

AD-A142 448

PROTON-INDUCED SINGLE EVENT UPSETS IN THE SBP0000
MICROPROCESSOR(U) NAVAL RESEARCH LAB WASHINGTON DC
L T MYERS ET AL. 12 JUN 84 NRL-MR-9213

1/1

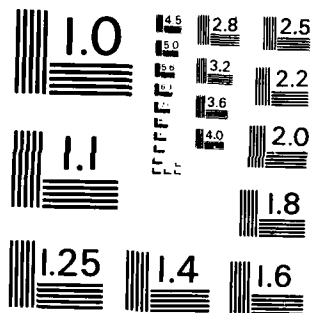
UNCLASSIFIED

F/G 9/2

NL

1

END
DATE
FILMED
8 '84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

AD-A142 448

REPORT DOCUMENTATION PAGE				
1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION AVAILABILITY OF REPORT		
2b DECLASSIFICATION DOWNGRADING SCHEDULE		Approved for public release; distribution unlimited.		
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 5213		5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a NAME OF PERFORMING ORGANIZATION Naval Research Laboratory	6b OFFICE SYMBOL <i>(If applicable)</i> Code 6608	7a NAME OF MONITORING ORGANIZATION		
6c ADDRESS (City, State and ZIP Code) Washington, DC 20375		7b ADDRESS (City, State and ZIP Code)		
8a NAME OF FUNDING SPONSORING ORGANIZATION Naval Electronic Systems Command	8b OFFICE SYMBOL <i>(If applicable)</i>	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c ADDRESS (City, State and ZIP Code) Washington, DC 20360		10 SOURCE OF FUNDING NOS		
		PROGRAM ELEMENT NO. 62712N	PROJECT NO. XF121413 10	TASK NO.
				WORK UNIT NO. DN380-034
11 TITLE (Include Security Classification) Proton-Induced Single Event Upsets in the SBP9989 Microprocessor				
12 PERSONAL AUTHOR(S) L. T. Myers, P. Shapiro, E. L. Petersen, A. B. Campbell, and D. Emily*				
13a TYPE OF REPORT Final	13b TIME COVERED FROM TO	14 DATE OF REPORT (Yr., Mo., Day) June 12, 1984	15 PAGE COUNT 29	
16 SUPPLEMENTARY NOTATION *Naval Weapons Supply Center, Crane, Indiana 47522				
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB GR	Single event upsets Radiation belts	
			Microelectronics Proton dose	
19 ABSTRACT (Continue on reverse if necessary and identify by block number)				
<p>This report describes the investigation of proton-induced single event upsets in the SBP9989 micro-processor fabricated by Texas Instruments, Inc. Ten different devices with various date codes were tested. Total dose failure occurred in five of these devices after 10^{12} protons/cm². An average upset cross section of 6×10^{-12} upset-cm²/proton-device was observed. Comparison with cosmic ray-induced upsets indicates that proton-induced upsets represent a more severe problem for satellites with orbits between 600 and 2500 nautical miles.</p> <p><i>10 June 1984</i></p> <p><i>6 June 1984</i></p>				
20 DISTRIBUTION AVAILABILITY OF ABSTRACT UNCLASSIFIED UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS <input type="checkbox"/>			21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL E. L. Petersen		22b TELEPHONE NUMBER <i>(Include Area Code)</i> (202) 767-3938	22c OFFICE SYMBOL Code 6608	

CONTENTS

INTRODUCTION 1

EXPERIMENTAL METHOD 1

EXPERIMENTAL RESULTS 2

CONCLUSIONS 4

ACKNOWLEDGMENTS 4

REFERENCES 4

APPENDIX 10

S DTIC ELECTE **D**
 JUN 27 1984
B

DTIC ELECTE JUN 27 1984	✓
Available for Distribution	
Dist. A-1	

DTIC
 COPY
 INSPECTED
 2

PROTON-INDUCED SINGLE EVENT UPSETS IN THE SBP9989 MICROPROCESSOR

INTRODUCTION

Large scale integrated (LSI) circuit devices that utilize integrated injection logic (I²L) are being considered for use in satellite systems. The use of LSI devices in the space environment requires an understanding of the susceptibility of these devices to soft upsets induced by both cosmic rays and protons. An earlier study¹ examined the susceptibility of I²L devices to cosmic ray-induced upsets. That work described the error mechanism, determined the sensitive region within the device, described the heavy ion upset experiment, and determined the probability for cosmic ray-induced upset in space.

This report supplements that work by examining proton-induced soft upsets in the SBP9989 microprocessor. The experiments performed at the NRL cyclotron soft upset facility were a collaborative effort by personnel from NRL, and NWSC, and NASA. The results of these experiments are compared with those from reference 1 and indicate that proton-induced upsets represent a more severe problem for this microprocessor in certain satellite orbits.

EXPERIMENTAL METHOD

The hardware test setup for these experiments is shown in Fig. 1. Two SBP9989 microprocessors are run in synchronism. The device under test (DUT) is exposed to 40 MeV protons while the remaining apparatus, which includes a working reference device, is placed behind a lead shield outside the radiation field. A remote terminal in the cyclotron control room monitors and controls the equipment.

A test program originated by Texas Instruments, Inc. for the SBP9900 was modified for the SBP9989 testing. Four instructions exclusive to the SBP9989, signed multiply, signed divide, load workspace pointer, and load status register, were added to the 9900 basic instruction set so that all 73 instructions are executed. Figure 2 outlines the flow of the test program. The test program, executed by the dual microprocessors is listed in the

Manuscript approved September 18, 1983.

appendix. This program is stored in PROM (programmable read-only memory) on the hardware instruction tester (HIT). The modifications to this program for testing the SBP9989 are available upon request. Outputs from both devices were compared at the conclusion of each machine cycle. If the outputs differ, the DUT is halted and the failing output is stored. The dual microprocessors are then resynchronized and the test program is restarted. Both the hardware test setup and the testing program have been described in reference 2.

The 40 MeV protons were provided by the NRL proton upset facility, which has been described in another report³. Briefly, a 45 MeV proton beam is generated by the NRL sector-focussed cyclotron. This beam passes through a degrader that lowers the beam energy to 40 MeV and diffuses the beam spot to make it more uniform. This beam is then bent with a 135° analyzing magnet, passes through a 0.002" tin scattering foil, and a kapton foil to irradiate the DUT in air.

The dose to the device is monitored indirectly. A solid state detector measures the particle flux that passes through the tin foil by counting elastically scattered protons. Dose calibrations are performed at the beginning, at the end, and throughout the run as needed. During calibration, dose at the device is measured with an array of thermoluminescent dosimeters (TLDs) and normalized to elastically scattered protons detected with the solid state detector. The array allows the beam uniformity to be determined also. Particle flux and dose to the device can then be determined during a test from the number of elastically scattered protons detected.

EXPERIMENTAL RESULTS

Table 1 summarizes the results of the eleven tests conducted on the SBP9989 microprocessors. The ten different devices that were tested are grouped according to date code. Device 110 had previously received 10^{11} protons/cm² without upset. Average particle flux during a run ranged from 2 to 9×10^{10} p/cm²-min. The injection current was 400 mA for all runs. The clock frequency was 3 MHz for ten runs and 1 MHz for the final run. Device 13 was tested at two different clock frequencies. Upset fluence ranged from 1.1 to 2.8×10^{11} protons/cm²-upset with the average upset fluence being 1.6×10^{11} protons/cm²-upset.

