

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
2				2 *****
3	*			3 *
4	*			4 * CUSE Performance instruction tests
5	*			5 *
6				6 *****
7	*			7 *
8	*			8 * This program ONLY tests the performance of the CUSE instructions.
9	*			9 *
10	*			10 *
11	*			11 * *****
12	*			12 * ** IMPORTANT! **
13	*			13 * *****
14	*			14 *
15	*			15 * This test uses the Hercules Diagnose X'008' interface
16	*			16 * to display messages and thus your .tst runtest script
17	*			17 * MUST contain a "DIAG8CMD ENABLE" statement within it!
18	*			18 *
19	*			19 *
20	*			20 * NOTE: This test is based on the CLCL-et-al Test but modified to
21	*			21 * only test the CUSE instruction. -- James Wekel November 2022
22	*			22 *
23				23 *****
24	*			24 *
25	*			25 * Example Hercules Testcase:
26	*			26 *
27	*			27 *
28	*			28 * *Testcase CUSE-02-performance (Test CUSE instructions)
29	*			29 *
30	*	mainsize	16	30 * mainsize 16
31	*	numcpu	1	31 * numcpu 1
32	*	sysclear		32 * sysclear
33	*	archlvl	z/Arch	33 * archlvl z/Arch
34	*	loadcore	"\$(testpath)/CUSE-02-performance.core" 0x0	34 * loadcore "\$(testpath)/CUSE-02-performance.core" 0x0
35	*	diag8cmd	enable # (needed for messages to Hercules console)	35 * diag8cmd enable # (needed for messages to Hercules console)
36	*	#r	408=ff # (enable timing tests)	36 * #r 408=ff # (enable timing tests)
37	*	runttest	500 # (test duration, depends on host)	37 * runtest 500 # (test duration, depends on host)
38	*	diag8cmd	disable # (reset back to default)	38 * diag8cmd disable # (reset back to default)
39	*	*Done		39 * *Done
40	*			40 *
41	*			41 *
42				42 *****

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				44 **** 45 *	
				46 * Tests: 47 * Both operand-1 and operand-2 cross a page boundary. 48 *	
				49 * 1. CUSE of 512 bytes - substring length 1 50 * 2. CUSE of 512 bytes - substring length 4 51 * 3. CUSE of 512 bytes - substring length 8 52 * 4. CUSE of 512 bytes - substring length 32 53 * 5. CUSE of 512 bytes - substring length 32 (different strings) 54 * 6. CUSE of 4160 (4096+64) bytes - substring length 32 55 * which results in a CC=3, and a branch back 56 * to complete the CUSE instruction. 57 *	
				58 *****	
00000000		00000000 0000E9AF 00000000		60 CUSE2TST START 0 61 USING CUSE2TST,R0	Low core addressability
00000000 000001A0 000001A8	00000001 80000000 00000000 00000200	00000000 000001A0		63 ORG CUSE2TST+X'1A0' 64 DC X'0000000180000000' 65 DC AD(BEGIN)	z/Architecture RESTART PSW
000001B0 000001D0 000001D8	00020001 80000000 00000000 0000DEAD	000001B0 000001D0		67 ORG CUSE2TST+X'1D0' 68 DC X'0002000180000000' 69 DC AD(X'DEAD')	z/Architecture PROGRAM CHECK PSW
000001E0		000001E0 00000200	71	ORG CUSE2TST+X'200'	Start of actual test program...

