

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

2				*
3				*Testcase str-001-srst
4				* Test cases for variations on the SRST (Search String) instruction.
5				*
6				*****
7				*
8				* str-001-srst.asm
9				*
10				* Created and placed into public domain 2018-12-27 by Bob Polmanter.
11				* Remove runtest *Compare dependency on 2022-03-08 by Fish
12				*
13				* The SRSTT instruction is tested against the definition in the
14				* z/ArchitecturePrinciples of Operation, SA22-7832.
15				*
16				* Test data is assembled into this program, and some test data is
17				* generated by this program. The program itself verifies the resulting
18				* status of registers and condition codes via simple CLC comparison.
19				*
20				*****
21				*
22				* Tests performed with SRST (Search String):
23				*
24				* 1. R0 bits 32-55 non-zero gives PIC06
25				* 2. Search char found; no operands cross page boundary
26				* 3. Search char not found; no operands cross page boundary
27				* 4. Search char found; operands 1&2 are equal (1 byte search)
28				* 5. Search char not found; operands 1&2 are equal (1 byte search)
29				* 6. Search char found; Operand 1 crosses page boundary
30				* 7. Search char not found; Operand 1 crosses page boundary
31				* 8. Search char found; large multi-page search
32				* 9. Search char not found; large multi-page search
33				*
34				*
35				* NOTE - the nature of the string instructions is such that this test
36				* case will only validate properly for the string instruction
37				* improvement modifications committed in December 2018. The
38				* computation of the CPU determined number of bytes is an
39				* unpredictable number on real hardware (at least above the
40				* minimum value) and the method used in Hercules prior to
41				* instruction improvements calculated it differently than the
42				* improved method. As a result, the operand registers will
43				* likely contain different values when compared by the test
44				* script due to the different CPU number of bytes
45				* determined. None of the methods are wrong, and failing
46				* results in the test script are not necessarily wrong.
47				* But this program and the resulting test script comparisons
48				* were written for the method used by the improved string
49				* instructions (CLST, MVST, SRST).
50				*
51				*
52				*****
53				*

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				54 *
		00000000	0000087F	55 SRST001 START 0
		00000000	00000001	56 STRTLABL EQU *
		00000000	00000001	57 R0 EQU 0
		00000001	00000001	58 R1 EQU 1
		00000002	00000001	59 R2 EQU 2
		00000003	00000001	60 R3 EQU 3
		00000004	00000001	61 R4 EQU 4
		00000005	00000001	62 R5 EQU 5
		00000006	00000001	63 R6 EQU 6
		00000007	00000001	64 R7 EQU 7
		00000008	00000001	65 R8 EQU 8
		00000009	00000001	66 R9 EQU 9
		0000000A	00000001	67 R10 EQU 10
		0000000B	00000001	68 R11 EQU 11
		0000000C	00000001	69 R12 EQU 12
		0000000D	00000001	70 R13 EQU 13
		0000000E	00000001	71 R14 EQU 14
		0000000F	00000001	72 R15 EQU 15
				73 *
				74 *
00000000		00000000		75 USING *,R15
				76 *
				77 * Selected z/Arch low core layout
				78 *
00000000		00000000	0000008C	79 ORG STRTLABL+X'8C' Program check interruption code
0000008C	00000000			80 PGMINTC DS F
				81 *
		00000150	00000001	82 PGMOPSW EQU STRTLABL+X'150' z/Arch Program check old PSW
				83 *
00000090		00000090	000001A0	84 ORG STRTLABL+X'1A0' z/Arch Restart PSW
000001A0	00000001 80000000			85 DC X'0000000180000000',A(0,START)
				86 *
000001B0		000001B0	000001D0	87 ORG STRTLABL+X'1D0' z/Arch Program check new PSW
000001D0	00000001 80000000			88 PGMNPSW DC X'0000000180000000',A(0,PROGCHK)
				89 *
				90 * Program check routine. We are looking for a single specification
				91 * exeception. Any other program check is not expected to occur and
				92 * results in a hard wait.
				93 *
000001E0		000001E0	00000200	94 ORG STRTLABL+X'200'
00000200				95 PROGCHK DS 0H Program check occured...
00000200	9500 F21C	0000021C		96 CLI DIDTHIS,X'00' First/only time here?
00000204	4770 F218	00000218		97 BNE FAIL No?! Then something is wrong!
00000208	9506 F08F	0000008F		98 CLI PGMINTC+3,X'06' Specification Exception?
0000020C	4770 F218	00000218		99 BNE FAIL No?! Then something is wrong!
00000210	92FF F21C	0000021C	100	100 MVI DIDTHIS,X'FF' Remember we did this once already
00000214	47F0 F230	00000230	101	101 B CONTINUE Continue, as this is expected (once!)
00000218	B2B2 F368	00000368	102	102 FAIL LPSWE FAILPSW Unexpected PIC, disabled wait
0000021C	00		103	103 DIDTHIS DC X'00' X'FF' == we already did this

