —EB

ZEROS TO M
3 TO MB
TN OFF MPY R L STAT

00 — 0D08
R MB=3
E4— *X —ED

MORE THAN 4 BYTES
IN OP2 FIELD AND
G2 HAS NOT UFLOWED
CHECK FOR SPEC ERROR
(IF M IS FILLED,
G2 WAS GT 7)

0X — 0D48
A R-4•R•A
D MOT1•ADDER
L WS2•LSA
L R•LS
E5— XX —EE

M NOT FILLED,
FETCH NEXT OP2 WD
OP2 ADR-4 TO SAR•WS2

X1 — 0D4D
E 1100
A R+L•R
L WS•E•LSA
L R•LS
C S47QE
R AQ(B=0)•A
R G1<0 G2<0
E6— ** —EF

SAME AS SM10C PLUS
TN ON SINGLE STAT

10 — 0D02
E 0110
B E•WR
B W•MD•F
E7— 0X —EG

INTERRUPT
G2 WAS GTR THAN 7 OR
G2 WAS GTR THAN G1
LOAD CODE FOR SPEC
ERROR INTO MD F

DOUBLE WORD MPCN(DVR) ASSEMBLED
QS402 — EFD
(01)
OP2 ASSEMBLY
COMPLETE

INTERRUPT FOR SPECIFICATION
QT310 — AFE
(0X)
TO INTERRUPT
ROUTINE

•STO MPCN/DVR IN WS
10 — D06
A M•T
A DCO
B LLB•U
B U•W
B WR•F
C WR•IVD
R S5
J3— *1 —JC

S1•ADDER
CARRY(0)•S1
DEC ADD CORRN•L

M SHOULD BE FILLED
AND READY TO BE
STORED— G1 SHOULD
NOT HAVE UFLOWED,
SIXES TO L
RIGHT DIGIT OF
NEXT BYTE TO F

DOUBLE WD MPCN DVR
01 — 0D09
E 0000
A M•R
A SR4•F
L WS•E•LSA
L R•LS
R T=0
J4— X* —JD

HO MPCN(DVR) WORD
READY TO BE STORED
SHIFT M RIGHT 4,
ENTER ZEROS IN HIGH
ORDER DIGIT POS,
SPILL LO DIG TO F
TAKE RESULTING WD
TO R AND WS 0

BR ON HO ZEROS

X0 — 0D4C
E 1100
A R+L•R
L WS•E•LSA
L R•LS
R AQ(B=0)•A
R G1<0 G2<0
J6— ** —JF

HO MPCN(DVR) WORD
/ SIXES TO R•WS12
BRANCH ON—
—G1 LESS THAN ZERO
—G2 LESS THAN ZERO

11 — 0D03
E 0110
B E•WR
B W•MD•F
J7— 0X —JG

INTERRUPT
G2 WAS GTR THAN G1
OR G2 WAS = TO G1
LOAD CODE FOR SPEC
ERROR INTO MD F

11 — D07
A M•T
A DCO
L WS2•LSA
L LS•R•LS
R S5 G2<0
N3— ** —NC

S1•ADDER
CARRY(0)•S1
DEC ADD CORRN•L

MPCN(DVR) COMPLETED
OR OP2 FETCH RQD
(M MAY ALSO BE
FILLED OR G1 MAY
HAVE UFLOWED)
SIXES TO L
OP 2 ADR TO R
ZEROS TO F

10 — 0D0A
A R-4•R•A
D MOT1•ADDER
L WS2•LSA
L R•LS
N4— Xx —ND

FETCH NEXT OP2 WD
OP2 ADR-4 TO
SAR• WS 2

XX — 0D0D
E 0101
B WR•F
D D•L
L WS•E•LSA
L L•LS
R MB=3 0
N5— *0 —NE

NEXT OP2 WORD
TO L• WS 5
BR ON M FILLED AND
READY TO BE STORED

LO MPCN(DVR) WORD
READY TO BE STORED
SHIFT M RT 4, ENTER
F IN HO DIG POSITION
TAKE RESULTING WD
TO R AND WS 6

11 — 0D0B
E 0110
A M•R
A F•SR4•F
L WS•E•LSA
L R•LS
S4— XX —SD

LO MPCN(DVR) WORD
+ SIXES TO L• WS 10
ZEROS TO F
BRANCH ON—
—G1 LESS THAN ZERO
—G2 LESS THAN ZERO

XX — 0D11
E 1010
A R+L•L
B WR•F
L WS•E•LSA
L L•LS
R BQ(A=1)•B
R G1<0 G2<0
S5— ** —SE

*
SINGLE WORD MPCN(DVR) ASSEMBLED
QS402 — SEE
(01)
OP2 ASSEMBLY
COMPLETE

00 — 0D0O
E 0101
A 0•M
L WS•E•LSA
L LS•L•LS
C S47•¬E
S6— 00 —SF

ASSEMBLE NEXT
MPCN DVR WORD

RELOAD OP 2 WORD
INTO L
ZEROS TO M
TN OFF SINGLE STAT AND
SINGLE DIV CK TEST STAT

WORKING STORE FOR MULTIPLY DEC (AND DIVIDE WHERE COMMON• INDICATES DOUBLE USAGE•)

WS 0 — HO MPCN(DVR) DBLED HO MPCN / 6S
WS 1 — OPERAND 1 ADDRESS (INITIALLY THE RIGHTMOST ADR)
WS 2 — RIGHTMOST OP2 ADR HO PARTIAL PRODUCT
WS 3 — LO PARTIAL PRODUCT
WS 4 — SIXES
WS 5 — TEMPORARY OPND 2 WORD STORAGE
WS 6 — LO MPCN(DVR) MPLR PROD WORD (MPLR DIGITS ARE REPLACED AS USED WITH PRODUCT DIGITS)
WS 8 — DOUBLED LO MPCN / 6S
WS10 — LOW ORDER MPCN + 6S
WS11 — LENGTH BUFFER
WS12 — HIGH ORDER MPCN / 6S

STATS FOR MULTIPLY (AND DIVIDE WHERE COMMON)

STAT 0 — RIGHT LEFT STAT• ON STATE INDICATES NEXT PROD DIGIT TO BE INSERTED IN THE LEFT DIGIT POSITION OF THE NEXT BYTE•
ALSO USED (INITIALLY) AS THE MULTIPLY INSTR STAT ON STATE THEN INDICATES MULTIPLY BEING PERFORMED
STAT 1 — DECIMAL CARRY STAT•
STAT 5 — SINGLE STAT ON INDICATES A SINGLE WORD MPCN (DVR)•

QS400

254760	12•21/64	MACH	C2050	DATE 10•01/65	SHEET 1 QS400
254770	02•01/65	NAME		LOG 902	VERSION
255094	04•01/65	MODE	MANUAL		
255449	09•20/65	P•N•	5364821	MULTIPLY• DIVIDE DEC (MP• DP)	
		TBM CORP•	SDD	OPND 2 ASSEMBLY (MPCN/DVR)	

