**IBM 1620 Jr.**

**Diagnostic Manual**

**­­Version 0.9**

**Introduction**

The IBM 1620 was always shipped with a full set of diagnostic programs. These programs were typically used by the IBM Customer Engineer to check the correct operation of the machine and to diagnose / isolate / exercise failures when they occurred. They cover every aspect of the machine from console switches & lights, core memory, and instruction execution to operation of peripherals. A properly maintained IBM 1620 rarely failed, but it did happen. The diagnostics were also used when the computer was first installed to verify that it was fully operational.

While the hardware of the IBM 1620 Jr. is several orders of magnitude more reliable than the original 1960’s IBM 1620, the diagnostic programs are still very useful to validate the simulation software in the main unit, the typewriter, and the card reader/punch device. Whenever changes are made to the simulators, the diagnostics should be run.

The Computer History Museum has in its collection, most of the manuals for the IBM 1620 Model 1 Diagnostics and a few paper tape images of the basic ones. Fortunately, for the diagnostics without binary images, the manuals include a simplified machine language listing of each program. One notable exception is the CU06/DT106 floating point diagnostic which is completely missing from the collection. As a replacement, a volunteer, John M. Bohn, wrote an SPS program (FP01) which thoroughly tests all of the 1620’s floating point instructions. In addition, the CU02 Error Check diagnostic only tests the memory addressing register (MAR) subset of the IBM 1620’s “Check Stop” conditions and depends on an internal maintenance switch, CE #9, which changed the computer’s behavior. For the IBM 1620 Jr. it was replaced with a new, comprehensive error checking program (CS01).

IBM supplied all of the diagnostics on paper tape, punched cards, and the customer engineering disk pack. For the IBM 1620 Jr. all of the diagnostic programs have been “built into” the simulator for ease of use. When the IBM 1620 Jr. is powered off, pressing the RESET button loads into simulated core memory one of the diagnostic programs based on the setting of the PROGRAM SWITCHES. When the machine is turned on, pressing the START button will run the program.

Here are the programs available:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PS 1** | **PS 2** | **PS 3** | **PS 4** | **Name** | **Description** |
| off | off | off | off | PowerOf2 | Power of 2 Calculator Demo Program |
| off | off | off | on | CU00 | Console Diagnostic |
| off | off | on | off | CU01 | General Op Codes Diagnostic |
| off | off | on | on | CU05 | Special Instructions Diagnostic |
| off | on | off | off | FP01 | Floating Point Diagnostic |
| off | on | off | on | CU03 | Indirect Addressing Diagnostic |
| off | on | on | off | CS01 | Check Stops Diagnostic |
| off | on | on | on | DX05L | Core Storage 20K Low Diagnostic |
| on | off | off | off | DX05H | Core Storage 20K High Diagnostic |
| on | off | off | on | CU04 | Additional Core Diagnostic |
| on | off | on | off | DX03 | Typewriter Diagnostic |
| on | off | on | on | IO02 | Card I/O Diagnostic |
| on | on | off | off | IO03 | Card I/O Reliability Diagnostic |

The following sections describe all of the diagnostic programs and how to run them.

**Lights and Switches Tests**

**Lights and Switches Tests**

Built into the IBM 1620 Jr. simulator is a way to manually test all of the lights and switches on the front panel. This is the IBM 1620 Jr.’s version of a “Lamp Test” switch found on other computers. The IBM 1620 did not have this feature, but it is a useful capability for maintenance, so it was added to the IBM 1620 Jr. Unlike the IBM 1620 diagnostic test programs, these tests do not involve executing any IBM 1620 code.

There are four different tests available. They are activated when the machine is powered off, the left-most (unlabeled) toggle switch is on, and one PROGRAM SWITCH is turned on.

PROGRAM SWITCH 1: This is the actual “Lights and Switches” test. It begins with all wired lights turned on. This tests all of the lights. As each switch is turned on or button is pressed, one or more lights are turned off. For the toggle switches [not the PROGRAM SWITCHes], the light(s) above it are turned off when the switch is on. The left-most toggle switches and the PROGRAM SWITCHes are checked by selecting the various tests. The button / light correspondence is:

RESET: POWER ON, POWER READY, THERMAL lights

DISPLAY MAR: Pressing this button tests the large MEMORY ADDRESS REGISTER DISPLAY SELECTOR rotary switch. The HUNDREDS row of the MEMORY ADDRESS REGISTER displays a binary value representing the position of the knob as it is rotated, where: OR-1 = 0, OR-2 = 1, OR-3 = 2, …, IR-2 = 11

SAVE: SAVE light

INSERT: INSERT light

RELEASE: PUNCH NO FEED, READER NO FEED lights

START: AUTOMATIC light

STOP / SIE: MANUAL light

INSTANT STOP / SCE: CHECK STOP light

PROGRAM SWITCH 2: This test displays the full LED brightness range. Each column of lights in the upper front panel displays a different brightness level from off in the left-most column to fully on in the right-most column. The columns in-between are 36 of the 254 intermediate brightness levels available.

PROGRAM SWITCH 3: This “snake” demo moves multiple lights across the upper and lower front panels. The two lights in the upper panels move randomly. The two lights in the lower panels move back and forth across the available wired lights. The speed of animation is controlled by the MEMORY ADDRESS REGISTER DISPLAY SELECTOR rotary switch, where PR-3 is the fastest and OR-4 is the slowest.

PROGRAM SWITCH 4: This one is just for fun. It lights “IBM 1620” in the upper front panels.

One thing to note about the IBM 1620 Jr.’s front panel – not all of the lights are wired and therefore they cannot be turned on. This is due to a combination of design decisions made when Jr. was built. None of the inactive lights were used on a real IBM 1620 Model 1 Level F machine, so it does not affect correct operation of the IBM 1620 Jr. However, the unlit lights are noticeable in the “lights and switches” tests. The inactive lights are:

INSTRUCTION AND EXECUTE CYCLE: E-24, E-25

CONTROL GATES: SET 00080, VRC GATE, END COMPR

INPUT-OUTPUT: *row 1 light 3, row 3 light 1, row 5 light 1*

DIGIT REGISTER: *row 1 light 2, row 2 light 2*

Lower Left Panel: *row 1 light 1, row 1 light 5, row 2 light 1, row 3 light 1*, MBR-E CHK, MBR-O CHK

Status Lights: *light 8, light 11, light 13*

**CU00 - Console Diagnostic**

//==============================================================================

//

// CU00 - Console Diagnostic

//

// Program Switch settings:

//

// PS1: not used

// PS2: not used

// PS3: not used

// PS4: not used

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - STOP

//

// Start addresses:

//

// 00000 - Full test

//

// Directions:

//

// 1. Load CC diagnostic

// 2. Press RESET

// 3. Press START

// 4. After printing the diagnostic program, the machine will halt with the

// MANUAL light on

// 5. Verify the register contents:

// OPERATION REGISTER - 48

// MULTIPLIER - 0

// SENSE AND BRANCH - 0

// MEMORY BUFFER REGISTER - 22

// MEMORY DATA REGISTER - 2

// DIGIT REGISTER - 22

// MEMORY ADDRESS REGISTER - 11111

// 6. Press SAVE

// 7. Verify the MEMORY ADDRESS REGISTER is 11112

// 8. Turn the MEMORY ADDRESS REGISER DISPLAY SELECTOR, press DISPLAY MAR, and

// verity the MAR register contents:

// IR-1 - 11112

// IR-2 - 00000

// OR-1 - 02222

// OR-2 - 04444

// OR-3 - 04444

// PR-1 - 11112

// PR-2 - 00000

// PR-3 - 00000

// 9. Press INSERT

// 10. Type 4911112

// 11. Press RELEASE-START

// 12. Verify AUTOMATIC light is on and the MANUAL light is off

// 13. Type ABCDEFGHIJKLMNOPQRSTUVWXYZ <record mark>

// 14. Press RELEASE

// 15. Press START

// 16. After printing the alphabet, the machine will wait for alphameric input

// 17. Verify the AUTOMATIC light is on and the SAVE light is off

// 18. Verify the MEMORY ADDRESS REGISTER is 11123

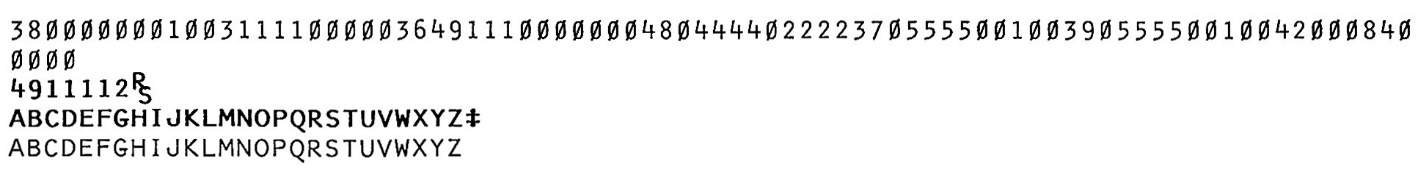
// 19. Press RELEASE

// 20. Verify the AUTOMATIC light is off and the MANUAL light is on

//

//==============================================================================

Sample Output – CU00



**CU01 - General Op Codes Diagnostic**

//==============================================================================

//

// CU01 - General Op Codes Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Bypass error type out

// OFF - Type out routine number on error

// PS2: ON - Loop in routine

// OFF - Continue to next routine

// PS3: ON - Stop on error

// OFF - Do not stop on error, continue

// PS4: ON - Repeat test CU01

// OFF - Run test CU01 once

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - PROGRAM

//

// Start addresses:

//

// 00828 - Full test w/o automatic divide

// 14004 - Full test w/ automatic divide

//

// Directions:

//

// 1. Load CU01 diagnostic

// 2. A tab stop is automatically set at column 10

// 3. Press START

// 4. Load blank cards in card punch

// 5. Press START

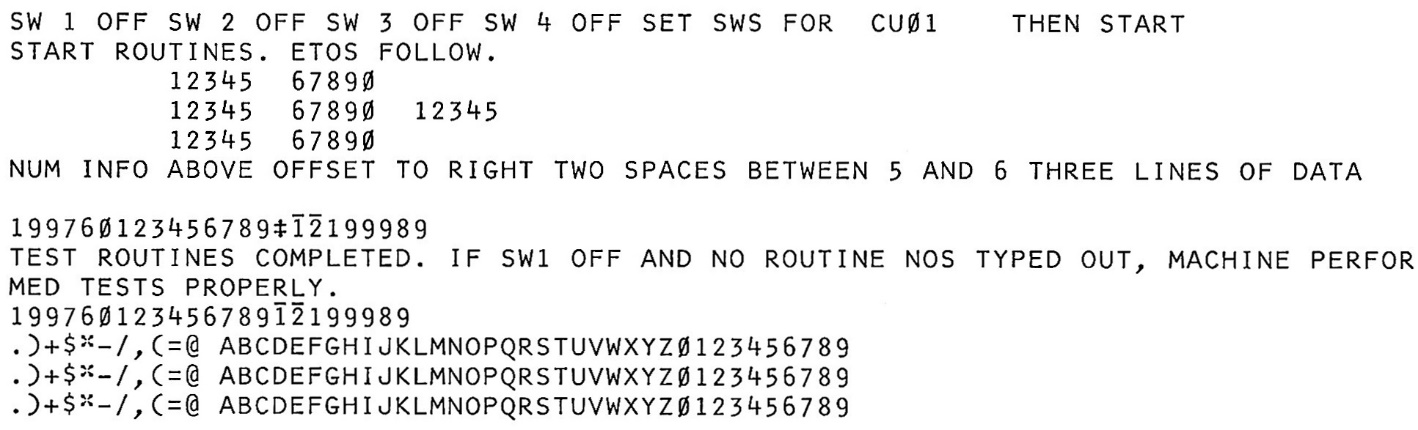
// 6. After 1620 stops, move card deck from punch to reader

// 7. Press START

//

//==============================================================================

Sample Output – CU01



**CU05 - Special Instructions Diagnostic**

//==============================================================================

//

// CU05 - Special Instructions Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Bypass error type out

// OFF - Type out routine number on error

// PS2: ON - Loop in routine

// OFF - Continue to next routine

// PS3: ON - Stop on error

// OFF - Do not stop on error, continue

// PS4: ON - Repeat test CU05

// OFF - Run test CU05 once

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - PROGRAM

//

// Start addresses:

//

// 00828 - Full test

//

// Directions:

//

// 1. Load CU05 diagnostic

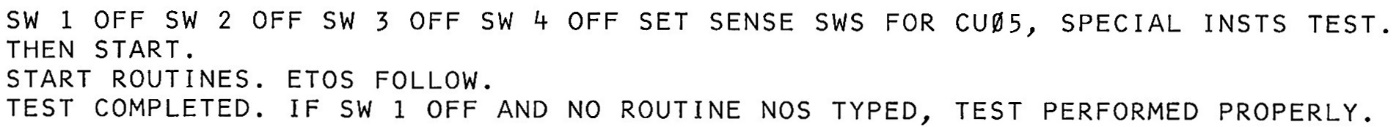
// 2. Press START

// 3. Press START

//

//==============================================================================

Sample Output – CU05



**FP01 - Floating Point Instructions Diagnostic**

//==============================================================================

//

// FP01 - Floating Point Instructions Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Bypass error type out

// OFF - Type out routine number on error

// PS2: ON - Loop in routine

// OFF - Continue to next routine

// PS3: ON - Stop on error

// OFF - Do not stop on error, continue

// PS4: not used

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - PROGRAM

//

// Start addresses:

//

// 00402 - Full test

//

// Directions:

//

// 1. Load FP01 diagnostic

// 2. Press START

//

//==============================================================================

Sample Output – FP01

= = = = = = = = = = = = = = = = = = = =

To be supplied when Floating Point

is implemented in the Simulator.

= = = = = = = = = = = = = = = = = = = =

IBM 1620 Jr. SPS Assembler (v0.99) Source: fp10-main.sps,... Assembled: 8/23/2020 @ 21:52

------ 1: fp10-main.sps --------------------------------------------------

1:1 \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1:2 \* IBM 1620 FLOATING PONT PACKAGE TEST

1:3 \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1:4 \*

1:5 \* -----------------------------------

1:6 \* RUNNING IN 1620 SIMH

1:7 \* SET CPU FP

1:8 \* SET CPU DIV

1:9 \* SET CPU IA

1:10 \* SET CPU MOD1

1:11 \* SET CPU 60K

1:12 \*

1:13 \* TURN ARITHMETIC CHECK STOP SWITCH OFF

1:14 \* DEPOSIT ARSTOP 0

1:15 \*

1:16 \* SENSE SWITCHES OFF

1:17 \* DEPOSIT SW1 0

1:18 \* DEPOSIT SW2 0

1:19 \* DEPOSIT SW3 0

1:20 \* DEPOSIT SW4 0

1:21 \*

1:22 \* -----------------------------------

1:23 \* -----------------------------------

1:24 \* SENSE SWITCH SETTINGS

1:25 \* SW1 ON = DO NOT PRINT ERROR

1:26 \* SW2 ON = REPEAT TEST

1:27 \* SW3 ON = HALT ON ERROR

1:28 \*

1:29 \*

1:30 \* ==========================================================

1:31 \* REF 1. 227-5630-1 IBM 1620 FLOATING POINT FEATURE CE MANUAL 1962

1:32 \* 2. G26-5595-0 AUTOMATIC FLOATING POINT OPERATIONS SEP61

1:33 \*

1:34 \* P = P ADDRESS

1:35 \* Q = Q ADDRESS

1:36 \* PF (MP) = CONTENTS OF P

1:37 \* QF (MQ) = CONTENTS OF Q

1:38 \* PE (EP) = EXPONENT OF THE FIELD AT THE P ADDRESS

1:39 \* QE (EQ) = EXPONENT OF THE FIELD AT THE Q ADDRESS

1:40 \* L = NUMBER OF DIGITS IN THE MANTISSA

1:41 \* D = PE-QE (EP-EQ)

1:42 \*

1:43 \* REF 2 USES MP, MQ, AND EP, EQ

1:44 \*

1:45 \* ==========================================================

1:46 \* SUMMARY

1:47 \* DATA

1:48 \* FLOATING SHIFT RIGHT (FSR-08)

1:49 \* FLOATING SHIFT LEFT (FSL-05)

1:50 \* TRANSMIT FLOATING (TFL-06)

1:51 \*

1:52 \* BRANCH AND TRANSMIT FLOATING (BTFL-07)

1:53 \*

1:54 \* ARITHMETIC - FROM REF 1 FLOATING POINT ARITHMETIC EXAMPLES

1:55 \* FLOATING POINT ARITHMETIC COMMAND RESULTS

1:56 \* ARE VERIFIED WITH FIXED POINT COMMANDS

1:57 \* ----------------------------------------------------------

1:58 \* FLOATING ADD (FADD-O1)

1:59 \* EXAMPLES 1-5 EXPONENT CONFIGURATION

1:60 \* EXAMPLES 6-11 FRACTION (MANTISSA) CONFIGURATION

1:61 \*

1:62 \* EXAMPLE 1 PE = QE / D = 0

1:63 \* EXAMPLE 2 (PE GT QE) SCAN Q -- FRACTION (MANTISSA) ADD-- D LT L

1:64 \* QF SHIFTED D (PE-QE) RIGHT TO ALIGN DECIMAL POINTS

1:65 \* EXAMPLE 3 (PE GT QE) SCAN Q -- FP EXIT -- D GT L

1:66 \* QE IS TOO SMALL-- Q DISCARDED-RESULT IS P

1:67 \* EXAMPLE 4 (PE LT QE) EXP TRANSMIT SCAN P - FRACTION (MANTISSA) ADD-

1:68 \* D LT L - PF SHIFTED D (PE-QE) RIGHT TO ALIGN DECIMALS

1:69 \* EXAMPLE 5 (PE LT QE) EXP TRANSMIT SCAN P -- FP EXIT -- D GT L

1:70 \* PE IS TOO SMALL-- P DISCARDED-RESULT IS Q

1:71 \* EXAMPLE 6 NO NORMALIZING REQUIRED---NO CARRY OUT

1:72 \* EXAMPLE 7 NORMALIZING SHIFT RIGHT REQUIRED -- CARRY OUT

1:73 \* EXPONENT ADJUST

1:74 \* EXAMPLE 8 NORMALIZING SHIFT RIGHT REQUIRED -- MACHINE INFINITY

1:75 \* 9...9 99 - CARRY OUT CAUSES EXPONENT OVERFLOW

1:76 \* EXAMPLE 9 ZERO FRACTION (MANTISSA) RESULT -- MACHINE ZERO 0...0 -99

1:77 \* EXAMPLE 10 NORMALIZING SHIFT LEFT -- SHIFT LEFT -- ZERO FILL

1:78 \* EXPONENT ADJUST

1:79 \* EXAMPLE 11 NORMALIZING SHIFT LEFT -- EXPONENT OVERFLOW

1:80 \* MACHINE ZERO

1:81 \*

1:82 \* FLOATING SUBTRACT (FSUB-02)

1:83 \* SAME AS FLOATING ADD EXCEPT THAT SIGN CONTROL

1:84 \* PROCEDURES FOR MQ ARE REVERSED.

1:85 \*

1:86 \* FLOATING MULTIPLY (FMUL-03)

1:87 \* EXAMPLE 1. NO EXPONENT MODIFY REQUIRED

1:88 \* EXAMPLE 2. EXPONENT MODIFY REQUIRED

1:89 \* EXAMPLE 3. DIGIT FORCE -- MACHINE INFINITY

1:90 \* EXAMPLE 4. EXPONENT OVERFLOW -- MODIFY PE TO 99

1:91 \* EXAMPLE 5. SPECIAL CASE 999 X 199

1:92 \*

1:93 \* FLOATING DIVIDE (FDIV-09)

1:94 \*

1:95

1:96 \* ==========================================================

1:97

1:98 00402 41 00000 00000 FPMAIN NOP 00000, 00000

1:99

1:100 00414 34 00000 00102 RCTY

1:101 00426 34 00000 00102 RCTY

1:102 00438 39 10319 00100 WATY TITLE

1:103 00450 34 00000 00102 RCTY

1:104 \* ==========================================================

1:105 \* RUN ALL TESTS

1:106 \*

1:107 \*

1:108 \*ALL TESTS

1:109 \* ==========================================================

1:110

1:111 00462 41 00000 00000 FIRST NOP

1:112

\_

1:113 00474 17 08614 00486 BTM SETS1, \*+12,, SET SW SUBROUTINE

1:114

1:115

\_

1:116 00486 17 01236 00600 BTM TFTST, \*+114,, TFL TEST

\_

1:117 00499 06 DC 2, 06,,, TFL INSTRUCTION

\_

1:118 00501 333333..33330@ DAC 50,======= TRANSMIT FLOATING (TFL 06) TESTS ========@

1:119

1:120

\_

1:121 00600 17 01236 00714 BTM TFTST, \*+114,, FSL TEST

\_

1:122 00613 05 DC 2, 05,,, FSL INSTRUCTION

\_

1:123 00615 333333..33330@ DAC 50,====== FLOATING SHIFT LEFT (FSL 05) TESTS =======@

1:124

1:125

\_

1:126 00714 17 01236 00828 BTM TFTST, \*+114,, FSR TEST

\_

1:127 00727 08 DC 2, 08,,, FSR INSTRUCTION

\_

1:128 00729 333333..33330@ DAC 50,====== FLOATING SHIFT RIGHT (FSR 08) TESTS ======@

1:129

1:130

\_

1:131 00828 17 01236 00942 BTM TFTST, \*+114,, BTFL TEST

\_

1:132 00841 07 DC 2, 07,,, BTFL INSTRUCTION

\_

1:133 00843 333333..33330@ DAC 50,=== BRANCH AND TRANSMIT FLOAT (BTFL 07) TESTS ===@

1:134

1:135

1:136

\_

1:137 00942 17 02466 01038 BTM FATST, \*+96,, FADD TEST

\_

1:138 00955 01 DC 2, 01,,, FADD INSTRUCTION

\_

1:139 00957 21 DC 2, 21,,, ADD INSTRUCTION

\_

1:140 00959 333333..33330@ DAC 40,====== FLOATING ADD (FADD) TESTS ======@

1:141

1:142

1:143

\_

1:144 01038 17 02466 01134 BTM FATST, \*+96,, FSUB TEST

\_

1:145 01051 02 DC 2, 02,,, FSUB INSTRUCTION

\_

1:146 01053 22 DC 2, 22,,, SUBTRACT INSTRUCTION

\_

1:147 01055 333333..33330@ DAC 40,====== FLOATING SUB (FSUB) TESTS ======@

1:148

1:149

\_

1:150 01134 17 06372 01146 BTM FMTST, \*+12,, FMUL TEST

1:151

1:152 \* ------ PRINT END OF TESTING MESSAGE AND HALT

1:153

1:154 01146 41 00000 00000 ALDON NOP

1:155 01158 34 00000 00102 RCTY

1:156 01170 34 00000 00102 RCTY

1:157 01182 39 10389 00100 WATY DONE

1:158 01194 34 00000 00102 RCTY

1:159 01206 48 00000 00000 H 00000, 00000,, END OF PROGRAM

1:160 01218 48 00000 00000 H 00000, 00000,, WILL SHOW (H) AS NEXT INSTRUCTION

1:161

1:162

1:163 \* ==========================================================

1:164 \* TEST SUBROUTINES FOLLOW

1:165 \* ==========================================================

1:166 \* ==========================================================

1:167 \* ==========================================================

1:168

------ 2: fp20-tftest.sps --------------------------------------------------

2:1 \* ========================================================

2:2 \* TFL, BTFL, FSL, FSR TESTS

2:3 \*

2:4 \*TFTST

2:5 \* ========================================================

2:6 01234 00005 DS 5,,, RETURN ADDRESS FROM BTM

2:7 01236 41 00000 00000 TFTST NOP 00000, 00000

2:8

2:9 \* INSTRUCTION ARG

2:10 01248 26 01283 01235 TF \*+35, TFTST-1,,

\_

2:11 01260 12 01283 00101 SM \*+23, 101,,

2:12 01272 26 01290 00000 TF TFINS+6, 0,, TO FLOATING POINT INSTRUCTION

2:13 01284 41 00000 00000 TFINS NOP 00000,00000,, KEEP INSTRUCTION ARG IN P ADDRESS

2:14

2:15 \* TEST TITLE ARG

2:16 01296 26 01331 01235 TF \*+35, TFTST-1,,

\_

2:17 01308 12 01331 00099 SM \*+23, 99,,

\_

2:18 01320 16 01362 00000 TFM TFWT+6, 0,, TO WRITE TITLE

2:19

2:20 \* TITLE

2:21

2:22 01332 34 00000 00102 RCTY

2:23 01344 34 00000 00102 RCTY

\_

2:24 01356 39 00000 00100 TFWT WATY 00000,,6, MAIN TEST TITLE

2:25 01368 34 00000 00102 RCTY

2:26

2:27

2:28 \* INITIALIZE TABLE POINTER

2:29

\_

2:30 01380 16 10767 00000 TFM TFINC, 0,, TABLE POINTER

2:31

2:32 \* POINT TO TEST DATA TABLES

2:33 \* RESET

\_

2:34 01392 16 10782 10802 TFM TFTBL, TFTBI

\_

2:35 01404 16 10787 10837 TFM TFTBL+5, TFTBD

2:36 01416 26 10777 10782 TF TFTBP, TFTBL

2:37 01428 26 10782 10787 TF TFTBP+5, TFTBL+5

2:38

2:39

2:40 01440 41 00000 00000 TFNXT NOP

2:41

2:42 \* ADVANCE TO NEXT ADDRESS IN TABLES

2:43

2:44 01452 21 10777 10767 A TFTBP, TFINC,, ID MESSAGE TABLE

\_

2:45 01464 14 10777 00000 CM TFTBP, 00000, 6, RETURN ON ZERO ADDRESS

\_

2:46 01476 46 01235 01200 BE TFTST-1,,6, RETURN

2:47

2:48 01488 21 10782 10767 A TFTBP+5, TFINC,, DATA TABLE

2:49

\_

2:50 01500 16 10767 00005 TFM TFINC, 5

2:51

\_

2:52 01512 26 10792 10777 TF TFI, TFTBP,11

\_

2:53 01524 26 10797 10782 TF TFD, TFTBP+5,11

2:54

2:55 \* START OF CURRENT TEST

2:56 01536 34 00000 00102 TFRPT RCTY

\_

2:57 01548 39 10792 00100 WATY TFI,,6, TEST TITLE

2:58 01560 34 00000 00101 SPTY

2:59

2:60 \* FP INSTRUCTION TEST FOLLOW.

