

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
-----	-------------	-------	-------	------

```

2 *****
3 *
4 *                               SKEY390Z
5 *
6 *****
7 *
8 *   This program verifies proper functioning of the following
9 *   System/390 and z/Architecture Storage Key instructions:
10 *
11 *       ISKE, IVSK, RRBE, SSKE, TB, TPROT      (both S/390 & z/Arch)
12 *       IRBM, RRBm, PFMF                      (z/Architecture only)
13 *
14 *   NOTE: due to varying support for certain instructions under
15 *   certain situations, some tests may crash at certain points.
16 *   If the crash is expected, then the crash is ignored and the
17 *   test that was being attempted is simply skipped.
18 *
19 *   PLEASE ALSO NOTE the program is purposely designed to branch to
20 *   an odd address should any test fail (such as the condition code
21 *   not being the expected value). The Program Check handler routine
22 *   when it notices the Program Old PSW is an odd address, backs up
23 *   the address by 5 bytes and uses that as the test's failing PSW.
24 *
25 *   Thus when any test fails, the disabled wait PSW points directly
26 *   to the failing instruction (i.e. the branch following the failed
27 *   comparison). ALSO NOTE that Hercules also issues a "Instruction
28 *   fetch error" message to its hardware console too whenever this
29 *   occurs (due to the PSW address being odd causing it to be unable
30 *   to fetch the next instruction), which is expected.
31 *
32 *   FINALLY, in order to support successfully running on non-Hercules
33 *   systems, we utilize the Hercules "CPUVERID xx FORCE" statement
34 *   to allow us to detect if we're running under Hercules. On "real
35 *   iron" (including zPDT and RD&T) the CPUID "Version code" (which
36 *   they're now calling the "Environment" field) will be either 00,
37 *   C1 or D3, whereas on Hercules it will be the value specified on
38 *   your "CPUVERID xx FORCE" statement (which is C8 for SKEY390Z).
39 *
40 *****

```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT				
				42	*****			
				43	* LOW CORE			
				44	*****			
00000000		00000000	0000307F	46	TEST	START 0		
		00000000		47		USING TEST,0	Use absolute addressing	
00000000		00000000	00000000	49	ORG	TEST+X'00'	S/390 Restart new PSW	
00000000	00080000			50	DC	XL4'00080000'	S/390 Restart new PSW	
00000004	00000200			51	DC	A(BEGIN)	S/390 Restart new PSW	
00000008		00000008	00000028	53	ORG	TEST+X'28'	S/390 Program old PSW	
		00000028	00000001	54	PGMOLD	EQU *	S/390 Program old PSW	
00000028		00000028	00000068	56	ORG	TEST+X'68'	S/390 Program new PSW	
00000068	00080000			57	DC	XL4'00080000'	S/390 Program new PSW	
0000006C	000006A0			58	DC	A(PGMCHK)	S/390 Program new PSW	
00000070		00000070	0000008C	60	ORG	TEST+X'8C'	Program interrupt code	
0000008C	00000000			61	PGMCODE	DC F'0'	Program interrupt code	
		00000001	00000001	63	PGM_OPERATION_EXCEPTION	EQU	X'0001'	
		00000006	00000001	64	PGM_SPECIFICATION_EXCEPTION	EQU	X'0006'	
00000090		00000090	00000150	66	ORG	TEST+X'150'	z/Arch Program OLD PSW	
		00000150	00000001	67	ZPGMOLD	EQU *	z/Arch Program OLD PSW	
00000150		00000150	000001A0	69	ORG	TEST+X'1A0'	z/Arch Restart new PSW	
000001A0	00000001			70	DC	XL4'00000001'	z/Arch Restart new PSW	
000001A4	80000000			71	DC	XL4'80000000'	z/Arch Restart new PSW	
000001A8	00000000			72	DC	XL4'00000000'	z/Arch Restart new PSW	
000001AC	00000370			73	DC	A(ZARCH)	z/Arch Restart new PSW	
000001B0		000001B0	000001D0	75	ORG	TEST+X'1D0'	z/Arch Program new PSW	
000001D0	00000001			76	DC	XL4'00000001'	z/Arch Program new PSW	
000001D4	80000000			77	DC	XL4'80000000'	z/Arch Program new PSW	
000001D8	00000000			78	DC	XL4'00000000'	z/Arch Program new PSW	
000001DC	00000724			79	DC	A(ZPGMCHK)	z/Arch Program new PSW	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				81 *****
				82 * INITIALIZATION
				83 *****
000001E0		000001E0	00000200	85 ORG TEST+X'200' Start of test program
00000200	B202 07A0		000007A0	87 BEGIN STIDP CPUID Save CPU ID (for later test for VM)
				88 *
				89 * The below STFL will fail with an Operation Exception on those
				90 * operating systems that were not written with support for the
				91 * 'N3' series of instructions (i.e. z/Architecture instructions
				92 * backported to ESA/390, such as the STFL instruction itself),
				93 * such as VM/ESA (which doesn't support 'N3' instructions).
				94 *
00000204	B2B1 0000		00000000	95 STFL 0 Store Facility List (just for fun)
00000208	D203 07B0 00C8	000007B0	000000C8	96 STFLPC MVC STFL390,X'C8' Save STFL results for posterity
				98 * Fall through to begin S/390 mode tests.....
				100 *
				101 *
				102 *
				103 *
				104 *
				105 *
				106 *
				107 *
				108 *
				109 *
				110 *
				111 *
				112 *
				113 *
				114 *
				115 *
				116 *
				117 *
				118 *
				119 *
				120 *
				121 *
				122 *
				V
				V
				V