QS400.EFD (01) FROM OP2 ASSEMBLY

QS400.SEE (01) FROM OP2 ASSEMBLY

01 — 0D01
E 0110
A 0→M
B WR→F
L WS→E→LSA
L LS→R→LS
R SO
A1— X* —AA

LO MPCN(DVR) WORD TO R
ZEROS TO F
BR ON MULTIPLY STAT

ZEROS TO H

X0 — 0E86
A R→R
A DDCO F→SL1→Q
A2— XX —AB

S1→ADDER
CARRY(0)→S1
DEC DBL CORRN→L

MULTIPLY
1ST STEP OF LO MPCN DBLING
SHIFT LO MPCN LEFT ONE
CORRECTION FACTOR TO L

XX — D98
E 0100
A R/L→R
A BCO
L WS→E→LSA
L LS→L→LS
A3— XX —AC

SAVE CAR FROM 0
FINISH DBLING— CORRECTION + SHIFTED MPCN TO R
SIXES TO L

XX — 0D99
E 1000
A R+L→R
B E→WR
B WR→F
L WS→E→LSA
L R→LS
R S5
A4— X* —AD

SIXES / DBLED LO MPCN TO R, WS 8 (TO BE USED IN EXCESS 6S ADDITION)
BR ON SINGLE STAT

X0 — 0D50
E 0000
L WS→E→LSA
L LS→R→LS
A5— XX —AE

DBL WURD MPCN— MO MPCN WD TO R

*
SINGLE WORD MPCN

XX — 0D9A
A R→R
A DDCO Q→SL1→F
A6— XX —AF

S1→ADDER
CARRY(0)→S1
DEC DBL CORRN→L

1ST STEP OF MO MPCN DBLING
SHIFT MO MPCN LEFT ONE
CORRECTION FACTOR TO L

XX — D98
E 0100
A R+L→R
D CSTAT→ADDER
L WS→E→LSA
L LS→L→LS
A7— XX —AG

FINISH DBLING— CORRECTION FACTOR / SHIFTED MPCN TO R
SIXES TO L

XX — 0D9C
E 0000
A R+L→R
L WS→E→LSA
L R→LS
A8— X1 —AH

SIXES / DBLED MO MPCN TO R, WS=0

X1 — 0D51
E 1000
L WS1→LSA
L LS→R→LS
C S03ØE
A9— XX —AI

RIGHTMOST UP1 ADR TO R

R ON X1 — 0D4F
E 1101
B E→WR
B W→MMB(E?)
E6— XX —EF

ALTER EMIT IF ASCII BIT ON MPCN AND MPLR SIGNS UNALIKE, MINUS SIGN TO RIGHTMOST DIGIT POSITION OF OP1

QS300 (X1) TO DECIMAL DIVIDE — AAE

R OFF

XX — 0D34
A R→A
B R3→U
B U→W
B W67→LB
J1— XX —JA

OP 1 ADR TO SAR
LO MPLR BYTE ADR TO LB

XX — 0D38
E 0110
B R3→U
B U→W
B W67→MB
D D→L
L WS→E→LSA
L L→LS
J3— XX —JC

LO MPLR WORD TO L AND WS6
LO MPLR BYTE ADR TO MB

XX — 0D40
E 0010
A O→R
B LLB→U
B U→W
L WS→E→LSA
L R→LS
C IVD/RSGNS
J4— XX —JD

IF —, ¬RSGN
IF NOT SGN.TRAP
ZEROS TO R TO WS2
TEST OP1 SIGN
R SIGN IS OFF
IF OP2 SIGN WAS PLUS

XX — 0D44
E 0011
A L→M
L WS→E→LSA
L R→LS
R RSGNS
J5— X* —JE

LO MPLR WORD TO M
ZEROS TO WS3
BR ON R SIGN STAT

X0 — 0D4E
E 1100
B E→WR
B W→MMB(E?)
J6— XX —JF

ALTER EMIT IF ASCII BIT ON MPCN AND MPLR SIGNS ALIKE, PLUS SIGN TO RIGHTMOST DIGIT POSITION OF OP1

XX — 0D9D
E 0110
A M→L
B LLB→U
B U→W
B WL→MD
L WS→E→LSA
L L→LS
C WL→IVD
R S5
J7— X* —JG

LO OP1 WORD (W PROD SIGN) TO L, WS 6
1ST MPLR DIGIT TO MD

QS404.AED (XX) FROM SINGLE WORD LOOP

MD NOT = 0, GO TO SINGLE WD LOOP
BR ON MD EVEN
QS404 (00,10) TO SINGLE WORD LOOP — NBE

QS404.NFE (0X) FROM MPLR FETCH

0X — 0D55
E 0011
B LLB→U
B ?→WL
B WL→MD
L WS→E→LSA
L LS→R→LS
C WL→IVD
R S5
N1— X* —NA

LO PARTIAL PROD TO R
NEXT MPLR DIGIT (RT) TO MD

SINGLE WORD MPCN
X1 — 0D53
E 1010
L WS→E→LSA
L LS→L→LS
R MD3=0 MD=0
N2— ** —NB

(LO MPCN / 6S) TO L
BRANCH ON—
—MPLR DIGIT = 0
—MPLR DIGIT EVEN

11 — 0D13
E 0110
A R→R
A SR4→F
L WS→E→LSA
L LS→L→LS
N3— XX —NC

MPLR DIGIT = 0
SHIFT PARTIAL PROD RIGHT 4, SPILL LO DIGIT TO F
MPLR PROD TO L

XX — 0D9E
E 1000
A L→M
C S03,¬E
R SO
N5— X* —NE

MPLR PROD TO M
PRESET R L STAT TO INDICATE RIGHT MPLR DIGIT NEXT
BRANCH ON R L STAT

NOTE— R/L STAT WAS TURNED ON (SET FOR LEFT DIGIT) DURING 2ND LEVEL I-FETCH

0X — 0D49
E 1000
B MD.F→U MMB→V
B VL→WL UR→WR
B W→MMB
C S03ØE
N6— XX —NF

LO PARTIAL PROD DIGIT FROM F TO RIGHT DIG POS OF MPLR PROD WD PER MB
SET R+L STAT ON TO INDICATE NEXT PROD DIGIT IS TO BE INSERTED IN LEFT SIDE OF BYTE

MD NOT = 0, GO TO DBL WD MULTIPLY LOOP
BR ON MD EVEN
QS404 (00,10) TO DOUBLE WORD LOOP — SBE

1X — 0D4B
B MD.F→U MMB→V
B ?→WL VR→WR
B W→MMB
D G1-1
D LB.MB-1
C 1→BS*MB
R GIMBZ
Q6— X* —SF

(G1=0)Ω(MB=0)→A
SET LOWEST ORDER PROD DIGIT FROM PARTIAL PROD INTO LEFT DIG POS OF MPLR PROD WD PER MB
LEAVE L R STAT SET TO INDICATE RIGHT PROD DIGIT NEXT