2:61 \* TEST DATA FIELD IN ADDRESS REFERENCED

2:62 \* BY TFD ASSUMED TO BE A 100 DIGIT MANTISSA

2:63

2:64 \* SELECT INSTRUCTION (TFINS)

2:65 01572 25 14955 14953 TD STAT, FALSE

2:66 01584 26 10658 10554 TF WK1, WK1Z,, ZERO WK1

2:67 01596 33 10558 00000 CF WK1-100

2:68 01608 26 10761 10554 TF WK2, WK1Z,, ZERO WK2

2:69 01620 33 10661 00000 CF WK2-100

2:70

2:71 01632 26 02195 10797 TF TEXCK+11, TFD,, EXP COMPARE

2:72

2:73 \* WHEN TFL(06)

\_

2:74 01644 14 01290 00006 CM TFINS+6, 06, 10,, TFL

2:75 01656 47 01740 01200 BNE TFSL

\_

2:76 01668 06 10658 10797 TFL WK1, TFD,11

2:77 01680 26 01715 10797 TF \*+35, TFD,, MANTISSA COMPARE

\_

2:78 01692 12 01715 00002 SM \*+23, 2

2:79 01704 24 10656 00000 C WK1-2, 00000,, MANTISSA

2:80 01716 46 02184 01200 BE TEXCK,,, EXPONENT

2:81 01728 49 02220 00000 B TER

2:82

2:83 \* WHEN FSL(05)

2:84 \* FSL LEAVES SIGN FLAG IN ORIGINAL POSITION

2:85 \* THE COMPARE (C) COMPARES THE SHIFTED DATA

2:86 \* TO THE ORIGINAL. IF THE ORIGINAL IS NEGATIVE

2:87 \* THEN TO MAKE THE COMPARE WORK THE SHIFTED DATA

2:88 \* NEEDS A SIGN (FLAG LOW ORDER)

\_

2:89 01740 14 01290 00005 TFSL CM TFINS+6, 05, 10,, FSL

2:90 01752 47 01896 01200 BNE TFSR

2:91 01764 26 01799 10797 TF \*+35, TFD

\_

2:92 01776 12 01799 00101 SM \*+23, 101

2:93 01788 31 10557 00000 TR WK1-101, 00000,, COPY FP NUMBER TO WORK

2:94 01800 05 10578 10656 FSL WK1-80, WK1-2

2:95 01812 26 01871 10797 TF TFSLC+11, TFD

\_

2:96 01824 12 01871 00002 SM TFSLC+11, 2

\_

2:97 01836 44 01860 01871 BNF TFSLC, TFSLC+11, 11

2:98 01848 32 10627 00000 SF WK1-31

2:99 01860 24 10627 00000 TFSLC C WK1-31, 00000,, MANTISSA

2:100 01872 46 02184 01200 BE TEXCK,,, EXPONENT

2:101 01884 49 02220 00000 B TER

2:102

2:103 \* WHEN FSR(08)

2:104 \* FIRST MAKE LEFT SHIFTED COPY TO WK1 AND CORRECT SIGN

\_

2:105 01896 14 01290 00008 TFSR CM TFINS+6, 08, 10,, FSR

2:106 01908 47 02112 01200 BNE TBTFL

2:107 01920 26 01955 10797 TF \*+35, TFD

\_

2:108 01932 12 01955 00002 SM \*+23, 2

2:109 01944 26 10608 00000 TF WK1-50, 00000,, COPY FP NUMBER TO WORK

2:110 01956 26 01979 10797 TF \*+23, TFD

2:111 01968 26 10658 00000 TF WK1, 00000,, AND EXPONENT

2:112 01980 26 02015 10797 TF \*+35, TFD

\_

2:113 01992 12 02015 00002 SM \*+23, 2

2:114 02004 44 02040 00000 BNF TFSRF, 00000

2:115 02016 33 10608 00000 CF WK1-50

2:116 02028 32 10656 00000 SF WK1-2

2:117 02040 08 10656 10608 TFSRF FSR WK1-2, WK1-50

2:118 02052 26 02087 10797 TF TFSRC+11, TFD

\_

2:119 02064 12 02087 00002 SM TFSRC+11, 2

2:120 02076 24 10656 00000 TFSRC C WK1-2, 00000,, MANTISSA

2:121 02088 46 02184 01200 BE TEXCK,,, EXPONENT

2:122 02100 49 02220 00000 B TER

2:123

2:124

2:125 \* WHEN BTFL(07)

2:126 02112 26 02183 10797 TBTFL TF \*+71, TFD

2:127 02124 26 02159 10797 TF \*+35, TFD

\_

2:128 02136 12 02159 00101 SM \*+23, 101

2:129 02148 31 10557 00000 TR WK1-101, 00000,, COPY FP NUMBER TO WORK

\_

2:130 02160 16 02460 02184 TFM RETAD, \*+24,, BTFL RETURN

2:131 02172 07 02392 00000 BTFL BTSUB, 00000,, USES WK1 AND SETS STAT

2:132

2:133 02184 24 10658 00000 TEXCK C WK1, 00000,, EXPONENT

2:134 02196 47 02220 01200 BNE TER

2:135 02208 25 14955 14952 TD STAT, TRUE

2:136

2:137

2:138

2:139 \* CHECK COMPARE STATUS

2:140 \* CALL ERROR ROUTINE AND RETURN ITS STATUS IN ESTAT

\_

2:141 02220 17 09518 02242 TER BTM ERTN, \*+22,, TRANSMIT ADDRESS OF NEXT INSTRUCTION

\_ \_

2:142 02236 1495514954 DSA STAT, ESTAT

2:143

2:144 02242 43 02266 14954 BD TFEND, ESTAT

2:145 02254 49 01536 00000 B TFRPT,,, RERUN TEST

2:146

2:147 02266 41 00000 00000 TFEND NOP 00000, 00000

2:148

2:149 02278 49 01440 00000 B TFNXT,,, GOTO NEXT TEST

2:150

2:151

------ 3: fp21-tftest.sps --------------------------------------------------

3:1 \* ---------------------------------------------

3:2 \* SUBROUTINE STATUS = BTSUB(FP1, WK1)

3:3 \* FP1 IS COPIED TO WK1

3:4 \* BRANCH AND TRANSMIT COPIES WK1 TO SUBROUTINE-BTSUB

3:5 \* BTSUB GETS WK1 CONTENTS JUST BEFORE BTSUB ADDRESS (BTSUB-1)

3:6 \* BTSUB CALLS CF -COMPARE FLOATING SUBROUTINE TO

3:7 \* COMPARE WK1 TO ORIGINAL FP1

3:8 \* CF RETURNS COMPARE STATUS IN STAT

3:9

3:10 \*

3:11 \*

3:12 \*BTSUB

3:13 \* ---------------------------------------------

3:14

\_

3:15 02339 000000..000000 DC 50, 0

3:16 02340 000000..000000 DSC 50, 0

3:17 02390 00 DSC 2, 0

3:18 02392 41 00000 00000 BTSUB NOP

3:19

3:20

3:21 \*COMPARE MANTISSA AND EXPONENT SEPARATELY

3:22 \* WITH SUBROUTINE CF (COMPARE FLOATING)

\_

3:23 02404 17 09922 02432 BTM CF, \*+28,, TRANSMIT RETURN ADDRESS

\_ \_ \_

3:24 02420 023911..814955 DSA BTSUB-1, WK1, STAT

3:25

3:26 02432 41 00000 00000 NOP 00000, 00000,, RETURN TO HERE

3:27

3:28 \* RETURN

3:29 \* NOTE BB WILL NOT WORK HERE,

3:30 \* BECAUSE THIS SUBROUTINE CONTAINS A BTM

\_

3:31 02444 49 02460 00000 B RETAD,,6

3:32

3:33 \* --------------------

3:34 \* SUBROUTINE BTFL RETURN ADDRESS

3:35 02460 00005 RETAD DS 5

3:36

3:37

------ 4: fp30-fadd.sps --------------------------------------------------

4:1 \* ========================================================

4:2 \* FADD TEST

4:3 \*

4:4 \*FADD

4:5 \* ========================================================

4:6 02465 00005 DS 5,,, RETURN ADDRESS FROM BTM

4:7 02466 41 00000 00000 FATST NOP 00000, 00000

4:8

4:9 \* GET ADD / SUBTRACT INST AND TITLE

4:10 02478 26 02513 02465 TF \*+35, FATST-1,,

\_

4:11 02490 12 02513 00083 SM \*+23, 83,,

4:12 02502 26 03043 00000 TF FARIT+1, 0,, TO FLOATING POINT INSTRUCTION

4:13

4:14 02514 26 02549 02465 TF \*+35, FATST-1,,

\_

4:15 02526 12 02549 00081 SM \*+23, 81,,

\_

4:16 02538 16 03118 00000 TFM ADSA, 0,, FIXED INST TO FIXED ADD/SUB SUB

4:17

4:18 02550 26 02585 02465 TF \*+35, FATST-1,,

\_

4:19 02562 12 02585 00079 SM \*+23, 79,,

\_

4:20 02574 16 02616 00000 TFM WTITL+6, 0,, TO WRITE TITLE

4:21

4:22 \* TITLE

4:23

4:24 02586 34 00000 00102 RCTY

4:25 02598 34 00000 00102 RCTY

\_

4:26 02610 39 00000 00100 WTITL WATY 00000,,6, MAIN FADD / FSUB TEST TITLE

4:27 02622 34 00000 00102 RCTY

4:28

4:29

4:30 \* INITIALIZE TABLE POINTER

4:31

\_

4:32 02634 16 11713 00000 TFM FAINC, 0,, TABLE POINTER

4:33

\_

4:34 02646 14 03043 00002 CM FARIT+1, 02, 10, CHOOSE FADD / SUB TABLES

4:35 02658 46 02754 01200 BE SUBTB

4:36

4:37 \* POINT TO FADD TEST DATA TABLES

4:38 \* RESET

\_

4:39 02670 16 11733 11778 TFM FATBL, FATBI

\_

4:40 02682 16 11738 11833 TFM FATBL+5, FATBP

\_

4:41 02694 16 11743 11888 TFM FATBL+10, FATBQ

4:42 02706 26 11728 11733 TF ASTBL, FATBL

4:43 02718 26 11733 11738 TF ASTBL+5, FATBL+5

4:44 02730 26 11738 11743 TF ASTBL+10, FATBL+10

4:45 02742 49 02826 00000 B FANXT

4:46

4:47 \* POINT TO FSUB TEST DATA TABLES

4:48 \* RESET

\_

4:49 02754 16 11748 11943 SUBTB TFM FSTBL, FSTBI

\_

4:50 02766 16 11753 11998 TFM FSTBL+5, FSTBP

\_

4:51 02778 16 11758 12053 TFM FSTBL+10, FSTBQ

4:52 02790 26 11728 11748 TF ASTBL, FSTBL

4:53 02802 26 11733 11753 TF ASTBL+5, FSTBL+5

4:54 02814 26 11738 11758 TF ASTBL+10, FSTBL+10

4:55

4:56

4:57 02826 41 00000 00000 FANXT NOP

4:58

4:59 \* ADVANCE TO NEXT ADDRESS IN TABLES

4:60 02838 21 11728 11713 A ASTBL, FAINC,, ID MESSAGE TABLE

\_

4:61 02850 14 11728 00000 CM ASTBL, 00000, 6, RETURN ON ZERO ADDRESS

\_

4:62 02862 46 02465 01200 BE FATST-1,,6, RETURN

4:63

4:64 02874 21 11733 11713 A ASTBL+5, FAINC,, P DATA TABLE

4:65 02886 21 11738 11713 A ASTBL+10, FAINC,, Q DATA TABLE

4:66

\_

4:67 02898 16 11713 00005 TFM FAINC, 5

4:68

\_

4:69 02910 26 11763 11728 TF FAI, ASTBL,11

\_

4:70 02922 26 11768 11733 TF FAP, ASTBL+5,11

\_

4:71 02934 26 11773 11738 TF FAQ, ASTBL+10,11

4:72

4:73 \* START OF CURRENT FADD TEST

4:74 02946 41 00000 00000 FARPT NOP

4:75 02958 34 00000 00102 RCTY

\_

4:76 02970 39 11763 00100 WATY FAI,,6, FADD TEST TITLE

4:77 02982 34 00000 00101 SPTY

4:78

4:79 \*--- NEED TO PRESERVE AUGEND - MOVE TO FASUM FIRST

4:80 \*--- AVOID TFL FOR FLOATING POINT TESTS

4:81

\_

4:82 02994 26 12205 11768 TF FASUM, FAP, 11, P POINTS TO EXPONENT

\_

4:83 03006 12 11768 00002 SM FAP, 00002,, P NOW POINTS TO MANTISSA

\_

4:84 03018 26 12203 11768 TF FASUM-2, FAP, 11,

\_

4:85 03030 11 11768 00002 AM FAP, 00002,, P AGAIN POINTS TO EXPONENT

4:86

4:87 \* FADD / FSUB INST ARE PASSED IN ARGS, WHICH REPLACE FADD BELOW

4:88

\_

4:89 03042 01 12205 11773 FARIT FADD FASUM, FAQ, 11, FLOAT ARIITHMETIC

4:90

4:91

4:92 \* INDICATOR CHECK

4:93 03054 47 03078 01400 BNV \*+24

4:94 03066 39 15015 00100 WATY OVM1,,, OVERFLOW MESSAGE

4:95 03078 47 03102 01500 BNXV \*+24

4:96 03090 39 15037 00100 WATY XCKM1,,, EXPONENT OVERFLOW MESSAGE

4:97

4:98

4:99 \* CALL SUBROUTINE A21

4:100 \* A (21) MANTASSAS AND COMPUTE EXPONENTS

\_

4:101 03102 17 03216 03140 BTM A21, \*+38,, TRANSMIT RETURN ADDRESS

\_ \_ \_

4:102 03118 000001..514955 ADSA DSA 00000, -FAP, -FAQ, FASUM, STAT

4:103

4:104

4:105 \* CHECK COMPARE STATUS

4:106 \* CALL ERROR ROUTINE AND RETURN ITS STATUS IN ESTAT

\_

4:107 03140 17 09518 03162 BTM ERTN, \*+22,, TRANSMIT ADDRESS OF NEXT INSTRUCTION

\_ \_

4:108 03156 1495514954 DSA STAT, ESTAT

4:109

4:110 03162 43 03186 14954 BD FAEND, ESTAT

4:111 03174 49 02946 00000 B FARPT,,, RERUN TEST

4:112

4:113 03186 41 00000 00000 FAEND NOP 00000, 00000

4:114

4:115 03198 49 02826 00000 B FANXT,,, GOTO NEXT FADD TEST

4:116

------ 5: fp31-fixadd.sps --------------------------------------------------

5:1 \* ---------------------------------------------

5:2 \* SUBROUTINE STATUS = A21(FXARI, FP1, FP2, FP-SUM)

5:3 \* COMPARES FADD SUM WITH A(21)/EX (EX ADJUST)

5:4 \* -- OR -- FSUB WITH S(22) PER FIRST ARGUMENT (FXARI)

5:5 \*

5:6 \* REF (1) 227-5630-1 IBM 1620 FLOATING POINT FEATURE CE MANUAL 1962

5:7 \*

5:8 \* VERIFIES FADD RESULT BY USING A(21) TO ADD MANTISSAS

5:9 \* AND SEPARATELY HANDLES EXPONENTS.

5:10 \* FLOATING POINT NUMBER IS CREATED FROM A(21) SUM AND

5:11 \* (ADJUSTED) EXPONENTS

5:12 \*

5:13 \* HANDLES CASES DESCRIBED IN REF (1) FLOATING ADD EXAMPLES

5:14 \*

5:15 \* EXAMPLE 1 PE = QE / D = 0 NO NORMALIZING - NO CARRY OUT

5:16 \* EXAMPLE 2 PE GT QE - D LT L QF SHIFTED RT TO ALIGN DECIMAL POINTS

5:17 \* EXAMPLE 3 PE GT QE - D GT L - Q DISCARDED-RESULT IS P

5:18 \* EXAMPLE 4 PE LT QE- D LT L PF SHIFTED RT TO ALIGN DECIMAL POINTS

5:19 \* EXAMPLE 5 PE LT QE - D GT L P DISCARDED-RESULT IS Q

5:20 \*

5:21 \* EXAMPLE 6 NO NORMALIZING REQUIRED-NO CARRY OUT-SAME AS EXAMPLE 1

5:22 \*

5:23 \* EXAMPLE 7 NORMALIZING SHIFT RT - CARRY OUT- EXP ADJUST

5:24 \* EXAMPLE 8 NORM SHIFT RT-CARRY OT CAUSES EXP OV MACH INFINITY (9...9 99)

5:25 \* EXAMPLE 9 ZERO FRACTION RESULT - MACHINE ZERO (0...0 -99)

5:26 \* EXAMPLE 10 NORMALIZING SHIFT LEFT - SHIFT LEFT- EXP ADJ

5:27 \* EXAMPLE 11 NORMALIZING SHIFT LEFT- EXP OV MACHINE ZERO (0...0 -99)

5:28 \*

5:29 \* USAGE

5:30 \* BTM A21, \*+38,, TRANSMIT RETURN ADDRESS

5:31 \* DSA FXARI, -FAP, -FAQ, FASUM, STAT

5:32 \* NOP 00000, 00000,, RETURN TO HERE

5:33 \*

5:34 \* FXARI = A OR S ARITHMETIC INSTRUCTION

5:35 \* A21A1 = -FAP (INDIRECT ADDRESS) FLOATING POINT NUMBER

5:36 \* A21A2 = -FAQ (INDIRECT ADDRESS) FLOATING POINT NUMBER

5:37 \* A21D1 = FASUM (DATA FIELD) FLOATING POINT FADD SUM

5:38 \* A21RC = STAT ( DATA, RETURN CODE )

5:39 \* RETURNS STATUS OF COMPARE

5:40 \*

5:41 \*

5:42 \*A21

5:43 \* ---------------------------------------------

5:44 03214 00005 DS 5,, RETURN ADDRESS

5:45 03216 41 00000 00000 A21 NOP 00000, 00000

5:46

5:47

5:48 \* GET FIRST ARGUMENT

5:49 03228 26 03263 03215 TF \*+35, A21-1,,

\_

5:50 03240 12 03263 00022 SM \*+23, 00022,,

5:51 03252 26 03275 00000 TF \*+23, 0,,

5:52 03264 26 04165 00000 TF FXAR1+1, 0,, -- A / S FIXED POINT ARITH INSTRUCTION

5:53 03276 26 03299 03275 TF \*+23, \*-1

5:54 03288 26 04789 00000 TF FXAR2+1, 0,, -- A / S FIXED POINT ARITH INSTRUCTION

5:55

5:56

5:57 \* GET SECOND ARGUMENT (INDIRECT)

5:58 03300 26 03335 03215 TF \*+35, A21-1,,

\_

5:59 03312 12 03335 00017 SM \*+23, 00017,,

5:60 03324 26 03359 00000 TF \*+35, 0,,

5:61 03336 33 03359 00000 CF \*+23

5:62 03348 26 06253 00000 TF A21A1, 0,,

5:63

5:64

5:65 \* GET THIRD ARGUMENT (INDIRECT)

5:66 03360 26 03395 03215 TF \*+35, A21-1,,

\_

5:67 03372 12 03395 00012 SM \*+23, 00012,,

5:68 03384 26 03419 00000 TF \*+35, 0,,

5:69 03396 33 03419 00000 CF \*+23

5:70 03408 26 06258 00000 TF A21A2, 0,,

5:71

5:72

5:73 \* GET FOURTH ARGUMENT

5:74 03420 26 03455 03215 TF \*+35, A21-1,,

\_

5:75 03432 12 03455 00007 SM \*+23, 00007,,

5:76 03444 26 03467 00000 TF \*+23, 0,,

5:77 03456 26 06360 00000 TF A21D1, 0,, -- GETS EXP

5:78

5:79 03468 26 03503 03467 TF \*+35, \*-1,,

\_

5:80 03480 12 03503 00002 SM \*+23, 00002,

5:81 03492 26 06358 00000 TF A21D1-2, 0,, -- GETS MANTISSA

5:82

5:83

5:84 \* EXPONENT

5:85 \* HANDLE RESULTANT EXPONENT BASED ON L AND D

5:86 \* AML = MANTISSA LENGTH AED = ABS(EP-EQ)

5:87

5:88 03504 26 05403 06042 TF AEP, AZ5,, CLEAR

5:89 03516 26 05511 06042 TF AEQ, AZ5,, CLEAR

\_

5:90 03528 21 05403 06253 A AEP, A21A1, 11, EXPONENT P

\_

5:91 03540 21 05511 06258 A AEQ, A21A2, 11, EXPONENT Q

5:92

5:93

5:94 \* SUBTRACT MP AND MQ EXPONENTS AED = ABS(EP-EQ)