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				124 *****
				125 * SSKE, ISKE, RRBE (390 mode)
				126 *****
0000020E	BF11 093A		0000093A	128 ICM R1,B'0001',=X'1C'
00000212	5820 08E8		000008E8	129 L R2,=A((4*_4K)+X'900')
00000216	B22B 0012			130 SSKE R1,R2
0000021A	BF11 093B		0000093B	131 ICM R1,B'0001',=X'26'
0000021E	5820 08EC		000008EC	132 L R2,=A((5*_4K)+X'A00')
00000222	B22B 0012			133 SSKE R1,R2
00000226	BF11 093C		0000093C	134 ICM R1,B'0001',=X'4E'
0000022A	5820 08F0		000008F0	135 L R2,=A((6*_4K)+X'B00')
0000022E	B22B 0012			136 SSKE R1,R2
				137 *****
00000232	5820 08E8		000008E8	138 L R2,=A((4*_4K)+X'900')
00000236	B229 0012			139 ISKE R1,R2
0000023A	BD11 093A		0000093A	140 CLM R1,B'0001',=X'1C'
0000023E	4770 023F		0000023F	141 BNE *+1
00000242	5820 08EC		000008EC	142 L R2,=A((5*_4K)+X'A00')
00000246	B229 0012			143 ISKE R1,R2
0000024A	BD11 093B		0000093B	144 CLM R1,B'0001',=X'26'
0000024E	4770 024F		0000024F	145 BNE *+1
00000252	5820 08F0		000008F0	146 L R2,=A((6*_4K)+X'B00')
00000256	B229 0012			147 ISKE R1,R2
0000025A	BD11 093C		0000093C	148 CLM R1,B'0001',=X'4E'
0000025E	4770 025F		0000025F	149 BNE *+1
				150 *****
00000262	5820 08E8		000008E8	151 L R2,=A((4*_4K)+X'900')
00000266	B22A 0002			152 RRBE R0,R2
0000026A	47D0 026B		0000026B	153 BC B'1101',*+1 NOT CC2 = was REF 1, CHG 0
0000026E	5820 08EC		000008EC	154 L R2,=A((5*_4K)+X'A00')
00000272	B22A 0002			155 RRBE R0,R2
00000276	47E0 0277		00000277	156 BC B'1110',*+1 NOT CC3 = was REF 1, CHG 1
0000027A	5820 08F0		000008F0	157 L R2,=A((6*_4K)+X'B00')
0000027E	B22A 0002			158 RRBE R0,R2
00000282	47E0 0283		00000283	159 BC B'1110',*+1 NOT CC3 = was REF 1, CHG 1
				160 *****
00000286	5820 08E8		000008E8	161 L R2,=A((4*_4K)+X'900')
0000028A	B229 0012			162 ISKE R1,R2
0000028E	BD11 093D		0000093D	163 CLM R1,B'0001',=X'18'
00000292	4770 0293		00000293	164 BNE *+1
00000296	5820 08EC		000008EC	165 L R2,=A((5*_4K)+X'A00')
0000029A	B229 0012			166 ISKE R1,R2
0000029E	BD11 093E		0000093E	167 CLM R1,B'0001',=X'22'
000002A2	4770 02A3		000002A3	168 BNE *+1
000002A6	5820 08F0		000008F0	169 L R2,=A((6*_4K)+X'B00')
000002AA	B229 0012			170 ISKE R1,R2
000002AE	BD11 093F		0000093F	171 CLM R1,B'0001',=X'4A'
000002B2	4770 02B3		000002B3	172 BNE *+1

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				174 *****
				175 * IVSK, TPROT, TB (390 mode)
				176 *****
000002B6	BF11 0940		00000940	178 ICM R1,B'0001',=X'6E'
000002BA	5820 08F4		000008F4	179 L R2,=A((7*_4K)+X'900')
000002BE	B22B 0012			180 SSKE R1,R2
				181 * The below could fail on systems with "ESA/390 Compatibility
				182 * Mode" installed/enabled, which does not support ESA/390 DAT.
000002C2	B701 08B8		000008B8	183 LCTL R0,R1,CR0_1_39 Configure DAT
000002C6	8000 0941		00000941	184 SSM =X'04' Enable DAT
000002CA	B223 0012			185 SSMPC IVSK R1,R2
000002CE	8000 0942		00000942	186 SSM =X'00' Disable DAT
000002D2	BD11 0943		00000943	187 CLM R1,B'0001',=X'68'
000002D6	4770 02D7		000002D7	188 BNE *+1
				189 *****
000002DA	BF11 0944		00000944	190 SKIPIVSK ICM R1,B'0001',=X'10'
000002DE	5820 08F8		000008F8	191 L R2,=A(4*_4K)
000002E2	B22B 0012			192 SSKE R1,R2
000002E6	5810 08E8		000008E8	193 L R1,=A((4*_4K)+X'900')
000002EA	BF21 0944		00000944	194 ICM R2,B'0001',=X'10'
000002EE	E501 1000 2000	00000000	00000000	195 TPROT 0(R1),0(R2)
000002F4	4770 02F5		000002F5	196 BC B'0111',*+1 NOT CC0 = FETCH OK, STORE OK
000002F8	BF21 0945		00000945	197 ICM R2,B'0001',=X'20'
000002FC	E501 1000 2000	00000000	00000000	198 TPROT 0(R1),0(R2)
00000302	47B0 0303		00000303	199 BC B'1011',*+1 NOT CC1 = FETCH OK, STORE NO
00000306	BF11 093D		0000093D	200 ICM R1,B'0001',=X'18' (set fetch protect)
0000030A	5820 08F8		000008F8	201 L R2,=A(4*_4K)
0000030E	B22B 0012			202 SSKE R1,R2
00000312	5810 08F8		000008F8	203 L R1,=A(4*_4K)
00000316	BF21 0945		00000945	204 ICM R2,B'0001',=X'20'
0000031A	E501 1000 2000	00000000	00000000	205 TPROT 0(R1),0(R2)
00000320	47D0 0321		00000321	206 BC B'1101',*+1 NOT CC2 = FETCH NO, STORE NO
				207 * We must skip the 'TB' (Test Block) test if we're running under
				208 * VM or are NOT running under Hercules since neither environment
				209 * has any command to sets block(s) of storage to "unusable".
00000324	95FF 07A0		000007A0	210 CLI CPUID,X'FF' Are we running under VM?
00000328	4780 0350		00000350	211 BE SKIPTB39 Yes, then skip 'TB' tests
0000032C	95C8 07A0		000007A0	212 CLI CPUID,X'C8' Are we running under Hercules?
00000330	4770 0350		00000350	213 BNE SKIPTB39 No, then skip 'TB' tests
00000334	1F00			214 SLR R0,R0 Required by TB instruction
00000336	5820 08FC		000008FC	215 L R2,=A((10*_4K)+X'DEF') Requires Herc 'f- A000' cmd
0000033A	B22C 0012			216 TB R1,R2
0000033E	47B0 033F		0000033F	217 BC B'1011',*+1 NOT CC1 = Unusable/BAD block
00000342	1F00			218 SLR R0,R0 Required by TB instruction
00000344	5820 0900		00000900	219 L R2,=A((11*_4K)+X'FED') Requires Herc 'f- B000' cmd
00000348	B22C 0012			220 TB R1,R2
0000034C	47B0 034D		0000034D	221 BC B'1011',*+1 NOT CC1 = Unusable/BAD block
		00000350	00000001	222 SKIPTB39 EQU *