0X — 0D54
E 0110
A M→L
B LLB→U
B ?→WL
B WL→MD
L WS→E→LSA
L L→LS
R S5
Q7— X* —SG

DO NOT STORE M YET
UPDATED MPLR PROD WORD TO L, WS 6
NEXT MPLR DIGIT (FROM RT DIG POS) TO MD
BR ON SINGLE STAT

STORE M
QS404 (1X) STORE PRODUCT — SFE

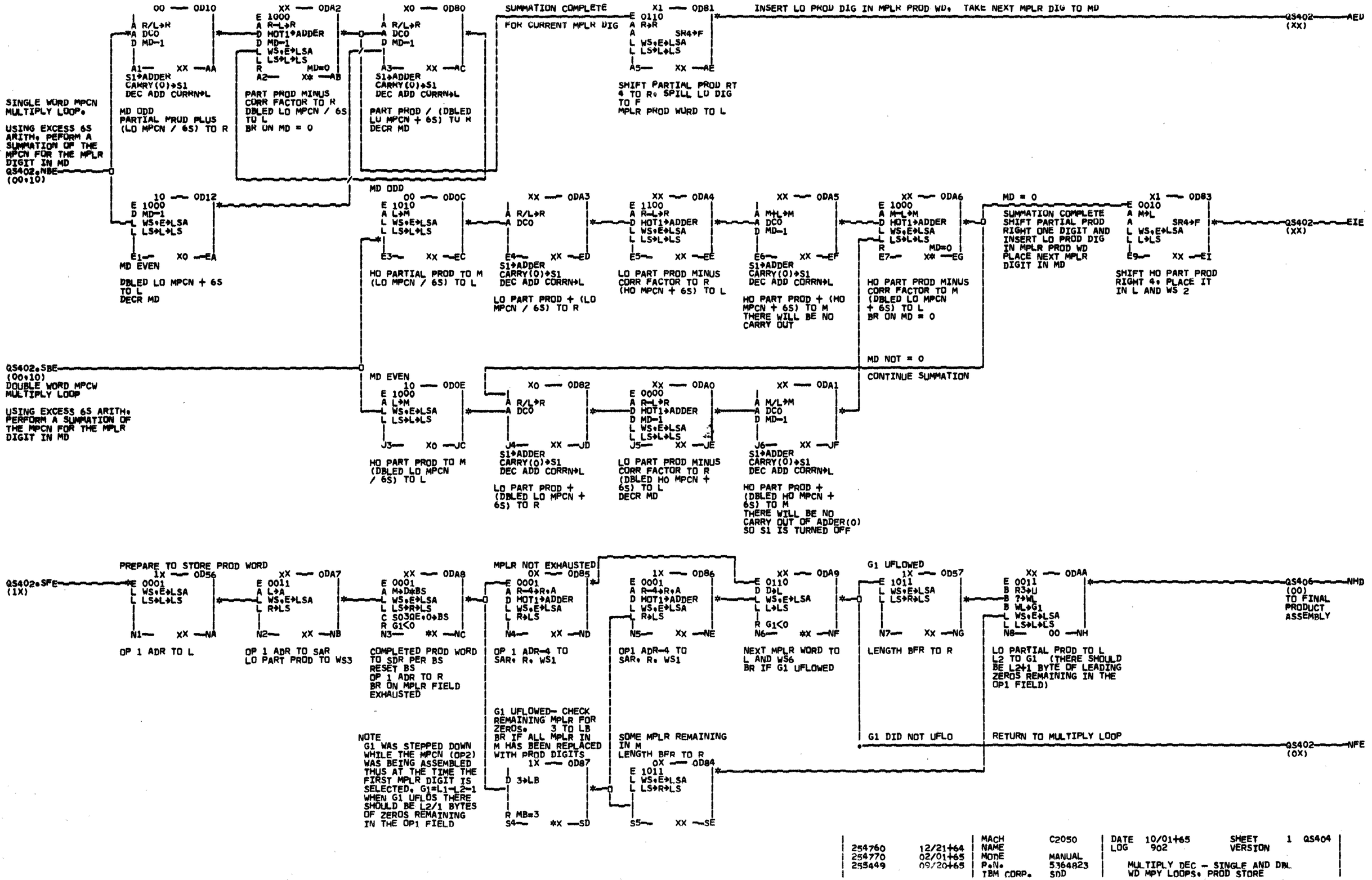
QS404.EIE (XX) FROM DOUBLE WORD LOOP

HO PART PROD TO L
BR ON—MPLR DIG = 0
—MPLR DIG EVEN
XO — 0D52
E 0010
L WS→E→LSA
L LS→L→LS
R MD3=0 MD=0
S2— ** —SB

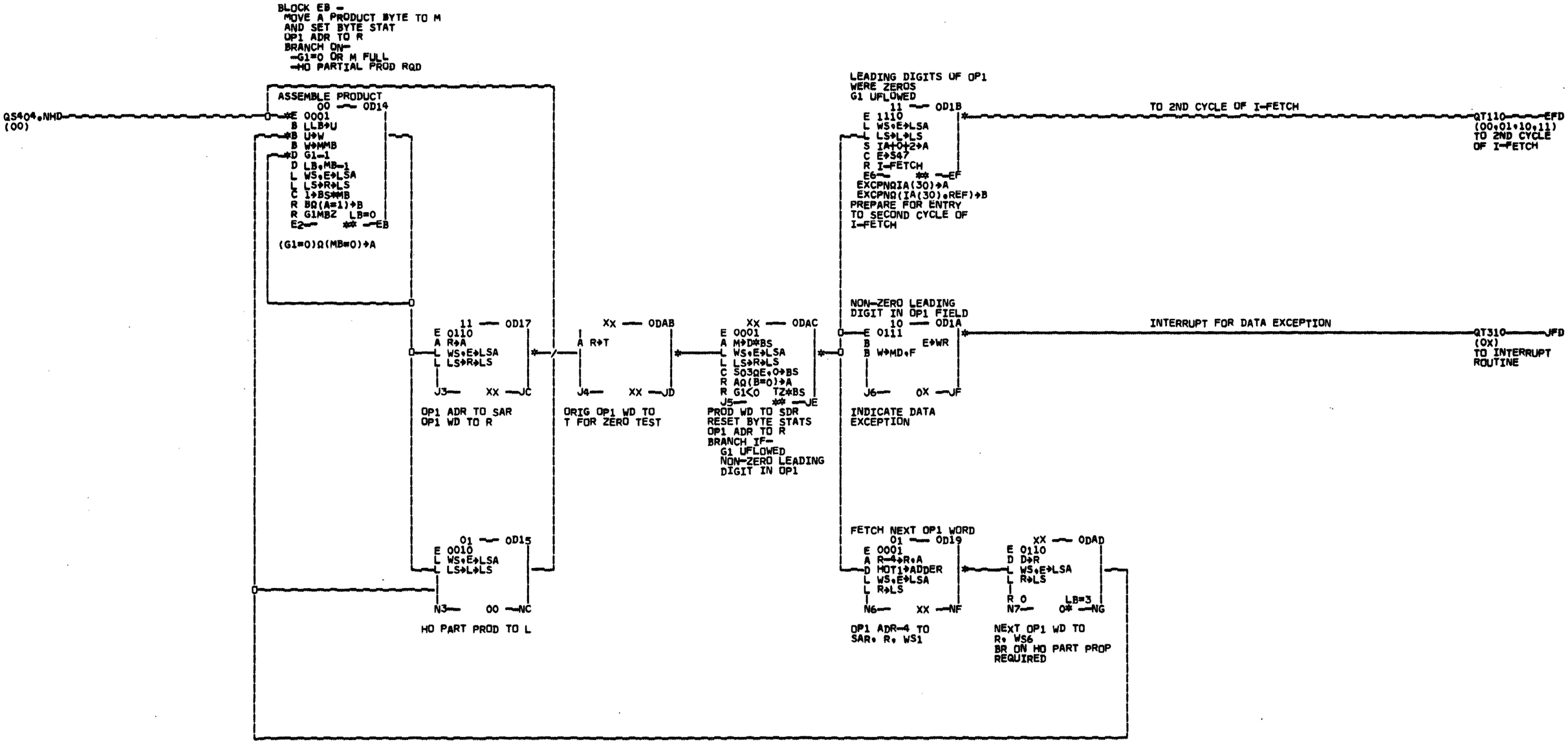
MPLR DIG = 0
SHIFT HO PART PROD RT 4, SPILL LO DIG TO F, TAKE NEW HO PART PROD TO L, WS2
11 — 0D0F
E 0010
A L→L
A SR4→F
L WS→E→LSA
L L→LS
S3— XX —SC

SHIFT LO PART PROD RIGHT 4 ENTER DIG FROM F TO HO POSITION SPILL LO DIG TO F
MPLR PROD WORD TO L
XX — 0D9F
E 0110
A R→R
A F→SR4→F
L WS→E→LSA
L LS→L→LS
S4— XX —SD

255449	09/20/65			MACH	C2050	DATE 10/01/65	SHEET 1 QS402
		254760	12/21/64	NAME		LOG 902	VERSION
		254765	01/13/65	MODE	MANUAL		
		255094	03/01/65	P.N.	536482?	MULT.DIV DEC— MPCN DBLING, SET	
		255103	03/31/65	IBM CORP.	SND	QUOT SIGN, SEL MPLR DIGIT	



254760	12/21+64	MACH	C2050	DATE 10/01+65	SHEET 1 QS404
254770	02/01+65	NAME		LOG 902	VERSION
255449	09/20+65	MODE	MANUAL	MULTIPLY DEC - SINGLE AND DBL WD MPY LOOPS. PROD STORE	
		P.N.	5364823		
		IBM CORP.	SND		



254760	12/21+64	MACH	C2050	DATE	10/01+65	SHEET	1 QS406
254765	01/13+65	NAME		LOG	902	VERSION	
255449	09/20+65	MODE	MANUAL				
		P.N.	5364824		MULTIPLY DECIMAL (MP)		
		IBM CORP.	SDD		FINAL PRODUCT ASSEMBLY		

QS406

QS010•NIE (0010XX) FROM I-FETCH

0010XX — C08
E 0000
D D→L•M
L WS•E→LSA
L L→LS
A1— XX —AA
PACK• 1ST OP 2 WD TO L• M FROM SDR• SET MVR FCTN REG TO 000 (CROSS-GT)