5:95 03552 26 05281 06042 TF AED, AZ5,, CLEAR

5:96 03564 21 05281 05403 A AED, AEP,, EXPONENT P

5:97 03576 22 05281 05511 S AED, AEQ,, EXPONENT Q

5:98 03588 33 05281 00000 CF AED,,, CLEAR SIGN FLAG TO GET ABS VAL

5:99

5:100

5:101 \* MANTISSA

\_

5:102 03600 12 06253 00002 SM A21A1, 00002

\_

5:103 03612 12 06258 00002 SM A21A2, 00002

5:104 03624 26 05397 06243 TF AMP, AZ101

5:105 03636 26 05505 06243 TF AMQ, AZ101

\_

5:106 03648 26 05397 06253 TF AMP , A21A1, 11, MANTISSA P

\_

5:107 03660 26 05505 06258 TF AMQ , A21A2, 11, MANTISSA Q

5:108

5:109

5:110 \* MANTISSA LENGTH OF FIRST ARG IN AML

5:111 \* L REPRESENTS MANTISSA LENGTH

\_

5:112 03672 16 05276 00001 TFM AML, 00001,

5:113 03684 26 05296 06253 TF AMPA, A21A1,, AMPA HIGH ORDER ADDRESS OF AMP

5:114

\_

5:115 03696 12 05296 00001 AL SM AMPA, 1

\_

5:116 03708 11 05276 00001 AM AML, 1

\_

5:117 03720 44 03696 05296 BNF AL, AMPA, 11

5:118

5:119

5:120 \* ALIGN DECIMAL POINTS

5:121 03732 24 05403 05511 C AEP, AEQ

5:122 03744 46 04104 01200 BE ADDM,,, NO ALIGN-ADD MANTISSAS UNCHANGED

5:123 03756 47 03780 01300 BL ADJP,,, PE LT QE

5:124 03768 46 03936 01100 BH ADJQ,,, PE GT QE

5:125

5:126

5:127 \* ADJUST P MANTISSA

5:128 \* PE LT QE - D GT L - P DISCARDED-RESULT IS Q

5:129 03780 24 05281 05276 ADJP C AED, AML,, COMPARE D TO L

5:130 03792 47 03840 01300 BL ADJP1

5:131 03804 26 05613 05505 TF AMPQ, AMQ

5:132 03816 26 05619 05511 TF AEPQ, AEQ

5:133 03828 49 05148 00000 B AMKFP,,, GOTO MAKE FP NUMBER FROM Q ONLY

5:134

\_

5:135 03840 16 03875 05397 ADJP1 TFM \*+35, AMP,, RESTORE ADDRESS IN TF BELOW

5:136 03852 22 03875 05281 S \*+23, AED,, ALIGN DECIMAL POINTS

5:137 03864 26 05397 05397 TF AMP, AMP

\_

5:138 03876 16 03918 05397 TFM \*+42, AMP,, RESTORE ADDRESS IN MF BELOW

5:139 03888 22 03918 05276 S \*+30, AML

\_

5:140 03900 11 03918 00001 AM \*+18, 00001

5:141 03912 71 05397 05397 MF AMP, AMP

5:142 03924 49 04104 00000 B ADDM

5:143

5:144

5:145 \* ADJUST Q MANTISSA

5:146 \* PE GT QE - D GT L - Q DISCARDED-RESULT IS P

5:147 03936 41 00000 00000 ADJQ NOP 00000, 00000,,

5:148 03948 24 05281 05276 C AED, AML,, COMPARE D TO L

5:149 03960 47 04008 01300 BL ADJQ1

5:150 03972 26 05613 05397 TF AMPQ, AMP

5:151 03984 26 05619 05403 TF AEPQ, AEP

5:152 03996 49 05148 00000 B AMKFP,,, GOTO MAKE FP NUMBER FROM P ONLY

5:153

\_

5:154 04008 16 04043 05505 ADJQ1 TFM \*+35, AMQ,, RESTORE ADDRESS IN TF BELOW

5:155 04020 22 04043 05281 S \*+23, AED,, ALIGN DECIMAL POINTS

5:156 04032 26 05505 05505 TF AMQ, AMQ

\_

5:157 04044 16 04086 05505 TFM \*+42, AMQ,, RESTORE ADDRESS IN MF BELOW

5:158 04056 22 04086 05276 S \*+30, AML

\_

5:159 04068 11 04086 00001 AM \*+18, 00001

5:160 04080 71 05505 05505 MF AMQ, AMQ

5:161 04092 49 04104 00000 B ADDM

5:162

5:163

5:164 \* ADD (ADJUSTED) P TO Q MANTISSAS

5:165 04104 41 00000 00000 ADDM NOP 00000, 00000,,

5:166 04116 25 05622 14953 TD ANRM, FALSE,, CLEAR NORMALIZE FLAG

5:167 04128 25 05621 14953 TD AOV, FALSE,, CLEAR OVERFLOW FLAG

5:168 04140 26 05613 06243 TF AMPQ, AZ101,, CLEAR

5:169 04152 26 05613 05397 TF AMPQ, AMP

5:170 04164 21 05613 05505 FXAR1 A AMPQ, AMQ,, ARGS MAY REPLACE A WITH S

5:171

5:172

5:173 \* CARRY-OUT OVERFLOW INDICATOR CHECK

5:174 04176 47 04212 01400 BNV ASHFT

5:175 04188 25 05621 14952 TD AOV, TRUE,, SET OVERFLOW FLAG

5:176 04200 49 04548 00000 B AZCK

5:177

5:178

5:179 \* NOMRALIZE MANTISSA IN AMPQ

5:180 \* SHIFT LEFT

5:181 \* COUNT LEADING ZEROS IN AZCNT

5:182 \* REQUIRES EXPONENT ADJUST

5:183 04212 41 00000 00000 ASHFT NOP 00000, 00000

5:184 04224 24 05613 06243 C AMPQ, AZ101,, ZERO MANTISSA CHECK

5:185 04236 46 04572 01200 BE AZRO

5:186

5:187 04248 25 05622 14953 TD ANRM, FALSE,, CLEAR NORMALIZE FLAG

\_

5:188 04260 16 05291 00000 TFM AZCNT, 00000,, CLEAR LEADING ZERO COUNTER

5:189

\_

5:190 04272 16 04319 05613 TFM \*+47, AMPQ

5:191 04284 22 04319 05276 S \*+35, AML

\_

5:192 04296 11 04319 00001 AM \*+23, 00001

5:193 04308 43 04548 05613 BD AZCK, AMPQ,, LEADING ZERO CHECK

5:194

5:195 04320 26 06034 06243 TF AWK1, AZ101,, CLEAR WORK

5:196 04332 25 05622 14952 TD ANRM, TRUE,, SET NORMALIZE FLAG

\_

5:197 04344 16 05286 00000 TFM AFLG, 00000

\_

5:198 04356 16 05296 05613 TFM AMPA, AMPQ,, POINTS HI MANTISSA ADDR

5:199

\_

5:200 04368 12 05296 00001 AFLAG SM AMPA, 00001,, SEARCH FOR ORIGINAL FLAG

\_

5:201 04380 44 04368 05296 BNF AFLAG, AMPA, 11

\_

5:202 04392 33 05296 00000 CF AMPA, 00000, 6, CLEAR IT

5:203

\_

5:204 04404 43 04464 05296 ADIGT BD ASF, AMPA, 11, 1ST DIGIT TO SHIFT LEFT

\_

5:205 04416 11 05291 00001 AM AZCNT, 00001,, COUNT LEADING ZEROS

\_

5:206 04428 11 05286 00001 AM AFLG, 00001

\_

5:207 04440 11 05296 00001 AM AMPA, 00001

5:208 04452 49 04404 00000 B ADIGT

5:209

\_

5:210 04464 16 04530 05613 ASF TFM ASF2+6, AMPQ,, INITIALIZE TF P ADDRESS

\_

5:211 04476 32 05296 00000 SF AMPA, 00000, 6, FLAG 1ST DIGIT

5:212 04488 26 06034 05613 TF AWK1, AMPQ,, NOW LEFT SHIFT MANTISSA

5:213 04500 26 05613 06243 TF AMPQ, AZ101

5:214 04512 22 04530 05286 S \*+18, AFLG

5:215 04524 26 05613 06034 ASF2 TF AMPQ, AWK1,, SHIFT COMPLETE

5:216 04536 49 04824 00000 B EXADJ,,, GOTO EXPONENT ADJUST

5:217

5:218

5:219

5:220 \* ZERO CHECK GOTO MACHINE ZERO

5:221 04548 24 05613 06243 AZCK C AMPQ, AZ101,, ZERO MANTISSA CHECK

5:222 04560 47 04596 01200 BNE ACO

\_ \_

5:223 04572 16 05619 00099 AZRO TFM AEPQ, -99,, MAKE MACHINE ZERO

5:224 04584 49 05148 00000 B AMKFP,,, GOTO MAKE FP

5:225

5:226

5:227 \* NORMALIZE SHIFT RIGHT

5:228 \* - CARRY OUT- EXP ADJUST

\_

5:229 04596 16 04710 05397 ACO TFM ACMF1+6, AMP,, INITIALIZE MF INSTRUCTIONS

\_

5:230 04608 16 04715 05397 TFM ACMF1+11, AMP

\_

5:231 04620 16 04758 05397 TFM ACMF2+6, AMP,,

\_

5:232 04632 16 04763 05397 TFM ACMF2+11, AMP

5:233 04644 43 04668 05621 BD \*+24, AOV

5:234 04656 49 04824 00000 B EXADJ,,, NO CARRY OUT GOTO EXP ADJ

5:235 04668 22 04715 05276 S \*+47, AML

\_

5:236 04680 11 04715 00001 AM \*+35, 00001

5:237 04692 22 04710 05276 S \*+18, AML

5:238 04704 71 05397 05397 ACMF1 MF AMP, AMP

5:239 04716 22 04763 05276 S \*+47, AML

\_

5:240 04728 11 04763 00001 AM \*+35, 00001

5:241 04740 22 04758 05276 S \*+18, AML

5:242 04752 71 05505 05505 ACMF2 MF AMQ, AMQ

5:243 04764 26 05613 06243 TF AMPQ, AZ101,, CLEAR

5:244 04776 26 05613 05397 TF AMPQ, AMP

5:245 04788 21 05613 05505 FXAR2 A AMPQ, AMQ,, ARGS MAY REPLACE A WITH S

5:246 04800 71 05612 05613 MF AMPQ-1, AMPQ

5:247 04812 26 05613 05612 TF AMPQ, AMPQ-1

5:248

5:249

5:250 \* ADJUST EXPONENT

5:251 \* ARITHMETIC OVERFLOW

5:252 \* MACHINE INFINITY

5:253 \* MANTISSA LEFT SHIFT FOR NORMALIZATION

5:254 04824 26 05619 06042 EXADJ TF AEPQ, AZ5

5:255 04836 26 05619 05403 TF AEPQ, AEP

5:256 04848 24 05403 05511 C AEP, AEQ

5:257 04860 46 04884 01300 BNL \*+24

5:258 04872 26 05619 05511 TF AEPQ, AEQ

5:259 04884 43 05088 05622 BD AELS, ANRM,, NORMALIZE LEFT SHIFT CHECK

5:260 04896 43 04920 05621 BD \*+24, AOV,,, ARITHMETIC OVERFLOW CHECK

5:261 04908 49 05148 00000 B AMKFP

5:262

\_

5:263 04920 14 05619 00099 CM AEPQ, 00099,, EXP 99 THEN MACH INF

5:264 04932 46 04968 01300 BNL AINF

\_

5:265 04944 11 05619 00001 AM AEPQ, 1,, ARITH OV EXP ADJUST

5:266 04956 49 05148 00000 B AMKFP

5:267

5:268 \* NORMALIZE SHIFT RIGHT-CARRY OUT

5:269 \* EXPONENT OVERFLOW MACH INFINITY

\_

5:270 04968 16 05070 05613 AINF TFM AINF2+6, AMPQ,, INITIALIZE SF

5:271 04980 44 05028 05613 BNF AINF1, AMPQ

5:272 04992 26 05613 05929 TF AMPQ, A9-3,, MOVE NINES

5:273 05004 32 05613 00000 SF AMPQ,,, REPLACE SIGN FLAG

5:274 05016 49 05040 00000 B \*+24

5:275 05028 26 05613 05929 AINF1 TF AMPQ, A9-3,, MOVE NINES

5:276 05040 22 05070 05276 S \*+30, AML

\_

5:277 05052 11 05070 00001 AM \*+18, 00001

5:278 05064 32 05613 00000 AINF2 SF AMPQ

5:279 05076 49 05148 00000 B AMKFP

5:280

5:281 \* NORMALIZING SHIFT LEFT

5:282 \* CHECK FOR EXP OVERFLOW

5:283 \* SET TO MACHINE ZERO

5:284 05088 22 05619 05291 AELS S AEPQ, AZCNT,, ADJ EXP FOR LEFT SHIFT

\_ \_

5:285 05100 14 05619 00100 CM AEPQ, -100

5:286 05112 47 05148 01200 BNE AMKFP

\_ \_

5:287 05124 16 05619 00099 TFM AEPQ, -99

5:288 05136 26 05613 06243 TF AMPQ, AZ101

5:289

5:290

5:291 \* PUT SUM AND ADJUSTED EXPONENT

5:292 \* INTO FLOATING POINT MM...MM EE

5:293 05148 71 05618 05615 AMKFP MF AEPQ-1, AEPQ-4

5:294 05160 26 08153 05619 TF PQFP, AEPQ

5:295 05172 26 08151 05613 TF PQFP-2, AMPQ

5:296

5:297

5:298 \* COMPARE FADD SUM TO A21 SUM

5:299 \* A21RC EQ TRUE IF EQUAL OTW FALSE

\_

5:300 05184 17 09922 05212 BTM CF, \*+28,, COMPARE FLOATING SUBROUTINE

\_ \_ \_

5:301 05200 063600..306366 DSA A21D1, PQFP, A21RC

5:302

5:303

5:304 \* SEND RETURN CODE TO CALLING PROGRAM

5:305 05212 26 05247 03215 TF \*+35, A21-1,,

\_

5:306 05224 12 05247 00002 SM \*+23, 2,,

5:307 05236 26 05254 00000 TF \*+18, 0,,

5:308 05248 25 00000 06366 TD 0, A21RC,,

5:309

5:310

5:311 \* RETURN

\_

5:312 05260 49 03215 00000 FACR B A21-1,,6, RETURN TO CALLING PROGRAM

5:313

5:314

5:315

5:316

5:317

5:318 \* ---- -- WORK AREAS FOR A (21) INSTRUCTION

\_

5:319 05276 00000 AML DC 5, 0,,, MANTISSA LENGTH

\_

5:320 05281 00000 AED DC 5, 0,,, EXP DIFFERENCE D = ABS(EP-EQ)

5:321

\_

5:322 05286 00000 AFLG DC 5, 0,,, MANTISSA FLAG ADDRESS

\_

5:323 05291 00000 AZCNT DC 5, 0,,, MANTISSA LEADING ZEROS COUNT

5:324

\_

5:325 05296 00000 AMPA DC 5,0,, POINTER TO AMP

5:326

\_

5:327 05346 000000..000000 DC 50, 0,,, MP MANTISSA FIELD

5:328 05347 000000..000000 DSC 50, 0,,, SIZE IS 101 FOR CARRY/OVERFLOW

5:329 05397 0 AMP DSC 1, 0

5:330 05398 @ DC 1, @

5:331

\_

5:332 05403 00000 AEP DC 5,0,,, P EXPONENT

5:333 05404 @ DC 1, @

5:334

\_

5:335 05454 000000..000000 DC 50, 0,,, MQ MANTISSA FIELD

5:336 05455 000000..000000 DSC 50, 0,,, SIZE IS 101 FOR CARRY/OVERFLOW

5:337 05505 0 AMQ DSC 1, 0

5:338 05506 @ DC 1, @

5:339

\_

5:340 05511 00000 AEQ DC 5,0,,, Q EXPONENT

5:341 05512 @ DC 1, @

5:342

\_

5:343 05562 000000..000000 DC 50, 0,,, MP + MQ SUM MANTISSA FIELD

5:344 05563 000000..000000 DSC 50, 0,,,

5:345 05613 0 AMPQ DSC 1, 0

5:346 05614 @ DC 1, @

5:347

\_

5:348 05619 00000 AEPQ DC 5,0,,, FINAL EXPONENT

5:349 05620 @ DC 1, @

5:350

\_

5:351 05621 0 AOV DC 1, 0,,, OVERFLOW FLAG FOR CARRY OUT

\_

5:352 05622 0 ANRM DC 1, 0,,, NORMALIZE FLAG

5:353

5:354 \* 123456789 123456789 123456789 123456789 1234567890

\_

5:355 05625 417271..47450@ AOVM DAC 21,A21 OVERFLOW MESSAGE@

\_

5:356 05667 417271..45000@ AEOVM DAC 31,A21 EXPONENT OVERFLOW MESSAGE@

\_

5:357 05777 000000..000000 DC 50, 0,,, MANTISSA + EXP

5:358 05778 000000..000000 DSC 50, 0

\_

5:359 05829 00 APQFP DC 2, 0

5:360 05830 @ DC 1, @

5:361

5:362

5:363 \* 123456789 123456789 123456789 123456789 1234567890

\_

5:364 05880 999999..999999 DC 50, 99999999999999999999999999999999999999999999999999

5:365 05881 999999..999999 DSC 50, 99999999999999999999999999999999999999999999999999

\_

5:366 05932 99 A9 DC 2, 99

5:367 05933 @ DC 1, @

5:368

\_

5:369 05983 000000..000000 DC 50, 0

5:370 05984 000000..000000 DSC 50, 0

\_

5:371 06034 0 AWK1 DC 1, 0

5:372 06035 @ DC 1, @

5:373

\_

5:374 06037 00 AZ2 DC 2, 0

\_

5:375 06042 00000 AZ5 DC 5, 0

5:376

\_

5:377 06092 000000..000000 DC 50, 0,,, 100 ZEROS

5:378 06093 000000..000000 DSC 49, 0

5:379 06142 0 AZ100 DSC 1, 0

5:380

\_

5:381 06192 000000..000000 DC 50, 0,,, 101 ZEROS

5:382 06193 000000..000000 DSC 50, 0

5:383 06243 0 AZ101 DSC 1, 0

5:384

5:385

\_

5:386 06248 00000 AMPAA DC 5, 0,,, HIGH ORDER ADDRESS OF AMPMA

5:387

\_

5:388 06253 00000 A21A1 DC 5, 0,,, ARG1

\_

5:389 06258 00000 A21A2 DC 5, 0,,, ARG2

5:390

\_

5:391 06308 000000..000000 DC 50, 0

5:392 06309 000000..000000 DSC 50, 0

\_

5:393 06360 00 A21D1 DC 2, 0 ,,, FADD CALLING PROGRAM PRODUCT

5:394 06361 @ DC 1, @

5:395

\_

5:396 06366 00000 A21RC DC 5, 0,,, RETURN CODE

5:397

------ 6: fp40-fmul.sps --------------------------------------------------

6:1 \* ========================================================

6:2 \* FMUL TEST

6:3 \*

6:4 \*FMUL

6:5 \* ========================================================

6:6

6:7 06371 00005 DS 5,,, RETURN ADDRESS FROM BTM

6:8 06372 41 00000 00000 FMTST NOP 00000, 00000

6:9

6:10 06384 34 00000 00102 RCTY

6:11 06396 34 00000 00102 RCTY

6:12 06408 39 13551 00100 WATY FMULT,,, MAIN FMUL TEST TITLE

6:13 06420 34 00000 00102 RCTY

6:14

6:15

6:16 \* INITIALIZE TABLE POINTER

6:17

\_

6:18 06432 16 13644 00000 TFM FMINC, 0,, TABLE POINTER

6:19

6:20 \* POINT TO TEST DATA TABLES

6:21 \* RESET

\_

6:22 06444 16 13664 13694 TFM FMTBL, FMTBI

\_

6:23 06456 16 13669 13734 TFM FMTBL+5, FMTBP

\_

6:24 06468 16 13674 13774 TFM FMTBL+10, FMTBQ

6:25 06480 26 13659 13664 TF FMTP, FMTBL

6:26 06492 26 13664 13669 TF FMTP+5, FMTBL+5

6:27 06504 26 13669 13674 TF FMTP+10, FMTBL+10

6:28

6:29 06516 41 00000 00000 FMNXT NOP

6:30

6:31 06528 21 13659 13644 A FMTP, FMINC,, ID MESSAGE TABLE

\_

6:32 06540 14 13659 00000 CM FMTP, 00000, 6, RETURN ON ZERO ADDRESS

\_

6:33 06552 46 06371 01200 BE FMTST-1,,6, RETURN

6:34

6:35 06564 21 13664 13644 A FMTP+5, FMINC,, MULTIPLICAND TABLE

6:36 06576 21 13669 13644 A FMTP+10, FMINC,, MULTIPLIERS TABLE

6:37

\_

6:38 06588 16 13644 00005 TFM FMINC, 5

6:39

\_

6:40 06600 26 13679 13659 TF FMI, FMTP,11

\_

6:41 06612 26 13684 13664 TF FMP, FMTP+5,11

\_

6:42 06624 26 13689 13669 TF FMQ, FMTP+10,11

6:43

6:44 \* START OF CURRENT FMUL TEST

6:45 06636 41 00000 00000 FMRPT NOP

6:46 06648 34 00000 00102 RCTY

\_

6:47 06660 39 13679 00100 WATY FMI,,6, TEST TITLE

6:48 06672 34 00000 00101 SPTY

6:49

6:50 \*--- NEED TO PRESERVE MULTIPLICAND - MOVE TO FMPRD FIRST

6:51 \*--- AVOID TFL FOR FLOATING POINT TESTS

6:52

\_

6:53 06684 26 13911 13684 TF FMPRD, FMP, 11, FMP POINTS TO EXPONENT

\_

6:54 06696 12 13684 00002 SM FMP, 00002,, FMP NOW POINTS TO MANTISSA

\_

6:55 06708 26 13909 13684 TF FMPRD-2, FMP, 11,

\_

6:56 06720 11 13684 00002 AM FMP, 00002,, FMA AGAIN POINTS TO EXPONENT

6:57

\_

6:58 06732 03 13911 13689 FMUL FMPRD, FMQ, 11

6:59

6:60 \* INDICATOR CHECK

6:61 06744 47 06768 01400 BNV \*+24

6:62 06756 39 15015 00100 WATY OVM1,,, OVERFLOW MESSAGE

6:63 06768 47 06792 01500 BNXV \*+24

6:64 06780 39 15037 00100 WATY XCKM1,,, EXPONENT OVERFLOW MESSAGE

6:65

6:66 \* CALL SUBROUTINE M23

6:67 \* M (23) MANTASSAS AND COMPUTE EXPONENTS

\_

6:68 06792 17 06900 06824 BTM M23, \*+32,, TRANSMIT RETURN ADDRESS

\_ \_\_ \_

6:69 06808 136841..114955 DSA -FMP, -FMQ, FMPRD, STAT

6:70

6:71

6:72 \* CHECK COMPARE STATUS

6:73 \* CALL ERROR ROUTINE AND RETURN ITS STATUS IN ESTAT

\_

6:74 06824 17 09518 06846 BTM ERTN, \*+22,, TRANSMIT ADDRESS OF NEXT INSTRUCTION

\_ \_

6:75 06840 1495514954 DSA STAT, ESTAT

6:76

6:77 06846 43 06870 14954 BD FMEND, ESTAT

6:78 06858 49 06636 00000 B FMRPT,,, RERUN TEST

6:79

6:80 06870 41 00000 00000 FMEND NOP 00000, 00000

6:81 06882 49 06516 00000 B FMNXT,,, GOTO NEXT FMUL TEST

6:82

------ 7: fp41-fixmul.sps --------------------------------------------------

7:1 \* ---------------------------------------------

7:2 \* SUBROUTINE STATUS = M23(FP1, FP2, FP-PRODUCT)

7:3 \* COMPARES FMUL PRODUCT WITH M(23)/EX-SUM

7:4 \* REF (1) 227-5630-1 IBM 1620 FLOATING POINT FEATURE CE MANUAL 1962

7:5 \*

7:6 \* VERIFIES FMUL RESULT BY USING M(23) TO MULTIPLY MANTISSAS

7:7 \* AND SEPARATELY ADDS EXPONENTS.