Five of the devices that were tested failed permanently during the test. The average total dose failure (TDF) was 180 kRads. The remaining devices were tested well past this dose without indication of permanent failure. Total dose susceptibility for the SBP9989 to gammas and electrons was studied by Woods and MacPhee.⁴ They found wide variation in TDF for the 9989, 500 to 4000 kRads for date codes 8120 and 20 to 300 kRads for date codes 8132. Our measurements on devices with these date codes are consistent with their observations. Our measurements on devices with date code 8152, which were not measured by Woods and MacPhee, indicates a total dose susceptibility intermediate between 8120's and 8132's. Total dose failure occurred on one device at 145 kRads, while two others survived past 400 kRad.

Operating the final device at a lower clock frequency seemed to lower the upset rate. The first ten tests were done with the clock frequency set at 3 MHz. The final device was also tested at 1 MHz clock frequency. This single data point at a lower frequency indicates that upset cross section varies with clock frequency. The lack of any strong correlation between heavy ion induced upsets and clock frequency in reference 1 indicates that the errors are associated with memory cell flip-flops. The frequency dependence observed in this experiment may indicate that the proton induced upsets occur in the random logic gates. An error is propagated in a random logic gate only if the upset is present when the next gate is clocked. This requires that the upset occur near a clock edge implying a lower probability for upset at a lower clock frequency.

Table 2 shows the types of errors that occurred with each device. Errors involving more than one bit are counted separately. In calculating upset fluence and cross section, multiple bit errors were counted only once. The vast majority of the errors involved a single bit in either the address or data register.

The average upset fluence of 1.6×10^{11} protons/cm²-upset corresponds to an upset cross section of 6.25×10^{-12} cm². This upset cross section is significantly less than that observed for heavy ions. Under similar conditions, reference 1 indicates that the upset cross section is 5.82×10^{-5} cm² for 144 MeV Kr ions and 5.55×10^{-7} cm² for 104 MeV O ions. Although the upset cross section is appreciably lower for 40 MeV protons, the upset rate due to protons in the radiation belts can be higher than the upset rate due to cosmic rays.

Figure 3 shows a plot of upset/bit-day due to both protons and cosmic rays for a 60° circular orbit versus altitude. Upset calculations for the cosmic rays are based upon cosmic ray fluence calculations of Adams⁵ for the 60° circular orbit and the critical charge for the 9989 determined in reference 1. Upset calculations for the protons are based upon calculations by Petersen⁶ for the 93425A. The proton upset cross section per bit of the 93425A is similar to the proton upset cross section per device of the 9989. The calculations involve extrapolating the upset cross section to higher energies and integrating the upset cross section over the proton energies found in the 60° circular orbit. The assumptions used in these calculations about the upset mechanism in the 9989 are certainly valid for indicating the relative effects due to cosmic rays and protons; however, the curves shown in figure 3 are primarily qualitative rather than quantitative.

These calculations indicate the importance of determining the upset cross sections both for protons and heavy ions. Either may have a dominant influence upon the behavior of the microprocessor, depending upon the altitude of the orbit.

CONCLUSIONS

The average proton upset cross section observed in these experiments was 6.25×10^{-12} cm². Although this is substantially less than the heavy ion upset cross section, proton induced upsets represent a more severe problem than cosmic ray-induced upsets for satellites orbiting within the proton radiation belts between 600 and 2500 nautical miles.

Total dose failure observed with these devices is consistent with that measured earlier. SBP9989 devices with date codes 8152 exhibit total dose failure between 100 and 450 kRads.

The dependence of upset cross section on clock frequency observed in these experiments was not observed with heavy ions and may indicate a different upset mechanism involving random logic gates.

ACKNOWLEDGMENTS

We would like to thank NASA for furnishing the SBP9989 parts for testing as well as Code 7750 of NRL for providing the beam time on the cyclotron. The efforts of the NRL cyclotron staff throughout the runs were greatly appreciated. Thanks to Jim Adams for his calculations on cosmic ray flux vs. altitude.

REFERENCES

1. W.E. Price, J.C. Pickel, T. Ellis, F.B. Frazee, "Cosmic Ray Induced Errors in I²L Microprocessors and Logic Devices," IEEE Trans. Nucl. Sci. NS-28 (1981) 3946.
2. Tom Ellis, "Radiation Effects Characterization of the SBP9900A 16-Bit Microprocessor," IEEE Trans. Nucl. Sci. NS-26 (1979) 4735.
3. P. Shapiro, A.B. Campbell, E.L. Petersen, L.T. Myers, "Proton-Induced Single Event Upsets in NMOS Microprocessors," IEEE Trans. Nucl. Sci. NS-29 (1982)
4. J.P. Woods and J.V. MacPhee, "Total Dose Susceptibility of the SBP9989 Microprocessor," IEEE Trans. Nucl. Sci. NS-29 (1982) 1740.
5. J.H. Adams, private communication.
6. E.L. Petersen, "Soft Errors Due to Protons in the Radiation Belt," IEEE Trans. Nucl. Sci. NS-28 (1981) 3981.

TABLE 1

Device	Date Code	Dose (KRad)	Fluence (10^{11} protons/cm ²)	Number of Upsets	Upset Fluence (10^{11} protons/cm ² - upset)	Upset Cross Section (10^{-12} cm ²)	Average Flux (10^{10} protons/cm ² min)	Clock Frequency (MHz)	Total Dose to Failure (KRad)
506	8132	208	11.1	4	2.8±1.4	3.6	2.3	3	221
468*	8132	179	9.5	9	1.1±0.4	9.1	2.4	3	184
1524	8152	412	21.9	15	1.5±0.4	6.7	3.0	3	No TDF
1547	8152	141	7.5	6	1.3±0.5	7.7	6.4	3	145
1559	8152	425	22.6	18	1.3±0.2	7.7	5.5	3	No TDF
1548	8152	210	11.2	5	2.2±1.0	4.5	8.9	3	217
1443	8152	174	9.2	5	1.9±0.8	5.3	6.7	3	205
110*	8120	269	14.3	12	1.2±0.3	8.3	2.4	3	No TDF
95	8120	727	38.7	18	2.2±0.5	4.5	7.5	3	No TDF
13	8120	312	16.6	12	1.4±0.4	7.1	7.3	3	No TDF
13	8120	346	18.4	8	2.3±0.8	4.3	8.1	1	No TDF

*Error rate increased sharply after 9.8×10^{11} protons/cm². These data are not included in the table because the errors are probably due to total dose effects.

**Device previously tested and received 10^{11} protons/cm² without upset.

TABLE II

Types of Errors	506	1524	0110	0468	1547	1559	1548	1448	0015	0013 (3MHz)	0013 (1MHz)
	Multiple Bits (18)	1				1		1			1
Multiple Bits (4-8)	1					1					
Multiple Bits (2-3)	1		1	1					1		
Memem									1		
CRU			1						1		
Address Bits	2		4	2	1	3		2	7		
Data Bits	2	5	6	6	3	14	4	3	7	10	
Not Determined		7							1		8
Total Number of Upsets	4	15	12	9	6	18	5	5	18	12	8