XX — 07B5
B LLB→U
B VL→WL ?→WR
B W→MMB
L WS2→LSA
L LS→L→LS
A2— XX —AB
OP 2 ADR TO L• CROSS LF HALF OF 1ST BYTE TO RT HF OF BYTE IN M•

XX — 07B6
E 0010
A M-LWA→T
D HOT1→ADDER
C E*S03
A3— XX —AC
11→XG•Y(30-31)
SUBTRACT OP 2 ADR FROM OP 1 ADR FOR ADR COMP• (R-L)• TN ON STAT 2 (R L STAT TO L)

XX — 07B7
E 0000
L WS•E→LSA
L LS→L→LS
R T13=0
A4— X✲ —AD
FETCH 1ST OP 2 WD

XO — 079C
B LLB→U MMB→V
B ?→WL VR→WR
B W→MMB
D G1•2-1
D LB•MB-1
C 1→BS✲MB
R G1MBZ G2LBZ
A5— ✲✲ —AE
(G1=0)Ω(MB=0)→A
(G2=0)Ω(LB=0)→B
XFER OP 2 DIGIT TO M LF HF BYTE•

QS502 — ADE (X1) OVERLAP CASE

QS502•ECE (X0•X1)

00 — 78C
B LLB→U
B VL→WL UR→WR
B W→MMB
D G2-1
D LB-1
R G2LBZ
E1— X✲ —EA
(G2=0)Ω(LB=0)→B

X1 — 07A1
E 0010
A R-4→L
B MMB→V
B VR→WR
B WR→F
D HOT1→ADDER
L WS2→LSA
L L→LS
C S03ΩE
R G2<0
E2— X✲ —EB
UPDATE OP2 ADR R/L STAT TO LEFT SAVE LAST DIGIT IN F

X0 — 07A2
A M-LWA→T
D HOT1→ADDER
L WS2→LSA
L LS→R→LS
E3— XX —EC
11→XG•Y(30-31)
SUB OP ADDRESSES

XX — 7B4
E 0001
A R→A
C S03•¬E
R AΩ(B=0)→A
R S3 T13=0
E5— ✲✲ —EE
OP2 ADR TO SAR• BR ON OP 2 NEEDED AND WD ADDRESSES EQUAL•

TN OFF STAT 3•

QS502 — JFE (11) MOVE WITH OFFSET

00 — 788
B LLB→U
B VL→WL UR→WR
B W→MMB
D G2-1
D LB-1
R G2=0
E7— X✲ —EG
OP 2 DIGIT TO M• DECR G2• LB•

01 — 078D
E 0010
A R-4→L
D HOT1→ADDER
L WS2→LSA
L L→LS
C S03•¬E
R G2<0
J1— X✲ —JA
OP 1 ADR TO L FROM R• OP 2 ADR TO R• TN OFF STAT 2 (R+L STAT RIGHT)•

X0 — 07A0
B LLB→U MMB→V
B ?→WL VR→WR
B W→MMB
D G1•2-1
D LB•MB-1
C 1→BS✲MB
R G1MBZ G2LBZ
J2— ✲✲ —JB
(G1=0)Ω(MB=0)→A
(G2=0)Ω(LB=0)→B
OP 2 DIGIT TO M LEFT BYTE• DECR G1• G2• LB• MB• SET BYTE STAT•

X1 — 07A3
E 0001
D 0→LB•MD
C S03ΩE
R 0 S2
J3— 0✲ —JC
OP 1 ADR TO R FROM L• TN ON STAT 3 (NEED OP 2)

00 — 0790
A 0→L
B W→MMB
D G1-1
D MB-1
C 1→BS✲MB
R G1MBZ 0
J4— ✲0 —JD
(G1=0)Ω(MB=0)→A
ZERO BYTE TO M•

10 — 0786
D D→L
R S1 S2
J6— ✲✲ —JF
OP 2 TO L FROM SDR• BR ON STAT 1 (MOVE WITH OFFSET INST) AND STAT 2 (R/L TO LEFT)•

01 — 0789
B LLB→U MMB→V
B ?→WL VR→WR
B W→MMB
D G1•2-1
D LB•MB-1
C 1→BS✲MB
R G1MBZ G2=0
J7— ✲✲ —JG
(G1=0)Ω(MB=0)→A
OP 2 DIGIT TO M LEFT HALF BYTE•

10 — 078E
E 0001
A H→A
C S03•¬E
R G1<0
N1— ✲X —NA
OP 1 ADR TO SAR•

0X — 0719
A H-4→H
D HOT1→ADDER
R S1
N2— X✲ —NB
INCR OP 1 ADR (R + 4 TO R)• BR ON STAT 1 (MOVE WITH OFFSET)

X0 — 079E
E 0000
A M→D✲BS
C E→BS
R 0 S3
N3— 0✲ —NC
M TO SDR PER BYTE STATS• RESET BS• BR ON STAT 3 (NEED OP 2)•

01 — 0791
A 0→L
B MMB→V
B UL→WL VR→WR
B W→MMB
D G1-1
D MB-1
C 1→BS✲MB
R G1MBZ 0
N4— ✲0 —ND
(G1=0)Ω(MB=0)→A
INSERT ZERO DIGIT INTO LEFT HALF BYTE IN M• DECR

10 — 0792
A H→A
R G1<0
N5— ✲X —NE
OP 1 ADR TO SAR•

01 — 0785
E 0000
A M→D✲BS
C E→BS
N6— XX —NF
M TO SDR PER BYTE STATS

XX — 07B3
A R→A
N7— 11 —NG
OP2 ADR TO SAR

11 — 0787
D D→L•M
R S1 S2
N8— ✲✲ —NH
OP 1 WD TO L• M FROM SDR•

QS502 — NHD (00•01•11)

QS502•JCF (0X•1X)