7:8 \* FLOATING POINT NUMBER IS CREATED FROM M(23) PRODUCT AND

7:9 \* (ADJUSTED) SUM OF EXPONENTS

7:10 \*

7:11 \* HANDLES CASES DESCRIBED IN REF (1) FLOATING MULTIPLY EXAMPLES

7:12 \*

7:13 \* FMUL EXAMPLE 1. NO EXPONENT MODIFY REQUIRED

7:14 \* FMUL EXAMPLE 2. EXPONENT MODIFY REQUIRED

7:15 \* FMUL EXAMPLE 3. DIGIT FORCE -- MACHINE INFINITY

7:16 \* FMUL EXAMPLE 4. EXPONENT OVERFLOW -- MODIFY PE TO 99

7:17 \* FMUL EXAMPLE 5. SPECIAL CASE 999 X 199

7:18 \*

7:19 \*

7:20 \* USAGE

7:21 \* BTM M23, \*+32,, TRANSMIT RETURN ADDRESS

7:22 \* DSA -FMP, -FMQ, FMPRD, STAT

7:23 \* NOP 00000, 00000,, RETURN TO HERE

7:24 \*

7:25 \* M23A1 = -FMP (INDIRECT ADDRESS) FLOATING POINT NUMBER

7:26 \* M23A2 = -FMQ (INDIRECT ADDRESS) FLOATING POINT NUMBER

7:27 \* M23D1 = FMPRD (DATA FIELD) FLOATING POINT FMUL PRODUCT

7:28 \* M23RC = STAT ( DATA, RETURN CODE )

7:29 \* RETURNS STATUS OF COMPARE

7:30 \*

7:31 \*

7:32 \*M23

7:33 \* ---------------------------------------------

7:34 06898 00005 DS 5,, RETURN ADDRESS

7:35 06900 41 00000 00000 M23 NOP 00000, 00000

7:36

7:37 \* GET FIRST ARGUMENT (INDIRECT)

7:38 06912 26 06947 06899 TF \*+35, M23-1,,

\_

7:39 06924 12 06947 00016 SM \*+23, 00016,,

7:40 06936 26 06971 00000 TF \*+35, 0,,

7:41 06948 33 06971 00000 CF \*+23

7:42 06960 26 08498 00000 TF M23A1, 0,,

7:43

7:44

7:45 \* GET SECOND ARGUMENT (INDIRECT)

7:46 06972 26 07007 06899 TF \*+35, M23-1,,

\_

7:47 06984 12 07007 00011 SM \*+23, 00011,,

7:48 06996 26 07031 00000 TF \*+35, 0,,

7:49 07008 33 07031 00000 CF \*+23

7:50 07020 26 08503 00000 TF M23A2, 0,,

7:51

7:52

7:53 \* GET THIRD ARGUMENT

7:54 07032 26 07067 06899 TF \*+35, M23-1,,

\_

7:55 07044 12 07067 00006 SM \*+23, 00006,,

7:56 07056 26 07079 00000 TF \*+23, 0,,

7:57 07068 26 08605 00000 TF M23D1, 0,, -- GETS EXP

7:58

7:59 07080 26 07115 07079 TF \*+35, \*-1,,

\_

7:60 07092 12 07115 00002 SM \*+23, 00002,

7:61 07104 26 08603 00000 TF M23D1-2, 0,, -- GETS MANTISSA

7:62

7:63

7:64 \* ADD MP AND MQ EXPONENTS

7:65 07116 26 08159 08373 TF SUMEX, MZ5,, CLEAR

\_

7:66 07128 21 08159 08498 A SUMEX, M23A1, 11, EXPONENT 1

\_

7:67 07140 21 08159 08503 A SUMEX, M23A2, 11, EXPONENT 2

7:68

7:69 \* MANTISSA

\_

7:70 07152 12 08498 00002 SM M23A1, 00002

\_

7:71 07164 12 08503 00002 SM M23A2, 00002

\_

7:72 07176 26 07848 08498 TF MPMA, M23A1, 11, MANTISSA 1

\_

7:73 07188 26 07949 08503 TF MQMA, M23A2, 11, MANTISSA 2

7:74

7:75 \* MANTISSA LENGTH OF FIRST ARG

7:76 \* L REPRESENTS MANTISSA LENGTH

\_

7:77 07200 16 07748 00001 TFM FML, 00001

7:78 07212 26 08493 08498 TF MPMAA, M23A1

7:79

\_

7:80 07224 12 08493 00001 ML SM MPMAA, 1

\_

7:81 07236 11 07748 00001 AM FML, 1

\_

7:82 07248 44 07224 08493 BNF ML, MPMAA, 11

7:83

7:84 \* IF EXP SUM GT 100 THEN RESULT IS MACHINE INFINITY 99...9 99

7:85 \* IF EXP SUM EQ 100 ONLY SET EXP TO 99

\_

7:86 07260 14 08159 00100 CM SUMEX, 100

7:87 07272 47 07392 01300 BL MUL,,, SUM EXP LT 100

7:88 07284 46 07380 01200 BE ME100,,, SUM EXP EQ 100

7:89

7:90 07296 26 08159 08262 TF SUMEX, M9,, SUM EXP GT 100 SET SUM EXP TO 99

7:91 07308 26 08363 08260 TF MWK1-2, M9-2,, COPY 9S TO WORK AREA

7:92 07320 22 07350 07748 S \*+30, FML

\_

7:93 07332 11 07350 00001 AM \*+18, 00001,

7:94 07344 32 08363 00000 SF MWK1-2

7:95 07356 26 08050 08363 TF PQMA, MWK1-2,, 9S IN MANTISSA PRODUCT FIELD

7:96 07368 49 07596 00000 B MKFP,,, GOTO MAKE FP NUM FOR COMPARE

7:97

7:98 07380 26 08159 08262 ME100 TF SUMEX, M9,, SUM EXP EQ 100 SET SUM EXP TO 99

7:99

7:100 \* MULTIPLY MANTISSAS (P,Q) USING M (23)

\_

7:101 07392 26 08478 08424 MUL TF PAREA, MZ100,6, 100 ZEROS TO PRODUCT AREA (99)

7:102 07404 23 07848 07949 M MPMA, MQMA,, P MANTISSA X Q MANTISSA

7:103

7:104 \* CHECK FOR EXPONENT CASE - PRODUCT LENGTH LT 2L

7:105 \* HIGH ORDER 0

7:106 \* PROD-HIGH = 100 -2L

\_

7:107 07416 16 08483 00100 TFM PAR1, 00100,, PRODUCT AREA (99) + 1

7:108 07428 22 08483 07748 S PAR1, FML,, 100 - L

7:109 07440 22 08483 07748 S PAR1, FML,, 100 - L

\_

7:110 07452 43 07548 08483 BD PRSH, PAR1,11, IF HIGH ORDER 0 THEN MOVE FLAG

7:111 07464 26 08488 08483 TF PARW, PAR1

\_

7:112 07476 11 08488 00001 AM PARW, 00001,

\_ \_

7:113 07488 71 08488 08483 MF PARW, PAR1,611

\_

7:114 07500 12 07748 00001 SM FML, 00001,

7:115

\_

7:116 07512 14 08159 00099 CM SUMEX, 00099,, IF OV KEEP 99

7:117 07524 46 07548 01200 BE PRSH

\_

7:118 07536 12 08159 00001 SM SUMEX, 00001,, MODIFY EXP IF HI ORD DIGIT EQ 0

7:119

7:120 \* SHIFT PRODUCT L (MANTISSA LENGTH) AND MOVE SIGN FLAG

7:121 07548 26 08488 08478 PRSH TF PARW, PAREA

7:122 07560 22 08488 07748 S PARW, FML,,

\_ \_

7:123 07572 71 08488 08478 MF PARW, PAREA,611

7:124

7:125 \* MOVE PRODUCT TO RESULT AREA

\_

7:126 07584 26 08050 08488 TF PQMA, PARW , 11,

7:127

7:128 \* PUT PRODUCT AND EXPONENT SUM INTO FLOATING POINT MM...MM EE

7:129 07596 41 00000 00000 MKFP NOP 0000,0000

7:130 07608 71 08158 08155 MF SUMEX-1, SUMEX-4

7:131 07620 26 08153 08159 TF PQFP, SUMEX

7:132 07632 26 08151 08050 TF PQFP-2, PQMA

7:133

7:134 \* COMPARE FMUL PRODUCT TO M23 PRODUCT

7:135 \* M23RC EQ TRUE IF EQUAL OTW FALSE

\_

7:136 07644 17 09922 07672 BTM CF, \*+28,, TRANSMIT RETURN ADDRESS

\_ \_ \_

7:137 07660 086050..308607 DSA M23D1, PQFP, M23RC

7:138

7:139 \* SEND RETURN CODE TO CALLING PROGRAM

7:140 07672 26 07707 06899 TF \*+35, M23-1,,

\_

7:141 07684 12 07707 00001 SM \*+23, 1,,

7:142 07696 26 07714 00000 TF \*+18, 0,,

7:143 07708 25 00000 08607 TD 0, M23RC,,

7:144

7:145 07720 41 00000 00000 FMCR NOP 00000, 00000

7:146

7:147 \* RETURN

\_

7:148 07732 49 06899 00000 B M23-1,,6, RETURN TO CALLING PROGRAM

7:149

7:150 \* ---- -- WORK AREAS FOR M (23) INSTRUCTION

\_

7:151 07748 00000 FML DC 5, 0,,, MANTISSA LENGTH

7:152

\_

7:153 07798 000000..000000 DC 50, 0,,, MP MANTISSA FIELD

7:154 07799 000000..000000 DSC 49, 0,,,

7:155 07848 0 MPMA DSC 1, 0,,,

7:156 07849 @ DC 1, @

7:157

\_

7:158 07899 000000..000000 DC 50, 0,,, MQ MANTISSA FIELD

7:159 07900 000000..000000 DSC 49, 0,,,

7:160 07949 0 MQMA DSC 1, 0,,,

7:161 07950 @ DC 1, @

7:162

\_

7:163 08000 000000..000000 DC 50, 0,,, MP X MQ PRODUCT MANTISSA FIELD

7:164 08001 000000..000000 DSC 49, 0,,,

7:165 08050 0 PQMA DSC 1, 0,,,

7:166 08051 @ DC 1, @

7:167

\_

7:168 08101 000000..000000 DC 50, 0,,, MANTISSA + EXP

7:169 08102 000000..000000 DSC 50, 0

\_

7:170 08153 00 PQFP DC 2, 0

7:171 08154 @ DC 1, @

7:172

\_

7:173 08159 00000 SUMEX DC 5, 0,,, EXPONENT SUM FIELD

7:174 08160 @ DC 1, @

7:175

7:176 \* 123456789 123456789 123456789 123456789 1234567890

\_

7:177 08210 999999..999999 DC 50, 99999999999999999999999999999999999999999999999999

7:178 08211 999999..999999 DSC 50, 99999999999999999999999999999999999999999999999999

\_

7:179 08262 99 M9 DC 2, 99

7:180 08263 @ DC 1, @

7:181

\_

7:182 08313 000000..000000 DC 50, 0

7:183 08314 000000..000000 DSC 50, 0

\_

7:184 08365 00 MWK1 DC 2, 0

7:185 08366 @ DC 1, @

7:186

\_

7:187 08368 00 MZ2 DC 2, 0

\_

7:188 08373 00000 MZ5 DC 5, 0

\_

7:189 08423 000000..000000 DC 50, 0,,, 100 ZEROS

7:190 08424 000000..000000 MZ100 DSC 50, 0

7:191

7:192

\_

7:193 08478 00099 PAREA DC 5, 00099,, MACHINE PRODUCT AREA

\_

7:194 08483 00100 PAR1 DC 5, 00100,,

\_

7:195 08488 00099 PARW DC 5, 00099,, MACHINE PRODUCT AREA

7:196

\_

7:197 08493 00000 MPMAA DC 5, 0,,, HIGH ORDER ADDRESS OF MPMA

7:198

\_

7:199 08498 00000 M23A1 DC 5, 0,,, ARG1

\_

7:200 08503 00000 M23A2 DC 5, 0,,, ARG2

7:201

\_

7:202 08553 000000..000000 DC 50, 0

7:203 08554 000000..000000 DSC 50, 0

\_

7:204 08605 00 M23D1 DC 2, 0 ,,, FMUL CALLING PROGRAM PRODUCT

7:205 08606 @ DC 1, @

7:206

\_

7:207 08607 0 M23RC DC 1, 0,,, RETURN CODE

------ 8: fp50-subroutines.sps --------------------------------------------------

8:1 \* ==========================================================

8:2 \* SET SENSE SWITCHES SUBROUTINE

8:3 \* ==========================================================

8:4

8:5 08612 00005 DS 5,,, RETURN ADDRESS FROM BTM

8:6 08614 41 00000 00000 SETS1 NOP

8:7 \* CALL SUBROUTINE PRINT SENSE SWITCH STATUS

\_

8:8 08626 17 08704 08638 BTM SWSTA, \*+12,, TRANSMIT RETURN ADDRESS

8:9 08638 34 00000 00102 RCTY

8:10

8:11 \* SET SENSE SWITCHES AND CONTINUE

8:12 08650 39 09341 00100 WATY SWMS5

8:13 08662 34 00000 00102 RCTY

8:14 08674 48 00000 00000 H

8:15

\_

8:16 08686 49 08613 00000 B SETS1-1,,6, RETURN

8:17

8:18

8:19 \* ---------------------------------------------

8:20 \* SUBROUTINE PRINT SENSE SWITCH STATUS

8:21 \*

8:22 \*SWSTA

8:23 \* ---------------------------------------------

8:24

8:25 08702 00005 DS 5,, SWATA RETURN ADDRESS

8:26 08704 41 00000 00000 SWSTA NOP

8:27 08716 34 00000 00102 RCTY

8:28 08728 39 09185 00100 WATY SWMS2

8:29 08740 34 00000 00102 RCTY

8:30

8:31 08752 39 09245 00100 WATY SWMS3

8:32 08764 34 00000 00102 RCTY

8:33

8:34 08776 39 09291 00100 WATY SWMS4

8:35 08788 34 00000 00102 RCTY

8:36

8:37 08800 39 09065 00100 WATY SWST,,, SENSE SWITCH STATUS

8:38 08812 34 00000 00102 RCTY

8:39 08824 39 09105 00100 WATY SWMS1

8:40 08836 34 00000 00102 RCTY

8:41

8:42 08848 46 08884 00100 BC1 \*+36,,, SW1

8:43 08860 39 09169 00100 WATY OFF

8:44 08872 49 08896 00000 B \*+24

8:45 08884 39 09153 00100 WATY ON

8:46

8:47 08896 46 08932 00200 BC2 \*+36,,, SW2

8:48 08908 39 09169 00100 WATY OFF

8:49 08920 49 08944 00000 B \*+24

8:50 08932 39 09153 00100 WATY ON

8:51

8:52 08944 46 08980 00300 BC3 \*+36,,, SW3

8:53 08956 39 09169 00100 WATY OFF

8:54 08968 49 08992 00000 B \*+24

8:55 08980 39 09153 00100 WATY ON

8:56

8:57 08992 46 09028 00400 BC4 \*+36,,, SW4

8:58 09004 39 09169 00100 WATY OFF

8:59 09016 49 09040 00000 B \*+24

8:60 09028 39 09153 00100 WATY ON

8:61

8:62 09040 34 00000 00102 RCTY

8:63

\_

8:64 09052 49 08703 00000 B SWSTA-1,,6, RETURN TO CALLING PROGRAM

8:65

8:66

\_

8:67 09065 624555..64620@ SWST DAC 20,SENSE SWITCH STATUS@

8:68

\_

8:69 09105 007100..00740@ SWMS1 DAC 24, 1 2 3 4@

\_

8:70 09153 565500..00000@ ON DAC 8,ON @

\_

8:71 09169 564646..00000@ OFF DAC 8,OFF @

8:72

\_

8:73 09185 006266..56590@ SWMS2 DAC 30, SW1 ON = DO NOT PRINT ERROR@

\_

8:74 09245 006266..62630@ SWMS3 DAC 23, SW2 ON = REPEAT TEST@

\_

8:75 09291 006266..56590@ SWMS4 DAC 25, SW3 ON = HALT ON ERROR@

\_

8:76 09341 624563..00000@ SWMS5 DAC 43,SET SENSE SWITCHES THEN CONTINUE@

\_

8:77 09427 594559..00000@ SWMS6 DAC 43,RERUN MODE. SW2 OFF = EXIT RERUN@

8:78

8:79

8:80

8:81

8:82 \* ---------------------------------------------

8:83 \* SUBROUTINE FL COMMAND TESTS ERROR ROUTINE

8:84 \* ERRC = ERTN(FL-CMD-TEST-STAT)

8:85 \*

8:86 \* CALLING

8:87 \* BTM ERTN, \*+23,, TRANSMIT ADDRESS OF NEXT INSTRUCTION

8:88 \* DSA FL-CMD-TEST-STAT, ESTAT

8:89 \* NEXT INSTRUCTION

8:90 \*

8:91 \* ARGS

8:92 \* FL-CMD-TEST-STAT IS EA1

8:93 \*

8:94 \* RETURN CODE

8:95 \* ERRC 0 = RERUN TEST 1 = GOTO NEXT TEST

8:96 \*

8:97 \* ERROR ROUTINE

8:98 \* SW1 ON = DO NOT PRINT ERROR

8:99 \* SW2 ON = REPEAT TEST

8:100 \* SW3 ON = HALT ON ERROR

8:101 \*

8:102 \*ERTN

8:103 \* ---------------------------------------------

8:104

8:105 09516 00005 DS 5,, ERTN RETURN ADDRESS

8:106 09518 41 00000 00000 ERTN NOP

8:107

8:108

8:109 \* GET FIRST ARGUMENT

8:110 09530 26 09565 09517 TF \*+35, ERTN-1,,

\_

8:111 09542 12 09565 00006 SM \*+23, 6,,

8:112 09554 26 09577 00000 TF \*+23, 0,,

8:113 09566 26 09914 00000 TF EA1, 0,, FL-CMD-TEST-STAT 1=PASS 0=FAIL

8:114

8:115 09578 25 09915 14953 TD ERRC, FALSE,, RET CD DEFAULT IS RERUN TEST

8:116

8:117 09590 43 09662 09914 BD ERPAS, EA1,, CHECK PASS OR FAIL

8:118

8:119 09602 46 09626 00100 BC1 ERSW3

8:120 09614 39 14957 00100 WATY FAIL

8:121

8:122 09626 47 09686 00300 ERSW3 BNC3 ERSW2

8:123 09638 48 00000 00000 H 00000, 00000

8:124 09650 49 09686 00000 B ERSW2

8:125

8:126 09662 41 00000 00000 ERPAS NOP 00000, 00000

8:127 09674 39 14991 00100 WATY PASS

8:128

8:129 09686 41 00000 00000 ERSW2 NOP

8:130 09698 46 09734 00200 BC2 ERPSW,,, IF SW2 ON REPEAT TEST

8:131 09710 25 09915 14952 TD ERRC, TRUE,, GOTO NEXT TEST

8:132 09722 49 09842 00000 B ERRT,,, RETURN

8:133

8:134 \* CALL SUBROUTINE PRINT SENSE SWITCH STATUS

\_

8:135 09734 17 08704 09746 ERPSW BTM SWSTA, \*+12,, TRANSMIT RETURN ADDRESS

8:136 09746 34 00000 00102 RCTY

8:137

8:138 \* SET SENSE SWITCHES AND CONTINUE

8:139 09758 34 00000 00102 RCTY

8:140 09770 34 00000 00102 RCTY

8:141 09782 39 09427 00100 WATY SWMS6

8:142 09794 34 00000 00102 RCTY

8:143 09806 39 09341 00100 WATY SWMS5

8:144 09818 34 00000 00102 RCTY

8:145 09830 48 00000 00000 H

8:146

8:147

8:148 09842 41 00000 00000 ERRT NOP

8:149 \* MOVE RETURN CODE TO THIRD ARGUMENT

8:150 09854 26 09889 09517 TF \*+35, ERTN-1,,

\_

8:151 09866 12 09889 00001 SM \*+23, 1,,

8:152 09878 26 09896 00000 TF \*+18, 0,,

8:153 09890 25 00000 09915 TD 0, ERRC,,

8:154

8:155 \* RETURN

\_

8:156 09902 49 09517 00000 B ERTN-1,,6, RETURN TO CALLING PROGRAM

8:157

8:158 \* =============

\_

8:159 09914 0 EA1 DC 1, 0,,, FL-CMD-TEST-STAT

\_

8:160 09915 0 ERRC DC 1, 0,,, ERROR RTN RETURN CODE 0 = RERUN TEST 1 = GOTO NEXT TEST

8:161

8:162

8:163 \* ---------------------------------------------

8:164 \* SUBROUTINE STATUS = CF(FP1, FP2)

8:165 \* COMPARE FLOATING

8:166 \*

8:167 \* COMPARES TWO FLOATING POINT NUMBERS BY

8:168 \* COMPARING THE MANTISSA FIELDS AND EXPONENT FIELDS

8:169 \*

8:170 \* CALLING

8:171 \* BTM CF, \*+28,, TRANSMIT ADDRESS OF NEXT INSTRUCTION

8:172 \* DSA FP1, FP2, STATUS

8:173 \* NEXT INSTRUCTION

8:174 \*

8:175 \* CFA1 ADDRESS OF FLOATING POINT NUMBER 1

8:176 \*

8:177 \* CFA2 ADDRESS OF FLOATING POINT NUMBER 2

8:178 \*

8:179 \* CFRC ADDRESS OF RETURN CODE (STATUS)

8:180 \* TRUE = EQ, FALSE = NE

8:181 \*

8:182 \* RETURN ADDRESS (BTM) CF-1

8:183 \*

8:184 \*CF

8:185 \* ---------------------------------------------

8:186

8:187 09920 00005 DS 5,, CF RETURN ADDRESS

8:188 09922 41 00000 00000 CF NOP 00000, 00000

8:189

8:190 \* GET FIRST ARGUMENT

8:191 09934 26 09969 09921 TF \*+35, CF-1,,

\_

8:192 09946 12 09969 00012 SM \*+23, 12,,

8:193 09958 26 09981 00000 TF \*+23, 0,,

\_

8:194 09970 16 10310 00000 TFM CFA1, 0,,

8:195 09982 44 10030 09981 BNF CFSEC, \*-1,, IF NOT INDIRECT GET 2ND ARG

8:196 09994 26 10029 09981 TF \*+35, \*-13

8:197 10006 33 10029 00000 CF \*+23

8:198 10018 26 10310 00000 TF CFA1, 0

8:199

8:200 \* GET SECOND ARGUMENT

8:201 10030 26 10065 09921 CFSEC TF \*+35, CF-1,,

\_

8:202 10042 12 10065 00007 SM \*+23, 7,,

8:203 10054 26 10077 00000 TF \*+23, 0,,

\_

8:204 10066 16 10315 00000 TFM CFA2, 0,,

8:205 10078 44 10126 10077 BNF CFINI, \*-1,, IF NOT INDIRECT GOTO INIT RC

8:206 10090 26 10125 10077 TF \*+35, \*-13

8:207 10102 33 10125 00000 CF \*+23

8:208 10114 26 10315 00000 TF CFA2, 0

8:209

8:210

8:211 \* INITIALIZATION

8:212 10126 25 10316 14952 CFINI TD CFRC, TRUE,, SET RETURN CODE

8:213

8:214 \* COMPARE EXPONENTS

\_ \_

8:215 10138 24 10310 10315 C CFA1, CFA2, 611, COMPARE EXPS

8:216 10150 46 10174 01200 BE MANT,,, EXPS ARE EQUAL GOTO MANTISSA

8:217 10162 25 10316 14953 TD CFRC, FALSE,, EXPS ARE NE SET RETURN CODE

8:218

8:219 \* GET MANTISSA ADDRESSES

\_

8:220 10174 12 10310 00002 MANT SM CFA1, 00002,, BACKUP 2 TO GET MANTISSA OF A

\_

8:221 10186 12 10315 00002 SM CFA2, 00002,, BACKUP 2 TO GET MANTISSA OF B

8:222

8:223 \* COMPARE MANTASSAS

\_ \_

8:224 10198 24 10310 10315 C CFA1, CFA2,611

8:225 10210 46 10234 01200 BE CFRT,,, RETURN IF EQ

8:226 10222 25 10316 14953 TD CFRC, FALSE,, SET RETURN CODE

8:227

8:228 10234 41 00000 00000 CFRT NOP

8:229 \* MOVE RETURN CODE (CFRC) TO THIRD ARGUMENT

8:230 10246 26 10281 09921 TF \*+35, CF-1,,

\_

8:231 10258 12 10281 00002 SM \*+23, 2,,

8:232 10270 26 10288 00000 TF \*+18, 0,,

8:233 10282 25 00000 10316 TD 0, CFRC,,

8:234

8:235 \* RETURN

\_

8:236 10294 49 09921 00000 B CF-1,,6, RETURN TO CALLING PROGRAM

8:237

8:238 \* =============

\_

8:239 10310 00000 CFA1 DC 5, 0

\_

8:240 10315 00000 CFA2 DC 5, 0

\_

8:241 10316 0 CFRC DC 1, 0,,, RETURN CODE 1=TRUE 0=FALSE

8:242

8:243

------ 9: fp60-symbols-dend.sps --------------------------------------------------