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT				
					225	*****			
					226	* Switch to z/Architecture mode...			
					227	*****			
00000350	1F00				229	SLR	R0,R0	Start clean	
00000352	4110	0001		00000001	230	LA	R1,1	Request z/Arch mode	
00000356	1F22				231	SLR	R2,R2	Start clean	
00000358	1F33				232	SLR	R3,R3	Start clean	
0000035A	AE02	0012		00000012	234	SIGP	R0,R2,X'12'	Request z/Arch mode	
0000035E	4780	0370		00000370	235	BE	ZARCH	Success! Begin z/Arch tests	
00000362	8200	0368		00000368	237	LPSW	GOODPSW	No z/Arch? Then we're done!	
00000368	000A0000				239	GOODPSW	DC 0D'0',XL4'000A0000'	S/390 SUCCESS disabled wait PSW	
0000036C	00000000				240	DC	A(0)	S/390 SUCCESS disabled wait PSW	
00000370	B202	07A0		000007A0	242	ZARCH	STIDP CPUID	Save CPU ID (for later test for VM)	
00000374	B2B1	0000		00000000	243		STFL 0	Store Facility List (just for fun)	
00000378	D203	07B4	00C8	000007B4	244		MVC STFLZ,X'C8'	Save STFL results for posterity	
0000037E	4100	001F		0000001F	245		LA R0,(L'FACLIST/8)-1	Store Facility List Extended	
00000382	B2B0	07B8		000007B8	246		STFLE FACLIST	Store Facility List Extended	
					248	*	Fall through to begin z/Architecture mode tests...		
					250	*			
					251	*			
					252	*			
					253	*			
					254	*			
					255	*			
					256	*			
					257	*			
					258	*			
					259	*			
					260	*			
					261	*			
					262	*			
					263	*			
					264	*			
					265	*	V	V	V

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				267 *****
				268 * SSKE, ISKE, RRBE (z/Arch mode)
				269 *****
00000386	BF11 093A		0000093A	271 ICM R1,B'0001',=X'1C'
0000038A	5820 08E8		000008E8	272 L R2,=A((4*_4K)+X'900')
0000038E	B22B 0012			273 SSKE R1,R2
00000392	BF11 093B		0000093B	274 ICM R1,B'0001',=X'26'
00000396	5820 08EC		000008EC	275 L R2,=A((5*_4K)+X'A00')
0000039A	B22B 0012			276 SSKE R1,R2
0000039E	BF11 093C		0000093C	277 ICM R1,B'0001',=X'4E'
000003A2	5820 08F0		000008F0	278 L R2,=A((6*_4K)+X'B00')
000003A6	B22B 0012			279 SSKE R1,R2
				280 *****
000003AA	5820 08E8		000008E8	281 L R2,=A((4*_4K)+X'900')
000003AE	B229 0012			282 ISKE R1,R2
000003B2	BD11 093A		0000093A	283 CLM R1,B'0001',=X'1C'
000003B6	4770 03B7		000003B7	284 BNE *+1
000003BA	5820 08EC		000008EC	285 L R2,=A((5*_4K)+X'A00')
000003BE	B229 0012			286 ISKE R1,R2
000003C2	BD11 093B		0000093B	287 CLM R1,B'0001',=X'26'
000003C6	4770 03C7		000003C7	288 BNE *+1
000003CA	5820 08F0		000008F0	289 L R2,=A((6*_4K)+X'B00')
000003CE	B229 0012			290 ISKE R1,R2
000003D2	BD11 093C		0000093C	291 CLM R1,B'0001',=X'4E'
000003D6	4770 03D7		000003D7	292 BNE *+1
				293 *****
000003DA	5820 08E8		000008E8	294 L R2,=A((4*_4K)+X'900')
000003DE	B22A 0002			295 RRBE R0,R2
000003E2	47D0 03E3		000003E3	296 BC B'1101',*+1 NOT CC2 = was REF 1, CHG 0
000003E6	5820 08EC		000008EC	297 L R2,=A((5*_4K)+X'A00')
000003EA	B22A 0002			298 RRBE R0,R2
000003EE	47E0 03EF		000003EF	299 BC B'1110',*+1 NOT CC3 = was REF 1, CHG 1
000003F2	5820 08F0		000008F0	300 L R2,=A((6*_4K)+X'B00')
000003F6	B22A 0002			301 RRBE R0,R2
000003FA	47E0 03FB		000003FB	302 BC B'1110',*+1 NOT CC3 = was REF 1, CHG 1
				303 *****
000003FE	5820 08E8		000008E8	304 L R2,=A((4*_4K)+X'900')
00000402	B229 0012			305 ISKE R1,R2
00000406	BD11 093D		0000093D	306 CLM R1,B'0001',=X'18'
0000040A	4770 040B		0000040B	307 BNE *+1
0000040E	5820 08EC		000008EC	308 L R2,=A((5*_4K)+X'A00')
00000412	B229 0012			309 ISKE R1,R2
00000416	BD11 093E		0000093E	310 CLM R1,B'0001',=X'22'
0000041A	4770 041B		0000041B	311 BNE *+1
0000041E	5820 08F0		000008F0	312 L R2,=A((6*_4K)+X'B00')
00000422	B229 0012			313 ISKE R1,R2
00000426	BD11 093F		0000093F	314 CLM R1,B'0001',=X'4A'
0000042A	4770 042B		0000042B	315 BNE *+1



LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	
					317	*****
					318	* IVSK, TPROT, TB (z/Arch mode)
					319	*****
0000042E	BF11	0940		00000940	321	ICM R1,B'0001',=X'6E'
00000432	5820	08F4		000008F4	322	L R2,=A((7*_4K)+X'900')
00000436	B22B	0012			323	SSKE R1,R2
0000043A	EB11	08C0	002F	000008C0	324	LCTLG R1,R1,CR1_Z Configure DAT
00000440	8000	0941		00000941	325	SSM =X'04' Enable DAT
00000444	B223	0012			326	IVSK R1,R2
00000448	8000	0942		00000942	327	SSM =X'00' Disable DAT
0000044C	BD11	0943		00000943	328	CLM R1,B'0001',=X'68'
00000450	4770	0451		00000451	329	BNE *+1
					330	*****
00000454	BF11	0944		00000944	331	ICM R1,B'0001',=X'10'
00000458	5820	0904		00000904	332	L R2,=A(6*_4K)
0000045C	B22B	0012			333	SSKE R1,R2
00000460	5810	0908		00000908	334	L R1,=A((6*_4K)+X'900')
00000464	BF21	0944		00000944	335	ICM R2,B'0001',=X'10'
00000468	E501	1000	2000	00000000	336	TPROT 0(R1),0(R2)
0000046E	4770	046F		0000046F	337	BC B'0111',*+1 NOT CC0 = FETCH OK, STORE OK
00000472	BF21	0945		00000945	338	ICM R2,B'0001',=X'20'
00000476	E501	1000	2000	00000000	339	TPROT 0(R1),0(R2)
0000047C	47B0	047D		0000047D	340	BC B'1011',*+1 NOT CC1 = FETCH OK, STORE NO
00000480	BF11	093D		0000093D	341	ICM R1,B'0001',=X'18' (set fetch protect)
00000484	5820	0904		00000904	342	L R2,=A(6*_4K)
00000488	B22B	0012			343	SSKE R1,R2
0000048C	5810	0904		00000904	344	L R1,=A(6*_4K)
00000490	BF21	0945		00000945	345	ICM R2,B'0001',=X'20'
00000494	E501	1000	2000	00000000	346	TPROT 0(R1),0(R2)
0000049A	47D0	049B		0000049B	347	BC B'1101',*+1 NOT CC2 = FETCH NO, STORE NO
					348	* We must skip the 'TB' (Test Block) test if we're running under
					349	* VM or are NOT running under Hercules since neither environment
					350	* has any command to sets block(s) of storage to "unusable".
0000049E	95FF	07A0		000007A0	351	CLI CPUID,X'FF' Are we running under VM?
000004A2	4780	04CA		000004CA	352	BE SKIPTBZ Yes, then skip 'TB' tests
000004A6	95C8	07A0		000007A0	353	CLI CPUID,X'C8' Are we running under Hercules?
000004AA	4770	04CA		000004CA	354	BNE SKIPTBZ No, then skip 'TB' tests
000004AE	1F00				355	SLR R0,R0 Required by TB instruction
000004B0	5820	08FC		000008FC	356	L R2,=A((10*_4K)+X'DEF') Requires Herc 'f- A000' cmd
000004B4	B22C	0012			357	TB R1,R2
000004B8	47B0	04B9		000004B9	358	BC B'1011',*+1 NOT CC1 = Unusable/BAD block
000004BC	1F00				359	SLR R0,R0 Required by TB instruction
000004BE	5820	0900		00000900	360	L R2,=A((11*_4K)+X'FED') Requires Herc 'f- B000' cmd
000004C2	B22C	0012			361	TB R1,R2
000004C6	47B0	04C7		000004C7	362	BC B'1011',*+1 NOT CC1 = Unusable/BAD block
			000004CA	00000001	363	SKIPTBZ EQU *



LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				365	*****
				366	* SSKE with mask (z/Arch mode)
				367	*****
000004CA	BF11 0946		00000946	369	ICM R1,B'0001',=X'33' Low-order X'01' bit s/b ignored
000004CE	5820 090C		0000090C	370	L R2,=A(_1M-(3*_4K)+7) (with some low-order bits set too)
000004D2	4100 0002		00000002	371	LA R0,2 (set CC2...)
000004D6	8900 001C		0000001C	372	SLL R0,32-4 (shift into proper position)
000004DA	0400			373	SPM R0 (set Condition Code 2 in PSW)
000004DC	B22B 1012			374	SSKE R1,R2,SSKE_MB Now do Multi-block SSKE
000004E0	47D0 04E1	000004E1		375	BC B'1101',*+1 FAIL if not still CC2!!
000004E4	5830 0910	00000910		376	L R3,=A(_1M+7) Check if R2 now has expected value
000004E8	1523			377	CLR R2,R3 Does it?
000004EA	4770 04EB	000004EB		378	BNE *+1 FAIL if not
000004EE	5820 090C	0000090C		379	L R2,=A(_1M-(3*_4K)+7)
000004F2	B229 0012			380	ISKE R1,R2
000004F6	BD11 0947	00000947		381	CLM R1,B'0001',=X'32'
000004FA	4770 04FB	000004FB		382	BNE *+1
000004FE	5820 0914	00000914		383	L R2,=A(_1M-(2*_4K)+7)
00000502	B229 0012			384	ISKE R1,R2
00000506	BD11 0947	00000947		385	CLM R1,B'0001',=X'32'
0000050A	4770 050B	0000050B		386	BNE *+1
0000050E	5820 0918	00000918		387	L R2,=A(_1M-(1*_4K)+7)
00000512	B229 0012			388	ISKE R1,R2
00000516	BD11 0947	00000947		389	CLM R1,B'0001',=X'32'
0000051A	4770 051B	0000051B		390	BNE *+1
0000051E	5820 091C	0000091C		391	L R2,=A(_1M-(0*_4K)+7)
00000522	B229 0012			392	ISKE R1,R2
00000526	BD11 0942	00000942		393	CLM R1,B'0001',=X'00'
0000052A	4770 052B	0000052B		394	BNE *+1
				395	*****
0000052E	BF11 0948	00000948		396	ICM R1,B'0001',=X'34'
00000532	5820 0920	00000920		397	L R2,=A(_1M-(1*_4K))
00000536	B22B 0012			398	SSKE R1,R2
0000053A	5820 0920	00000920		399	L R2,=A(_1M-(1*_4K))
0000053E	B229 0012			400	ISKE R1,R2
00000542	BD11 0948	00000948		401	CLM R1,B'0001',=X'34'
00000546	4770 0547	00000547		402	BNE *+1
0000054A	BF11 0949	00000949		403	ICM R1,B'0001',=X'36' Ref + Chg
0000054E	5820 0924	00000924		404	L R2,=A(_1M-(2*_4K))
00000552	4100 0002	00000002		405	LA R0,2
00000556	8900 001C	0000001C		406	SLL R0,32-4
0000055A	0400			407	SPM R0
0000055C	B22B 5012			408	SSKE R1,R2,SSKE_MB+SSKE_MR Multi-block MR=1, MC=0
00000560	47E0 0561	00000561		409	BC B'1110',*+1 NOT CC3 = MB + MR/MC success?
00000564	5830 0928	00000928		410	L R3,=A(_1M)
00000568	1523			411	CLR R2,R3
0000056A	4770 056B	0000056B		412	BNE *+1
0000056E	5820 0924	00000924		413	L R2,=A(_1M-(2*_4K))
00000572	B229 0012			414	ISKE R1,R2
00000576	BD11 0947	00000947		415	CLM R1,B'0001',=X'32' Should still be X'32' due to
0000057A	4770 057B	0000057B		416	BNE *+1 MR ignore = same = no change



LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				422 *****
				423 * PFMF (z/Arch mode)
				424 *****
				426 * UNCONDITIONALLY set all keys (in 2nd 1MB) to X'F4'...
				427 *
				428 * Since they're all currently X'00' (different from the value
				429 * we want) and neither conditional option bit is on (meaning
				430 * the decision whether to update the key or not should be done
				431 * SOLELY on whether or not the key and fetch bits are already
				432 * the value we want them to be or not, i.e. no conditional
				433 * reference or change bit ignoring is requested)
				434 *
0000058E	5810 08C8		000008C8	435 L R1,PFMF_1 KEY=F4 MR=0 MC=0
00000592	5820 092C		0000092C	436 L R2,=A(2*_1M)
00000596	B9AF 0012			437 PFMF R1,R2
0000059A	5820 092C		0000092C	438 L R2,=A(2*_1M)
0000059E	B229 0012			439 ISKE R1,R2
000005A2	BD11 094A		0000094A	440 CLM R1,B'0001',=X'F4' (spot check)
000005A6	4770 05A7		000005A7	441 BNE *+1
000005AA	5820 0930		00000930	442 L R2,=A(2*_1M+(128*_4K))
000005AE	B229 0012			443 ISKE R1,R2
000005B2	BD11 094A		0000094A	444 CLM R1,B'0001',=X'F4' (spot check)
000005B6	4770 05B7		000005B7	445 BNE *+1
000005BA	5820 0934		00000934	446 L R2,=A(2*_1M+_1M-1)
000005BE	B229 0012			447 ISKE R1,R2
000005C2	BD11 094A		0000094A	448 CLM R1,B'0001',=X'F4' (spot check)
000005C6	4770 05C7		000005C7	449 BNE *+1
				450 *
				451 * Now set them all to X'F2' by specifying the option to ignore
				452 * any differences in the reference bit and only update the key
				453 * if the CHANGE bit is different from what we want (since the
				454 * key and fetch bits are already what we want)
				455 *
000005CA	5810 08CC		000008CC	456 L R1,PFMF_2 KEY=F2 MR=1 MC=0
000005CE	5820 092C		0000092C	457 L R2,=A(2*_1M)
000005D2	B9AF 0012			458 PFMF R1,R2
000005D6	5820 092C		0000092C	459 L R2,=A(2*_1M)
000005DA	B229 0012			460 ISKE R1,R2
000005DE	BD11 094B		0000094B	461 CLM R1,B'0001',=X'F2' (spot check)
000005E2	4770 05E3		000005E3	462 BNE *+1
000005E6	5820 0930		00000930	463 L R2,=A(2*_1M+(128*_4K))
000005EA	B229 0012			464 ISKE R1,R2
000005EE	BD11 094B		0000094B	465 CLM R1,B'0001',=X'F2' (spot check)
000005F2	4770 05F3		000005F3	466 BNE *+1
000005F6	5820 0934		00000934	467 L R2,=A(2*_1M+_1M-1)
000005FA	B229 0012			468 ISKE R1,R2
000005FE	BD11 094B		0000094B	469 CLM R1,B'0001',=X'F2' (spot check)
00000602	4770 0603		00000603	470 BNE *+1

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				472 *
				473 *
				474 * Finally, set them all back to X'F4' again by specifying the
				475 * option to ignore any differences in the change bit and only
				476 * update the key if the REFERENCE bit is different from what
				477 * we want.
00000606	5810 08D0		000008D0	478 L R1,PFMF_3 KEY=F4 MR=0 MC=1
0000060A	5820 092C		0000092C	479 L R2,=A(2*_1M)
0000060E	B9AF 0012			480 PFMF R1,R2
00000612	5820 092C		0000092C	481 L R2,=A(2*_1M)
00000616	B229 0012			482 ISKE R1,R2
0000061A	BD11 094A		0000094A	483 CLM R1,B'0001',=X'F4' (spot check)
0000061E	4770 061F		0000061F	484 BNE *+1
00000622	5820 0930		00000930	485 L R2,=A(2*_1M+(128*_4K))
00000626	B229 0012			486 ISKE R1,R2
0000062A	BD11 094A		0000094A	487 CLM R1,B'0001',=X'F4' (spot check)
0000062E	4770 062F		0000062F	488 BNE *+1
00000632	5820 0934		00000934	489 L R2,=A(2*_1M+_1M-1)
00000636	B229 0012			490 ISKE R1,R2
0000063A	BD11 094A		0000094A	491 CLM R1,B'0001',=X'F4' (spot check)
0000063E	4770 063F		0000063F	492 BNE *+1
				494 *****
				495 * IRBM, RRBMM (z/Arch mode)
				496 *****
00000642	9140 07CA		000007CA	498 TM FACLIST+IRBMBYT,IRBMBIT Is facility available?
00000646	4780 0660		00000660	499 BZ SKIPIRBM No, then skip this test
0000064A	B90B 0011			500 SLGR R1,R1
0000064E	5820 092C		0000092C	501 L R2,=A(2*_1M)
00000652	B9AC 0012			502 IRBM R1,R2
00000656	E310 08D8 0020		000008D8	503 CG R1,=XL8'FFFFFFFFFFFFFFFF'
0000065C	4770 065D		0000065D	504 BNE *+1
				505 *****
00000660	9120 07C0		000007C0	506 SKIPIRBM TM FACLIST+RRBMBYT,RRBMBIT Is facility available?
00000664	4780 068C		0000068C	507 BZ SKIPRRBM No, then skip this test
00000668	B90B 0011			508 SLGR R1,R1
0000066C	5820 092C		0000092C	509 L R2,=A(2*_1M)
00000670	B9AE 0012			510 RRBMM R1,R2
00000674	E310 08D8 0020		000008D8	511 CG R1,=XL8'FFFFFFFFFFFFFFFF'
0000067A	4770 067B		0000067B	512 BNE *+1
0000067E	B9AE 0012			513 RRBMM R1,R2
00000682	E310 08E0 0020		000008E0	514 CG R1,=XL8'0000000000000000'
00000688	4770 0689		00000689	515 BNE *+1
		0000068C	00000001	516 SKIPRRBM EQU *



LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	
					534	*****
					535	* ESA/390 PROGRAM CHECK ROUTINE
					536	*****
000006A0	5010	07A8		000007A8	538	PGMCHK ST R1,SAVER1 Save original R1
000006A4	4110	0700		00000700	539	LA R1,OKPGMS R1 --> Expected PGMCHKs table
000006A8	9101	002F		0000002F	541	TM PGMOLD+8-1,X'01' Test failure? (odd branch address?)
000006AC	4780	06C4		000006C4	542	BZ PGMTAB No, something else; check table
000006B0	5810	002C		0000002C	544	L R1,PGMOLD+4 Yes, get program check address
000006B4	4B10	0938		00000938	545	SH R1,=H'5' Backup to failing branch instruction
000006B8	5010	002C		0000002C	546	ST R1,PGMOLD+4 Put back into PGM OLD PSW
000006BC	47F0	06F4		000006F4	547	B PGMFAIL Go load disabled wait PSW
000006C0	4110	100C		0000000C	549	PGMNEXT LA R1,12(,R1) Bump to next entry
000006C4	D50B	1000 094C	00000000	0000094C	550	PGMTAB CLC 0(12,R1),=12X'00' End of table?
000006CA	4780	06F4		000006F4	551	BE PGMFAIL Yes, bonafide program check!
000006CE	D501	1000 008E	00000000	0000008E	552	CLC 0(2,R1),PGMCODE+2 Expected Program Interrupt Code?
000006D4	4770	06C0		000006C0	553	BNE PGMNEXT No, try next entry
000006D8	D503	1004 002C	00000004	0000002C	554	CLC 4(4,R1),PGMOLD+4 Expected Program Interrupt Address?
000006DE	4770	06C0		000006C0	555	BNE PGMNEXT No, try next entry
000006E2	D203	002C 1008	0000002C	00000008	557	MVC PGMOLD+4(4),8(R1) Yes! Move continue address into PSW
000006E8	94FB	0028		00000028	558	NI PGMOLD,X'FF'-X'04' Turn off DAT in case it's on
000006EC	5810	07A8		000007A8	559	L R1,SAVER1 Restore original R1
000006F0	8200	0028		00000028	560	LPSW PGMOLD Ignore the crash and continue
000006F4	9602	0029		00000029	562	PGMFAIL OI PGMOLD+1,X'02' Convert to disabled wait PSW
000006F8	5810	07A8		000007A8	563	L R1,SAVER1 Restore original R1
000006FC	8200	0028		00000028	564	LPSW PGMOLD Load disabled wait crash PSW
					566	*****
					567	* Table of allowable program checks...
					568	*****
00000700					570	OKPGMS DC 0D'0'
00000700	00010001	00000208			571	DC 2AL2(PGM_OPERATION_EXCEPTION),A(STFLPC),A(STFLPC)
0000070C	00060006	000002CA			572	DC 2AL2(PGM_SPECIFICATION_EXCEPTION),A(SSMPC),A(SKIPIVSK)
00000718	00000000	00000000			573	DC 2AL2(0),A(0),A(0) End of table

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT
					575 *****
					576 * z/Architecture PROGRAM CHECK ROUTINE
					577 *****
00000724	E310 07A8	0024		000007A8	579 ZPGMCHK STG R1,SAVER1 Save original R1
0000072A	4110 0790			00000790	580 LA R1,ZOKPGMS R1 --> Expected PGMCHKs table
0000072E	9101 015F			0000015F	582 TM ZPGMOLD+16-1,X'01' Test failure? (odd branch address?)
00000732	4780 074A			0000074A	583 BZ ZPGMTAB No, something else; check table
00000736	5810 015C			0000015C	585 L R1,ZPGMOLD+12 Yes, get program check address
0000073A	4B10 0938			00000938	586 SH R1,=H'5' Backup to failing branch instruction
0000073E	5010 015C			0000015C	587 ST R1,ZPGMOLD+12 Put back into PGM OLD PSW
00000742	47F0 077C			0000077C	588 B ZPGMFAIL Go load disabled wait PSW
00000746	4110 100C			0000000C	590 ZPGMNEXT LA R1,12(,R1) Bump to next entry
0000074A	D50B 1000 094C		00000000	0000094C	591 ZPGMTAB CLC 0(12,R1),=12X'00' End of table?
00000750	4780 077C			0000077C	592 BE ZPGMFAIL Yes, bonafide program check!
00000754	D501 1000 008E		00000000	0000008E	593 CLC 0(2,R1),PGMCODE+2 Expected Program Interrupt Code?
0000075A	4770 0746			00000746	594 BNE ZPGMNEXT No, try next entry
0000075E	D503 1004 015C		00000004	0000015C	595 CLC 4(4,R1),ZPGMOLD+12 Expected Program Interrupt Address?
00000764	4770 0746			00000746	596 BNE ZPGMNEXT No, try next entry
00000768	D203 015C 1008		0000015C	00000008	598 MVC ZPGMOLD+12(4),8(R1) Yes! Move continue address into PSW
0000076E	94FB 0150			00000150	599 NI ZPGMOLD,X'FF'-X'04' Turn off DAT in case it's on
00000772	E310 07A8 0004			000007A8	600 LG R1,SAVER1 Restore original R1
00000778	B2B2 0150			00000150	601 LPSWE ZPGMOLD Ignore the crash and continue
0000077C	9602 0151			00000151	603 ZPGMFAIL OI ZPGMOLD+1,X'02' Convert to disabled wait PSW
00000780	E310 07A8 0004			000007A8	604 LG R1,SAVER1 Restore original R1
00000786	B2B2 0150			00000150	605 LPSWE ZPGMOLD Load disabled wait crash PSW
					607 *****
					608 * Table of allowable program checks...
					609 *****
00000790					611 ZOKPGMS DC 0D'0'
00000790	00000000 00000000				612 DC 2AL2(0),A(0),A(0) End of table



LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				614 *****	
				615 *	
				616 *****	
		00001000	00000001	618 _4K EQU 4096	Constant 4K
		00100000	00000001	619 _1M EQU (1024*1024)	Constant 1M
				620 *	
000007A0	00000000	00000000		621 CPUID DC D'0'	CPU Identification
000007A8	00000000	00000000		622 SAVER1 DC D'0'	Saved original R1 value
000007B0	99999999			623 STFL390 DC XL4'99999999'	Saved S/390 STFL results
000007B4	99999999			624 STFLZ DC XL4'99999999'	Saved z/Arch STFL results
000007B8	99999999	99999999		626 FACLIST DC 0XL256'00',256X'99'	Extended Facilities List...
		00000008	00000001	628 RRBMBYT EQU X'08'	Facility 66 byte (RRBM instruction)
		00000020	00000001	629 RRBMBIT EQU X'20'	Facility 66 bit
		00000012	00000001	630 IRBMBYT EQU X'12'	Facility 145 byte (IRBM instruction)
		00000040	00000001	631 IRBMBIT EQU X'40'	Facility 145 bit
000008B8	00B00000	00001000		633 CR0_1_39 DC 0F'0',XL4'00B00000',A(SEGTAB39)	ESA/390 ASD
000008C0	00000000	00002000		634 CR1_Z DC 0D'0',A(0),A(SEGTABZ)	z/Arch. ASD
000008C8	000210F4			636 PFMF_1 DC A(PFMF_SK+PFMF_1M+X'F4')	
000008CC	000214F2			637 PFMF_2 DC A(PFMF_SK+PFMF_1M+PFMF_MR+X'F2')	
000008D0	000212F4			638 PFMF_3 DC A(PFMF_SK+PFMF_1M+PFMF_MC+X'F4')	
		00020000	00000001	640 PFMF_SK EQU X'00020000'	Set the storage key
		00010000	00000001	641 PFMF_CF EQU X'00010000'	Zero page frame too
		00000000	00000001	643 PFMF_4K EQU X'00000000'	Process just one single 4K page
		00001000	00000001	644 PFMF_1M EQU X'00001000'	Process 1MB frame of 4K pages
		00000400	00000001	646 PFMF_MR EQU X'00000400'	Don't compare Reference bits
		00000200	00000001	647 PFMF_MC EQU X'00000200'	Don't compare Change bits
		00000004	00000001	649 SSKE_MR EQU X'04'	Reference Bit Update Mask
		00000002	00000001	650 SSKE_MC EQU X'02'	Change Bit Update Mask
		00000001	00000001	651 SSKE_MB EQU X'01'	Multiple Blocks Option