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				105 *****
				106 *
				107 * Main program.
				108 *
0000021E				109 START DS 0H
				110 *
				111 *****
				112 * PREP * Prepare a multi-page frame area
				113 *****
				114 *
0000021E	9825 F700		00000700	115 LM R2,R5,AREA -> large area and length
00000222	0E24			116 MVCL R2,R4 Pad it full of X'AA'
				117 *
				118 *****
				119 * TEST 1 * Ensure any non-zero bits in R0 bits 32-55 gives PIC 06
				120 *****
				121 *
00000224	4100 0400		00000400	122 LA R0,X'400' Set invalid termination char
00000228	9857 F710		00000710	123 LM R5,R7,TEST1 Get string area ptrs
0000022C	B25E 0076			124 SRST R7,R6 Attempt a SRST, should get PIC 6
				125 *
00000230	95FF F21C	00000230	00000001	126 CONTINUE EQU *
			0000021C	127 CLI DIDTHIS,X'FF' Did PIC 06 happen?
00000234	4770 F218		00000218	128 BNE FAIL No?! Then something is wrong!
00000238	D207 F1D0 F368	000001D0	00000368	129 MVC PGMNPSW,FAILPSW All other p checks should halt
				130 *
				131 *****
				132 * TEST 2 * Search char found; no operands cross page boundary
				133 *****
				134 *
0000023E	9857 F71C		0000071C	135 LM R5,R7,TEST2 Get string area ptrs
00000242	925B 5000		00000000	136 MVI 0(R5),C'\$' Set search char
00000246	4D90 F332		00000332	137 BAS R9,SEARCH search the string
0000024A	9068 F800		00000800	138 STM R6,R8,RESULT2 Save test result regs
0000024E	92AA 5000		00000000	139 MVI 0(R5),X'AA' Reset the search char
				140 *
				141 *****
				142 * TEST 3 * Search char not found; no operands cross page boundary
				143 *****
				144 *
00000252	9857 F728		00000728	145 LM R5,R7,TEST3 Get string area ptrs
00000256	925B 5000		00000000	146 MVI 0(R5),C'\$' Set search char
0000025A	4D90 F332		00000332	147 BAS R9,SEARCH search the string
0000025E	9068 F810		00000810	148 STM R6,R8,RESULT3 Save test result regs
00000262	92AA 5000		00000000	149 MVI 0(R5),X'AA' Reset the search char
				150 *
				151 *****
				152 * TEST 4 * Search char found; operands contain equal addresses
				153 *****
				154 *
00000266	9857 F734		00000734	155 LM R5,R7,TEST4 Get string area ptrs
0000026A	925B 5000		00000000	156 MVI 0(R5),C'\$' Set search char