9:1 \* ======================================================

9:2 \* ======================================================

9:3 \* ------------------------------------------------------

9:4 \*SYMBOLS

9:5 \* ------------------------------------------------------

9:6 \* ======================================================

9:7

\_

9:8 10319 465356..59630@ TITLE DAC 35,FLOATING POINT DIAGNOSTICS - START@

\_

9:9 10389 465356..55440@ DONE DAC 33,FLOATING POINT DIAGNOSTICS - END@

9:10

9:11

9:12 \* WORK AREAS

9:13

\_

9:14 10503 000000..000000 DC 50, 0

9:15 10504 000000..000000 DSC 50, 0

9:16 10554 00 WK1Z DSC 2, 0,,, TO ZERO WK1

9:17 10556 @ DC 1,@

9:18

9:19 10557 000000..000000 DSC 50, 0

9:20 10607 000000..000000 DSC 50, 0

\_

9:21 10658 00 WK1 DC 2, 0

9:22 10659 @ DC 1,@

9:23

9:24 10660 000000..000000 DSC 50, 0

9:25 10710 000000..000000 DSC 50, 0

\_

9:26 10761 00 WK2 DC 2, 0

9:27 10762 @ DC 1,@

9:28

9:29 \*TFL, BTFL, FSR, FSL DATA START

9:30

9:31

\_

9:32 10767 00000 TFINC DC 5, 0,, TABLE POINTER

9:33

9:34 \* TABLE POINTER

9:35 10777 00010 TFTBP DS 10

9:36

9:37 \* LIST OF DATA TABLES

\_ \_

9:38 10782 1080210837 TFTBL DSA TFTBI, TFTBD

9:39

9:40

9:41 10792 00005 TFI DS 5

9:42 10797 00005 TFD DS 5

9:43

9:44

9:45 \* TF TEST IDENTIFICATION MESSAGES

\_ \_ \_

9:46 10802 108691..100000 TFTBI DSA TFI1, TFI2, TFI3, TFI4, TFI5,

\_

9:47 10832 00000 DSA 00000

9:48

9:49 \* TF TEST DATA TABLE

\_ \_ \_

9:50 10837 110351..700000 TFTBD DSA TFD1, TFD2, TFD3, TFD4, TFD5,

\_

9:51 10867 00000 DSA 00000

9:52

9:53

9:54 \* TF TEST DATA

9:55

\_

9:56 10869 757000..79000@ TFI1 DAC 33,50 DIGIT NUMBER -9999...99 -99 @

9:57

9:58 10934 000000..000000 DSC 50, 0

\_ \_

9:59 11033 999999..999999 DC 50, -99999999999999999999999999999999999999999999999999

\_\_

9:60 11035 99 TFD1 DC 2, -99

9:61 11036 @ DC 1, @

9:62

\_

9:63 11039 757000..63000@ TFI2 DAC 49,50 DIGIT NEGATIVE NUMBER WITH NEGATIVE EXPONENT @

9:64

9:65 11136 000000..000000 DSC 50, 0

\_ \_

9:66 11235 123456..999999 DC 50, -12345678901234567890123459999999999999999999999999

\_\_

9:67 11237 04 TFD2 DC 2, -04

9:68 11238 @ DC 1, @

9:69

\_

9:70 11241 757000..59000@ TFI3 DAC 17,50 DIGIT NUMBER @

9:71

9:72 11274 000000..000000 DSC 50, 0

\_

9:73 11373 222222..567898 DC 50, 22222222222222222222222222222222222222221234567898

\_

9:74 11375 02 TFD3 DC 2, 02

9:75 11376 @ DC 1, @

9:76

9:77

\_

9:78 11379 757000..79000@ TFI4 DAC 29,50 DIGIT NUMBER - EXP = -99 @

9:79

9:80 11436 000000..000000 DSC 50, 0

\_

9:81 11535 000000..000000 DC 50, 00000000000000000000000000000000000000000000000000

\_\_

9:82 11537 99 TFD4 DC 2, -99

9:83 11538 @ DC 1, @

9:84

9:85

9:86

9:87 \* 123456789 123456789 123456789 123456789 1234567890

\_

9:88 11541 757000..00000@ TFI5 DAC 33,50 DIGIT NUMBER 9999...99 99 @

9:89

9:90 11606 000000..000000 DSC 50, 0

\_

9:91 11705 999999..999999 DC 50, 99999999999999999999999999999999999999999999999999

\_

9:92 11707 99 TFD5 DC 2, 99

9:93 11708 @ DC 1, @

9:94

9:95

9:96

9:97 \*TFL, BTFL, FSR, FSL DATA END

9:98

9:99

9:100 \*FADD TEST DATA START

9:101

9:102

9:103 \* FADD TESTS TABLE

9:104

9:105

9:106 \* NUMBER OF ELEMENTS IN TABLES

9:107 \* TABLE POINTER

\_

9:108 11713 00000 FAINC DC 5, 0,, TABLE POINTER

9:109

9:110 \* TABLE POINTER

9:111 11728 00015 ASTBL DS 15

9:112

9:113 \* LIST OF FADD DATA TABLES

\_ \_ \_

9:114 11733 117781..311888 FATBL DSA FATBI, FATBP, FATBQ

9:115

9:116 \* LIST OF FSUB DATA TABLES

\_ \_ \_

9:117 11748 119431..812053 FSTBL DSA FSTBI, FSTBP, FSTBQ

9:118

9:119

9:120 11763 00005 FAI DS 5

9:121 11768 00005 FAP DS 5

9:122 11773 00005 FAQ DS 5

9:123

9:124 \* FADD TEST IDENTIFICATION MESSAGES

\_ \_ \_

9:125 11778 122091..713439 FATBI DSA FAI1, FAI2, FAI3, FAI4, FAI5, FAI7, FAI8, FAI9, FAI10, FAI11

\_

9:126 11828 00000 DSA 00000

9:127

9:128 \* P DATA

\_ \_ \_

9:129 11833 123121..013542 FATBP DSA FAP1, FAP2, FAP3, FAP4, FAP5, FAP7, FAP8, FAP9, FAP10, FAP11

\_

9:130 11883 00000 DSA 00000

9:131

9:132 \* Q DATA

\_ \_ \_

9:133 11888 123181..613548 FATBQ DSA FAQ1, FAQ2, FAQ3, FAQ4, FAQ5, FAQ7, FAQ8, FAQ9, FAQ10, FAQ11

\_

9:134 11938 00000 DSA 00000

9:135

9:136

9:137 \*\*\*\*\*\*\*\*\*\* FSUB TEST DATA 7 AND 8 DIFFER -- P NEGATIVE

9:138 \* FSUB TEST IDENTIFICATION MESSAGES

\_ \_ \_

9:139 11943 122091..713439 FSTBI DSA FAI1, FAI2, FAI3, FAI4, FAI5, FSI7, FSI8, FAI9, FAI10, FAI11

\_

9:140 11993 00000 DSA 00000

9:141

9:142 \* P DATA

\_ \_ \_

9:143 11998 123121..013542 FSTBP DSA FAP1, FAP2, FAP3, FAP4, FAP5, FSP7, FSP8, FAP9, FAP10, FAP11

\_

9:144 12048 00000 DSA 00000

9:145

9:146 \* Q DATA

\_ \_ \_

9:147 12053 123181..613548 FSTBQ DSA FAQ1, FAQ2, FAQ3, FAQ4, FAQ5, FSQ7, FSQ8, FAQ9, FAQ10, FAQ11

\_

9:148 12103 00000 DSA 00000

9:149

9:150

9:151

9:152 \* FADD DATA SYMBOLS

9:153

\_

9:154 12153 000000..000000 DC 50, 0

9:155 12154 000000..000000 DSC 50, 0

\_

9:156 12205 00 FASUM DC 2, 0

9:157 12206 @ DC 1, @

9:158

9:159 \* REF 1. 227-5630-1 IBM 1620 FLOATING POINT FEATURE CE MANUAL 1962

9:160 \* EXAMPLES 1-5 EXPONENT CONFIGURATION

9:161 \* EXAMPLES 6-11 FRACTION (MANTISSA) CONFIGURATION

9:162 \* EXAMPLES 1 AND 6 SAME

9:163

9:164

\_

9:165 12209 456771..64630@ FAI1 DAC 50,EX1 PE = QE / D = 0 NO NORMALIZING - NO CARRY OUT@

9:166

\_

9:167 12310 123 DC 3, 123

\_

9:168 12312 04 FAP1 DC 2, 04

9:169 12313 @ DC 1, @

9:170

\_

9:171 12316 246 DC 3, 246

\_

9:172 12318 04 FAQ1 DC 2, 04

9:173 12319 @ DC 1, @

9:174

9:175

\_

9:176 12321 456772..43490@ FAI2 DAC 50,EX2 PE GT QE - D LT L QF SHIFTED RT TO ALIGN DECI@

9:177

\_

9:178 12422 123 DC 3, 123

\_

9:179 12424 06 FAP2 DC 2, 06

9:180 12425 @ DC 1, @

9:181

\_

9:182 12428 246 DC 3, 246

\_

9:183 12430 04 FAQ2 DC 2, 04

9:184 12431 @ DC 1, @

9:185

9:186

\_

9:187 12433 456773..00000@ FAI3 DAC 50,EX3 PE GT QE - D GT L - Q DISCARDED-RESULT IS P @

9:188

\_

9:189 12534 123 DC 3, 123

\_

9:190 12536 08 FAP3 DC 2, 08

9:191 12537 @ DC 1, @

9:192

\_

9:193 12540 246 DC 3, 246

\_

9:194 12542 03 FAQ3 DC 2, 03

9:195 12543 @ DC 1, @

9:196

9:197

\_

9:198 12545 456774..49540@ FAI4 DAC 50,EX4 PE LT QE- D LT L PF SHIFTED RT TO ALIGN DECIM@

9:199

\_

9:200 12646 123 DC 3, 123

\_

9:201 12648 02 FAP4 DC 2, 02

9:202 12649 @ DC 1, @

9:203

\_

9:204 12652 246 DC 3, 246

\_

9:205 12654 04 FAQ4 DC 2, 04

9:206 12655 @ DC 1, @

9:207

9:208

\_

9:209 12657 456775..00000@ FAI5 DAC 50,EX5 PE LT QE - D GT L P DISCARDED-RESULT IS Q @

9:210

\_

9:211 12758 123 DC 3, 123

\_

9:212 12760 03 FAP5 DC 2, 03

9:213 12761 @ DC 1, @

9:214

\_

9:215 12764 246 DC 3, 246

\_

9:216 12766 08 FAQ5 DC 2, 08

9:217 12767 @ DC 1, @

9:218

9:219

9:220 \* EX 6 NO NORMALIZING REQUIRED---NO CARRY OUT -- SAME AS EX1

9:221

\_

9:222 12769 456777..63000@ FAI7 DAC 50,EX7 NORMALIZING SHIFT RT - CARRY OUT- EXP ADJUST @

9:223

\_

9:224 12870 423 DC 3, 423

\_

9:225 12872 04 FAP7 DC 2, 04

9:226 12873 @ DC 1, @

9:227

\_

9:228 12876 745 DC 3, 745

\_

9:229 12878 04 FAQ7 DC 2, 04

9:230 12879 @ DC 1, @

9:231

9:232

\_

9:233 12881 456777..63000@ FSI7 DAC 49,EX7 (SUB) NORM SHIFT RT - CARRY OUT- EXP ADJUST @

9:234

9:235 \* FSUB TEST - P IS NEGATIVE

9:236

\_ \_

9:237 12980 423 DC 3, -423

\_

9:238 12982 04 FSP7 DC 2, 04

9:239 12983 @ DC 1, @

9:240

\_

9:241 12986 745 DC 3, 745

\_

9:242 12988 04 FSQ7 DC 2, 04

9:243 12989 @ DC 1, @

9:244

9:245

\_

9:246 12991 456778..55460@ FAI8 DAC 50,EX8 NORM SHIFT RT-CARRY OT CAUSES EXP OV MACH INF@

9:247

\_

9:248 13092 423 DC 3, 423

\_

9:249 13094 99 FAP8 DC 2, 99

9:250 13095 @ DC 1, @

9:251

\_

9:252 13098 745 DC 3, 745

\_

9:253 13100 99 FAQ8 DC 2, 99

9:254 13101 @ DC 1, @

9:255

\_

9:256 13103 456778..55460@ FSI8 DAC 50,EX8 (SUB) NORM SHIFT RT-CARRY OT- EXP OV MACH INF@

9:257

9:258 \* FSUB TEST - P IS NEGATIVE

9:259

\_ \_

9:260 13204 423 DC 3, -423

\_

9:261 13206 99 FSP8 DC 2, 99

9:262 13207 @ DC 1, @

9:263

\_

9:264 13210 745 DC 3, 745

\_

9:265 13212 99 FSQ8 DC 2, 99

9:266 13213 @ DC 1, @

9:267

\_

9:268 13215 456779..79790@ FAI9 DAC 50,EX9 ZERO FRACTION RESULT - MACHINE ZERO 0...0 -99@

9:269

\_

9:270 13316 345 DC 3, 345

\_

9:271 13318 04 FAP9 DC 2, 04

9:272 13319 @ DC 1, @

9:273

\_ \_

9:274 13322 345 DC 3, -345

\_

9:275 13324 04 FAQ9 DC 2, 04

9:276 13325 @ DC 1, @

9:277

9:278

\_

9:279 13327 456771..44510@ FAI10 DAC 50,EX10 NORMALIZING SHIFT LEFT - SHIFT LEFT- EXP ADJ@

9:280

\_

9:281 13428 345 DC 3, 345

\_

9:282 13430 04 FAP10 DC 2, 04

9:283 13431 @ DC 1, @

9:284

\_ \_

9:285 13434 321 DC 3, -321

\_

9:286 13436 04 FAQ10 DC 2, 04

9:287 13437 @ DC 1, @

9:288

9:289

\_

9:290 13439 456771..56000@ FAI11 DAC 50,EX11 NORMALIZING SHIFT LEFT- EXP OV MACHINE ZERO @

9:291

\_

9:292 13540 345 DC 3, 345

\_\_

9:293 13542 99 FAP11 DC 2, -99

9:294 13543 @ DC 1, @

9:295

\_ \_

9:296 13546 321 DC 3, -321

\_\_

9:297 13548 99 FAQ11 DC 2, -99

9:298 13549 @ DC 1, @

9:299

9:300 \*

9:301

9:302 \*FADD TEST DATA END --------------------

9:303

9:304

9:305 \*FMUL TEST DATA START

9:306

\_

9:307 13551 333333..33330@ FMULT DAC 45,====== FLOATING MULTIPLY (FMUL) TESTS ======@

9:308

9:309 \* FMUL TESTS TABLE

9:310

\_

9:311 13644 00000 FMINC DC 5, 0,, TABLE POINTER

9:312

9:313 \* TABLE POINTER

9:314 13659 00015 FMTP DS 15

9:315

9:316 \* LIST OF TABLES

\_ \_ \_

9:317 13664 136941..413774 FMTBL DSA FMTBI, FMTBP, FMTBQ

9:318

9:319 13679 00005 FMI DS 5

9:320 13684 00005 FMP DS 5

9:321 13689 00005 FMQ DS 5

9:322

9:323

9:324 \* FMUL TEST IDENTIFICATION MESSAGES

\_ \_ \_

9:325 13694 142151..114871 FMTBI DSA FMI1, FMI10, FMI11, FMI2, FMI3, FMI4, FMI5

\_

9:326 13729 00000 DSA 00000

9:327

9:328 \*MULTIPLICANDS

\_ \_ \_

9:329 13734 143181..214944 FMTBP DSA FMP1, FMP10, FMP11, FMP2, FMP3, FMP4, FMP5

\_

9:330 13769 00000 DSA 00000

9:331

9:332 \*MULTIPLIERS

\_ \_ \_

9:333 13774 143241..814950 FMTBQ DSA FMQ1, FMQ10, FMQ11, FMQ2, FMQ3, FMQ4, FMQ5

\_

9:334 13809 00000 DSA 00000

9:335

9:336

9:337 \* FMUL DATA SYMBOLS

9:338

\_

9:339 13859 000000..000000 DC 50, 0

9:340 13860 000000..000000 DSC 50, 0

\_

9:341 13911 00 FMPRD DC 2, 0

9:342 13912 @ DC 1, @

9:343

9:344

9:345 \* REF 227-5630-1 IBM 1620 FLOATING POINT FEATURE CE MANUAL 1962

9:346 \* --------------------------------------------------------

9:347

9:348 \* 123456789 123456789 123456789 123456789 1234567890

\_

9:349 13915 717070..04000@ FMI0 DAC 47,100 DIGIT NUMBER TEST +MP,-MQ, +PE,-MQ (FMUL) @

9:350 \* 123456789 123456789 123456789 123456789 123456789

\_

9:351 14057 123456..222222 DC 50, 12345678222222222222222222222222222222222222222222

9:352 14058 222222..222222 DSC 50, 22222222222222222222222222222222222222222222222222

\_

9:353 14109 99 FMP0 DC 2, 99

9:354 14110 @ DC 1, @

9:355

9:356 \* 123456789 123456789 123456789 123456789 123456789

\_

9:357 14160 123456..333333 DC 50, 12345669333333333333333333333333333333333333333333

9:358 14161 333333..333333 DSC 49, 3333333333333333333333333333333333333333333333333

\_

9:359 14210 3 DSC 1, -3

\_\_

9:360 14212 50 FMQ0 DC 2, -50

9:361 14213 @ DC 1, @

9:362

9:363

9:364 \* FMUL EXAMPLE 1. NO EXPONENT MODIFY REQUIRED

9:365

9:366 \* 123456789 123456789 123456789 123456789 1234567890

\_

9:367 14215 456771..04000@ FMI1 DAC 50,EX1A NO EXP MOD REQUIRED +MP,+MQ, +EP,+EQ (FMUL) @

9:368

\_

9:369 14316 345 DC 3, 345

\_

9:370 14318 07 FMP1 DC 2, 07

9:371 14319 @ DC 1, @

9:372

\_

9:373 14322 432 DC 3, 432

\_

9:374 14324 03 FMQ1 DC 2, 03

9:375 14325 @ DC 1, @

9:376

9:377

\_

9:378 14327 456771..04000@ FMI10 DAC 50,EX1B NO EXP MOD REQUIRED -MP,+MQ, -EP,+EQ (FMUL) @

9:379

\_ \_

9:380 14428 345 DC 3, -345

\_\_

9:381 14430 07 FMP10 DC 2, -07

9:382 14431 @ DC 1, @

9:383

\_

9:384 14434 432 DC 3, 432

\_

9:385 14436 03 FMQ10 DC 2, 03

9:386 14437 @ DC 1, @

9:387

9:388

\_

9:389 14439 456771..04000@ FMI11 DAC 50,EX1C NO EXP MOD REQUIRED -MP,-MQ, -EP,-EQ (FMUL) @

9:390

\_ \_

9:391 14540 345 DC 3, -345

\_\_

9:392 14542 07 FMP11 DC 2, -07

9:393 14543 @ DC 1, @

9:394

\_ \_

9:395 14546 432 DC 3, -432

\_\_

9:396 14548 03 FMQ11 DC 2, -03

9:397 14549 @ DC 1, @

9:398

9:399

9:400 \* FMUL EXAMPLE 2. EXPONENT MODIFY REQUIRED

9:401

9:402 \* 123456789 123456789 123456789 123456789 1234567890

\_

9:403 14551 456772..04000@ FMI2 DAC 49,EX2 EXPONENT MOD REQ-ED +MP,+MQ, +EP,+EQ (FMUL) @

9:404

\_

9:405 14650 123 DC 3, 123

\_

9:406 14652 02 FMP2 DC 2, 02

9:407 14653 @ DC 1, @

9:408

\_

9:409 14656 246 DC 3, 246

\_

9:410 14658 04 FMQ2 DC 2, 04

9:411 14659 @ DC 1, @

9:412

9:413

9:414 \* FMUL EXAMPLE 3. DIGIT FORCE -- MACHINE INFINITY

9:415

9:416 \* 123456789 123456789 123456789 123456789 1234

\_

9:417 14661 456773..04000@ FMI3 DAC 44,EX3 DIGIT FORCE -- MACHINE INFINITY (FMUL) @

9:418

\_

9:419 14750 345 DC 3, 345

\_

9:420 14752 43 FMP3 DC 2, 43

9:421 14753 @ DC 1, @

9:422

\_

9:423 14756 432 DC 3, 432

\_

9:424 14758 64 FMQ3 DC 2, 64

9:425 14759 @ DC 1, @

9:426

9:427

9:428 \* FMUL EXAMPLE 4. EXPONENT OVERFLOW -- MODIFY PE TO 99

9:429

9:430 \* 123456789 123456789 123456789 123456789 123456789

\_

9:431 14761 456774..04000@ FMI4 DAC 49,EX4 EXPONENT OVERFLOW -- MODIFY PE TO 99 (FMUL) @

9:432

\_

9:433 14860 123 DC 3, 123

\_

9:434 14862 38 FMP4 DC 2, 38

9:435 14863 @ DC 1, @

9:436

\_

9:437 14866 246 DC 3, 246

\_

9:438 14868 62 FMQ4 DC 2, 62

9:439 14869 @ DC 1, @

9:440

9:441

9:442 \* FMUL EXAMPLE 5. SPECIAL CASE 999 X 199

9:443

9:444 \* 123456789 123456789 123456789 123456789 12345

\_

9:445 14871 456775..04000@ FMI5 DAC 35,EX5 SPECIAL CASE 999 X 199 (FMUL) @

9:446

\_

9:447 14942 999 DC 3, 999

\_

9:448 14944 01 FMP5 DC 2, 01

9:449 14945 @ DC 1, @

9:450

\_

9:451 14948 199 DC 3, 199

\_

9:452 14950 01 FMQ5 DC 2, 01

9:453 14951 @ DC 1, @

9:454

9:455

9:456 \*FMUL TEST DATA END --------------------

9:457

9:458 \* CONSTANTS AND STATUS CODES

\_

9:459 14952 1 TRUE DC 1,1

\_

9:460 14953 0 FALSE DC 1,0

9:461

\_

9:462 14954 0 ESTAT DC 1,0,,, FL COMMAND TEST ERR RTN STATUS

\_

9:463 14955 0 STAT DC 1,0,,, FL COMMAND TEST STATUS

9:464

\_

9:465 14957 333333..33330@ FAIL DAC 17,==== FAILED ====@

\_

9:466 14991 202057..20200@ PASS DAC 11,--PASSED--@

9:467

9:468

\_

9:469 15012 0 OV DC 1,0,,, OVERFLOW FLAG

\_

9:470 15013 0 XCK DC 1,0,,, EXPONENT OVERFLOW FLAG

9:471

\_

9:472 15015 005665..66000@ OVM1 DAC 11, OVERFLOW @

\_

9:473 15037 004567..66000@ XCKM1 DAC 20, EXPONENT OVERFLOW @

9:474

9:475

9:476 \* -------------------------------------------------

9:477 \* -------------------------------------------------

9:478 \* -------------------------------------------------

9:479 00402 DEND FPMAIN

Symbol Cross-Reference Table

============================

Id Source File

-- --------------------------------------------------

1 fp10-main.sps

2 fp20-tftest.sps

3 fp21-tftest.sps

4 fp30-fadd.sps

5 fp31-fixadd.sps

6 fp40-fmul.sps

7 fp41-fixmul.sps

8 fp50-subroutines.sps

9 fp60-symbols-dend.sps

Symbol Addr. Type Defined References

------- ----- ------ ------- -------------------------------------------------------------