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				73 **** 74 * The actual "CUSE2TST" program itself... 75 **** 76 * 77 * Architecture Mode: z/Arch 78 * Register Usage: 79 * 80 * R0 (work) 81 * R1 (work) 82 * R2 (work) or MSG subroutine call 83 * R3 (work) 84 * R4 (work) 85 * R6 CUSETEST Base (of current test) 86 * R7 (work) 87 * R8 First base register 88 * R9 Second base register 89 * R10-R13 (work) 90 * R14 Subroutine call 91 * R15 Secondary Subroutine call or work 92 * 93 ****	
00000200		00000200		95 USING BEGIN,R8	FIRST Base Register
00000200		00001200		96 USING BEGIN+4096,R9	SECOND Base Register
00000200	0580			98 BEGIN BALR R8,0	Initialize FIRST base register
00000202	0680			99 BCTR R8,0	Initialize FIRST base register
00000204	0680			100 BCTR R8,0	Initialize FIRST base register
00000206	4190 8800		00000800	102 LA R9,2048(,R8)	Initialize SECOND base register
0000020A	4190 9800		00000800	103 LA R9,2048(,R9)	Initialize SECOND base register
				105 **** 106 * Run the performance test(s)... 107 ****	
0000020E	45E0 8328		00000528	109 BAL R14,TEST91	Time CUSE instruction (speed test)
				111 **** 112 * Test for normal or unexpected test completion... 113 ****	
00000212	95FF 8208		00000408	115 CLI TIMEOPT,X'FF'	Was this a timing run?
00000216	4770 8D80		00000F80	116 BNE EOJ	No, timing run; just go end normally
0000021A	95F4 8200		00000400	118 CLI TESTNUM,X'F4'	Did we end on expected test?
0000021E	4770 8D98		00000F98	119 BNE FAILTEST	No?! Then FAIL the test!
00000222	9599 8201		00000401	121 CLI SUBTEST,X'99'	Did we end on expected SUB-test?
00000226	4770 8D98		00000F98	122 BNE FAILTEST	No?! Then FAIL the test!
0000022A	47F0 8D80		00000F80	124 B EOJ	Yes, then normal completion!

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				126 ****	*****
				127 * Fixed test storage locations ...	
				128 *****	*****
0000022E		0000022E	00000400	130 ORG CUSE2TST+X'400'	
00000400				132 TESTADDR DS 0D	Where test/subtest numbers will go
00000400 99				133 TESTNUM DC X'99'	Test number of active test
00000401 99				134 SUBTEST DC X'99'	Active test sub-test number
00000408				136 TIMEOPT DS 0D	
00000408 00				137 TIMEOPT DC X'00'	Set to non-zero to run timing tests
00000410				139 DS 0D	
00000410 00000000 00000000				140 SAVE2T5 DC 4F'0'	
00000420 00000000				141 SAVER2 DC F'0'	
00000424 00000000				142 SAVER6 DC F'0'	
00000428		00000428	00000528	144 ORG *+X'100'	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT		
				146 **** 147 * TEST91 148 ****	Time CUSE instruction (speed test)	*****
00000528	91FF 8208		00000408	150 TEST91	TM TIMEOPT,X'FF'	Is timing tests option enabled?
0000052C	078E			151 BZR	R14	No, skip timing tests
0000052E	4160 8E40		00001040	153 LA	R6,CUSEPERF	Point R5 --> testing control table
00000532		00000000		154 USING	CUSETEST,R6	What each table entry looks like
			00000532	00000001	156 TST91LOP EQU *	
00000532	5060 8224		00000424	157 ST	R6,SAVER6	Save current pref table base
00000536	4370 6000		00000000	159 IC	R7,TNUM	Set test number
0000053A	4270 8200		00000400	160 STC	R7,TESTNUM	
				161 *		
				162 **	Initialize operand data (move data to testing address)	
				163 *		
				164 *	Build Operand-1	
0000053E	5820 6018		00000018	166 L	R2,OP1WHERE	Where to move operand-1 data to
00000542	5830 601C		0000001C	167 L	R3,OP1LEN	Get operand-1 length
00000546	58A0 6008		00000008	168 L	R10,SS1ADDR	Calculate OP 1 starting address
0000054A	1BA3			169 SR	R10,R3	
0000054C	5AA0 600C		0000000C	170 A	R10,SS1LEN	
00000550	58B0 601C		0000001C	171 L	R11,OP1LEN	
00000554	0E2A			172 MVCL	R2,R10	
00000556	0620			174 BCTR	R2,0	less one for last char addr
00000558	D200 2000 6006	00000000	00000006	175 MVC	0(0,R2),SS1LAST	set last char
				177 *	Build Operand-2	
0000055E	5840 6020		00000020	179 L	R4,OP2WHERE	Where to move operand-1 data to
00000562	5850 6024		00000024	180 L	R5,OP2LEN	Get operand-1 length
00000566	58A0 6010		00000010	181 L	R10,SS2ADDR	Calculate OP 2 starting address
0000056A	1BA5			182 SR	R10,R5	
0000056C	5AA0 6014		00000014	183 A	R10,SS2LEN	
00000570	58B0 6024		00000024	184 L	R11,OP2LEN	
00000574	0E4A			185 MVCL	R4,R10	
00000576	0640			187 BCTR	R4,0	less one for last char addr
00000578	D200 4000 6007	00000000	00000007	188 MVC	0(0,R4),SS2LAST	set last char
				190 *	Set Substring length and pad byte	
0000057E	4300 6004		00000004	192 IC	R0,SSLLEN	Set SS length
00000582	4310 6005		00000005	193 IC	R1,PAD	Set SS Pad byte