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
0000026E	4D90 F332		00000332	157 BAS R9,SEARCH search the string
00000272	9068 F820		00000820	158 STM R6,R8,RESULT4 Save test result regs
00000276	92AA 5000		00000000	159 MVI 0(R5),X'AA' Reset the search char
				160 *
				161 *****
				162 * TEST 5 * Search char not found; operands contain equal addresses
				163 *****
				164 *
0000027A	9857 F740		00000740	165 LM R5,R7,TEST5 Get string area ptrs
0000027E	925B 5000		00000000	166 MVI 0(R5),C'\$' Set search char
00000282	4D90 F332		00000332	167 BAS R9,SEARCH search the string
00000286	9068 F830		00000830	168 STM R6,R8,RESULT5 Save test result regs
0000028A	92AA 5000		00000000	169 MVI 0(R5),X'AA' Reset the search char
				170 *
				171 *****
				172 * TEST 6 * Search char found; Operand 1 crosses page boundary
				173 *****
				174 *
0000028E	9857 F74C		0000074C	175 LM R5,R7,TEST6 Get string area ptrs
00000292	925B 5000		00000000	176 MVI 0(R5),C'\$' Set search char
00000296	4D90 F332		00000332	177 BAS R9,SEARCH search the string
0000029A	9068 F840		00000840	178 STM R6,R8,RESULT6 Save test result regs
0000029E	92AA 5000		00000000	179 MVI 0(R5),X'AA' Reset the search char
				180 *
				181 *****
				182 * TEST 7 * Search char not found; Operand 1 crosses page boundary
				183 *****
				184 *
000002A2	9857 F758		00000758	185 LM R5,R7,TEST7 Get string area ptrs
000002A6	925B 5000		00000000	186 MVI 0(R5),C'\$' Set search char
000002AA	4D90 F332		00000332	187 BAS R9,SEARCH search the string
000002AE	9068 F850		00000850	188 STM R6,R8,RESULT7 Save test result regs
000002B2	92AA 5000		00000000	189 MVI 0(R5),X'AA' Reset the search char
				190 *
				191 *****
				192 * TEST 8 * Search char found; large multi-page search
				193 *****
				194 *
000002B6	9857 F764		00000764	195 LM R5,R7,TEST8 Get string area ptrs
000002BA	925B 5000		00000000	196 MVI 0(R5),C'\$' Set search char
000002BE	4D90 F332		00000332	197 BAS R9,SEARCH search the string
000002C2	9068 F860		00000860	198 STM R6,R8,RESULT8 Save test result regs
000002C6	92AA 5000		00000000	199 MVI 0(R5),X'AA' Reset the search char
				200 *
				201 *****
				202 * TEST 9 * Search char not found; large multi-page search
				203 *****
				204 *
000002CA	9857 F770		00000770	205 LM R5,R7,TEST9 Get string area ptrs
000002CE	925B 5000		00000000	206 MVI 0(R5),C'\$' Set search char
000002D2	4D90 F332		00000332	207 BAS R9,SEARCH search the string
000002D6	9068 F870		00000870	208 STM R6,R8,RESULT9 Save test result regs