A21 03216 <inst> 5:45 4:101 5:49 5:58 5:66 5:74 5:305 5:312

A21A1 06253 DC 5:388 5:62 5:90 5:102 5:106 5:113

A21A2 06258 DC 5:389 5:70 5:91 5:103 5:107

A21D1 06360 DC 5:393 5:77 5:81 5:301

A21RC 06366 DC 5:396 5:301 5:308

A9 05932 DC 5:366 5:272 5:275

ACMF1 04704 <inst> 5:238 5:229 5:230

ACMF2 04752 <inst> 5:242 5:231 5:232

ACO 04596 <inst> 5:229 5:222

ADDM 04104 <inst> 5:165 5:122 5:142 5:161

ADIGT 04404 <inst> 5:204 5:208

ADJP 03780 <inst> 5:129 5:123

ADJP1 03840 <inst> 5:135 5:130

ADJQ 03936 <inst> 5:147 5:124

ADJQ1 04008 <inst> 5:154 5:149

ADSA 03118 DSA 4:102 4:16

AED 05281 DC 5:320 5:95 5:96 5:97 5:98 5:129 5:136 5:148

5:155

AELS 05088 <inst> 5:284 5:259

AEOVM 05667 DAC 5:356

AEP 05403 DC 5:332 5:88 5:90 5:96 5:121 5:151 5:255 5:256

AEPQ 05619 DC 5:348 5:132 5:151 5:223 5:254 5:255 5:258 5:263

5:265 5:284 5:285 5:287 5:293 5:293 5:294

AEQ 05511 DC 5:340 5:89 5:91 5:97 5:121 5:132 5:256 5:258

AFLAG 04368 <inst> 5:200 5:201

AFLG 05286 DC 5:322 5:197 5:206 5:214

AINF 04968 <inst> 5:270 5:264

AINF1 05028 <inst> 5:275 5:271

AINF2 05064 <inst> 5:278 5:270

AL 03696 <inst> 5:115 5:117

ALDON 01146 <inst> 1:154

AMKFP 05148 <inst> 5:293 5:133 5:152 5:224 5:261 5:266 5:279 5:286

AML 05276 DC 5:319 5:112 5:116 5:129 5:139 5:148 5:158 5:191

5:235 5:237 5:239 5:241 5:276

AMP 05397 DSC 5:329 5:104 5:106 5:135 5:137 5:137 5:138 5:141

5:141 5:150 5:169 5:229 5:230 5:231 5:232

5:238 5:238 5:244

AMPA 05296 DC 5:325 5:113 5:115 5:117 5:198 5:200 5:201 5:202

5:204 5:207 5:211

AMPAA 06248 DC 5:386

AMPQ 05613 DSC 5:345 5:131 5:150 5:168 5:169 5:170 5:184 5:190

5:193 5:198 5:210 5:212 5:213 5:215 5:221

5:243 5:244 5:245 5:246 5:246 5:247 5:247

5:270 5:271 5:272 5:273 5:275 5:278 5:288

5:295

AMQ 05505 DSC 5:337 5:105 5:107 5:131 5:154 5:156 5:156 5:157

5:160 5:160 5:170 5:242 5:242 5:245

ANRM 05622 DC 5:352 5:166 5:187 5:196 5:259

AOV 05621 DC 5:351 5:167 5:175 5:233 5:260

AOVM 05625 DAC 5:355

APQFP 05829 DC 5:359

ASF 04464 <inst> 5:210 5:204

ASF2 04524 <inst> 5:215 5:210

ASHFT 04212 <inst> 5:183 5:174

ASTBL 11728 DS 9:111 4:42 4:43 4:44 4:52 4:53 4:54 4:60

4:61 4:64 4:65 4:69 4:70 4:71

AWK1 06034 DC 5:371 5:195 5:212 5:215

AZ100 06142 DSC 5:379

AZ101 06243 DSC 5:383 5:104 5:105 5:168 5:184 5:195 5:213 5:221

5:243 5:288

AZ2 06037 DC 5:374

AZ5 06042 DC 5:375 5:88 5:89 5:95 5:254

AZCK 04548 <inst> 5:221 5:176 5:193

AZCNT 05291 DC 5:323 5:188 5:205 5:284

AZRO 04572 <inst> 5:223 5:185

BTSUB 02392 <inst> 3:18 2:131 3:24

CF 09922 <inst> 8:188 3:23 5:300 7:136 8:191 8:201 8:230 8:236

CFA1 10310 DC 8:239 8:194 8:198 8:215 8:220 8:224

CFA2 10315 DC 8:240 8:204 8:208 8:215 8:221 8:224

CFINI 10126 <inst> 8:212 8:205

CFRC 10316 DC 8:241 8:212 8:217 8:226 8:233

CFRT 10234 <inst> 8:228 8:225

CFSEC 10030 <inst> 8:201 8:195

DONE 10389 DAC 9:9 1:157

EA1 09914 DC 8:159 8:113 8:117

ERPAS 09662 <inst> 8:126 8:117

ERPSW 09734 <inst> 8:135 8:130

ERRC 09915 DC 8:160 8:115 8:131 8:153

ERRT 09842 <inst> 8:148 8:132

ERSW2 09686 <inst> 8:129 8:122 8:124

ERSW3 09626 <inst> 8:122 8:119

ERTN 09518 <inst> 8:106 2:141 4:107 6:74 8:110 8:150 8:156

ESTAT 14954 DC 9:462 2:142 2:144 4:108 4:110 6:75 6:77

EXADJ 04824 <inst> 5:254 5:216 5:234

FACR 05260 <inst> 5:312

FAEND 03186 <inst> 4:113 4:110

FAI 11763 DS 9:120 4:69 4:76

FAI1 12209 DAC 9:165 9:125 9:139

FAI10 13327 DAC 9:279 9:125 9:139

FAI11 13439 DAC 9:290 9:125 9:139

FAI2 12321 DAC 9:176 9:125 9:139

FAI3 12433 DAC 9:187 9:125 9:139

FAI4 12545 DAC 9:198 9:125 9:139

FAI5 12657 DAC 9:209 9:125 9:139

FAI7 12769 DAC 9:222 9:125

FAI8 12991 DAC 9:246 9:125

FAI9 13215 DAC 9:268 9:125 9:139

FAIL 14957 DAC 9:465 8:120

FAINC 11713 DC 9:108 4:32 4:60 4:64 4:65 4:67

FALSE 14953 DC 9:460 2:65 5:166 5:167 5:187 8:115 8:217 8:226

FANXT 02826 <inst> 4:57 4:45 4:115

FAP 11768 DS 9:121 4:70 4:82 4:83 4:84 4:85 4:102

FAP1 12312 DC 9:168 9:129 9:143

FAP10 13430 DC 9:282 9:129 9:143

FAP11 13542 DC 9:293 9:129 9:143

FAP2 12424 DC 9:179 9:129 9:143

FAP3 12536 DC 9:190 9:129 9:143

FAP4 12648 DC 9:201 9:129 9:143

FAP5 12760 DC 9:212 9:129 9:143

FAP7 12872 DC 9:225 9:129

FAP8 13094 DC 9:249 9:129

FAP9 13318 DC 9:271 9:129 9:143

FAQ 11773 DS 9:122 4:71 4:89 4:102

FAQ1 12318 DC 9:172 9:133 9:147

FAQ10 13436 DC 9:286 9:133 9:147

FAQ11 13548 DC 9:297 9:133 9:147

FAQ2 12430 DC 9:183 9:133 9:147

FAQ3 12542 DC 9:194 9:133 9:147

FAQ4 12654 DC 9:205 9:133 9:147

FAQ5 12766 DC 9:216 9:133 9:147

FAQ7 12878 DC 9:229 9:133

FAQ8 13100 DC 9:253 9:133

FAQ9 13324 DC 9:275 9:133 9:147

FARIT 03042 <inst> 4:89 4:12 4:34

FARPT 02946 <inst> 4:74 4:111

FASUM 12205 DC 9:156 4:82 4:84 4:89 4:102

FATBI 11778 DSA 9:125 4:39 9:114

FATBL 11733 DSA 9:114 4:39 4:40 4:41 4:42 4:43 4:44

FATBP 11833 DSA 9:129 4:40 9:114

FATBQ 11888 DSA 9:133 4:41 9:114

FATST 02466 <inst> 4:7 1:137 1:144 4:10 4:14 4:18 4:62

FIRST 00462 <inst> 1:111

FMCR 07720 <inst> 7:145

FMEND 06870 <inst> 6:80 6:77

FMI 13679 DS 9:319 6:40 6:47

FMI0 13915 DAC 9:349

FMI1 14215 DAC 9:367 9:325

FMI10 14327 DAC 9:378 9:325

FMI11 14439 DAC 9:389 9:325

FMI2 14551 DAC 9:403 9:325

FMI3 14661 DAC 9:417 9:325

FMI4 14761 DAC 9:431 9:325

FMI5 14871 DAC 9:445 9:325

FMINC 13644 DC 9:311 6:18 6:31 6:35 6:36 6:38

FML 07748 DC 7:151 7:77 7:81 7:92 7:108 7:109 7:114 7:122

FMNXT 06516 <inst> 6:29 6:81

FMP 13684 DS 9:320 6:41 6:53 6:54 6:55 6:56 6:69

FMP0 14109 DC 9:353

FMP1 14318 DC 9:370 9:329

FMP10 14430 DC 9:381 9:329

FMP11 14542 DC 9:392 9:329

FMP2 14652 DC 9:406 9:329

FMP3 14752 DC 9:420 9:329

FMP4 14862 DC 9:434 9:329

FMP5 14944 DC 9:448 9:329

FMPRD 13911 DC 9:341 6:53 6:55 6:58 6:69

FMQ 13689 DS 9:321 6:42 6:58 6:69

FMQ0 14212 DC 9:360

FMQ1 14324 DC 9:374 9:333

FMQ10 14436 DC 9:385 9:333

FMQ11 14548 DC 9:396 9:333

FMQ2 14658 DC 9:410 9:333

FMQ3 14758 DC 9:424 9:333

FMQ4 14868 DC 9:438 9:333

FMQ5 14950 DC 9:452 9:333

FMRPT 06636 <inst> 6:45 6:78

FMTBI 13694 DSA 9:325 6:22 9:317

FMTBL 13664 DSA 9:317 6:22 6:23 6:24 6:25 6:26 6:27

FMTBP 13734 DSA 9:329 6:23 9:317

FMTBQ 13774 DSA 9:333 6:24 9:317

FMTP 13659 DS 9:314 6:25 6:26 6:27 6:31 6:32 6:35 6:36

6:40 6:41 6:42

FMTST 06372 <inst> 6:8 1:150 6:33

FMULT 13551 DAC 9:307 6:12

FPMAIN 00402 <inst> 1:98 9:479 9:479

FSI7 12881 DAC 9:233 9:139

FSI8 13103 DAC 9:256 9:139

FSP7 12982 DC 9:238 9:143

FSP8 13206 DC 9:261 9:143

FSQ7 12988 DC 9:242 9:147

FSQ8 13212 DC 9:265 9:147

FSTBI 11943 DSA 9:139 4:49 9:117

FSTBL 11748 DSA 9:117 4:49 4:50 4:51 4:52 4:53 4:54

FSTBP 11998 DSA 9:143 4:50 9:117

FSTBQ 12053 DSA 9:147 4:51 9:117

FXAR1 04164 <inst> 5:170 5:52

FXAR2 04788 <inst> 5:245 5:54

M23 06900 <inst> 7:35 6:68 7:38 7:46 7:54 7:140 7:148

M23A1 08498 DC 7:199 7:42 7:66 7:70 7:72 7:78

M23A2 08503 DC 7:200 7:50 7:67 7:71 7:73

M23D1 08605 DC 7:204 7:57 7:61 7:137

M23RC 08607 DC 7:207 7:137 7:143

M9 08262 DC 7:179 7:90 7:91 7:98

MANT 10174 <inst> 8:220 8:216

ME100 07380 <inst> 7:98 7:88

MKFP 07596 <inst> 7:129 7:96

ML 07224 <inst> 7:80 7:82

MPMA 07848 DSC 7:155 7:72 7:102

MPMAA 08493 DC 7:197 7:78 7:80 7:82

MQMA 07949 DSC 7:160 7:73 7:102

MUL 07392 <inst> 7:101 7:87

MWK1 08365 DC 7:184 7:91 7:94 7:95

MZ100 08424 DSC 7:190 7:101

MZ2 08368 DC 7:187

MZ5 08373 DC 7:188 7:65

OFF 09169 DAC 8:71 8:43 8:48 8:53 8:58

ON 09153 DAC 8:70 8:45 8:50 8:55 8:60

OV 15012 DC 9:469

OVM1 15015 DAC 9:472 4:94 6:62

PAR1 08483 DC 7:194 7:107 7:108 7:109 7:110 7:111 7:113

PAREA 08478 DC 7:193 7:101 7:121 7:123

PARW 08488 DC 7:195 7:111 7:112 7:113 7:121 7:122 7:123 7:126

PASS 14991 DAC 9:466 8:127

PQFP 08153 DC 7:170 5:294 5:295 5:301 7:131 7:132 7:137

PQMA 08050 DSC 7:165 7:95 7:126 7:132

PRSH 07548 <inst> 7:121 7:110 7:117

RETAD 02460 DS 3:35 2:130 3:31

SETS1 08614 <inst> 8:6 1:113 8:16

STAT 14955 DC 9:463 2:65 2:135 2:142 3:24 4:102 4:108 6:69

6:75

SUBTB 02754 <inst> 4:49 4:35

SUMEX 08159 DC 7:173 7:65 7:66 7:67 7:86 7:90 7:98 7:116

7:118 7:130 7:130 7:131

SWMS1 09105 DAC 8:69 8:39

SWMS2 09185 DAC 8:73 8:28

SWMS3 09245 DAC 8:74 8:31

SWMS4 09291 DAC 8:75 8:34

SWMS5 09341 DAC 8:76 8:12 8:143

SWMS6 09427 DAC 8:77 8:141

SWST 09065 DAC 8:67 8:37

SWSTA 08704 <inst> 8:26 8:8 8:64 8:135

TBTFL 02112 <inst> 2:126 2:106

TER 02220 <inst> 2:141 2:81 2:101 2:122 2:134

TEXCK 02184 <inst> 2:133 2:71 2:80 2:100 2:121

TFD 10797 DS 9:42 2:53 2:71 2:76 2:77 2:91 2:95 2:107

2:110 2:112 2:118 2:126 2:127

TFD1 11035 DC 9:60 9:50

TFD2 11237 DC 9:67 9:50

TFD3 11375 DC 9:74 9:50

TFD4 11537 DC 9:82 9:50

TFD5 11707 DC 9:92 9:50

TFEND 02266 <inst> 2:147 2:144

TFI 10792 DS 9:41 2:52 2:57

TFI1 10869 DAC 9:56 9:46

TFI2 11039 DAC 9:63 9:46

TFI3 11241 DAC 9:70 9:46

TFI4 11379 DAC 9:78 9:46

TFI5 11541 DAC 9:88 9:46

TFINC 10767 DC 9:32 2:30 2:44 2:48 2:50

TFINS 01284 <inst> 2:13 2:12 2:74 2:89 2:105

TFNXT 01440 <inst> 2:40 2:149

TFRPT 01536 <inst> 2:56 2:145

TFSL 01740 <inst> 2:89 2:75

TFSLC 01860 <inst> 2:99 2:95 2:96 2:97 2:97

TFSR 01896 <inst> 2:105 2:90

TFSRC 02076 <inst> 2:120 2:118 2:119

TFSRF 02040 <inst> 2:117 2:114

TFTBD 10837 DSA 9:50 2:35 9:38

TFTBI 10802 DSA 9:46 2:34 9:38

TFTBL 10782 DSA 9:38 2:34 2:35 2:36 2:37

TFTBP 10777 DS 9:35 2:36 2:37 2:44 2:45 2:48 2:52 2:53

TFTST 01236 <inst> 2:7 1:116 1:121 1:126 1:131 2:10 2:16 2:46

TFWT 01356 <inst> 2:24 2:18

TITLE 10319 DAC 9:8 1:102

TRUE 14952 DC 9:459 2:135 5:175 5:196 8:131 8:212

WK1 10658 DC 9:21 2:66 2:67 2:76 2:79 2:93 2:94 2:94

2:98 2:99 2:109 2:111 2:115 2:116 2:117

2:117 2:120 2:129 2:133 3:24

WK1Z 10554 DSC 9:16 2:66 2:68

WK2 10761 DC 9:26 2:68 2:69

WTITL 02610 <inst> 4:26 4:20

XCK 15013 DC 9:470

XCKM1 15037 DAC 9:473 4:96 6:64

**CU03 - Indirect Addressing Diagnostic**

//==============================================================================

//

// CU03 - Indirect Addressing Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Bypass error type out

// OFF - Type out routine number on error

// PS2: ON - Loop in routine

// OFF - Continue to next routine

// PS3: ON - Stop on error

// OFF - Do not stop on error, continue

// PS4: ON - Repeat test CU03

// OFF - Run test CU03 once

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - STOP

//

// Start addresses:

//

// 00828 - Full test

//

// Directions:

//

// 1. Load CU03 diagnostic

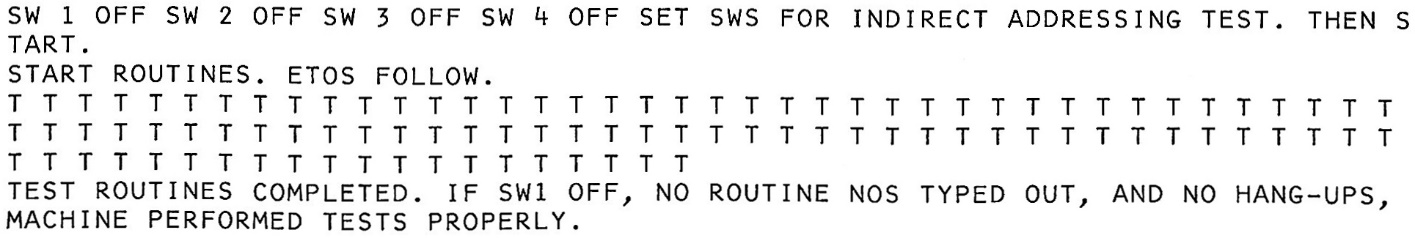
// 2. Press START

// 3. Press START

//

//==============================================================================

Sample Output – CU03



**CS01 - Check Stops Diagnostic**

//==============================================================================

//

// CS01 - Check Stops Diagnostic

//

// Program Switch settings:

//

// PS1: not used

// PS2: not used

// PS3: not used

// PS4: not used

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - STOP

//

// Start addresses:

//

// 00402 - Full test

//

// Directions:

//

// 1. Load CS01 diagnostic

// 2. Press START

// 3. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/C/C-8-2

// 4. Press RESET

// 5. Press START

// 6. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/C-8-2/C

// 7. Press RESET

// 8. Press START

// 9. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C-8-2/C/C

// 10. Press RESET

// 11. Press START

// 12. Verify CHECK STOP & MAR CHK are on and MAR = C/C-8-2/C/C/C

// 13. Press RESET

// 14. Press START

// 15. Verify CHECK STOP & MAR CHK are on and MAR = C-8-2/C/C/C/C

// 16. Press RESET

// 17. Press START

// 18. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/C/8-2-1

// 19. Press RESET

// 20. Press START

// 21. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/8-2-1/C

// 22. Press RESET

// 23. Press START

// 24. Verify CHECK STOP & MAR CHK are on and MAR = C/C/8-2-1/C/C

// 25. Press RESET

// 26. Press START

// 27. Verify CHECK STOP & MAR CHK are on and MAR = C/8-2-1/C/C/C

// 28. Press RESET

// 29. Press START

// 30. Verify CHECK STOP & MAR CHK are on and MAR = 8-2-1/C/C/C/C

// 31. Press RESET

// 32. Press START

// 33. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/C/C-8-4

// 34. Press RESET

// 35. Press START

// 36. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/C-8-4/C

// 37. Press RESET

// 38. Press START

// 39. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C-8-4/C/C

// 40. Press RESET

// 41. Press START

// 42. Verify CHECK STOP & MAR CHK are on and MAR = C/C-8-4/C/C/C

// 43. Press RESET

// 44. Press START

// 45. Verify CHECK STOP & MAR CHK are on and MAR = C-8-4/C/C/C/C

// 46. Press RESET

// 47. Press START

// 48. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/C/8-4-1

// 49. Press RESET

// 50. Press START

// 51. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/8-4-1/C

// 52. Press RESET

// 53. Press START

// 54. Verify CHECK STOP & MAR CHK are on and MAR = C/C/8-4-1/C/C

// 55. Press RESET

// 56. Press START

// 57. Verify CHECK STOP & MAR CHK are on and MAR = C/8-4-1/C/C/C

// 58. Press RESET

// 59. Press START

// 60. Verify CHECK STOP & MAR CHK are on and MAR = 8-4-1/C/C/C/C

// 61. Press RESET

// 62. Press START

// 63. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/C/8-4-2

// 64. Press RESET

// 65. Press START

// 66. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/8-4-2/C

// 67. Press RESET

// 68. Press START

// 69. Verify CHECK STOP & MAR CHK are on and MAR = C/C/8-4-2/C/C

// 70. Press RESET

// 71. Press START

// 72. Verify CHECK STOP & MAR CHK are on and MAR = C/8-4-2/C/C/C

// 73. Press RESET

// 74. Press START

// 75. Verify CHECK STOP & MAR CHK are on and MAR = 8-4-2/C/C/C/C

// 76. Press RESET

// 77. Press START

// 78. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/C/C-8-4-2-1

// 79. Press RESET

// 80. Press START

// 81. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C/C-8-4-2-1/C

// 82. Press RESET

// 83. Press START

// 84. Verify CHECK STOP & MAR CHK are on and MAR = C/C/C-8-4-2-1/C/C

// 85. Press RESET

// 86. Press START

// 87. Verify CHECK STOP & MAR CHK are on and MAR = C/C-8-4-2-1/C/C/C

// 88. Press RESET

// 89. Press START

// 90. Verify CHECK STOP & MAR CHK are on and MAR = C-8-4-2-1/C/C/C/C

// 91. Press RESET

// 92. Press START

// 93. Verify CHECK STOP is on and OPERATION REGISTER = 7/7

// 94. Press RESET

// 95. Press START

// 96. Verify CHECK STOP is on and OPERATION REGISTER = C/C-8-4-2-1

// 97. Press RESET

// 98. Press START

// 99. Verify CHECK STOP is on and OPERATION REGISTER = C-8-4-2-1/C

// 100. Press RESET

// 101. Press START

// 102. Verify CHECK STOP & MAR CHK are on and MAR = C/8/4/2/1

// 103. Press RESET

// 104. Press INSERT

// 105. Press RELEASE

// 106. Press START

// 107. Verify CHECK STOP & MAR CHK are on and MAR = C-4-2/C/C/C/C

// 108. Press RESET

// 109. Press START

// 110. Verify CHECK STOP & MAR CHK are on and MAR = C-8-1/C-8-1/C-8-1/C-8-1/C-8-1

// 111. Press RESET

// 112. Press START

// 113. Type A

// 114. Verify CHECK STOP & MAR CHK are on and MAR = 1/C/C/C/C

// 115. Press RESET

// 116. Press START

// 117. Verify CHECK STOP & MAR CHK are on and MAR = 1/C/C/C/C

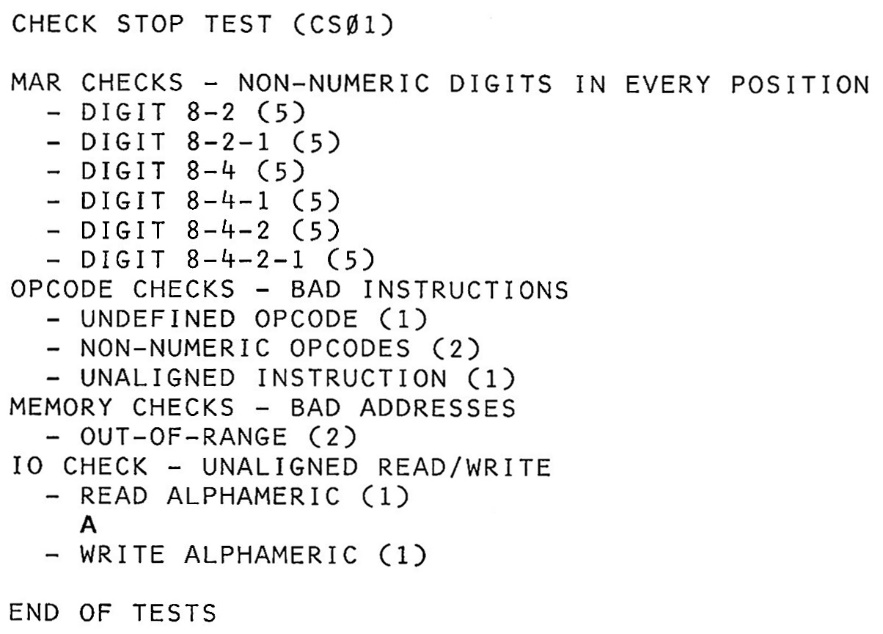
// 118. Press RESET

// 119. Press START

//

//==============================================================================

Sample Output – CS01



IBM 1620 Jr. SPS Assembler (v0.99) Source: CS01\_Check\_Stops.sps Assembled: 8/26/2020 @ 08:05

1 \*=====================================================================

2 \*

3 \* CHECK STOPS DIAGNOSTIC

4 \*

5 \* USAGE - TESTS ALL COMBINATION OF CHECK STOPS IN IBM 1620 JR.