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
000008D8				653	LTORG , Literals pool
000008D8	FFFFFFFF FFFFFFFF			654	=XL8'FFFFFFFFFFFFFFFF'
000008E0	00000000 00000000			655	=XL8'0000000000000000'
000008E8	00004900			656	=A( (4*_4K)+X'900' )
000008EC	00005A00			657	=A( (5*_4K)+X'A00' )
000008F0	00006B00			658	=A( (6*_4K)+X'B00' )
000008F4	00007900			659	=A( (7*_4K)+X'900' )
000008F8	00004000			660	=A(4*_4K)
000008FC	0000ADEF			661	=A( (10*_4K)+X'DEF' )
00000900	0000BFED			662	=A( (11*_4K)+X'FED' )
00000904	00006000			663	=A(6*_4K)
00000908	00006900			664	=A( (6*_4K)+X'900' )
0000090C	000FD007			665	=A( _1M-(3*_4K)+7 )
00000910	00100007			666	=A( _1M+7 )
00000914	000FE007			667	=A( _1M-(2*_4K)+7 )
00000918	000FF007			668	=A( _1M-(1*_4K)+7 )
0000091C	00100007			669	=A( _1M-(0*_4K)+7 )
00000920	000FF000			670	=A( _1M-(1*_4K) )
00000924	000FE000			671	=A( _1M-(2*_4K) )
00000928	00100000			672	=A( _1M )
0000092C	00200000			673	=A(2*_1M)
00000930	00280000			674	=A(2*_1M+(128*_4K) )
00000934	002FFFFFF			675	=A(2*_1M+_1M-1)
00000938	0005			676	=H'5'
0000093A	1C			677	=X'1C'
0000093B	26			678	=X'26'
0000093C	4E			679	=X'4E'
0000093D	18			680	=X'18'
0000093E	22			681	=X'22'
0000093F	4A			682	=X'4A'
00000940	6E			683	=X'6E'
00000941	04			684	=X'04'
00000942	00			685	=X'00'
00000943	68			686	=X'68'
00000944	10			687	=X'10'
00000945	20			688	=X'20'
00000946	33			689	=X'33'
00000947	32			690	=X'32'
00000948	34			691	=X'34'
00000949	36			692	=X'36'
0000094A	F4			693	=X'F4'
0000094B	F2			694	=X'F2'
0000094C	00000000 00000000			695	=12X'00'





LOC	OBJECT CODE	ADDR1	ADDR2	STMT
-----	-------------	-------	-------	------

```
753 *****
```

```
754 * Register equates
```

```
755 *****
```

```
00000000 00000001 757 R0      EQU    0
```

00000001	00000001	758 R1	EQU	1
----------	----------	--------	-----	---

00000002	00000001	759 R2	EQU	2
----------	----------	--------	-----	---

00000003	00000001	760 R3	EQU	3
----------	----------	--------	-----	---

00000004	00000001	761 R4	EQU	4
----------	----------	--------	-----	---

```
00000005  00000001  762 R5      EQU    5
```

00000006	00000001	763 R6	EQU	6
----------	----------	--------	-----	---

00000007	00000001	764 R7	EQU	7
----------	----------	--------	-----	---

```
00000008  00000001  765 R8      EQU    8
```

00000009	00000001	766 R9	EQU	9
----------	----------	--------	-----	---

0000000A	00000001	767 R10	EQU	10
----------	----------	---------	-----	----

```
00000000B  00000001    768 R11      EQU    11
```

0000000C	00000001	769 R12	EQU	12
----------	----------	---------	-----	----

0000000D	00000001	770 R13	EQU	13
----------	----------	---------	-----	----

```
0000000E  00000001  771 R14      EQU    14
```

0000000F	00000001	772 R15	EQU	15
----------	----------	---------	-----	----

774 END

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES															
BEGIN	I	000200	4	87	51															
CPUID	D	0007A0	8	621	87	210	212	242	351	353										
CR0_1_39	F	0008B8	4	633	183															
CR1_Z	D	0008C0	8	634	324															
FACLIST	X	0007B8	256	626	245	246	498	506												
GOODPSW	D	000368	8	239	237															
GOODPSWZ	D	000690	8	529	527															
IMAGE	1	000000	12416	0																
IRBMBIT	U	000040	1	631	498															
IRBMBYT	U	000012	1	630	498															
OKPGMS	D	000700	8	570	539															
PAGTAB39	A	001800	4	708	703															
PAGTABZ	A	003000	8	736	731															
PFMF_1	A	0008C8	4	636	435															
PFMF_1M	U	001000	1	644	636	637	638													
PFMF_2	A	0008CC	4	637	456															
PFMF_3	A	0008D0	4	638	478															
PFMF_4K	U	000000	1	643																
PFMF_CF	U	010000	1	641																
PFMF_MC	U	000200	1	647	638															
PFMF_MR	U	000400	1	646	637															
PFMF_SK	U	020000	1	640	636	637	638													
PGMCHK	I	0006A0	4	538	58															
PGMCODE	F	00008C	4	61	552	593														
PGMFAIL	I	0006F4	4	562	547	551														
PGMNEXT	I	0006C0	4	549	553	555														
PGMOLD	U	000028	1	54	541	544	546	554	557	558	560	562	564							
PGMTAB	I	0006C4	6	550	542															
PGM_OPERATION_EXCEPTION																				
	U	000001	1	63	571															
PGM_SPECIFICATION_EXCEPTION																				
	U	000006	1	64	572															
R0	U	000000	1	757	152	155	158	183	214	218	229	234	245	295	298	301	355	359	371	372
R1	U	000001	1	758	373	405	406	407												
					128	130	131	133	134	136	139	140	143	144	147	148	162	163	166	167
					170	171	178	180	183	185	187	190	192	193	195	198	200	202	203	205
					216	220	230	271	273	274	276	277	279	282	283	286	287	290	291	305
					306	309	310	313	314	321	323	324	326	328	331	333	334	336	339	341
					343	344	346	357	361	369	374	380	381	384	385	388	389	392	393	396
					398	400	401	403	408	414	415	418	419	435	437	439	440	443	444	447
					448	456	458	460	461	464	465	468	469	478	480	482	483	486	487	490
					491	500	502	503	508	510	511	513	514	538	539	544	545	546	549	550
					552	554	557	559	563	579	580	585	586	587	590	591	593	595	598	600
R10	U	00000A	1	767	604															
R11	U	00000B	1	768																
R12	U	00000C	1	769																
R13	U	00000D	1	770																
R14	U	00000E	1	771																
R15	U	00000F	1	772																
R2	U	000002	1	759	129	130	132	133	135	136	138	139	142	143	146	147	151	152	154	155
					157	158	161	162	165	166	169	170	179	180	185	191	192	194	195	197