OP 1 ADR TO SAR• TN ON STAT 3 (NEED OP 2)•

11 — 078F
E 0001
A H→A
C S03ΩE
R G1<0
S1— ✲X —SA

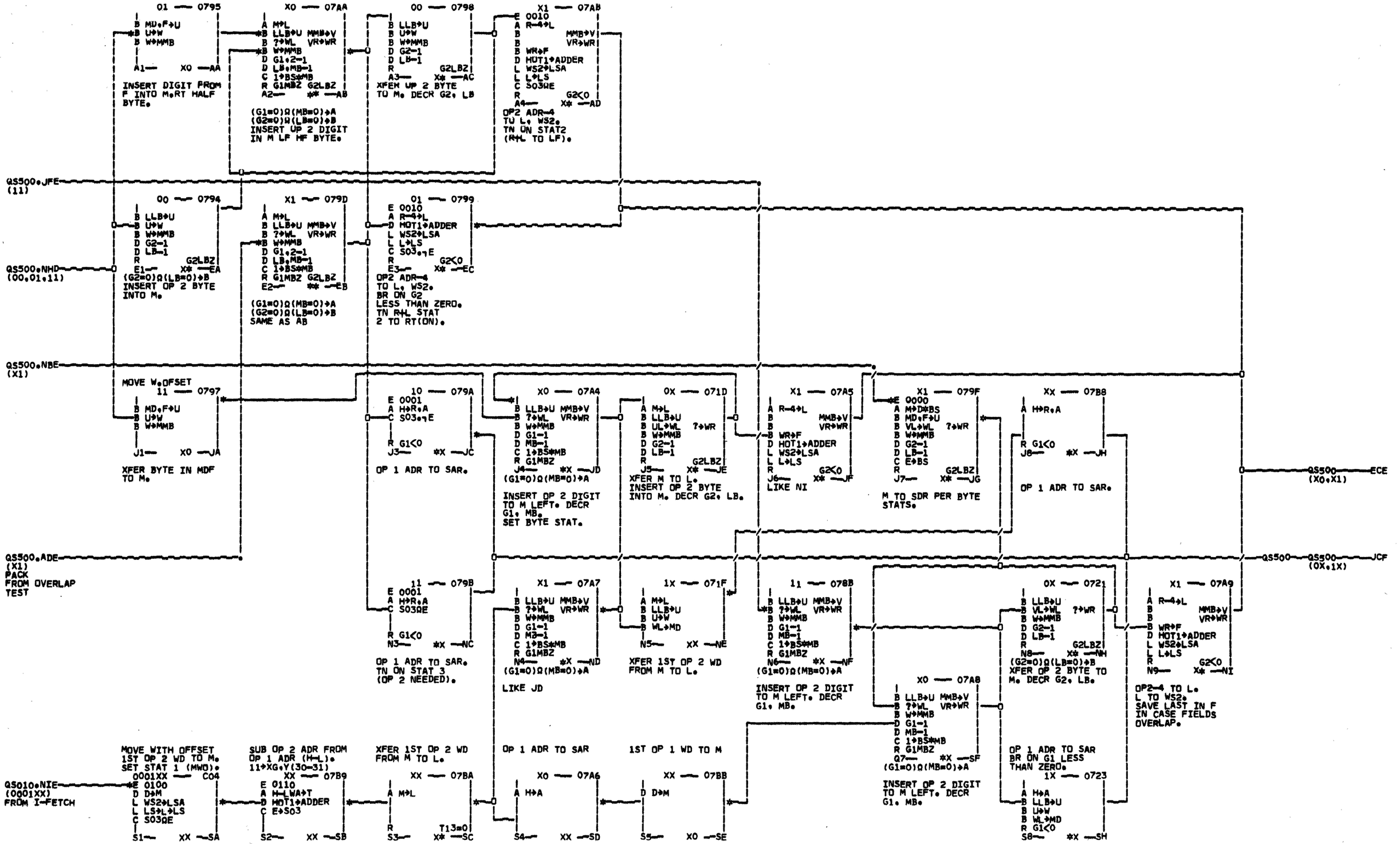
M TO SDR PER BYTE STATS• END•

1X — 071B
A M→D✲BS
S2— 1X —SB

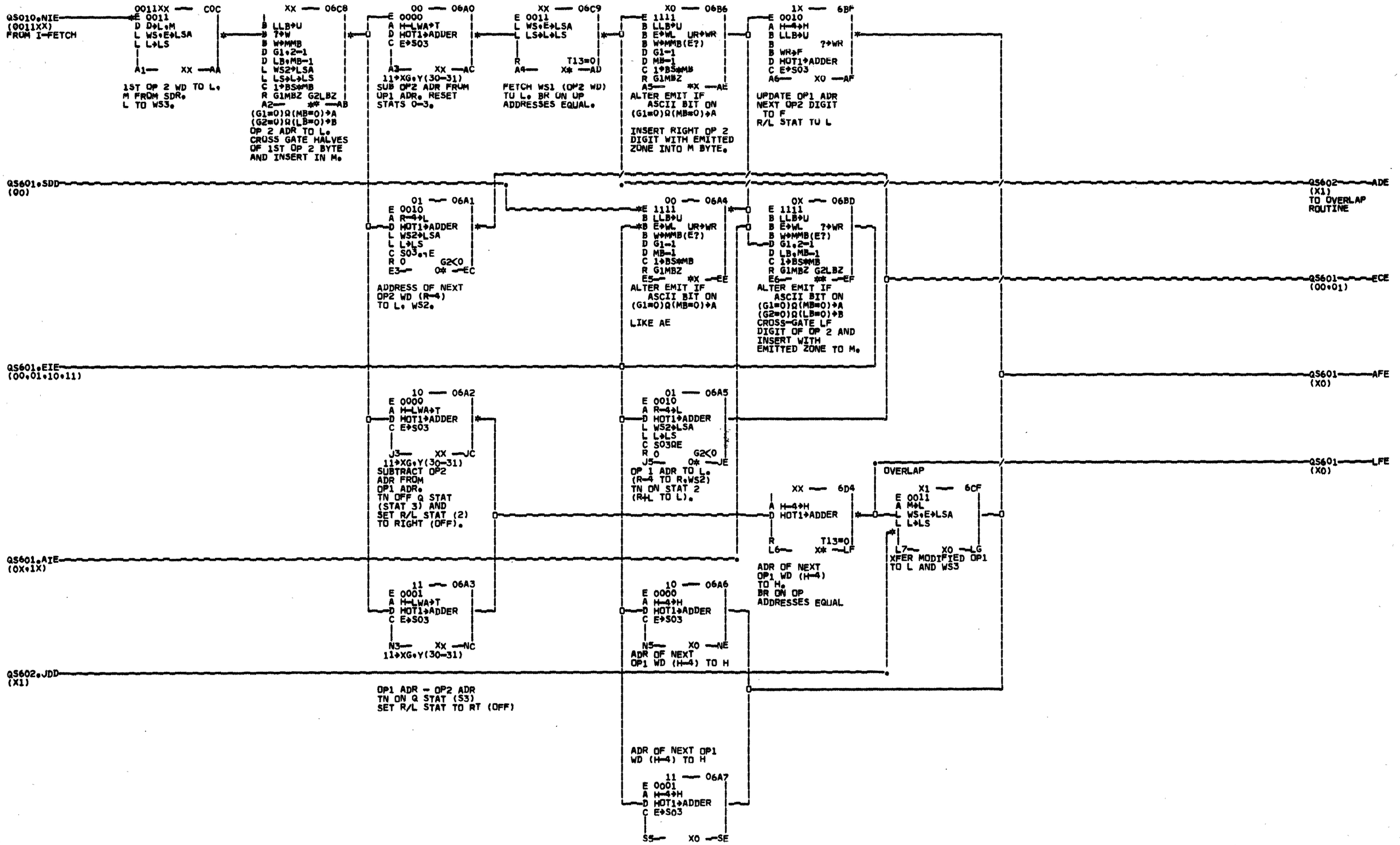
QS502 — NBE (X1) MOVE WITH OFFSET

QT100 — SBE (1X) TO I-FETCH

254765	01•13•65	MACH	C2050	DATE 10•01•65	SHEET 1 QS500
255094	03•01•65	NAME		LOG 902	VERSION
255449	09•20•65	MODE	MANUAL	PACK• MOVE WITH OFFSET	
		P•N•	5364825		
		IBM CORP•	SDD		



254760	12-21/64	MACH	C2050	DATE 10-01/65	SHEET 1 QS502
255449	09-20/65	NAME		LOG 902	VERSION
		MODE	MANUAL	PACK. MOVE WITH OFFSET	
		P.N.	5364826		
		IBM CORP.	SDD		



254760	12-21/64	MACH	C2050	DATE	10-01/65	SHEET	1 QS600
255437	04-26/65	NAME		LOG	902	VERSION	
255449	09-20/65	MODE	MANUAL				
		P.N.	5364827		UNPACK		
		IBM CORP.	SDD		NON-OVERLAPPED FIELDS		

QS600

QS600•AFE (X0)

QS602•NDE (X0)

QS600•LFE (X0)

QS602•EDD (00•01)

QS600•ECE (00•01)

1X — 06D2
A M→D*BS
A2— 1X —AB
M TO SDR PER BYTE STATS• END•

X0 — 06C2
E 0000
A M→D*BS
C E→BS
R S2 T13=0
A3— ** —AC
M TO SDR PER BYTE STATS• RESET BYTE STATS• BR ON R L STAT (2) TO L (ON) AND UP ADDRESSES EQUAL•

X0 — 06CE
A H/4→A
L WS2→LSA
L LS•L→LS
R G1<0
E1— *X —EA
ADR OF OP1 WORD (H/4) TO SAR•

0X — 06D0
E 0011
A H—LWA→T
D HOT1→ADDER
L WS•E→LSA
L LS•L→LS
R S3
E2— X* —EB
11→XG•Y(30—31) SUB OP2 ADR FROM OP 1 ADR• (R—L)• FETCH OP 2 WORD TO L FROM WS3•

X1 — 06C3
E 0000
A M→D*BS
C E→BS
E3— XX —EC
M TO SDR PER BYTE STATS• RESET BYTE STATS•

XX — 06CD
A R—4→L
D HOT1→ADDER
L WS2→LSA
R 0 G2<0
E4— 0* —ED
OP 1 ADR TO L FROM R• FETCH OP2 ADR TO R• BR ON G2 LESS THAN ZERO UPDATE OP2 ADR• R—4 TO R•WS2•