6 \*

7 \* ALGORITHM - TEST ALL POSSIBLE NON-NUMERIC DIGITS IN ALL POSITIONS

8 \* OF MEMORY ADDRESS REGISTER.

9 \*

10 \* TEST AN UNDEFINED OP CODE.

11 \*

12 \* TEST AN INVALID OP CODE.

13 \*

14 \* TEST AN INSTRUCTION AT AN ODD ADDRESS.

15 \*

16 \* TEST ACCESS TO MEMORY BEYOND 59999.

17 \*

18 \* TEST A READ ALPHAMERIC INSTRUCTION TO AN EVEN ADDRESS.

19 \*

20 \* TEST A WRITE ALPHAMERIC INSTRUCTION FROM AN EVEN ADDRESS .

21 \*

22 \* NOTE - DIGIT COMBINATIONS 0X0B, 0X0D, 0X0E, AND 0X0F MUST BE

23 \* MANUALLY ENTERED IN THE CHECK STOPS CMEM FILE.

24 \*

25 \*=====================================================================

26

27 00402 34 00000 00102 START RCTY

28 00414 39 01447 00100 WATY TITLE

29 00426 34 00000 00102 RCTY

30 00438 34 00000 00102 RCTY

31

32 \* MAR CHECKS

33 00450 39 01493 00100 MAR WATY MART

34 00462 34 00000 00102 RCTY

35

36 \* RECORD MARK (0XA) IN ALL POSITIONS

37 00474 39 01593 00100 WATY MAR10

38 00486 34 00000 00102 RCTY

39

40 00498 330000 DSC 6,330000

41 00504 @ DSC 1,@

42 00505 00000 DSC 5,00000

43

44 00510 33000 DSC 5,33000

45 00515 @ DSC 1,@

46 00516 000000 DSC 6,000000

47

48 00522 3300 DSC 4,3300

49 00526 @ DSC 1,@

50 00527 0000000 DSC 7,0000000

51

52 00534 330 DSC 3,330

53 00537 @ DSC 1,@

54 00538 00000000 DSC 8,00000000

55

56 00546 33 DSC 2,33

57 00548 @ DSC 1,@

58 00549 000000000 DSC 9,000000000

59

60 \* RECORD GROUP MARK (0XB) IN ALL POSITIONS

61 00558 39 01629 00100 WATY MAR11

62 00570 34 00000 00102 RCTY

63

64 00582 33 00001 00000 CF 00001

65 00594 33 00010 00000 CF 00010

66 00606 33 00100 00000 CF 00100

67 00618 33 01000 00000 CF 01000

68 00630 33 10000 00000 CF 10000

69

70 \* NUMERIC BLANK (0XC) IN ALL POSITIONS

71 00642 39 01669 00100 WATY MAR12

72 00654 34 00000 00102 RCTY

73

74 00666 330000 DSC 6,330000

75 00672 ~ DNB 1

76 00673 00000 DSC 5,00000

77

78 00678 33000 DSC 5,33000

79 00683 ~ DNB 1

80 00684 000000 DSC 6,000000

81

82 00690 3300 DSC 4,3300

83 00694 ~ DNB 1

84 00695 0000000 DSC 7,0000000

85

86 00702 330 DSC 3,330

87 00705 ~ DNB 1

88 00706 00000000 DSC 8,00000000

89

90 00714 33 DSC 2,33

91 00716 ~ DNB 1

92 00717 000000000 DSC 9,000000000

93

94 \* INVALID (0XD) IN ALL POSITIONS

95 00726 39 01705 00100 WATY MAR13

96 00738 34 00000 00102 RCTY

97

98 00750 33 00001 00000 CF 00001

99 00762 33 00010 00000 CF 00010

100 00774 33 00100 00000 CF 00100

101 00786 33 01000 00000 CF 01000

102 00798 33 10000 00000 CF 10000

103

104 \* INVALID (0XE) IN ALL POSITIONS

105 00810 39 01745 00100 WATY MAR14

106 00822 34 00000 00102 RCTY

107

108 00834 33 00001 00000 CF 00001

109 00846 33 00010 00000 CF 00010

110 00858 33 00100 00000 CF 00100

111 00870 33 01000 00000 CF 01000

112 00882 33 10000 00000 CF 10000

113

114 \* GROUP MARK (0XF) IN ALL POSITIONS

115 00894 39 01785 00100 WATY MAR15

116 00906 34 00000 00102 RCTY

117

118 00918 33 00001 00000 CF 00001

119 00930 33 00010 00000 CF 00010

120 00942 33 00100 00000 CF 00100

121 00954 33 01000 00000 CF 01000

122 00966 33 10000 00000 CF 10000

123

124 00978 39 01829 00100 INST WATY INSTT

125 00990 34 00000 00102 RCTY

126

127 01002 39 01895 00100 WATY INSTB

128 01014 34 00000 00102 RCTY

129

130 01026 770000000000 DSC 12,770000000000

131

132 01038 39 01945 00100 WATY INSTN

133 01050 34 00000 00102 RCTY

134

135 01062 010000000000 DSC 12,010000000000

136

137 01074 100000000000 DSC 12,100000000000

138

139 01086 39 02001 00100 WATY INSTU

140 01098 34 00000 00102 RCTY

141

142 01110 26 00011 02375 TF 11,IMEM+11

143 01122 49 08421 00000 B 08421

144

145 01134 26 00011 02363 MEM TF 11,ISTRT+11

146 01146 39 02061 00100 WATY MEMT

147 01158 34 00000 00102 RCTY

148

149 01170 39 02121 00100 WATY MEMH

150 01182 34 00000 00102 RCTY

151

152 01194 33 60000 00000 CF 60000

153 01206 33 99999 00000 CF 99999

154

155 01218 39 02163 00100 IO WATY IOT

156 01230 34 00000 00102 RCTY

157

158 01242 39 02227 00100 WATY IOR

159 01254 34 00000 00102 RCTY

160

161 01266 34 00000 00101 SPTY

162 01278 34 00000 00101 SPTY

163 01290 34 00000 00101 SPTY

164 01302 34 00000 00101 SPTY

165 01314 37 10000 00100 RATY 10000

166 01326 34 00000 00102 RCTY

167

168 01338 39 02275 00100 WATY IOW

169 01350 34 00000 00102 RCTY

170

171 01362 39 10000 00100 WATY 10000

172

173 01374 26 00011 02363 END TF 11,ISTRT+11

174 01386 34 00000 00102 RCTY

175 01398 39 02327 00100 WATY ENDT

176 01410 34 00000 00102 RCTY

177 01422 48 00000 00000 H

178 01434 49 00402 00000 B START

179

180 \*

181 \* DATA

182 \*

\_

183 01447 434845..71040@ TITLE DAC 23,CHECK STOP TEST (CS01)@

\_

184 01493 544159..56550@ MART DAC 50,MAR CHECKS - NON-NUMERIC DIGITS IN EVERY POSITION@

\_

185 01593 000020..75040@ MAR10 DAC 18, - DIGIT 8-2 (5)@

\_

186 01629 000020..75040@ MAR11 DAC 20, - DIGIT 8-2-1 (5)@

\_

187 01669 000020..75040@ MAR12 DAC 18, - DIGIT 8-4 (5)@

\_

188 01705 000020..75040@ MAR13 DAC 20, - DIGIT 8-4-1 (5)@

\_

189 01745 000020..75040@ MAR14 DAC 20, - DIGIT 8-4-2 (5)@

\_

190 01785 000020..75040@ MAR15 DAC 22, - DIGIT 8-4-2-1 (5)@

\_

191 01829 565743..55620@ INSTT DAC 33,OPCODE CHECKS - BAD INSTRUCTIONS@

\_

192 01895 000020..71040@ INSTB DAC 25, - UNDEFINED OPCODE (1)@

\_

193 01945 000020..72040@ INSTN DAC 28, - NON-NUMERIC OPCODES (2)@

\_

194 02001 000020..71040@ INSTU DAC 30, - UNALIGNED INSTRUCTION (1)@

\_

195 02061 544554..45620@ MEMT DAC 30,MEMORY CHECKS - BAD ADDRESSES@

\_

196 02121 000020..72040@ MEMH DAC 21, - OUT-OF-RANGE (2)@

\_

197 02163 495600..63450@ IOT DAC 32,IO CHECK - UNALIGNED READ/WRITE@

\_

198 02227 000020..71040@ IOR DAC 24, - READ ALPHAMERIC (1)@

\_

199 02275 000020..04000@ IOW DAC 26, - WRITE ALPHAMERIC (1)@

\_

200 02327 455544..63620@ ENDT DAC 13,END OF TESTS@

201

\_

202 02352 49 00402 00000 ISTRT B START,,0

\_

203 02364 49 01134 00000 IMEM B MEM,,0

204

205 00402 DEND START

Symbol Cross-Reference Table

============================

Symbol Addr. Type Defined References

------- ----- ------ ------- -------------------------------------------------------------

END 01374 <inst> 173

ENDT 02327 DAC 200 175

IMEM 02364 <inst> 203 142

INST 00978 <inst> 124

INSTB 01895 DAC 192 127

INSTN 01945 DAC 193 132

INSTT 01829 DAC 191 124

INSTU 02001 DAC 194 139

IO 01218 <inst> 155

IOR 02227 DAC 198 158

IOT 02163 DAC 197 155

IOW 02275 DAC 199 168

ISTRT 02352 <inst> 202 145 173

MAR 00450 <inst> 33

MAR10 01593 DAC 185 37

MAR11 01629 DAC 186 61

MAR12 01669 DAC 187 71

MAR13 01705 DAC 188 95

MAR14 01745 DAC 189 105

MAR15 01785 DAC 190 115

MART 01493 DAC 184 33

MEM 01134 <inst> 145 203

MEMH 02121 DAC 196 149

MEMT 02061 DAC 195 146

START 00402 <inst> 27 205 178 202 205

TITLE 01447 DAC 183 28

**DX05 - Core Storage 20K Diagnostic**

//==============================================================================

//

// DX05L - Core Storage L 20K Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Bypass error type out

// OFF - Type out routine number on error

// PS2: ON - Loop in routine

// OFF - Continue to next routine

// PS3: not used

// PS4: ON - Repeat test DX05L

// OFF - Run test DX05L once

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - STOP

//

// Start addresses:

//

// 05052 - Full test

//

// Directions:

//

// 1. Load DX05L diagnostic

// 2. Press START

//

//==============================================================================

//==============================================================================

//

// DX05H - Core Storage H 20K Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Bypass error type out

// OFF - Type out routine number on error

// PS2: ON - Loop in routine

// OFF - Continue to next routine

// PS3: not used

// PS4: ON - Repeat test DX05H

// OFF - Run test DX05H once

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - STOP

//

// Start addresses:

//

// 00402 - Full test

//

// Directions:

//

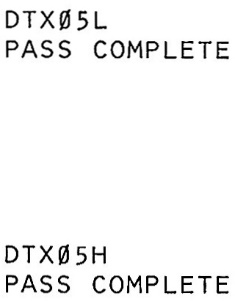
// 1. Load DX05H diagnostic

// 2. Press START

//

//==============================================================================

Sample Output – DX05



**CU04 - Additional Core Diagnostic**

//==============================================================================

//

// CU04 - Additional Core Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Bypass error type out

// OFF - Type out routine number on error

// PS2: ON - Loop in routine

// OFF - Continue to next routine

// PS3: ON - Stop on error

// OFF - Do not stop on error, continue

// PS4: not used

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - PROGRAM

//

// Start addresses:

//

// 00828 - Full test

//

// Directions:

//

// 1. Load CU04 diagnostic

// 2. Press START

// 3. Press START

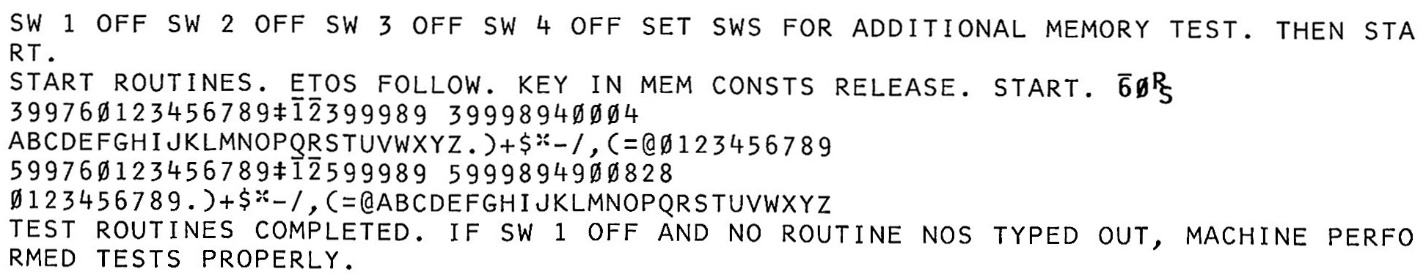
// 4. Type <flag 6> 0

// 5. Press RELEASE-START

//

//==============================================================================

Sample Output – CU04



// CU04 - Additional Core Diagnostic (IBM manual not available)

00000: 04 09 00 00 08 02 08 0A 00 00 00 00 00 00 00 00 00 00 00 00 // "I # "

00100: 00 00 00 00 00 00 00 00 00 00 00 00 01 00 02 00 03 00 04 00 // " +- "

00120: 00 00 02 00 04 00 06 00 08 00 00 00 03 00 06 00 09 00 02 01 // " - /"

00140: 00 00 04 00 08 00 02 01 06 01 00 00 05 00 00 01 05 01 00 02 // " / - J "

00160: 00 00 06 00 02 01 08 01 04 02 00 00 07 00 04 01 01 02 08 02 // " / B 0A "

00180: 00 00 08 00 06 01 04 02 02 03 00 00 09 00 08 01 07 02 06 03 // " B, 2T"

00200: 00 00 00 00 00 00 00 00 00 00 05 00 06 00 07 00 08 00 09 00 // " - 0 "

00220: 00 01 02 01 04 01 06 01 08 01 05 01 08 01 01 02 04 02 07 02 // " /A J B2"

00240: 00 02 04 02 08 02 02 03 06 03 05 02 00 03 05 03 00 04 05 04 // " B ,T K.L)M"

00260: 00 03 06 03 02 04 08 04 04 05 05 03 02 04 09 04 06 05 03 06 // ".T( E L( V "

00280: 00 04 08 04 06 05 04 06 02 07 05 04 04 05 03 06 02 07 01 08 // ") VF ME "

00300: 00 01 02 03 04 05 06 07 08 09 01 02 03 04 05 06 07 08 09 10 // " ,EX @O8 "

00320: 02 03 04 05 06 07 08 09 10 11 03 04 05 06 07 08 09 10 11 12 // ",EX @O8 "

00340: 04 05 06 07 08 09 10 11 12 13 05 06 07 08 09 10 11 12 13 14 // "EX , O8 @"

00360: 06 07 08 09 10 11 12 13 14 15 07 08 09 10 11 12 13 14 15 16 // "X ,E 8 @O"

00380: 08 09 10 11 12 13 14 15 16 17 09 10 11 12 13 14 15 16 17 18 // " ,EX @O8"

00400: 0A 00 00 00 00 00 00 00 05 02 04 05 06 08 00 00 04 09 05 05 // " K EY IN"

00420: 00 00 05 04 04 05 05 04 00 00 04 03 05 06 05 05 06 02 06 03 // " MEM CONST"

00440: 06 02 00 00 05 09 04 05 05 03 04 05 04 01 06 02 04 05 00 03 // "S REL EASE."

00460: 00 00 06 02 06 03 04 01 05 09 06 03 00 03 00 00 00 0A 00 00 // " STAR T. # "

00480: 00 00 00 00 00 00 00 00 00 00 00 00 06 02 06 03 04 01 05 09 // " STAR"

00500: 06 03 00 00 05 09 05 06 06 04 06 03 04 09 05 05 04 05 06 02 // "T ROU TINES"

00520: 00 03 00 00 04 05 06 03 05 06 06 02 00 00 04 06 05 06 05 03 // ". ETO S FOL"

00540: 05 03 05 06 06 06 00 03 00 00 00 0A 00 00 00 00 00 00 00 00 // "LOW. # "

00560: 00 00 00 00 06 02 06 06 00 00 07 01 00 00 05 06 05 05 00 00 // " SW 1 ON "

00580: 00 0A 06 02 06 06 00 00 07 01 00 00 05 06 04 06 04 06 00 00 // "#SW 1 OFF "

00600: 00 0A 06 02 06 06 00 00 07 02 00 00 05 06 05 05 00 00 00 0A // "#SW 2 ON #"

00620: 06 02 06 06 00 00 07 02 00 00 05 06 04 06 04 06 00 00 00 0A // "SW 2 OFF #"

00640: 06 02 06 06 00 00 07 03 00 00 05 06 05 05 00 00 00 0A 06 02 // "SW 3 ON #S"

00660: 06 06 00 00 07 03 00 00 05 06 04 06 04 06 00 00 00 0A 06 02 // "W 3 O FF #S"

00680: 06 06 00 00 07 04 00 00 05 06 05 05 00 00 00 0A 06 02 06 06 // "W 4 O N #SW"

00700: 00 00 07 04 00 00 05 06 04 06 04 06 00 00 00 0A 06 02 04 05 // " 4 OF F #SE"

00720: 06 03 00 00 06 02 06 06 06 02 00 00 04 06 05 06 05 09 00 00 // "T SWS FOR "

00740: 04 01 04 04 04 04 04 09 06 03 04 09 05 06 05 05 04 01 05 03 // "ADDIT IONAL"

00760: 00 00 05 04 04 05 05 04 05 06 05 09 06 08 00 00 06 03 04 05 // " MEMO RY TE"

00780: 06 02 06 03 00 03 00 00 06 03 04 08 04 05 05 05 00 00 06 02 // "ST. T HEN S"

00800: 06 03 04 01 05 09 06 03 00 03 00 00 00 0A 00 00 00 00 00 00 // "TART. # "

00820: 00 00 00 00 00 00 00 00 04 06 00 00 08 05 02 00 00 01 00 00 // " F - "

00840: 04 07 00 00 08 07 06 00 00 01 00 00 03 09 00 00 05 06 05 00 // "G O-"

00860: 00 01 00 00 04 09 00 00 08 08 08 00 00 00 00 00 03 09 00 00 // " I "

00880: 05 08 03 00 00 01 00 00 04 06 00 00 09 01 02 00 00 02 00 00 // "Q F - "

00900: 04 07 00 00 09 03 06 00 00 02 00 00 03 09 00 00 06 00 03 00 // "G "

00920: 00 01 00 00 04 09 00 00 09 04 08 00 00 00 00 00 03 09 00 00 // " I "

00940: 06 02 01 00 00 01 00 00 04 06 00 00 09 07 02 00 00 03 00 00 // "S+ F -. "

00960: 04 07 00 00 09 09 06 00 00 03 00 00 03 09 00 00 06 04 01 00 // "G . U+"

00980: 00 01 00 00 04 09 00 01 00 00 08 00 00 00 00 00 03 09 00 00 // " I "

01000: 06 05 09 00 00 01 00 00 04 06 00 01 00 03 02 00 00 04 00 00 // "V F .-) "

01020: 04 07 00 01 00 05 06 00 00 04 00 00 03 09 00 00 06 07 09 00 // "G ) X "

01040: 00 01 00 00 04 09 00 01 00 06 08 00 00 00 00 00 03 09 00 00 // " I "

01060: 06 09 07 00 00 01 00 00 03 09 00 00 07 01 07 00 00 01 00 00 // "Z0 10 "

01080: 04 08 00 00 00 00 00 00 00 00 00 00 03 04 00 00 00 00 00 00 // "H @ "

01100: 00 01 00 02 03 09 00 00 04 09 03 00 00 01 00 00 04 09 00 01 // " I I "

01120: 01 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 // "\* "

01140: 03 09 00 00 04 00 09 00 00 01 00 00 03 06 00 01 01 03 03 00 // " $ "

01160: 00 01 00 00 04 09 00 01 03 00 08 00 00 00 00 00 00 00 00 00 // " I "

01180: 00 00 00 00 00 00 00 00 10 00 00 03 00 04 01 00 01 03 01 04 // " .)+$\*"

01200: 02 00 02 01 02 03 02 04 03 03 03 04 04 01 04 02 04 03 04 04 // "-/,(= @ABCD"

01220: 04 05 04 06 04 07 04 08 04 09 05 01 05 02 05 03 05 04 05 05 // "EFGHI JKLMN"

01240: 05 06 05 07 05 08 05 09 06 01 06 02 06 03 06 04 06 05 06 06 // "OPQR STUVW"

01260: 06 07 06 08 06 09 07 00 07 01 07 02 07 03 07 04 07 05 07 06 // "XYZ01 23456"

01280: 07 07 07 08 07 09 06 09 00 0A 00 00 00 00 00 00 00 00 00 00 // "789Z# "

01300: 00 00 00 00 00 00 00 00 03 01 02 09 09 05 00 00 01 01 08 08 // " "

01320: 02 04 00 01 02 08 07 03 00 00 04 09 04 07 00 01 03 08 00 00 // "( 3 IG "

01340: 01 02 00 00 04 09 00 01 04 07 06 00 00 00 00 00 00 00 00 00 // " I G "

01380: 04 06 00 01 04 05 02 00 00 01 00 00 03 09 00 01 04 06 05 00 // "F E- F-"

01400: 00 01 00 00 03 04 00 00 00 00 00 00 00 01 00 02 03 09 00 01 // " @ "

01420: 01 08 09 00 00 01 00 00 03 04 00 00 00 00 00 00 00 01 00 02 // " @ "