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES															
					198	201	202	204	205	215	216	219	220	231	234	272	273	275	276	278
					279	281	282	285	286	289	290	294	295	297	298	300	301	304	305	308
					309	312	313	322	323	326	332	333	335	336	338	339	342	343	345	346
					356	357	360	361	370	374	377	379	380	383	384	387	388	391	392	397
					398	399	400	404	408	411	413	414	417	418	436	437	438	439	442	443
					446	447	457	458	459	460	463	464	467	468	479	480	481	482	485	486
					489	490	501	502	509	510	513									
R3	U	000003	1	760	232	376	377	410	411											
R4	U	000004	1	761																
R5	U	000005	1	762																
R6	U	000006	1	763																
R7	U	000007	1	764																
R8	U	000008	1	765																
R9	U	000009	1	766																
RRBMBIT	U	000020	1	629	506															
RRBMBYT	U	000008	1	628	506															
SAVER1	D	0007A8	8	622	538	559	563	579	600	604										
SEGTAB39	A	001000	4	703	633															
SEGTABZ	A	002000	8	731	634															
SKIPIRBM	I	000660	4	506	499															
SKIPIVSK	I	0002DA	4	190	572															
SKIPRRBM	U	00068C	1	516	507															
SKIPTB39	U	000350	1	222	211	213														
SKIPTBZ	U	0004CA	1	363	352	354														
SSKE_MB	U	000001	1	651	374	408														
SSKE_MC	U	000002	1	650																
SSKE_MR	U	000004	1	649	408															
SSMPC	I	0002CA	4	185	572															
STFL390	X	0007B0	4	623	96															
STFLPC	I	000208	6	96	571															
STFLZ	X	0007B4	4	624	244															
TEST	J	000000	12416	46	49	53	56	60	66	69	75	85	701	706	729	734	47			
ZARCH	I	000370	4	242	73	235														
ZOKPGMS	D	000790	8	611	580															
ZPGMCHK	I	000724	6	579	79															
ZPGMFAIL	I	00077C	4	603	588	592														
ZPGMNEXT	I	000746	4	590	594	596														
ZPGMOLD	U	000150	1	67	582	585	587	595	598	599	601	603	605							
ZPGMTAB	I	00074A	6	591	583															
_1M	U	100000	1	619	370	376	383	387	391	397	404	410	436	442	446					
_4K	U	001000	1	618	129	132	135	179	191	215	219	332	334	370	383	387	391	397	404	442
					708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723
					736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751
=12X'00'	X	00094C	1	695	550	591														
=A((10*_4K)+X'DEF')	A	0008FC	4	661	215	356														
=A((11*_4K)+X'FED')	A	000900	4	662	219	360														
=A((4*_4K)+X'900')	A	0008E8	4	656	129	138	151	161	193	272	281	294	304							
=A((5*_4K)+X'A00')	A	0008EC	4	657	132	142	154	165	275	285	297	308								



SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES							
=A((6*_4K)+X'900')	A	000908	4	664	334							
=A((6*_4K)+X'B00')	A	0008F0	4	658	135	146	157	169	278	289	300	312
=A((7*_4K)+X'900')	A	0008F4	4	659	179	322						
=A(2*_1M)	A	00092C	4	673	436	438	457	459	479	481	501	509
=A(2*_1M+(128*_4K))	A	000930	4	674	442	463	485					
=A(2*_1M+_1M-1)	A	000934	4	675	446	467	489					
=A(4*_4K)	A	0008F8	4	660	191	201	203					
=A(6*_4K)	A	000904	4	663	332	342	344					
=A(_1M)	A	000928	4	672	410							
=A(_1M+7)	A	000910	4	666	376							
=A(_1M-(0*_4K)+7)	A	00091C	4	669	391							
=A(_1M-(1*_4K))	A	000920	4	670	397	399	417					
=A(_1M-(1*_4K)+7)	A	000918	4	668	387							
=A(_1M-(2*_4K))	A	000924	4	671	404	413						
=A(_1M-(2*_4K)+7)	A	000914	4	667	383							
=A(_1M-(3*_4K)+7)	A	00090C	4	665	370	379						
=H'5'	H	000938	2	676	545	586						
=X'00'	X	000942	1	685	186	327	393					
=X'04'	X	000941	1	684	184	325						
=X'10'	X	000944	1	687	190	194	331	335				
=X'18'	X	00093D	1	680	163	200	306	341				
=X'1C'	X	00093A	1	677	128	140	271	283				
=X'20'	X	000945	1	688	197	204	338	345				
=X'22'	X	00093E	1	681	167	310						
=X'26'	X	00093B	1	678	131	144	274	287				
=X'32'	X	000947	1	690	381	385	389	415				
=X'33'	X	000946	1	689	369							
=X'34'	X	000948	1	691	396	401						
=X'36'	X	000949	1	692	403	419						
=X'4A'	X	00093F	1	682	171	314						
=X'4E'	X	00093C	1	679	134	148	277	291				
=X'68'	X	000943	1	686	187	328						
=X'6E'	X	000940	1	683	178	321						
=X'F2'	X	00094B	1	694	461	465	469					
=X'F4'	X	00094A	1	693	440	444	448	483	487	491		
=XL8'0000000000000000'	X	0008E0	8	655	514							
=XL8'FFFFFFFFFFFFFFFF'	X	0008D8	8	654	503	511						

MACRO    DEFN    REFERENCES

No defined macros

DESC	SYMBOL	SIZE	POS	ADDR
------	--------	------	-----	------

Entry: 0

Image	IMAGE	12416	0000-307F	0000-307F
Region		12416	0000-307F	0000-307F
CSECT	TEST	12416	0000-307F	0000-307F

STMT

FILE NAME

```
1 c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\skey390z\skey390z.asm
```

```
** NO ERRORS FOUND **
```