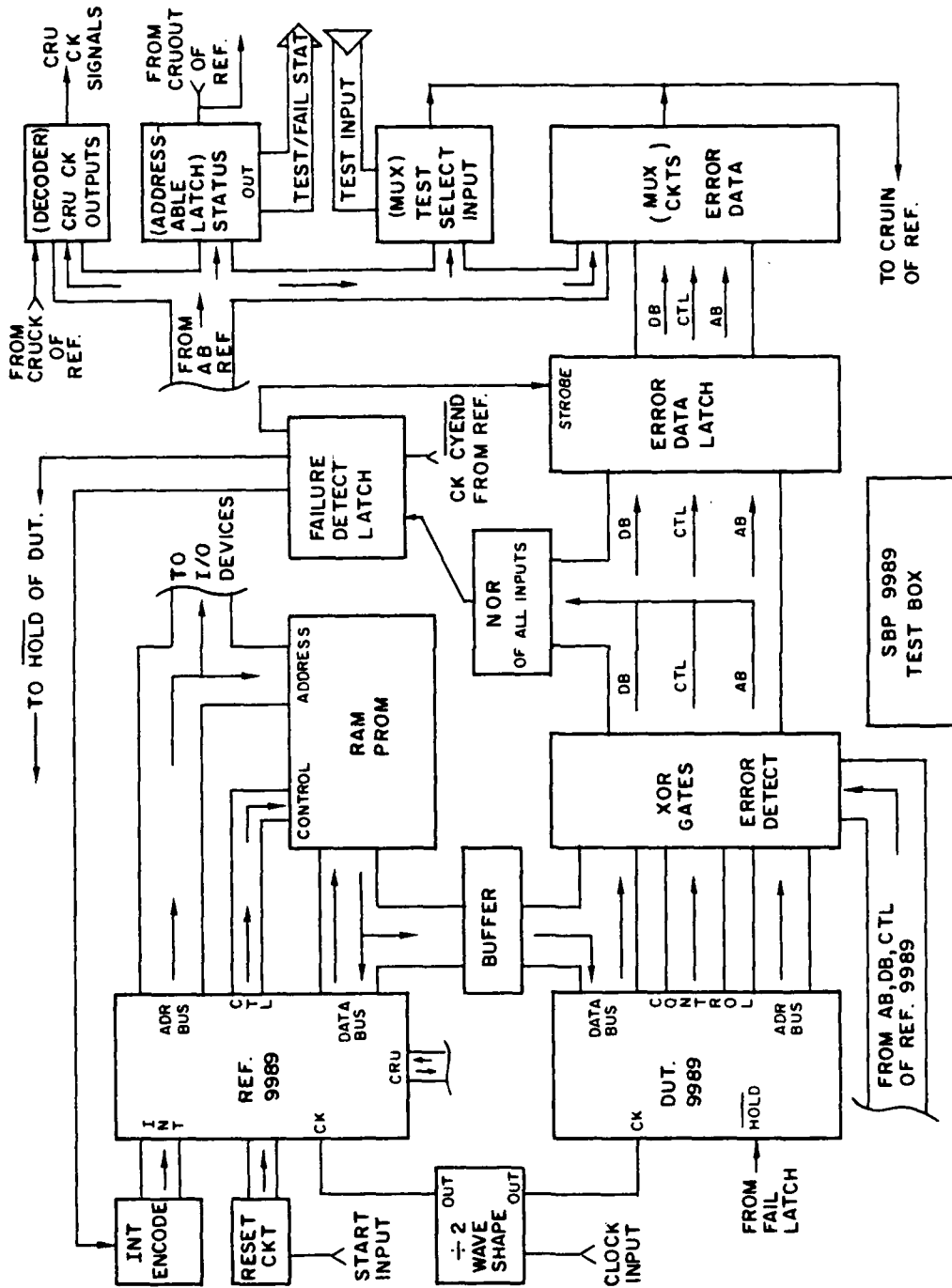


Figure 1. Hardware Configuration for Soft Upset Experiment with SBP9989.

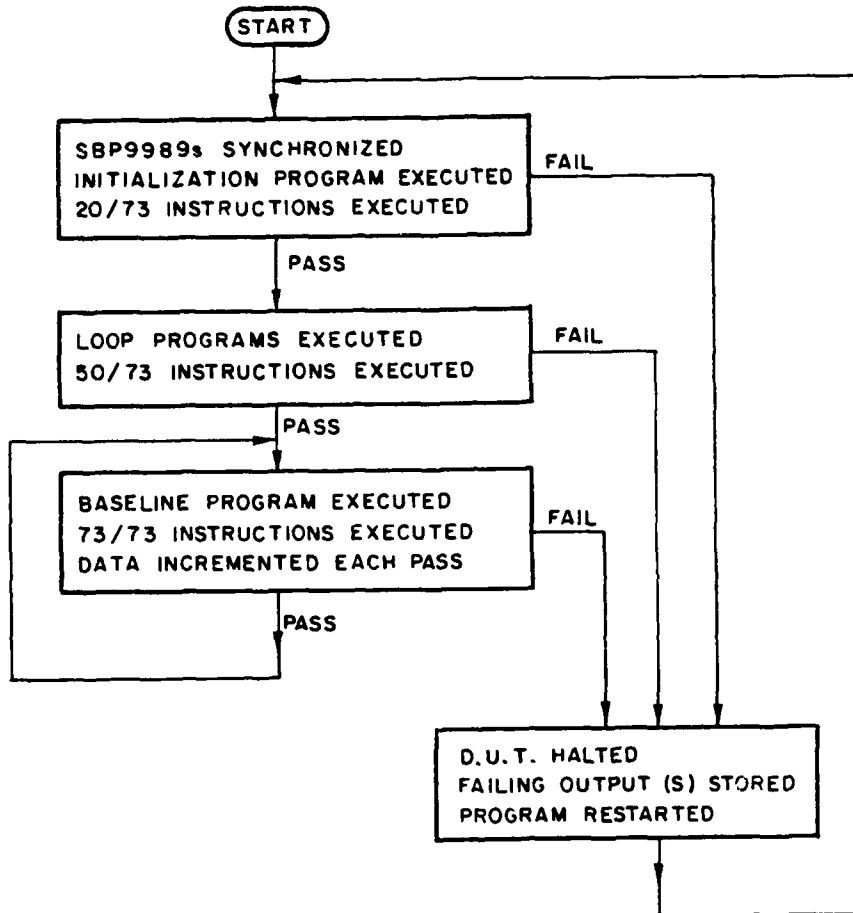


Figure 2. Flow Chart of Program for Testing SBP9989.