01440: 03 09 02 09 09 05 01 00 00 01 00 00 04 07 00 01 04 07 06 00 // " + G G "

01460: 00 03 00 00 04 08 07 00 07 00 07 03 00 00 00 0A 04 06 00 01 // ". H00 3 #F "

01480: 03 00 08 00 00 02 00 00 04 09 00 01 05 04 08 00 00 00 00 00 // " I M "

01540: 00 00 00 00 00 00 00 00 03 01 00 09 09 05 00 02 09 09 05 00 // " -"

01560: 02 04 03 00 00 04 09 01 00 00 04 09 04 07 00 01 06 03 02 00 // "( ) IG T-"

01580: 01 02 00 00 04 09 00 01 07 02 08 00 00 00 00 00 00 00 00 00 // " I 2 "

01620: 00 00 00 00 00 00 00 00 00 00 00 00 04 06 00 01 07 00 04 00 // " F 0 "

01640: 00 01 00 00 03 09 00 01 07 01 07 00 00 01 00 00 03 04 00 00 // " 1 0 @ "

01660: 00 00 00 00 00 01 00 02 03 09 02 09 09 05 01 00 00 01 00 00 // " + "

01680: 03 04 00 00 00 00 00 00 00 01 00 02 03 09 00 09 09 05 01 00 // "@ +"

01700: 00 01 00 00 04 07 00 01 07 02 08 00 00 03 00 00 04 08 07 00 // " G 2 . H0"

01720: 07 00 07 04 00 00 00 0A 04 06 00 01 05 04 08 00 00 02 00 00 // "04 #F M "

01740: 04 09 00 01 08 07 02 00 00 00 00 00 00 00 00 00 00 00 00 00 // "I - "

01760: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 03 05 06 // " CO"

01780: 05 05 06 02 06 03 06 02 00 00 04 09 05 05 04 03 05 06 05 09 // "NSTS INCOR"

01800: 05 09 00 03 00 00 05 02 04 05 06 08 00 00 04 09 05 05 00 00 // "R. KE Y IN "

01820: 04 03 05 06 05 09 05 09 00 00 04 03 05 06 05 05 06 02 06 03 // "CORR CONST"

01840: 06 06 00 00 04 01 05 05 04 04 00 00 06 02 06 03 04 01 05 09 // "W AND STAR"

01860: 04 05 06 03 00 03 00 00 00 0A 00 00 01 04 00 01 01 03 04 00 // "ET. # \* $ "

01880: 00 00 14 00 04 06 00 03 00 09 06 00 01 02 00 00 01 04 00 01 // " F. \* "

01900: 01 03 04 00 00 00 16 00 04 06 00 02 00 08 08 00 01 02 00 00 // "$ F "

01920: 03 09 00 01 07 07 07 00 00 01 00 00 04 09 00 01 01 04 00 00 // " 70 I \* "

01960: 00 00 00 00 00 00 00 00 16 09 07 09 07 08 07 07 07 06 07 05 // " Z 98765"

01980: 07 04 07 03 07 02 07 01 07 00 06 09 06 08 06 07 06 06 06 05 // "43210 ZYXWV"

02000: 06 04 06 03 06 02 06 01 05 09 05 08 05 07 05 06 05 05 05 04 // "UTS R QPONM"

02020: 05 03 05 02 05 01 04 09 04 08 04 07 04 06 04 05 04 04 04 03 // "LKJIH GFEDC"

02040: 04 02 04 01 03 04 03 03 02 04 02 03 02 01 02 00 01 04 01 03 // "BA@=( ,/-\*$"

02060: 01 00 00 04 00 00 00 03 00 0A 00 00 00 00 00 00 00 00 00 00 // "+) .# "

02080: 00 00 00 00 00 00 00 00 03 01 04 09 09 05 00 00 01 09 06 08 // " I Y"

02100: 02 04 00 02 00 06 07 05 00 00 04 09 04 07 00 02 01 06 00 00 // "( 5 IG "

02120: 01 02 00 00 04 09 00 02 02 05 06 00 00 00 00 00 00 00 00 00 // " I "

02160: 04 06 00 02 02 03 02 00 00 01 00 00 03 09 00 02 02 04 05 00 // "F ,- (-"

02180: 00 01 00 00 03 04 00 00 00 00 00 00 00 01 00 02 03 09 00 01 // " @ "

02200: 09 06 09 00 00 01 00 00 03 04 00 00 00 00 00 00 00 01 00 02 // " @ "

02220: 03 09 04 09 09 05 01 00 00 01 00 00 04 07 00 02 02 05 06 00 // " I + G "

02240: 00 03 00 00 04 08 07 00 07 00 07 06 00 00 00 0A 04 06 00 02 // ". H00 6 #F "

02260: 00 08 08 00 00 02 00 00 04 09 00 02 03 00 04 00 00 00 00 00 // " I "

02300: 00 00 00 00 03 01 00 09 09 05 00 04 09 09 05 00 02 04 05 00 // " ) -(-"

02320: 00 04 09 01 00 00 04 09 04 07 00 02 03 07 06 00 01 02 00 00 // ") IG "

02340: 04 09 00 02 04 07 02 00 00 00 00 00 00 00 00 00 00 00 00 00 // "I G- "

02360: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 06 00 02 // " F "

02380: 04 04 08 00 00 01 00 00 03 09 00 02 04 06 01 00 00 01 00 00 // "D F+ "

02400: 03 04 00 00 00 00 00 00 00 01 00 02 03 09 04 09 09 05 01 00 // "@ I +"

02420: 00 01 00 00 03 04 00 00 00 00 00 00 00 01 00 02 03 09 00 09 // " @ "

02440: 09 05 01 00 00 01 00 00 04 07 00 02 04 07 02 00 00 03 00 00 // " + G G-. "

02460: 04 08 07 00 07 00 07 03 00 00 00 0A 04 06 00 02 03 00 04 00 // "H003 #F "

02480: 00 02 00 00 04 09 00 02 06 04 00 00 00 00 00 00 00 00 00 00 // " I U "

02520: 14 01 04 02 04 03 04 04 04 05 04 06 04 07 04 08 04 09 05 01 // "ABCDE FGHIJ"

02540: 05 02 05 03 05 04 05 05 05 06 05 07 05 08 05 09 06 01 06 02 // "KLMNO PQR S"

02560: 06 03 06 04 06 05 06 06 06 07 06 08 06 09 00 00 00 03 00 04 // "TUVWX YZ .)"

02580: 01 00 01 03 01 04 02 00 02 01 02 03 02 04 03 03 03 04 07 00 // "+$\*-/ ,(=@0"

02600: 07 01 07 02 07 03 07 04 07 05 07 06 07 07 07 08 07 09 06 09 // "12345 6789Z"

02620: 00 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 // "# "

02640: 03 01 02 09 08 05 00 00 02 05 02 00 03 01 04 09 09 05 00 02 // " - I "

02660: 09 08 05 00 02 04 05 00 00 04 09 02 09 09 04 09 04 07 00 02 // " -(-) IG "

02680: 07 02 04 00 01 02 00 00 04 09 00 02 08 02 00 00 00 00 00 00 // "2 I "

02720: 00 00 00 00 04 06 00 02 07 09 06 00 00 01 00 00 03 09 00 02 // " F 9 "

02740: 08 00 09 00 00 01 00 00 03 04 00 00 00 00 00 00 00 01 00 02 // " @ "

02760: 03 09 02 09 08 05 01 00 00 01 00 00 03 04 00 00 00 00 00 00 // " + @ "

02780: 00 01 00 02 03 09 04 09 09 05 01 00 00 01 00 00 04 07 00 02 // " I + G "

02800: 08 02 00 00 00 03 00 00 04 08 07 00 07 00 07 08 00 00 00 0A // " . H 008 #"

02820: 04 06 00 02 06 04 00 00 00 02 00 00 04 09 00 02 08 06 08 00 // "F U I "

02860: 00 00 00 00 00 00 00 00 03 01 02 09 09 05 00 04 09 09 05 00 // " ) -"

02880: 02 04 03 00 00 04 09 05 00 00 04 09 04 07 00 02 09 04 00 00 // "( ) IG "

02900: 01 02 00 00 04 09 00 03 00 03 06 00 00 00 00 00 00 00 00 00 // " I.. "

02940: 04 06 00 03 00 01 02 00 00 01 00 00 03 09 00 03 00 02 03 00 // "F. - . "

02960: 00 01 00 00 03 04 00 00 00 00 00 00 00 01 00 02 03 09 04 09 // " @ I"

02980: 09 05 01 00 00 01 00 00 03 04 00 00 00 00 00 00 00 01 00 02 // " + @ "

03000: 03 09 02 09 09 05 01 00 00 01 00 00 04 07 00 03 00 03 06 00 // " + G.. "

03020: 00 03 00 00 04 08 07 00 07 00 07 09 00 00 00 0A 04 06 00 02 // ". H00 9 #F "

03040: 08 06 08 00 00 02 00 00 04 09 00 03 00 09 06 00 00 00 00 00 // " I . "

03080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 10 00 04 06 00 03 // " F."

03100: 01 00 08 00 01 04 00 00 01 01 00 03 00 09 05 00 00 00 10 01 // "+ \* . - "

03120: 04 07 00 01 03 00 08 00 01 04 00 00 04 09 00 03 01 06 08 00 // "G \* I. "

03160: 00 00 00 00 00 00 00 00 04 01 03 09 09 05 02 00 00 03 00 00 // " A -. "

03180: 03 04 00 00 00 00 00 00 00 01 00 02 03 05 03 09 09 07 06 00 // "@ "

03200: 00 01 00 00 03 04 00 00 00 00 00 00 00 01 00 01 04 09 03 09 // " @ I "

03220: 09 05 02 00 00 00 00 00 04 09 00 03 02 04 00 00 00 00 00 00 // " - I .( "

03240: 04 01 02 05 08 05 01 00 00 03 00 00 03 04 00 00 00 00 00 00 // "A +. @ "

03260: 00 01 00 02 03 09 02 05 08 05 01 00 00 01 00 00 04 09 00 03 // " + I."

03280: 02 08 08 00 00 00 00 00 01 04 00 01 01 03 04 00 00 00 14 00 // " \* $ "

03300: 04 06 00 03 06 08 04 00 01 02 00 00 01 04 00 01 01 03 04 00 // "F.Y \* $ "

03320: 00 00 16 00 04 06 00 03 03 06 00 00 01 02 00 00 03 09 00 01 // " F. "

03340: 07 07 07 00 00 01 00 00 04 09 00 01 01 04 00 00 00 00 00 00 // "70 I \* "

03360: 04 01 05 09 09 05 02 00 00 03 00 00 03 04 00 00 00 00 00 00 // "AR -. @ "

03380: 00 01 00 02 03 05 05 09 09 07 06 00 00 01 00 00 03 04 00 00 // " R @ "

03400: 00 00 00 00 00 01 00 01 04 09 05 09 09 05 02 00 00 00 00 00 // " I R - "

03420: 04 09 00 03 04 03 02 00 00 00 00 00 04 01 04 05 08 05 01 00 // "I.C- AE +"

03440: 00 03 00 00 03 04 00 00 00 00 00 00 00 01 00 02 03 09 04 05 // ". @ E"

03460: 08 05 01 00 00 01 00 00 04 09 00 03 06 08 04 00 00 00 00 00 // " + I .Y "

03480: 06 03 04 05 06 02 06 03 00 00 05 09 05 06 06 04 06 03 04 09 // "TEST ROUTI"

03500: 05 05 04 05 06 02 00 00 04 03 05 06 05 04 05 07 05 03 04 05 // "NES C OMPLE"

03520: 06 03 04 05 04 04 00 03 00 00 04 09 04 06 00 00 06 02 06 06 // "TED. IF SW"

03540: 00 00 07 01 00 00 05 06 04 06 04 06 00 00 04 01 05 05 04 04 // " 1 OF F AND"

03560: 00 00 05 05 05 06 00 00 05 09 05 06 06 04 06 03 04 09 05 05 // " NO R OUTIN"

03580: 04 05 00 00 05 05 05 06 06 02 00 00 06 03 06 08 05 07 04 05 // "E NOS TYPE"

03600: 04 04 00 00 05 06 06 04 06 03 02 03 00 00 05 04 04 01 04 03 // "D OUT , MAC"

03620: 04 08 04 09 05 05 04 05 00 00 05 07 04 05 05 09 04 06 05 06 // "HINE PERFO"

03640: 05 09 05 04 04 05 04 04 00 00 06 03 04 05 06 02 06 03 06 02 // "RMED TESTS"

03660: 00 00 05 07 05 09 05 06 05 07 04 05 05 09 05 03 06 08 00 03 // " PROP ERLY."

03680: 00 00 00 0A 03 04 00 00 00 00 00 00 00 01 00 02 03 09 00 03 // " #@ ."

03700: 04 08 01 00 00 01 00 00 04 08 00 00 00 00 00 00 00 00 00 00 // "H+ H "

03720: 00 00 00 00 00 00 00 00 00 00 00 0A 00 00 00 00 00 00 00 00 // " # "

25840: 00 00 00 00 00 00 00 00 00 00 04 01 04 02 04 03 04 04 04 05 // " ABCDE"

25860: 04 06 04 07 04 08 04 09 05 01 05 02 05 03 05 04 05 05 05 06 // "FGHIJ KLMNO"

25880: 05 07 05 08 05 09 06 02 06 03 06 04 06 05 06 06 06 07 06 08 // "PQRST UVWXY"

25900: 06 09 00 03 00 04 01 00 01 03 01 04 02 00 02 01 02 03 02 04 // "Z.)+$ \*-/,("

25920: 03 03 03 04 07 00 07 01 07 02 07 03 07 04 07 05 07 06 07 07 // "=@012 34567"

25940: 07 08 07 09 00 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 // "89# "

39940: 00 00 00 00 00 00 00 00 00 00 00 00 03 08 03 09 09 09 04 00 // " "

39960: 00 01 00 00 04 09 00 03 02 02 08 0A 00 00 00 00 03 09 09 07 // " I. # "

39980: 06 00 01 02 03 04 05 06 07 08 09 0A 11 12 03 09 09 09 08 09 // " @O8 # "

40000: 04 00 00 00 04 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 // " # "

45840: 00 00 00 00 00 00 00 00 00 00 07 00 07 01 07 02 07 03 07 04 // " 01234"

45860: 07 05 07 06 07 07 07 08 07 09 00 03 00 04 01 00 01 03 01 04 // "56789 .)+$\*"

45880: 02 00 02 01 02 03 02 04 03 03 03 04 04 01 04 02 04 03 04 04 // "-/,(= @ABCD"

45900: 04 05 04 06 04 07 04 08 04 09 05 01 05 02 05 03 05 04 05 05 // "EFGHI JKLMN"

45920: 05 06 05 07 05 08 05 09 06 02 06 03 06 04 06 05 06 06 06 07 // "OPQRS TUVWX"

45940: 06 08 06 09 00 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 // "YZ# "

59940: 00 00 00 00 00 00 00 00 00 00 00 00 03 08 05 09 09 09 04 00 // " R "

59960: 00 01 00 00 04 09 00 03 04 02 00 0A 00 00 00 00 05 09 09 07 // " I.B # R "

59980: 06 00 01 02 03 04 05 06 07 08 09 0A 11 12 05 09 09 09 08 09 // " @O8 # R "

// End

**DX03 – Console Typewriter Diagnostic**

//==============================================================================

//

// DX03 – Console Typewriter Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Bypass error type out

// OFF - Type out routine number on error

// PS2: ON - Loop in routine

// OFF - Continue to next routine

// PS3: ON - Stop on error

// OFF - Do not stop on error, continue

// PS4: ON - Repeat test DX03

// OFF - Run test DX03 once

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - STOP

// O'FLOW - PROGRAM

//

// Start addresses:

//

// 00440 - Full test w/ title

// 00726 - Full test w/o title

// 01982 - Alphameric echo test

// 02256 - Numeric echo test

//

// Directions:

//

// 1. Load DX03 diagnostic

// 2. Tab stops are automatically set at 45 & 75

// 3. Press START

// 4. Press START

// 5. Press INSERT

// 6. Type 4901982

// 7. Press RELEASE

// 8. Press START

// 9. Type alphameric characters, end with record mark

// 10. Press RELEASE-START

// 11. Press INSERT

// 12. Type 4902256

// 13. Press RELEASE

// 14. Press START

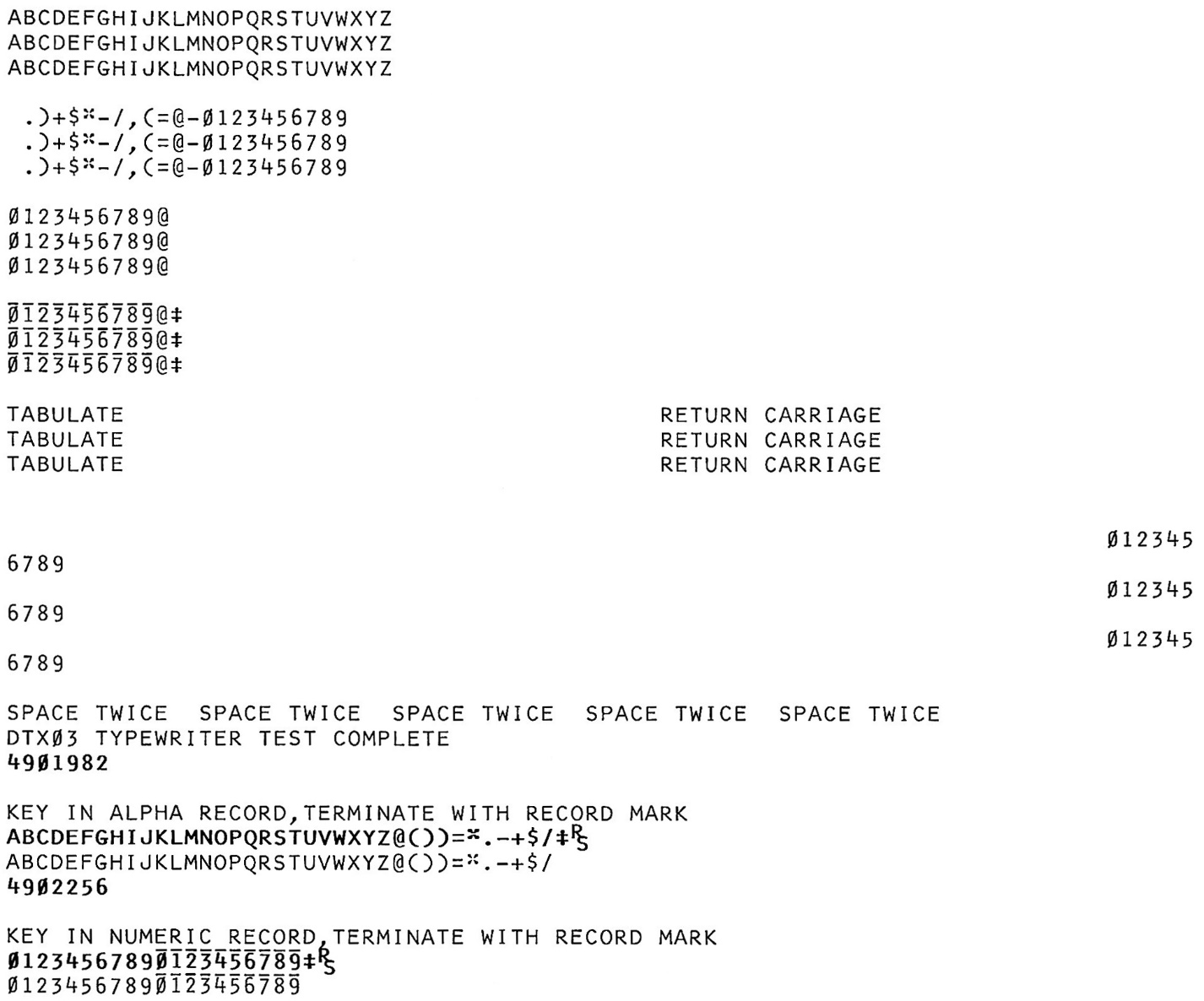
// 15. Type numeric characters, end with record mark

// 16. Press RELEASE-START

//

//==============================================================================

Sample Output – DX03



**IO02 - Card I/O Diagnostic**

//==============================================================================

//

// IO02 - Card I/O Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Bypass error type out

// OFF - Type out routine number on error

// PS2: ON - Loop in routine

// OFF - Continue to next routine

// PS3: ON - Stop on error

// OFF - Do not stop on error, continue

// PS4: ON - Repeat test IO02

// OFF - Run test IO02 once

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - PROGRAM

// O'FLOW - PROGRAM

//

// Start addresses:

//

// 00828 - Full test

// 07152 - Punch alphameric ripple deck

// 02796 - Verify alphameric ripple deck

// 07188 - Punch numeric ripple deck

// 05304 - Verify numeric ripple deck

//

// Directions:

//

// 1. Load IO02 diagnostic

// 2. Press START

// 3. Press START

// 3. Press RELEASE

// 5. Press RESET

// 6. Press INSERT

// 7. Type 4907152

// 8. Press RELEASE-START

// 9. Verify that PUNCH NO FEED light is lit

// 10. Insert empty card deck into punch

// 11. Press START

// 12. Verify that READER NO FEED light is lit

// 13. Move card deck from punch to reader

// 14. Press RELEASE

// 15. Press RESET

// 16. Press INSERT

// 17. Type 4907188

// 18. Press RELEASE-START

// 19. Verify that PUNCH NO FEED light is lit

// 20. Insert empty card deck into punch

// 21. Press START

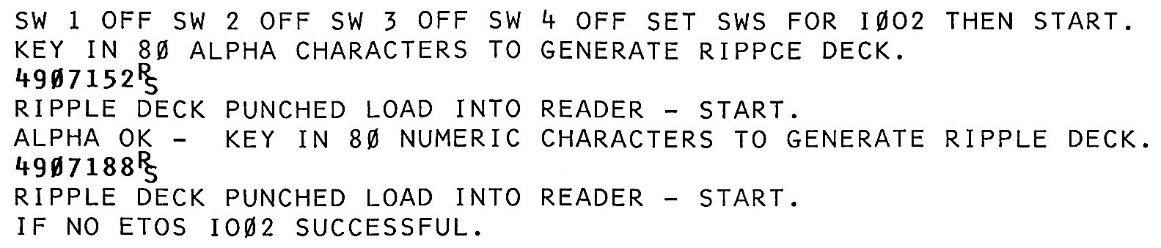
// 22. Verify that READER NO FEED light is lit

// 23. Move card deck from punch to reader

//

//==============================================================================

Sample Output – IO02



**IO03 - Card I/O Reliability Diagnostic**

//==============================================================================

//

// IO03 - Card I/O Reliability Diagnostic

//

// Program Switch settings:

//

// PS1: ON - Punch only

// OFF - Read and punch

// PS2: ON - Read only

// OFF - Read and punch

// PS3: ON - Don't print errors

// OFF - Print errors

// PS4: ON - Delay constant

// OFF - Delay changes

//

// Check switches settings:

//

// DISK I/O - STOP

// PARITY - STOP

// I/O - PROGRAM

// O'FLOW - PROGRAM

//

// Start addresses:

//

// 00652 - Full test

//

// Directions:

//

// 1. Load IO03 diagnostic

// 2. Turn on PROGRAM SWITCH 1

// 3. Press START

// 4. Press START

// 5. Verify that PUNCH NO FEED light is lit

// 6. Insert empty card deck into punch

// 7. After ~ 1 minute press STOP

// 8. Remove cards from punch

// 9. Turn off PROGRAM SWITCH 1

// 10. Press RESET

// 11. Press INSERT

// 12. Type 4900652

// 13. Press RELEASE

// 14. Press START

// 15. Verify that READER NO FEED light is lit

// 16. Insert newly punched cards in reader

// 17. Verify that PUNCH NO FEED light is lit

// 18. Insert empty card deck into punch

//

//==============================================================================

Sample Output – IO03