01 — 06AD
E 1111
B E→WL
B W→MMB(E?)
D G1—1
D MB—1
C 1→BS*MB
R G1MBZ 1
E5— *1 —EE
ALTER EMIT IF ASCII BIT ON (G1=0)Ω(MB=0)→A
INSERT ZERO BYTE TO M• DECR G1• MB XFER OP 1 ADR TO R FROM L•

11 — 06AF
E 0001
A M—4→M
D HOT1→ADDER
C SO3ΩE
E6— XO —EF

00 — 06AC
A H—LWA→T
D HOT1→ADDER
L WS2→LSA
L LS→R→LS
J1— XX —JA
11→XG•Y(30—31) COMPARE OP1 AND OP2 ADDRESSES (H—L)•
OP2 ADR (WS2) TO R•

XX — 06CA
A R→A
R T13=0
J3— X* —JC
ADR OF NEXT OP2 WD TO SAR

X1 — 06C1
E 0000
A M→D*BS
C E→BS
J4— XX —JD
M TO SDR PER BYTE STATS• RESET B•S• OP 1 AND OP2 OVLAP

XX — 06CB
A R→A
J5— XX —JE
OP 1 ADR TO SAR TO REFETCH WD JUST MODIFIED•

XX — 06CC
E 0011
D D→L•M
L WS•E→LSA
L L→LS
N5— Xx —NE
OP 1 WD TO L• M• WS1 FROM SDR•

NON—OVERLAP FIELDS OP 2 WD TO L• WS1

XO — 06CO
E 0011
D D→L
L WS•E→LSA
L L→LS
S4— 00 —SD

QS600 (00) — SDD

00 — 06A8
E 1111
B LLB→U
B E→WL UR→WR
B W→MMB(E?)
D G1—1
D MB—1
C 1→BS*MB
R G1MBZ
A9— *X —AI
ALTER EMIT IF ASCII BIT ON (G1=0)Ω(MB=0)→A
INSERT RT OP 2 DIGIT WITH EMITTED ZONE TO M• DECR G1• MB•

QS600 (0X•1X) — AIE

QT100 (1X) — ABF
TO I—FETCH

10 — 6AA
E 1111
B MD•F→U
B E→WL UR→WR
B W→MMB(E?)
D G1•2—1
D LB•MB—1
C 1→BS*MB
R G1MBZ G2LBZ
E9— ** —EI
ALTER EMIT IF ASCII BIT ON (G1=0)Ω(MB=0)→A (G2=0)Ω(LB=0)→B
INSERT OP2 DIGIT WITH EMITTED ZONE IN M

QS600 (00•01•10•11) — EIE

01 — 06A9
A L→M
J9— XX —JI
XFER L TO M•

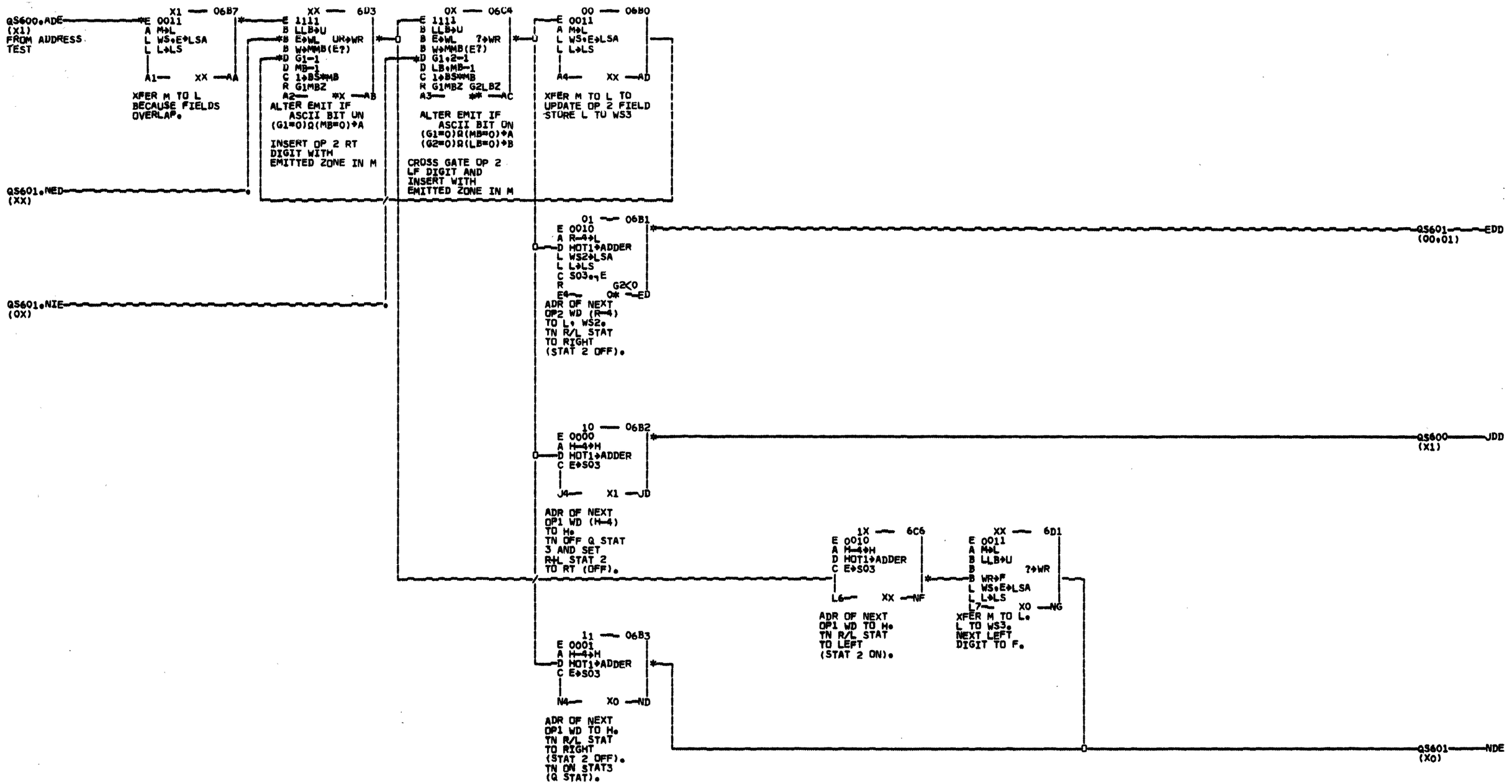
QS602 (XX) — NED

11 — 06AB
A L→M
N9— 0X —NI
XFER L TO M•

QS602 (0X) — NIE

254760	12/21/64	MACH	C2050	DATE 10•01/65	SHEET 1 QS601
254770	02/01/65	NAME		LOG 902	VERSION
255437	04/26/65	MODE	MANUAL	UNPACK — OP1 STORAGE	
255449	09/20/65	P•N•	5364828	OP2 FETCH• ZERO INSERTION	
		IBM CORP•	SDD		

QS601



254760	12/21+64	MACH	C2050	DATE 10/01+65	SHEET 1 QS602
285437	04/26+65	NAME		LOG 902	VERSION
255449	09/20+65	MODE	MANUAL		
		P.N.	5364829	UNPACK	
		IBM CORP.	SMD	OVERLAPPED FIELDS	

QS706•JFE
(X0)

QS706•JBD
(00•01•11)

QT310 —— AED
(0X)
TO INTERRUPT
ROUTINE

QS010•NIE
(1000XX)
FROM I-FETCH

OP2 WD TO L
L TO WS3
1000XX — C20
E 0011
D D→L
L WS•E→LSA
L L→LS
C1— XX —AA

GENERAL NOTES
OP2 BYTES ARE ASSEMBLED TO BE BYTE ALIGNED WITH THE OP1 FIELD• IF L2 GRT THAN G1 GO TO THE DECIMAL ADD PROGRAM TO CHK EXCESS OP2 BYTES FOR NON-ZERO• ALSO BR TO ADD PROGRAM AT COMPLETION TO STORE SIGN AND SET CR•