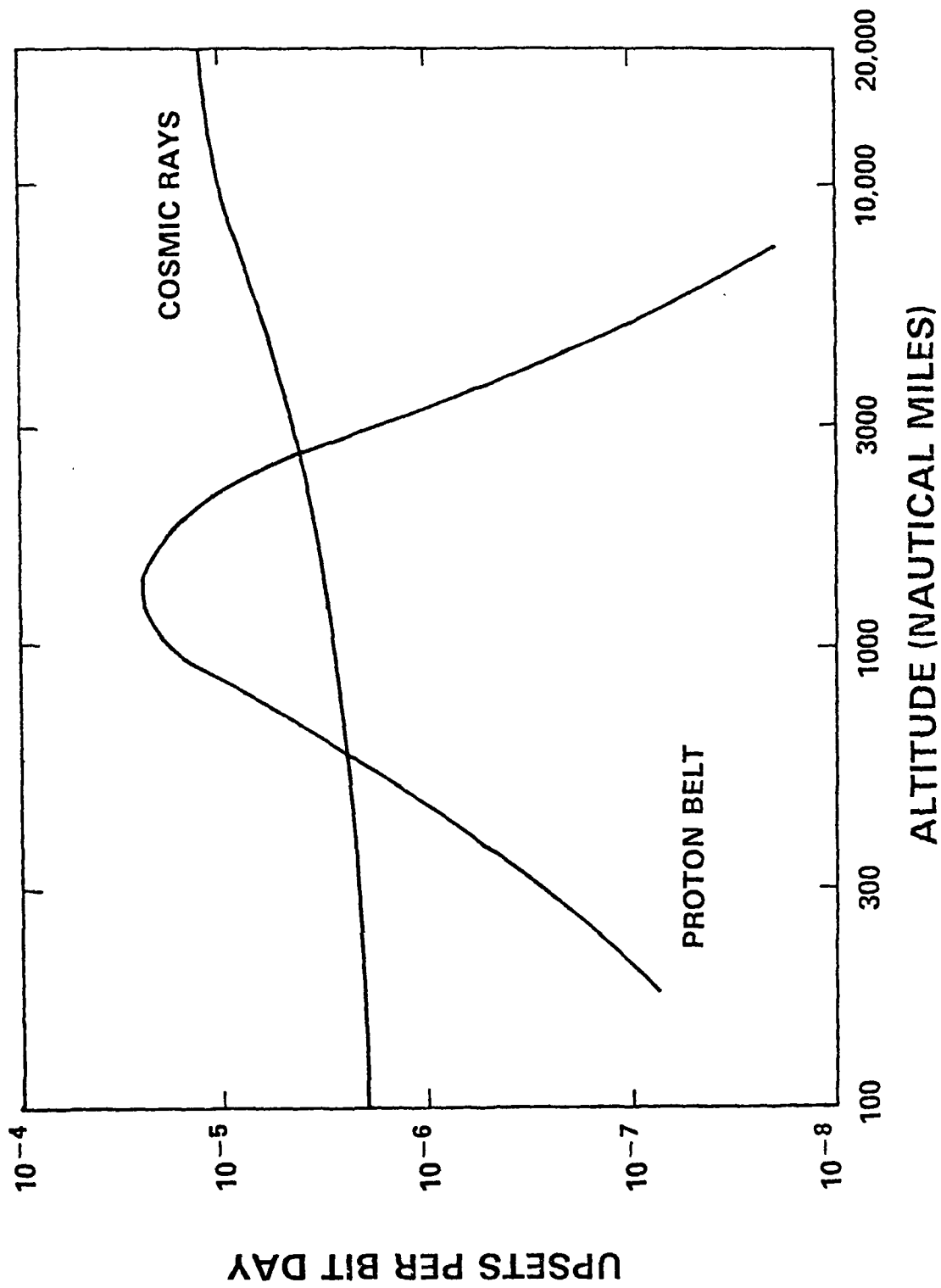


Figure 3. Single Event Upset Rate vs. Altitude for 600 Circular Orbit.

APPENDIX

SPECIAL VERSION SBP9989/SPR9000 DUAL MICROPROCESSOR TEST BOX

THIS PROGRAM IS DESIGNED TO OPERATE WITHIN A SPECIALLY DESIGNED HARDWARE INSTRUCTION TESTER (HIT). BASIC "HIT" HARDWARE CONSISTS OF A REFERENCE SBP9989/SBR9000 EXECUTING THIS PROGRAM, A SLAVE SBP9989/SBR9000 EXECUTING IN SYNCHRONIZATION, AND A COMPARISON CIRCUIT WHICH WILL DETECT DIFFERENCES. THE FOLLOWING ACTIONS OCCUR WHEN A FAILURE IS DETECTED:

1. DISABLE TEST FAILURE REGISTERS. THESE REGISTERS CONTAIN PASS/FAIL INFORMATION ON EACH DEVICE PIN.
2. PLACE SLAVE DEVICE IN "HOLD". THIS WILL "FREEZE" THE STATE OF THE FAILING DEVICE. (ALL OUTPUTS HI-Z)
3. INITIATE INTERRUPT (LEVEL 4) ROUTINE TO READ THE RESULTS FROM THE PASS/FAIL REGISTERS VIA THE "CRU-INPUT" INSTRUCTION. SOFTWARE DETERMINES WHETHER A FAILURE OCCURED ON THE "DATA BUS" AND SETS A STATUS INDICATOR IF ANY DATA PIN FAILED. LIKewise ANY "ADDRESS BUS" FAILURE RESULTS IN THE SETTING OF A STATUS INDICATOR. A THIRD STATUS INDICATOR IS PROVIDED FOR "CONTROL PIN" FAILURE. THE NORMAL OPERATIONAL PROCEDURE WILL BE AS FOLLOWS:
 - A. SELECT DESIRED INSTRUCTION TO BE TESTED. MAY BE SET LOCALLY ON SWITCHES OR REMOTELY BY PROVIDING A TTL LOGIC LEVEL ON THE "TEST SELECTION" INPUT PINS. (WITH THE ON-BOARD SWITCHES IN THE 'HIGH' POSITION.) THE BINARY CODE PROVIDES SELECTION OF 64 DIFFERENT TEST SEQUENCES. 46 TESTS (TEST01 TO 2E) ARE TWO INSTRUCTION LOOPS. TESTS 2F TO 3D ARE FOUR INSTRUCTION LOOPS. ALL 73 INSTRUCTIONS ARE EXECUTED BY THESE SHORT LOOPS. TEST 00 BRANCHES TO A LONGER DIAGNOSTIC LOOP PROGRAM. THE LAST TWO TESTS HAVE NOT BEEN IMPLEMENTED AND ARE AVAILABLE AS SPARE CAPABILITY. EACH OF THESE SPARES HAVE FOUR WORDS RESERVED IN PROM WHICH REMAIN UNPROGRAMMED FOR POSSIBLE FUTURE USE.
 - B. INITIATE TEST SEQUENCE. MAY BE DONE LOCALLY BY PRESSING THE "START" SWITCH. REMOTE OPERATION IS ALSO POSSIBLE BY APPLICATION OF A LOGIC LEVEL ZERO (GROUND) ON THE "START" INPUT PIN. ALL FAIL STATUS INDICATIONS ARE RESET AND THE TESTING STATUS BIT SET DURING THE INITIALIZATION. THE TESTING STATUS WILL REMAIN SET UNTIL A FAILURE IS DETECTED. THE 'TESTING' INDICATION MAY BE MONITORED REMOTELY ON THE TTL OUTPUT PIN PROVIDED.

IF A FAILURE IS DETECTED, A LEVEL 4 INTERRUPT WILL BE TAKEN BY THE REFERENCE DEVICE. FAILURE STATUS INDICATORS WILL BE SET AND THE TESTING STATUS RESET. THESE INDICATORS MAY BE MONITORED LOCALLY ON THE LED'S PROVIDED. THIS STATUS INFORMATION IS AVAILABLE REMOTELY VIA THE CRU OUTPUT PORT PROVIDED. TWO OUTPUT FORMATS ARE AVAILABLE. THE SHORT FORMAT PROVIDES ONE 16 BIT WORD WHICH CONTAINS THE "STATUS INDICATIONS" AND ALSO THE BINARY NUMBER REPRESENTING THE FAILING TEST. IF DESIRED, THE LONG FORMAT WILL PROVIDE THREE ADDITIONAL 16 BIT WORDS CONTAINING PASS/FAIL INDICATIONS ON ALL TESTED PINS. THE LONG/SHORT FUNCTION IS SWITCH SELECTED LOCALLY, AND ALSO REMOTELY AVAILABLE IF DESIRED.

ALL OF THE FOLLOWING OUTPUTS ARE TRANSMITTED VIA THE CRU OUTPUT PORT, BUT ONLY THE FIRST EIGHT ARE DISPLAYED ON-BOARD.