LOC	OBJECT CODE	ADDR1	ADDR2	STMT			
000002DA	92AA 5000		00000000	209	MVI	0(R5),X'AA'	Reset the search char
				210	*		
				211	**	Verify results...	
				212	*		
000002DE	D50B F388 F800	00000388	00000800	213	CLC	GRESLT2,RESULT2	Expected results?
000002E4	4770 F218		00000218	214	BNE	FAIL	No?! Then something is wrong!
000002E8	D50B F394 F810	00000394	00000810	215	CLC	GRESLT3,RESULT3	Expected results?
000002EE	4770 F218		00000218	216	BNE	FAIL	No?! Then something is wrong!
000002F2	D50B F3A0 F820	000003A0	00000820	217	CLC	GRESLT4,RESULT4	Expected results?
000002F8	4770 F218		00000218	218	BNE	FAIL	No?! Then something is wrong!
000002FC	D50B F3AC F830	000003AC	00000830	219	CLC	GRESLT5,RESULT5	Expected results?
00000302	4770 F218		00000218	220	BNE	FAIL	No?! Then something is wrong!
00000306	D50B F3B8 F840	000003B8	00000840	221	CLC	GRESLT6,RESULT6	Expected results?
0000030C	4770 F218		00000218	222	BNE	FAIL	No?! Then something is wrong!
00000310	D50B F3C4 F850	000003C4	00000850	223	CLC	GRESLT7,RESULT7	Expected results?
00000316	4770 F218		00000218	224	BNE	FAIL	No?! Then something is wrong!
0000031A	D50B F3D0 F860	000003D0	00000860	225	CLC	GRESLT8,RESULT8	Expected results?
00000320	4770 F218		00000218	226	BNE	FAIL	No?! Then something is wrong!
00000324	D50B F3DC F870	000003DC	00000870	227	CLC	GRESLT9,RESULT9	Expected results?
0000032A	4770 F218		00000218	228	BNE	FAIL	No?! Then something is wrong!
				229	*		
0000032E	B2B2 F358		00000358	230	LPSWE	GOODPSW	E0J, load disabled wait PSW
				231	*		
				232	*-- SRST routine used by tests		
				233	*		
00000332	4100 005B	00000332	00000001	234	SEARCH	EQU *	
00000336	1B88		0000005B	235	LA	R0,C'\$'	Load search character
				236	SR	R8,R8	Init SRST counter
				237	*		
		00000338	00000001	238	INVOKE	EQU *	
00000338	B25E 0076			239	SRST	R7,R6	Look for search char
0000033C	4180 8001		00000001	240	LA	R8,1(,R8)	Count executions of SRST
00000340	4780 F34E		0000034E	241	BC	8,BADCC	CC=0 SHOULD NEVER HAPPEN
00000344	4710 F338		00000338	242	BC	1,INVOKE	Restart the search
00000348	B222 0080			243	IPM	R8	Put final CC in high R8
0000034C	07F9			244	BR	R9	Return
				245	*		
0000034E	B2B2 F378		00000378	246	BADCC	LPSWE BADCCPSW	Stop on invalid CC
				247	*		
00000358				248	DS	0D	Ensure correct alignment for psw
00000358	00020000 00000000			249	GOODPSW	DC X'0002000000000000',A(0,0)	Normal end - disabled wait
00000368	00020000 00000000			250	FAILPSW	DC X'0002000000000000',XL4'00',X'0000DEAD'	Abnormal end
00000378	00020000 00000000			251	BADCCPSW	DC X'0002000000000000',XL4'00',X'000BADCC'	Abnormal end
				252	*		
				253	*		
00000388	00002500 00002532			254	GRESLT2	DC XL12'0000250000000253210000001'	
00000394	00002500 00002580			255	GRESLT3	DC XL12'0000250000000258020000001'	
000003A0	00002500 00002500			256	GRESLT4	DC XL12'0000250000000250020000001'	
000003AC	00002500 00002500			257	GRESLT5	DC XL12'0000250000000250020000001'	
000003B8	00003080 00003100			258	GRESLT6	DC XL12'0000308000000310010000002'	
000003C4	00003080 00003300			259	GRESLT7	DC XL12'0000308000000330020000002'	
000003D0	0000B000 0000BF80			260	GRESLT8	DC XL12'0000B00000000BF801000000A'	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT			
000003DC	0000B000 0000BFFF			261 GRESLT9	DC	XL12'0000B0000000BFFF2000000A'	
				262 *			
				263 *			
				264 *			
000003E8		000003E8	00000700	265	ORG	STRTLABL+X'700'	
				266 *			
00000700	00002000			267 AREA	DC	X'00002000'	-> start of multi-page area
00000704	00010000			268 AREALEN	DC	A(4096*16)	Size of multi=page area
00000708	00000000			269 ZERO	DC	A(0)	
0000070C	AA000000			270 PAD	DC	X'AA000000'	MVCL pad char
				271 *			
00000710				272 TEST1	DS	0F	Test should fail PIC06
00000710	00002532			273	DC	X'00002532'	-> where to place search char
00000714	00002500			274	DC	X'00002500'	-> where to start search
00000718	00002580			275	DC	X'00002580'	-> end of search area
				276 *			
0000071C				277 TEST2	DS	0F	Op 1 doesnt cross page
0000071C	00002532			278	DC	X'00002532'	-> where to place search char
00000720	00002500			279	DC	X'00002500'	-> where to start search
00000724	00002580			280	DC	X'00002580'	-> end of search area
				281 *			
00000728				282 TEST3	DS	0F	Op 1 doesnt cross page
00000728	00001000			283	DC	X'00001000'	-> search char outside of area
0000072C	00002500			284	DC	X'00002500'	-> where to start search
00000730	00002580			285	DC	X'00002580'	-> end of search area
				286 *			
00000734				287 TEST4	DS	0F	Op 1&2 are equal, search ok
00000734	00002500			288	DC	X'00002500'	-> where to place search char
00000738	00002500			289	DC	X'00002500'	-> where to start search
0000073C	00002500			290	DC	X'00002500'	-> end of search area
				291 *			
00000740				292 TEST5	DS	0F	Op 1&2 are equal, search fails
00000740	00001000			293	DC	X'00001000'	-> search char outside of area
00000744	00002500			294	DC	X'00002500'	-> where to start search
00000748	00002500			295	DC	X'00002500'	-> end of search area
				296 *			
0000074C				297 TEST6	DS	0F	Op 1 crosses page; search ok
0000074C	00003100			298	DC	X'00003100'	-> where to place search char
00000750	00002F80			299	DC	X'00002F80'	-> where to start search
00000754	00003300			300	DC	X'00003300'	-> end of search area
				301 *			
00000758				302 TEST7	DS	0F	Op 1 crosses page, search fails
00000758	00001000			303	DC	X'00001000'	-> search char outside of area
0000075C	00002F80			304	DC	X'00002F80'	-> where to start search
00000760	00003300			305	DC	X'00003300'	-> end of search area
				306 *			
00000764				307 TEST8	DS	0F	large multi-page; search ok
00000764	0000BF80			308	DC	X'0000BF80'	-> where to place search char
00000768	00002100			309	DC	X'00002100'	-> where to start search
0000076C	0000BFFF			310	DC	X'0000BFFF'	-> end of search area
				311 *			
00000770				312 TEST9	DS	0F	large multi-page, search fails