XX — 07BE
A R→L
B LLB→U
B ?→WR
B WR→F
L WS1→LSA
L LS→R→LS
C WR→IVD
C L•RSGNS
C2— XX —AB

IF —•1→LSGN•
¬RSGN
IF /•0→LSGN
IF NOT SGN•TRAP
ADR OF RIGHTMOST BYTE OF OP2 FIELD TO L FROM R•
INSERT 1ST OP2 DIGIT INTO F•

XX — 7CA
E 1010
A H—L→T
A BC8
D HOT1→ADDER
L WS•E→LSA
L LS→L→LS
C 0→RSGNS
R (CAR)
C3— X* —AC

SAVE CAR FROM 8

COMPARE RIGHTMOST BYTE ADDRESSES OF OP1 AND OP2 (H—L)
FETCH ADR OF LEFTMOST BYTE OF OP2 TO L•

X0 — 7C0
E 0011
A H—L→T
A BC8
D HOT1→ADDER
L WS•E→LSA
L LS→L→LS
C S47QE
R (CAR)
C5— X* —AD

SAVE CAR FROM 8

COMPARE RIGHTMOST BYTE ADR OF OP1 WITH LEFTMOST BYTE ADR OF OP2• FETCH OP2 WD TO L•

X1 — 07C3
E 0111
B E→WK
B W→MD•F
C6— 0X —AE

GO TO INVALID DATA INTERRUPT BECAUSE OF OVLAP OF FIELDS

00 — 7BC
B LLB→U
B U→W
B W→MMB
D G1•2—1
D LB•MB—1
L WS1→LSA
L LS→R→LS
C W→IVD
C 1→BS*MB
R BQ(A=1)→B
R G1MBZ G2LBZ
C7— ** —AF

(G1=0)Ω(MB=0)→A
(G2=0)Ω(LB=0)→B
ASSEMBLE OP2 BYTE IN M•

X1 — 7C5
B LLB→U
B U→W
B W→MMB
D G1•2—1
D LB•MB—1
L WS1→LSA
L LS→R→LS
C W→IVD
C 1→BS*MB
R BQ(A=1)→B
R G1MBZ G2LBZ
C8— ** —AG

(G1=0) MB=0)→A

LIKE AF

X1 — D97
D D→L
C9— 00 —JG

OP2 WORD TO L

01 — DC5
E 1100
A H→A
B MMB→V
B VL→WL E→WR
B W→MMB(E?)
D 3→MB
R TZ*BS
G1— X* —EH

ALTER EMIT IF ASCII BIT ON
OP1 ADR TO SAR•
PLUS SIGN TO M•

X0 — 7C6
E 0100
A H—4→H
D HOT1→ADDER
C E→S47
R BQ(A=1)→B
R G1<0 G2<0
G2— ** —JH

UPDATE OP1 ADR•
TN OFF 1ST TIME STAT 6•
TN ON NON-ZERO STAT•

00 — 7C8
E 0000
A M→D*BS
C E→BS
R 0 LB=3
G3— 0* —NI

M TO SDR PER BYTE STATS•

X1 — 7C1
E 0011
L WS•E→LSA
L LS→L→LS
C S47QE
G5— X0 —ED

OP2 WORD TO L TO WS3

X0 — 07C2
B LLB→U
B UL→WL VR→WR
B W→MMB
D G1•2—1
D LB•MB—1
C W→IVD
C 1→BS*MB
R BQ(A=1)→B
R G1MBZ G2LBZ
G6— ** —EE

(G1=0)Ω(MB=0)→A
(G2=0)Ω(LB=0)→B
ASSEMBLE OP2 BYTE IN M (ZERO SIGN DIGIT)•

01 — 07BD
A 0→L
L WS2→LSA
L LS→R→LS
R G2<0
G7— X* —EF

ZEROS TO L•
OP2 ADR TO R•

X0 — 7C4
A R—4→R•A
D HOT1→ADDER
L WS2→LSA
L R→LS
R S6
G8— X* —EG

R—4 TO R AND SAR•
R TO WS2•

X0 — D96
D D→L
R G2<=LB 1
G9— *1 —NG

BRANCH IGNORES G2 SIGN
OP2 WORD TO L

QS706 —— NGE
(01•11)

10 — DC6
A H→A
D 3→MB
R TZ*BS
L1— X* —AI

OP1 ADR TO SAR•
RESET MB=3•

X1 — 7C7
E 0010
A H—4→H
D HOT1→ADDER
C S47•¬E
R BQ(A=1)→B
R G1<0 G2<0
L2— ** —NH

UPDATE OP1 ADR
TN OFF 1ST TIME STAT 6•

01 — 7C9
E 0000
A M→D*BS
C E→BS
L3— 01 —JI

OP1 TO SDR
RESET BYTE STATS

11 — 7BF
A M→T
B R3→U
B U→W
B W67→MB
R AQ(B=0)→A
R LSGNS S6
L7— ** —AH

ASSEMBLED FIELD IN M TO LATCHES FOR ZERO TEST•
FIRST OP1 BYTE ADR TO MB•
BR ON L SIGN STAT•
FIRST TIME STAT•

11 — DC7
E 1101
A H→A
B MMB→V
B VL→WL E→WR
B W→MMB(E?)
D 3→MB
R TZ*BS
Q1— X* —EI

ALTER EMIT IF ASCII BIT ON
OP1 ADR TO SAR•
SET MB=3•
EMIT MINUS TO M•

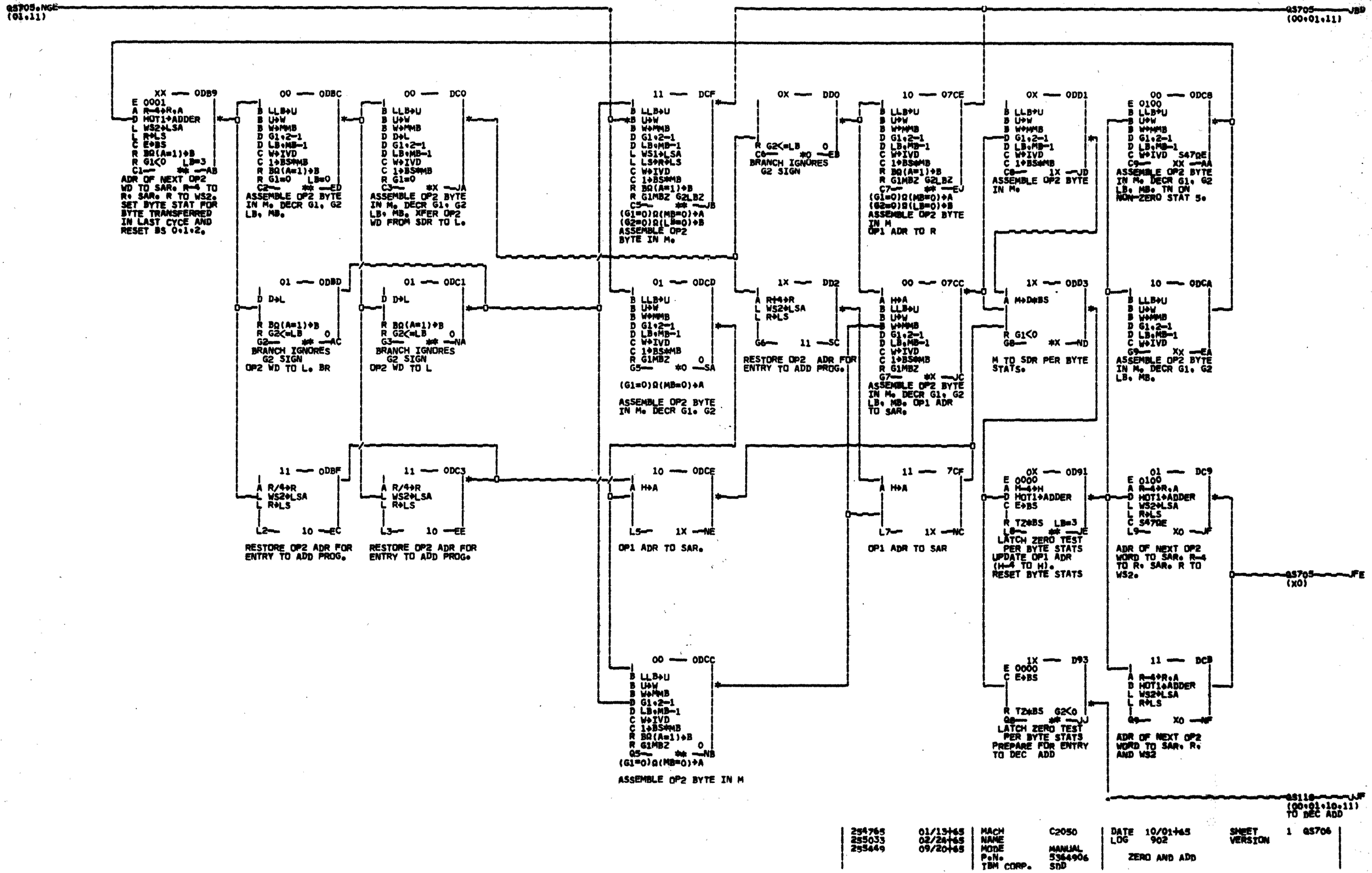
11 — 7CB
E 0000
A M→D*BS
C E→BS
R 1 G2<0
Q3— 1* —NA