00 TESTING-SET DURING TESTING, RESET ON FAIL
01 FAILURE-SET ON FAIL, RESET AT START OF TEST
02 DATA FAIL-SET ONLY ON FAILURE OF ANY DATA PIN
03 ADDRESS FAIL-SET ONLY ON FAILURE OF ANY ADDRESS PIN
04 CONTROL FAIL-SET ONLY ON FAILURE OF ANY CONTROL PIN
05 SPARE
06 SPARE
07 LOOP TEST-SET DURING DIAG LOOP TESTING, RESET AT END OF LOOP PGM
08 TEST NUMBER (LSB))
09 TEST NUMBER) SWITCH POSITION READ AT THE START
0A TEST NUMBER) OF TESTING AND STORED WITHIN MEMORY.
0B TEST NUMBER) VALUE IS OUTPUT WHEN FAILURE IS
0C TEST NUMBER) DETECTED.
0D TEST NUMBER (MSB))
0E "0"
0F "0"

LONG FORMAT INCLUDES THESE ADDITIONAL OUTPUTS:

10 SWITCH POSITION (LSB))
11 SWITCH POSITION) TEST NUMBER SELECTED WHEN
12 SWITCH POSITION) FAILURE IS DETECTED (MAY BE
13 SWITCH POSITION) DIFFERENT FROM TEST BEING RUN)
14 SWITCH POSITION)
15 SWITCH POSITION (MSB))
16 SPARE SWITCH
17 LONG/SHORT OUTPUT FORMAT SWITCH (1=LONG)
18 "0"
19 SPARE
1A SPARE
1B SPARE
1C IAQ PASS/FAIL (1=FAILURE)
1D HOLDA PASS/FAIL (1=FAILURE)
1E DBIN PASS/FAIL (1=FAILURE)
1F MEMEN PASS/FAIL (1=FAILURE)
20 CRUOUT PASS/FAIL (1=FAILURE)
21 THRU 2F A14 THRU A0 PASS/FAIL (1=FAILURE)
30 THRU 3F D15 THRU D0 PASS/FAIL (1=FAILURE)

TEST PROGRAM

LOCATION	OBJECT CODE	MNEMONIC	COMMENT
FC00	0000	>0000	WS0
FC02	5555	>5555	WS1
FC04	0F0F	>0F0F	WS2
FC06	A0A0	>A0A0	WS3
FC08	AA00	>AA00	WS4
FC0A	FFFF	>FFFF	WS5
FC0C	0001	>0001	WS6
FC0E	102E	JMP TST13	WS7
FC10	FC00	REGA	WS8
FC12	FC70	TST14	WS9
FC14	10FE	>10FE	WS10 PC RELATIVE JUMP
FC16	FC07	REGA+7	WS11
FC18	0060	>0060	WS12 CRU BASE ADDRESS
FC1A	FC00	REGA	WS13 WP
FC1C	FC88	TST1A	WS14 PC
FC1E	FFFF	>FFFF	WS15 ST
FC20	0460	TST00	B @DIAG
FC22	FE42		BEGIN LOOP PROG
FC24	4042	TST01	SZC R2,R1
FC26	15FE		JQT TST01
FC28	E043	TST02	SOC R3,R1
FC2A	11FE		JLT TST02
FC2C	A0C3	TST03	A R3,R3
FC2E	16FE		JNE TST03
FC30	60C3	TST04	S R3,R3
FC32	13FE		JEQ TST04
FC34	8103	TST05	C R3,R4
FC36	1AFE		JL TST05
FC38	C3C1	TST06	MOV R1,R15
FC3A	1BFE		JH TST06
FC3C	0B03	TST07	SRC R3,0
FC3E	14FE		JHE TST07
FC40	0A94	TST08	SLA R4,9
FC42	14FE		JHE TST08
FC44	0983	TST09	SRL R3,8
FC46	18FE		JOC TST09
FC48	0B41	TST0A	SRA R1,4
FC4A	17FE		JNC TST0A
FC4C	0741	TST0B	ABS R1
FC4E	19FE		JNO TST0B
FC50	0702	TST0C	SET0 R2
FC52	10FE		JMP TST0C
FC54	06C4	TST0D	SWPB R4
FC56	10FE		JMP TST0D
FC58	0646	TST0E	DECT R6
FC5A	17FE		JNC TST0E
FC5C	0585	TST0F	INC R5
FC5E	18FE		JOC TST0F
FC60	0543	TST10	INV R3
FC62	18FE		JH TST10
FC64	0506	TST11	NEG R6

FC66	11FE		JLT TST11	
FC68	04C1	TST12	CLR R1	
FC6A	11FE		JLT TST12	
FC6C	0447	TST13	B R7	
FC6E	0000		>0000	SHOULD NEVER EXECUTE
FC70	1000	TST14	NOP	
FC72	0408		BLWP R8	ST ALL (800F)
FC74	048A	TST15	X R10	
FC76	0000		>0000	SHOULD NEVER EXECUTE
FC78	06A0	TST16	BL @TST16	
FC7A	FC78			
FC7C	03E0	TST17	LREX	
FC7E	10FE		JMP TST17	
FC80	03C0	TST18	CKOF	
FC82	10FE		JMP TST18	
FC84	03A0	TST19	CKON	
FC86	10FE		JMP TST19	
FC88	0380	TST1A	RTWP	
FC8A	0000		>0000	SHOULD NEVER EXECUTE
FC8C	0340	TST1B	IDLE	
FC8E	0000		>0000	SHOULD NEVER EXECUTE
FC90	02CF	TST1C	STST R15	ST ALL (840F)
FC92	10FE		JMP TST1C	
FC94	02AD	TST1D	STWP R13	
FC96	10FE		JMP TST1D	
FC98	3C4F	TST1E	DIV R15,R1	
FC9A	10FE		JMP TST1E	
FC9C	3881	TST1F	MPY R1,R2	
FC9E	10FE		JMP TST1F	
FCA0	2985	TST20	XOR R5,R6	ST 1=0 & 2=0
FCA2	11FE		JLT STS20	
FCA4	2583	TST21	CZC R3,R6	ST 2=1
FCA6	12FE		JLE TST21	
FCA8	23C4	TST22	COC R4,R15	ST 2=1
FCAA	14FE		JHE TST22	
FCAC	1000	TST23	NOP	
FCAE	2C00		XOP R0,0	ST ALL (860F)
FCB0	35C6	TST24	STCR R6,7	ST 3=1
FCB2	1CFE		JOP TST24	
FCB4	3001	TST25	LDCR R1,0	(16 BITS)
FCB6	10FE		JMP TST25	
FCB8	1FDB	TST26	TB -40	ST 2=0
FCBA	16FE		JNE TST26	
FCBC	1E0F	TST27	SBZ 15	
FCBE	10FE		JMP TST27	
FCC0	1D0F	TST28	SBO 15	
FCC2	10FE		JMP TST28	
FCC4	1F0F	TST29	TB 15	ST 2=1
FCC6	13FE		JEQ TST29	
FCC8	B103	TST2A	AB R3,4	ST 3=1
FCCA	1CFE		JOP TST2A	
FCCC	B11B	TST2B	AB *R11,R4	ST 3=1
FCCE	1CFE		JOP TST2B	
FCD0	B6C4	TST2C	AB R4,*R11	ST 3=1
FCD2	1CFE		JOP TST2C	