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				195 **** 196 * Define come helpful macros to ensure our counts are correct 197 ****
				199 MACRO 200 OVERONLY &NUM &NUM = number of sets 201 LCLA &CTR 202 &CTR SETA &NUM 203 .LOOP ANOP 204 .* 205 * 206 LM R2,R5,OPSWHERE 207 BC B'0001',*+4 208 .* 209 &CTR SETA &CTR-1 210 AIF (&CTR GT 0).LOOP 211 MEND
				213 MACRO 214 DOINSTR &NUM &NUM = number of sets 215 LCLA &CTR 216 &CTR SETA &NUM 217 .LOOP ANOP 218 .* 219 * 220 LM R2,R5,OPSWHERE 221 CUSE R2,R4 222 BC B'0001',*-4 223 .* 224 &CTR SETA &CTR-1 225 AIF (&CTR GT 0).LOOP 226 MEND

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				228 **** 229 * Next, time the overhead... 230 ****
00000586	5870 8DB4	00000FB4	232	L R7 ,NUMLOOPS
0000058A	B205 8DB8	00000FB8	233	STCK BEGCLOCK
0000058E	9025 8210	00000410	234	STM R2 ,R5 ,SAVE2T5
00000592	05A0		235	BALR R10 ,0
			236 *	100 sets of overhead (first 2)
			237	OVERONLY 2
			238+*	
00000594	9825 6018	00000018	239+	LM R2 ,R5 ,OPSWHERE
00000598	4710 839C	0000059C	240+	BC B'0001' ,*+4
241+*				
0000059C	9825 6018	00000018	242+	LM R2 ,R5 ,OPSWHERE
000005A0	4710 83A4	000005A4	243+	BC B'0001' ,*+4
			245 *ETC.....
			247	PRINT OFF
			537	PRINT ON
			539	OVERONLY 2
			540+*	(last 2)
000008A4	9825 6018	00000018	541+	LM R2 ,R5 ,OPSWHERE
000008A8	4710 86AC	000008AC	542+	BC B'0001' ,*+4
543+*				
000008AC	9825 6018	00000018	544+	LM R2 ,R5 ,OPSWHERE
000008B0	4710 86B4	000008B4	545+	BC B'0001' ,*+4
000008B4	067A		547	BCTR R7 ,R10
000008B6	B205 8DC0	00000FC0	548	STCK ENDCLOCK
000008BA	45F0 8C30	00000E30	549	BAL R15 ,CALCDUR
000008BE	D207 8DD0 8DC8	00000FD0	550	MVC OVERHEAD ,DURATION

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				552 ****	*****
				553 * Now do the actual timing run...	
				554 ****	*****
000008C4	5870 8DB4		00000FB4	556 L R7,NUMLOOPS	
000008C8	B205 8DB8		00000FB8	557 STCK BEGCLOCK	
000008CC	05A0			558 BALR R10,0	
				559 *	100 sets of instructions
				560 DOINSTR 2	(first 2)
				561+*	
000008CE	9825 6018		00000018	562+ LM R2,R5,OPSWHERE	
000008D2	B257 0024			563+ CUSE R2,R4	
000008D6	4710 86D2		000008D2	564+ BC B'0001',*-4	
				565+*	
000008DA	9825 6018		00000018	566+ LM R2,R5,OPSWHERE	
000008DE	B257 0024			567+ CUSE R2,R4	
000008E2	4710 86DE		000008DE	568+ BC B'0001',*-4	
				570 *ETC.....
				572 PRINT OFF	
				958 PRINT ON	
				960 DOINSTR 2	(last 2)
				961+*	
00000D66	9825 6018		00000018	962+ LM R2,R5,OPSWHERE	
00000D6A	B257 0024			963+ CUSE R2,R4	
00000D6E	4710 8B6A		00000D6A	964+ BC B'0001',*-4	
				965+*	
00000D72	9825 6018		00000018	966+ LM R2,R5,OPSWHERE	
00000D76	B257 0024			967+ CUSE R2,R4	
00000D7A	4710 8B76		00000D76	968+ BC B'0001',*-4	
00000D7E	067A			970 BCTR R7,R10	
00000D80	B205 8DC0		00000FC0	971 STCK ENDCLOCK	
00000D84	9825 8210		00000410	973 LM R2,R5,SAVE2T5	
00000D88	D204 8E11 8DA8	00001011	00000FA8	974 MVC PRTLINE+33(5),=CL5'CUSE'	
00000D8E	45F0 8BA6		00000DA6	975 BAL R15,RPTSPEED	
				976 *	
				977 ** More performance tests?	
				978 *	
00000D92	5860 8224		00000424	979 L R6,SAVER6	Restore perf table base
00000D96	4160 603C		0000003C	980 LA R6,CUSENEXT	Go on to next table entry
00000D9A	D503 8D9C 6000	00000F9C	00000000	981 CLC =F'0',0(R6)	End of table?
00000DA0	4770 8332		00000532	982 BNE TST91L0P	No, loop...
00000DA4	07FE			983 BR R14	Return to caller or FAILTEST