STORE OP1 DATA
GO TO DEC ADD

QS118 —— NAE
(10•11)
TO DECIMAL ADD OVERFLOW ROUTINE

254770	02/01/65	MACH	C2050	DATE 10/01/65	SHEET 1 QS705
255033	02/24/65	NAME		LOG 902	VERSION
255449	09/20/65	MODE	MANUAL	ZERO AND ADD	
		P•N•	5364905		
		IBM CORP•	SMD		

QS705


QS705.NGE (01.11)
QS705 JBD (00.01.11)
XX — ODB9
E 0001
A R-4→R.A
D HOT1→ADDER
L WS2→LSA
L R→LS
C E→BS
R BQ(A=1)→B
R G1<0 LB=3
C1— ** —AB
ADR OF NEXT OP2 WD TO SAR. R-4 TO R. SAR. R TO WS2. SET BYTE STAT FOR BYTE TRANSFERRED IN LAST CYCE AND RESET BS 0.1.2.
00 — ODBC
B LLB→U
B U→W
B W→MMB
D G1.2-1
D LB.MB-1
C W→IVD
C 1→BS*MB
R BQ(A=1)→B
R G1=0 LB=0
C2— ** —ED
ASSEMBLE OP2 BYTE IN M. DECR G1. G2 LB. MB.
00 — DC0
B LLB→U
B U→W
B W→MMB
D D→L
D G1.2-1
D LB.MB-1
C W→IVD
C 1→BS*MB
R G1=0
C3— *X —JA
ASSEMBLE OP2 BYTE IN M. DECR G1. G2 LB. MB. XFER OP2 WD FROM SDR TO L.
11 — DCF
B LLB→U
B U→W
B W→MMB
D G1.2-1
D LB.MB-1
L WS1→LSA
L LS→R+LS
C W→IVD
C 1→BS*MB
R BQ(A=1)→B
R G1MBZ G2LBZ
C5— ** —JB
(G1=0)Ω(MB=0)→A
(G2=0)Ω(LB=0)→B
ASSEMBLE OP2 BYTE IN M.
0X — DD0
R G2<=LB 0
C6— *0 —EB
BRANCH IGNORES G2 SIGN
10 — 07CE
B LLB→U
B U→W
B W→MMB
D G1.2-1
D LB.MB-1
C W→IVD
C 1→BS*MB
R BQ(A=1)→B
R G1MBZ G2LBZ
C7— ** —EJ
(G1=0)Ω(MB=0)→A
(G2=0)Ω(LB=0)→B
ASSEMBLE OP2 BYTE IN M
OP1 ADR TO R
0X — DD1
B LLB→U
B U→W
B W→MMB
D G1.2-1
D LB.MB-1
C W→IVD
C 1→BS*MB
C8— 1X —JD
ASSEMBLE OP2 BYTE IN M.
00 — DDC8
E 0100
B LLB→U
B U→W
B W→MMB
D G1.2-1
D LB.MB-1
C W→IVD S470E
C9— XX —AA
ASSEMBLE OP2 BYTE IN M. DECR G1. G2 LB. MB. TN ON NON-ZERO STAT 5.
01 — ODBD
D D→L
R BQ(A=1)→B
R G2<=LB 0
G2— ** —AC
BRANCH IGNORES G2 SIGN
OP2 WD TO L. BR
01 — ODC1
D D→L
R BQ(A=1)→B
R G2<=LB 0
G3— ** —NA
BRANCH IGNORES G2 SIGN
OP2 WD TO L
01 — ODCD
B LLB→U
B U→W
B W→MMB
D G1.2-1
D LB.MB-1
C W→IVD
C 1→BS*MB
R G1MBZ 0
G5— *0 —SA
(G1=0)Ω(MB=0)→A
ASSEMBLE OP2 BYTE IN M. DECR G1. G2
1X — DD2
A R+4→R
L WS2→LSA
L R→LS
G6— 11 —SC
RESTORE OP2 ADR FOR ENTRY TO ADD PROG.
00 — 07CC
A M→A
B LLB→U
B U→W
B W→MMB
D G1.2-1
D LB.MB-1
C W→IVD
C 1→BS*MB
R G1MBZ
G7— *X —JC
ASSEMBLE OP2 BYTE IN M. DECR G1. G2 LB. MB. OP1 ADR TO SAR.
1X — DD3
A M→D*BS
R G1<0
G8— *X —ND
M TO SDR PER BYTE STATS.
10 — ODCA
B LLB→U
B U→W
B W→MMB
D G1.2-1
D LB.MB-1
C W→IVD
G9— XX —EA
ASSEMBLE OP2 BYTE IN M. DECR G1. G2 LB. MB.
11 — ODBF
A R/4→R
L WS2→LSA
L R→LS
L2— 10 —EC
RESTORE OP2 ADR FOR ENTRY TO ADD PROG.
11 — ODC3
A R/4→R
L WS2→LSA
L R→LS
L3— 10 —EE
RESTORE OP2 ADR FOR ENTRY TO ADD PROG.
10 — ODCE
A H→A
L5— 1X —NE
OP1 ADR TO SAR.
11 — 7CF
A H→A
L7— 1X —NC
OP1 ADR TO SAR
0X — OD91
E 0000
A H-4→H
D HOT1→ADDER
C E→BS
R TZ*BS LB=3
L8— ** —JE
LATCH ZERO TEST PER BYTE STATS UPDATE OP1 ADR (H-4 TO H). RESET BYTE STATS
01 — DC9
E 0100
A R-4→R.A
D HOT1→ADDER
L WS2→LSA
L R→LS
C S470E
L9— X0 —JF
ADR OF NEXT OP2 WORD TO SAR. R-4 TO R. SAR. R TO WS2.
QS705 JFE (X0)
00 — ODCC
B LLB→U
B U→W
B W→MMB
D G1.2-1
D LB.MB-1
C W→IVD
C 1→BS*MB
R BQ(A=1)→B
R G1MBZ 0
Q5— ** —NB
(G1=0)Ω(MB=0)→A
ASSEMBLE OP2 BYTE IN M
1X — D93
E 0000
C E→BS
R TZ*BS G2<0
Q8— ** —JJ
LATCH ZERO TEST PER BYTE STATS PREPARE FOR ENTRY TO DEC ADD
11 — DCB
A R-4→R.A
D HOT1→ADDER
L WS2→LSA
L R→LS
Q9— X0 —NF
ADR OF NEXT OP2 WORD TO SAR. R. AND WS2
QS110 JJF (00.01.10.11) TO DEC ADD
254765 01/13/65
255033 02/24/65
255449 09/20/65
MACH C2050
NAME
MODE MANUAL
P.N. 5364906
IBM CORP. SMD
DATE 10/01/65
LOG 902
SHEET 1 QS706
VERSION
ZERO AND ADD