FCD4	B13B	TST2D	AB *R11+,R4	ST 5=1
FCD6	1CFE		JOP TST2D	
FCD8	A0FB	TST2E	A *R11+,R3	ST 2=0
FCDA	16FE		JNE TST2E	
FCDC	0282	TST2F	CI R2,>0F10	ST 0=0 & ST 2=0
FCDE	0F10			
FCE0	12FD		JLE TST2F	
FCE2	0000		>0000	
FCE4	0261	TST30	ORI R1,>3333	
FCE6	3333			
FCEB	10FD		JMP TST30	
FCEA	0000		>0000	
FCEC	0243	TST31	ANDI R3,>CCCC	
FCEE	CCCC			
FCF0	10FD		JMP TST31	
FCF2	0000		>0000	
FCF4	0224	TST32	AI R4,>F000	ST 4=1
FCF6	F000			
FCF8	19FD		JNO TST32	
FCFA	0000		>0000	
FCFC	0205	TST33	LI R5,>OFF0	ST 1=1
FCFE	OFF0			
FD00	15FD		JGT TST33	
FD02	0000		>0000	
FD04	0360	TST34	RSET	
FD06	0300		LIMI 15	
FD08	000F			
FD0A	10FC		JMP TST34	
FD0C	02E0	TST35	LWPI REGA	
FD0E	FC00			
FD10	10FD		JMP TST35	
FD12	0000		>0000	
FD14	B820	TST36	C @REGA,@REGA+12	ST 1=0 & ST 2=0
FD16	FC00			
FD18	FC0C			
FD1A	11FC		JLT TST36	
FD1C	BEE6	TST37	AB @REGA+B(R6),*R11+	ST 3=0
FD1E	FC0B			
FD20	1C01		JOP #+4	
FD22	10FC		JMP	
FD24	00B4	TST38	LST R4	ST 3 =0
FD26	18FE		JOC TST38	
FD28	00B3		LST R3	ST 2 =1
FD2A	13FC		JEG TST38	
FD2C	0098	TST39	LWP R8	ST 0 =0 & ST 2=0
FD2E	9103		CB R3,R4	
FD30	1AFD		JL TST39	
FD32	0000		0000	
FD34	D040	TST3A	MOV8 R0,R1	ST 5 =0
FD36	1CFE		JOP TST3A	
FD38	90C6		CB R6,R3	ST 5=1
FD3A	1CFC		JOP TST3A	
FD3C	0180	TST3B	DIV8 R0	ST 4=1
FD3E	19FE		JNO TST3B	
FD40	01C1		MPYS R1	ST 0=0

FD42	12FC		JLE TST3B	
FD44	05C9	TST3C	INCT R5	ST 3=1
FD46	17FE		JNC TST3C	
FD48	0601		DEC R1	ST 3=0
FD4A	17FC		JNE TST3C	
FD4C	F043	TST3D	SQCB R3,R1	ST 1=0 & ST 2=0
FD4E	14FE		JHE TST3D	
FD50	5042		SZCB R2,R1	ST 1=1
FD52	15FC		JQT TST3D	
FD54	0000	TST3E	DATA>0000,>0000,>0000,>0000	
FD56	0000			
FD58	0000			
FD5A	0000			
FD5C	0000	TST3F	DATA>0000,>0000,>0000,>0000	
FD5E	0000			
FD60	0000			
FD62	0000			END OF TEST INSTRUCTIONS

* INTERRUPT VECTOR TABLE *

FD64	1E00		SBZ	
FD66	0404		BLWP,R4	
FD68	30E4		LDCRU	
FD6A	0000			
FD6C	0000			
FD6E	0000			
FD70	0000			
FD72	1484		JHE \$-122	
FD74	18B2		JOC \$-124	
FD76	1007		JMP	
FD78	0000			
FD7A	0000			
FD7C	0000			
FD7E	1030		JMP LVL4	
FD80	0000			
FD82	0000			
FD84	0000			

* INITIALIZATION ROUTINE *

FD86	02E0		LWPI TSTWP	
------	------	--	------------	--

* LOAD RAM WITH INT. VECTORS *

FD88	0100			
FD8A	04CC		CLR R12	SET CRU BASE ADDRESS = 0000
FD8C	04C0		CLR R0	
FD8E	0201		LI R1,TSTWP	
FD90	0100			
FD92	0202		LI R2,REF	
FD94	FD76			
FD96	CC01		MOV R1,*R0+	
FD98	CC02		MOV R2,*R0+	
FD9A	0221		AI R1,>0020	

FD9C	0020	INCT R2
FD9E	05C2	CI R0.>003E
FDA0	0280	
FDA2	003E	JLT LOOP1
FDA4	11FB	MOV @REGA+16, *R0+
FDA6	CC20	
FDA8	FC10	LI R1, TST23
FDAA	0201	
FDAC	FCAC	MOV R1, *R0+
FDAE	CC01	

* READ INPUT SWITCHES & DEVELOP BRANCH ADDRESS *

FDB0	358A	STCR R10,6	
FDB2	098A	SRL R10,8	
FDB4	C24A	MOV R10,R9	
FDB6	028A	CI R10,>002E	
FDB8	002E		
FDBA	1504	JOT FOUR	
FDBC	0A29	SLA R9,2	MULTIPLY SWITCH # BY 4
FDBE	0229	AI R9,TST00	
FDC0	FC20		
FDC2	1005	JMP SETST	
FDC4	0229	AI R9,>FFD1	SUBTRACT # OF 2-WORD INSTRUCTIONS
FDC6	FFD1		
FDC8	0A39	SLA R9,3	MULTIPLY BY 8
FDCA	0229	AI R9,TST2F	
FDCC	FCDC		

* SET TEST STATUS INDICATORS *

FDCE	C1CA	MOV R10,R7	SETST
FDD0	0A87	SLA R7,8	
FDD2	0587	INC R7	
FDD4	3007	LDCR R7,0	(16 BITS) TESTING & SWITCH
FDD6	0208	LI R8,REGA	POSITIONS
FDD8	FC00		
FDDA	0300	LIMI >F	
FDDC	000F		
FDDE	0408	BLWP R8	VECTOR TO SELECTED TEST

* TEST FAILURE INTERRUPT (LEVEL 4) WP=>0180

FDE0	0200	LI R0,TSTWP+14	
FDE2	010E		
FDE4	04C7	CLR R7	
FDE6	D1D0	MOV8 *R0,R7	FAILING TEST #
FDE8	020A	LI R10,32	
FDEA	0020		
FDEC	04CC	CLR R12	
FDEE	1E00	SBZ 0	TURN OFF TESTING LIGHT
FDFO	3406	STCR R6,0	(16 BITS) INPUT CONTROL
FDf2	A30A	A R10,R12	PASS/FAIL
FDf4	3405	STCR R5,0	(16 BITS) INPUT ADDRESS
FDf6	A30A	A R10,R12	PASS/FAIL
FDf8	3404	STCR R4,0	(16 BITS) INPUT DATA PASS/FAIL
FDFA	04C8	CLR R8	DEVELOP FAIL STATUS

* TEST CONTROL PINS FOR FAILURE *

FDFC	C0C5	MOV R5, R3	
FDFE	0B13	SRA R3, 1	TEST CRUOUT
FE00	1702	JNC CTRL	
FE02	0588	INC R8	
FE04	1005	JMP ADDR	
FE06	C0C6	MOV R6, R3	CTRL
FE08	0983	SRL R3, 8	
FE0A	C0C3	MOV R3, R3	
FE0C	1301	JEQ ADDR	
FE0E	0588	INC R8	
FE10	0A18	SLA R8, 1	ADDR