LOC	OBJECT CODE	ADDR1	ADDR2	STMT			
				985 ****			
				986 * RPTSPEED		Report instruction speed	
				987 ****			
00000DA6	50F0 8C10	00000E10	989	RPTSPEED ST	R15, RPTSAVE	Save return address	
00000DAA	9057 8C14	00000E14	990	STM	R5, R7, RPTSVR5T7	Save R5-7	
00000DAE	45F0 8C30	00000E30	992	BAL	R15, CALCDUR	Calculate duration	
00000DB2	4150 8DD0	00000FD0	994	LA	R5, OVERHEAD	Subtract overhead	
00000DB6	4160 8DC8	00000FC8	995	LA	R6, DURATION	From raw timing	
00000DBA	4170 8DC8	00000FC8	996	LA	R7, DURATION	Yielding true instruction timing	
00000DBE	45F0 8C84	00000E84	997	BAL	R15, SUBDWORD	Do it	
00000DC2	98AB 8DC8	00000FC8	999	LM	R10, R11, DURATION	Convert to...	
00000DC6	8CA0 000C	0000000C	1000	SRDL	R10, 12	... microseconds	
00000DCA	4EA0 8DD8	00000FD8	1002	CVD	R10, TICKSAAA	Convert HIGH part to decimal	
00000DCE	4EB0 8DE0	00000FE0	1003	CVD	R11, TICKSBBB	Convert LOW part to decimal	
00000DD2	F877 8DE8 8DD8	00000FE8	00000FD8	1005	ZAP	TICKSTOT, TICKSAAA	Calculate...
00000DD8	FC75 8DE8 8DAD	00000FE8	00000FAD	1006	MP	TICKSTOT, =P'4294967296'	... decimal...
00000DDE	FA77 8DE8 8DE0	00000FE8	00000FE0	1007	AP	TICKSTOT, TICKSBBB	... microseconds
00000DE4	D20B 8E1B 8E34	0000101B	00001034	1009	MVC	PRTLINE+43(L'EDIT), EDIT	(edit into...
00000DEA	DE0B 8E1B 8DEB	0000101B	00000FEB	1010	ED	PRTLINE+43(L'EDIT), TICKSTOT+3	... print line)
				1012 *			
				1013 *		Use Hercules Diagnose for Message to console	
				1014 *			
00000DF0	9002 8C20	00000E20	1015	STM	R0, R2, RPTDWSAV	Save regs used by MSG	
00000DF4	4100 0044	00000044	1016	LA	R0, PRTLNG	Message length	
00000DF8	4110 8DF0	00000FF0	1017	LA	R1, PRTLNE	Message address	
00000DFC	4520 8CB8	00000EB8	1018	BAL	R2, MSG	Call Hercules console MSG display	
00000E00	9802 8C20	00000E20	1019	LM	R0, R2, RPTDWSAV	Restore regs	
00000E04	9857 8C14	00000E14	1021	LM	R5, R7, RPTSVR5T7	Restore R5-7	
00000E08	58F0 8C10	00000E10	1022	L	R15, RPTSAVE	Restore return address	
00000E0C	07FF		1023	BR	R15	Return to caller	
00000E10	00000000		1025	RPTSAVE DC	F'0'	R15 save area	
00000E14	00000000 00000000		1026	RPTSVR5T7 DC	3F'0'	R5-R7 save area	
00000E20	00000000 00000000		1028	RPTDWSAV DC	2D'0'	R0-R2 save area for MSG call	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT		
				1030 ****	*****	*****
				1031 * CALCDUR	Calculate DURATION	
				1032 ****	*****	*****
00000E30	50F0 8C74	00000E74	1034	CALCDUR ST	R15,CALCRET	Save return address