LOC	OBJECT CODE	ADDR1	ADDR2	STMT						
00000770	00001000			313	DC	X'00001000'			-> search char outside of area	
00000774	00002100			314	DC	X'00002100'			-> where to start search	
00000778	0000BFFF			315	DC	X'0000BFFF'			-> end of search area	
				316	*					
				317	*					
				318	*	Locations for results				
				319	*					
				320	*	Result fields are kept on 16-byte boundaries to more easily				
				321	*	track their assembled offsets for use in the .tst script.				
				322	*					
				323	*		offset			
0000077C		0000077C	00000800	324	ORG	STRTLABL+X'800'	8xx			
00000800	00000000	00000000		325	RESULT2	DS	4F	00	Register results test 2	
00000810	00000000	00000000		326	RESULT3	DS	4F	10	Register results test 3	
00000820	00000000	00000000		327	RESULT4	DS	4F	20	Register results test 4	
00000830	00000000	00000000		328	RESULT5	DS	4F	30	Register results test 5	
00000840	00000000	00000000		329	RESULT6	DS	4F	40	Register results test 6	
00000850	00000000	00000000		330	RESULT7	DS	4F	50	Register results test 7	
00000860	00000000	00000000		331	RESULT8	DS	4F	60	Register results test 8	
00000870	00000000	00000000		332	RESULT9	DS	4F	70	Register results test 9	
				333	*					
				334	END					

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
AREA	X	000700	4	267	115
AREALEN	A	000704	4	268	
BADCC	I	00034E	4	246	241
BADCCPSW	X	000378	8	251	246
CONTINUE	U	000230	1	126	101
DIDTHIS	X	00021C	1	103	96 100 127
FAIL	I	000218	4	102	97 99 128 214 216 218 220 222 224 226 228
FAILPSW	X	000368	8	250	102 129
GOODPSW	X	000358	8	249	230
GRESLT2	X	000388	12	254	213
GRESLT3	X	000394	12	255	215
GRESLT4	X	0003A0	12	256	217
GRESLT5	X	0003AC	12	257	219
GRESLT6	X	0003B8	12	258	221
GRESLT7	X	0003C4	12	259	223
GRESLT8	X	0003D0	12	260	225
GRESLT9	X	0003DC	12	261	227
IMAGE	1	000000	2176	0	
INVOKE	U	000338	1	238	242
PAD	X	00070C	4	270	
PGMINTC	F	00008C	4	80	98
PGMNPSW	X	0001D0	8	88	129
PGMOPSW	U	000150	1	82	
PROGCHK	H	000200	2	95	88
R0	U	000000	1	57	122 235
R1	U	000001	1	58	
R10	U	00000A	1	67	
R11	U	00000B	1	68	
R12	U	00000C	1	69	
R13	U	00000D	1	70	
R14	U	00000E	1	71	
R15	U	00000F	1	72	75
R2	U	000002	1	59	115 116
R3	U	000003	1	60	
R4	U	000004	1	61	116
R5	U	000005	1	62	115 123 135 136 139 145 146 149 155 156 159 165 166 169 175 176 179
R6	U	000006	1	63	124 138 148 158 168 178 188 198 208 239
R7	U	000007	1	64	123 124 135 145 155 165 175 185 195 205 239
R8	U	000008	1	65	138 148 158 168 178 188 198 208 236 240 243
R9	U	000009	1	66	137 147 157 167 177 187 197 207 244
RESULT2	F	000800	4	325	138 213
RESULT3	F	000810	4	326	148 215
RESULT4	F	000820	4	327	158 217
RESULT5	F	000830	4	328	168 219
RESULT6	F	000840	4	329	178 221
RESULT7	F	000850	4	330	188 223
RESULT8	F	000860	4	331	198 225
RESULT9	F	000870	4	332	208 227
SEARCH	U	000332	1	234	137 147 157 167 177 187 197 207
SRST001	J	000000	2176	55	
START	H	00021E	2	109	85

MACRO DEFN REFERENCES

No defined macros

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

Entry: 0

Image	IMAGE	2176	000-87F	000-87F
Region		2176	000-87F	000-87F
CSECT	SRST001	2176	000-87F	000-87F

STMT

FILE NAME

```
1 c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\str-001-srst\str-001-srst.asm
```

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