* TEST ADDRESS CONTROL PINS FOR FAILURE *

FE12	C0C5	MOV R5, R3	
FE14	0913	SRL R3, 1	
FE16	C0C3	MOV R3, R3	
FE18	1301	JEQ	
FE1A	0588	INC R8	
FE1C	0A18	SLA R8, 1	
FE1E	C0C4	MOV R4, 3	
FE20	1301	JEQ FIX	
FE22	0588	INC R8	
FE24	0A18	SLA R8, 1	
FE26	0588	INC R8	
FE28	0A18	SLA R8, 1	
FE2A	D207	MOVB R7, R8	
FE2C	04CC	CLR R12	
FE2E	3008	LDCR R8, 0	(16 BITS)

* TEST FOR SHORT OR LONG OUTPUT FORMAT *

FE30	1F07	TB 7	
FE32	1606	JNE DONE	
FE34	A30A	A R10, R12	
FE36	3006	LDCR R6, 0	(16 BITS)
FE38	A30A	A R10, R12	
FE3A	3005	LDCR R5, 0	(16 BITS)
FE3C	A30A	A R10, R12	
FE3E	3004	LDCR R4, 0	(16 BITS)
FE40	0340	IDLE	DONE

* END OF CONTROL SECTION *

```

*****
*
* THIS PROGRAM IS DESIGNED TO BE EXECUTED FROM A 256-WORD
* ROM IN THE 990 CPU.
*
* THE FOLLOWING REGISTERS ARE NOT ALTERED:
*       R0,R9,R12,R13,R14,R15
*
* OPERATOR INDICATIONS:
*
*       NORMAL:           FAULT LIGHT ON AT START OF TEST,
*                          OFF AT END WITH IDLE ON.
*
*       CPU ERROR:       CPU WILL HANG IN A LOOP
*
*****

```

* FAULT LIGHT BIT DISPLACEMENT *

FE42	02E0	LWPI WPADR	INITIAL WORKSPACE
FE44	00E0		
FE46	04CC	CLR R12	CRU BASE = >0000
FE48	1D07	SBO FAULT	FAULT LIGHT ON

* PRELIMINARY CPU TESTS *

FE4A	0208	LI R8,>7FFF	
FE4C	7FFF		
FE4E	05C8	INCT R8	
FE50	19FF	JNO *	
FE52	C040	MOV R0,R1	SAVE R0
FE54	C008	MOV R8,R0	RO = SHIFT COUNT = 1
FE56	0858	SRA R8,5	DO SRA
FE58	0918	SRL R8,1	THEN A SRL
FE5A	0808	SRA R8,0	SHIFT ONCE MORE
FE5C	C001	MOV R1,R0	RESTORE R0
FE5E	0288	CI R8,>3F00	CHECK RESULT
FE60	3F00		
FE62	16FF	JNE *	
FE64	0704	SETO R4	
FE66	0244	ANDI R4,>5555	CHECK ANDI
FE68	5555		
FE6A	06A0	BL @CHKAL5	CHECK RESULT
FE6C	FE7E		
FE6E	0264	ORI R4,>5555	CHECK ORI
FE70	5555		
FE72	06A0	BL @CHKAL5	
FE74	FE7E		

* TEST STATUS REGISTER INSTRUCTIONS *

FE76	0300	LIMI >F	CHECK LIM1
FE78	000F		
FE7A	0420	BLWP @DBLWP	TEST BLWP
FE7C	FE96		

* SUBROUTINE TO CHECK R4 = >5555 *

```
FE7E      0264      ORI R4,>AAAA
FE80      AAAA
FE82      0584      INC R4
FE84      16FF      JNE $
FE86      045B      B *R11                RETURN
```

* ENTRY POINT FOR BLWP *

```
FE88      02CB      STST R11
FE8A      081B      SRA R11,1           CHECK INT MASK . NE. 0
FE8C      17FF      JNC $
FE8E      04CF      CLR R15             CLEAR INT LEVEL
FE90      020E      LI R14,RBLWP       AND PC
FE92      FE9A
FE94      0380      RTWP
```

*WP AND ENTRY FOR BLWP TEST *

* NOTE THAT R13-R15 DURING THE TEST IS REALLY R3-5 *

```
FE96      00CC      DATA WPADR-20
FE98      FE88      DATA TBLWP
```

* RETURN FROM BLWP TEST *

```
FE9A      02C7      STST R7
FE9C      C047      MOV R7,R1           CHECK STATUS = 0
FE9E      16FF      JNE $
FEA0      0300      LIMI >F           RE-ENABLE INTERRUPTS
FEA2      000F
```

* TWO NUMBERS, A AND B, ARE GENERATED FOR THE CPU MAIN TEST. *
* A IS A SEQUENTIAL NUMBER FROM 0 TO 32K. B IS THE VALUE OF A *
* ROTATED 5 PLACES TO THE RIGHT, AND THEN BYTE SWAPED. *

```
FEA4      C0B1      MOV R1,R2           R1 = A
FEA6      0852      SRC R2,5
FEA8      06C2      SWPB R2                R2 = B
```

* ABSOLUTE VALUE TEST. IF B IS NEGATIVE, IT IS NEGATED BY *
 * INVERSION, INCREMENTED BY 2, AND THEN DECREMENTED BY 1 *

FEAA	C2C2	MOV R2,R11	
FEAC	074B	ABS R11	ABS OF B IN R11
FEAE	C0C2	MOV R2,R3	
FEBO	0A13	SLA R3,1	
FEB2	1709	JNC WASPOS	CHECK IF POS OR NEG
FEB4	054B	INV R11	NEGATE TO MAKE POSITIVE
FEB6	808B	C R11,R2	
FEB8	18FF	JH \$	
FEBA	1101	JLT \$+4	
FEBC	10FF	JMP \$	
FEBE	82C2	C R2,R11	
FECO	1AFF	JL \$	
FEC2	05CB	INCT R11	
FEC4	060B	DEC R11	
FEC6	808B	C R11,R2	SEE IF B = B
FEC8	16FF	JNE \$	

* ADD A AND B USING ADD. SAVE SUM IN R3 *

FECA	C0C1	MOV R1,R3	
FECB	A0C2	A R2,R3	R3 = A + B
		* ADD A AND B BY: NEGATE A, SUBTRACT B FROM A,	*
		* NEGATE RESULT FOR ANSWER.	*

FECE	C101	MOV R1,R4	
FEDO	0504	NEG R4	NEGATE A
FED2	6102	S R2,R4	SUBTRACT B
FED4	0504	NEG R4	INVERT
FED6	9103	CB R3,R4	USE CB'S TO CHECK EQUAL
FED8	18FF	JH \$	
FEDA	1201	JLE \$+4	
FEDC	10FF	JMP \$	
FEDE	9820	CB @WPADR+7,@WPADR+9	
FEE0	00E7		
FEE2	00E9		
FEE4	16FF	JNE \$	
FEE6	20C4	C0C R4,R3	CHECK EQUAL AGAIN WITH
FEE8	16FF	JNE \$	C0C AND C2C
FEEA	0544	INV R4	
FEEC	24C4	C2C R4,R3	
FEED	16FF	JNE \$	
FEFO	0544	INV R4	
FEF2	7103	SB R3,R4	CHECK AGAIN WITH SB'S
FEF4	11FF	JLT \$	
FEF6	16FF	JNE \$	
FEF8	7820	SB @WPADR+7,@WPADR+9	
FEFA	00E7		
FEFC	00E9		
FEFE	16FF	JNE \$	

* ADD A AND B WITH ADD BYTE *

FF00	D102	MOVB R2, R4	
FF02	D820	MOVB @WPADR+5, @WPADR+9	B IN R4
FF04	00E3		
FF06	00E9		
FF08	B820	AB @WPADR+3, @WPADR+9	
FF0A	00E3		
FF0C	00E9		
FF0E	1702	JNC NOCRY	
FF10	0224	AI R4, >100	ADD CARRY
FF12	0100		
FF14	B101	AB R1, R4	
FF16	B103	C R3, R4	CHECK IF EQUAL
FF18	16FF	JNE *	
FF1A	0583	INC R3	CHECK INC AND DECT
FF1C	0583	INC R3	
FF1E	B0C4	C R4, R3	
FF20	13FF	JEQ *	SHOULD NOT BE EQUAL
FF22	15FF	JGT *	
FF24	0643	DECT R3	
FF26	B103	C R3, R4	
FF28	16FF	JNE *	SHOULD BE EQUAL AGAIN

* SWAP BYTE TEST *

* RESULT OF SWPB IS COMPARED WITH RESULT OF SHIFTING *
 * AND MERGING THE SAME WORD. *

FF2A	C143	MOV R3, R5	
FF2C	C185	MOV R5, R6	
FF2E	C1C5	MOV R5, R7	
FF30	06C7	SWPB R7	SWAP R7
FF32	0985	SRL R5, 8	SWAP WITH SHIFTS
FF34	0A86	SLA R6, 8	
FF36	E185	SOC R5, R6	MERGE WITH SOC
FF38	B1C6	C R6, R7	SEE IF EQUAL
FF3A	16FF	JNE *	
FF3C	08B6	SRC R6, 8	CHECK SRC
FF3E	B0C6	C R6, R3	
FF40	16FF	JNE *4	

* MULTIPLY TEST *

* A AND B ARE MULTIPLIED WITH A SHIFT-AND-ADD ROUTINE. *
 * THE DOUBLE-PRECISION RESULT IS THEN COMPARED AGAINST *
 * THE RESULT FROM THE MULTIPLY INSTRUCTION. *

FF42	020A	LI R10,16	R10 = SHIFT LOOP COUNTER
FF44	0010		
FF46	04CB	CLR R11	R11 = HIGH ORDER RESULT
FF48	04CB	CLR R8	R8 = LOW ORDER RESULT
FF4A	04C7	CLR R7	R7 = HIGH ORDER OF B
FF4C	C182	MOV R2,R6	R6 = LOW ORDER OF B
FF4E	C141	MOV R1,R5	R5 IS A
FF50	0B15	SRA R5,1	SHIFT A BIT FROM A
FF52	1704	JNC NOADD	SEE IF A ONE
FF54	A2C6	A R6,R11	DOUBLE PRECISION ADD
FF56	1701	JNC \$+4	
FF58	0588	INC R8	ADD CARRY IF ANY
FF5A	A207	A R7,R8	
FF5C	0A17	SLA R7,1	DOUBLE ARITHMETIC SHIFT
FF5E	0A16	SLA R6,1	LEFT ON B
FF60	1801	JOC \$+4	
FF62	1001	JMP NOCARY	
FF64	0587	INC R7	ADD CARRY IF ANY
FF66	060A	DEC R10	SEE IF DONE
FF68	16F3	JNE MULLUP	

* DD HARDWARE MULTIPLY *

FF6A	C0C1	MOV R1,R3	R3 = A
FF6C	38E0	MPY @WPADR+4,R3	DO THE MULTIPLY
FF6E	00E4		
FF70	8203	C R3,R8	CHECK BOTH WORDS OF RESULT
FF72	16FF	JNE \$	
FF74	82C4	C R4,R11	
FF76	16FF	JNE \$	

* DIVIDE TEST *

* RESULT OF MULTIPLY IS DIVIDED BY B TO GET A. *
 * IF A>B, A IS ADDED BEFORE DIVISION TO PRODUCE *
 * A REMAINDER OF A, UNLESS SUCH ADDITION CAUSES *
 * A CARRY. *

FF7B	80B1	C R1,R2	SEE IF A>B
FF7A	140B	JHE NADD	
FF7C	A101	A R1,R4	ADD A TO DIVIDEND
FF7E	1702	JNC \$+6	IF CARRY OCCURED
FF80	6101	S R1,R4	SUBTRACT A BACK OUT
FF82	1004	JMP NADD	
FF84	3CC2	DIV R2,R3	DO THE DIVIDE
FF86	04A0	X @CR1R4	COMPARE A TO REMAINDER
FF8B	FF80		
FF8A	1003	JMP CHKREM	
FF8C	3CE0	DIV @WPADR+4,R3	DO THE DIVIDE
FF8E	00E4		
FF90	C104	MOV R4,R4	SEE IF REMAINDER ZERO
FF92	16FF	JNE \$	

* PARITY TEST. LH BYTE OF A USED. *

FF94	D181	MOVB R1,R6	BYTE IN R6
FF96	0207	L1 R7,B	R7 = SHIFT COUNTER
FF98	000B		
FF9A	04CB	CLR R8	R8 = BIT COUNTER
FF9C	0A16	SLA R6,1	ROTATE ONCE
FF9E	1701	JNC \$+4	
FFA0	058B	INC R8	COUNT BIT IF A ONE
FFA2	0607	DEC R7	DECREMENT SHIFT COUNTER
FFA4	15FB	JGT FLOOP	
FFA6	0B1B	SRC R8,1	CHECK SOFTWARE PARITY
FFA8	1703	JNC EVNPAR	
FFAA	D041	MOVB R1,R1	GENERATE HARDWARE PARITY
FFAC	1C03	JOP PAROK	
FFAE	10FF	JMP \$	
FFB0	B1C1	AB R1,R7	GENERATE HARDWARE PARITY
FFB2	1CFF	JOP \$	

* CHECK B = B WITH SZC AND SZCB *

FFB4	C142	MOV R2,R5	
FFB6	4142	SZC R2,R5	
FFB8	16FF	JNE *	
FFBA	A142	A R2,R5	
FFBC	5142	SZCB R2,R5	
FFBE	16FF	JNE *	
FFC0	5820	SZCB @WPADR+5,@WPADR+11	
FFC2	00E3		
FFC4	00EB		
FFC6	16FF	JNE *	
FFC8	070B	SETC R11	
FFCA	42C2	SZC R2,R11	
FFCC	13FF	JEQ *	R11 SHOULD BE .NE. ZERO

* CHECK A = A WITH SOCB *

FFCE	F141	SOCB R1,R5	
FFD0	06C5	SWPB R5	
FFD2	F160	SOCB @WPADR+3,R5	
FFD4	00E3		
FFD6	0885	SRC R5,B	
FFD8	6141	S R1,R5	CHECK A = A
FFDA	16FF	JNE *	

*
* SET VALUE OF A FOR NEXT LOOP
*
* CHECK FOR VALUES OF A THAT WILL CAUSE
* OVERFLOW OF B
*

FFDC	02B1	CI R1>471B	
FFDE	471B		
FFE0	1601	JNE +1	
FFE2	05B1	INC R1	
FFE4	02B1	CI R1>0FFF	
FFE6	0FFF		
FFE8	1601	JNE +1	
FFEA	05B1	INC R1	
FFEC	05B1	INC R1	
FFEE	02B1	CI R1>8000	
FFF0	8000		
FFF2	1302	JEQ +2	
FFF4	0460	B to	
FFF6	FEA4	FEA4	
FFF8	1E07	SBZ FAULT	
FFFA	0340	IDLE	END OF TEST
FFFC	0100	WP = 0100	
FFFE	FDB6	PC = FDB6	