00000E34	9057 8C78	00000E78	1035	STM	R5,R7,CALCWORK	Save work registers
00000E38	9867 8DB8	00000FB8	1037	LM	R6,R7,BEGCLOCK	Remove CPU number from clock value
00000E3C	8C60 0006	00000006	1038	SRDL	R6,6	"
00000E40	8D60 0006	00000006	1039	SLDL	R6,6	"
00000E44	9067 8DB8	00000FB8	1040	STM	R6,R7,BEGCLOCK	"
00000E48	9867 8DC0	00000FC0	1042	LM	R6,R7,ENDCLOCK	Remove CPU number from clock value
00000E4C	8C60 0006	00000006	1043	SRDL	R6,6	"
00000E50	8D60 0006	00000006	1044	SLDL	R6,6	"
00000E54	9067 8DC0	00000FC0	1045	STM	R6,R7,ENDCLOCK	"
00000E58	4150 8DB8	00000FB8	1047	LA	R5,BEGCLOCK	Starting time
00000E5C	4160 8DC0	00000FC0	1048	LA	R6,ENDCLOCK	Ending time
00000E60	4170 8DC8	00000FC8	1049	LA	R7,DURATION	Difference
00000E64	45F0 8C84	00000E84	1050	BAL	R15,SUBDWORD	Calculate duration
00000E68	9857 8C78	00000E78	1052	LM	R5,R7,CALCWORK	Restore work registers
00000E6C	58F0 8C74	00000E74	1053	L	R15,CALCRET	Restore return address
00000E70	07FF		1054	BR	R15	Return to caller
00000E74	00000000		1056	CALCRET DC	F'0'	R15 save area
00000E78	00000000 00000000		1057	CALCWORK DC	3F'0'	R5-R7 save area
				1059 ****	*****	*****
				1060 * SUBDWORD	Subtract two doublewords	
				1061 * R5 --> subtrahend, R6 --> minuend, R7 --> result		
				1062 ****	*****	*****
00000E84	9014 8CA8	00000EA8	1064	SUBDWORD STM	R1,R4,SUBDWSAV	Save registers
00000E88	9812 5000	00000000	1066	LM	R1,R2,0(R5)	Subtrahend (value to subtract)
00000E8C	9834 6000	00000000	1067	LM	R3,R4,0(R6)	Minuend (what to subtract FROM)
00000E90	1F42		1068	SLR	R4,R2	Subtract LOW part
00000E92	47B0 8C9A	00000E9A	1069	BNM	*+4+4	(branch if no borrow)
00000E96	5F30 8DA0	00000FA0	1070	SL	R3,=F'1'	(otherwise do borrow)
00000E9A	1F31		1071	SLR	R3,R1	Subtract HIGH part
00000E9C	9034 7000	00000000	1072	STM	R3,R4,0(R7)	Store results
00000EA0	9814 8CA8	00000EA8	1074	LM	R1,R4,SUBDWSAV	Restore registers
00000EA4	07FF		1075	BR	R15	Return to caller
00000EA8	00000000 00000000		1077	SUBDWSAV DC	2D'0'	R1-R4 save area

LOC	OBJECT CODE	ADDR1	ADDR2	STMT		
				1079 **** 1080 * Issue HERCULES MESSAGE pointed to by R1, length in R0 1081 * R2 = return address 1082 ****		
00000EB8	4900 8DA4		00000FA4	1084 MSG CH R0,=H'0' 1085 BNHR R2		Do we even HAVE a message? No, ignore
00000EBC	07D2					
00000EBE	9002 8CF0		00000EF0	1087 STM R0,R2,MSGSAVE		Save registers
00000EC2	4900 8DA6		00000FA6	1089 CH R0,=AL2(L'MSGMSG)		Message length within limits?
00000EC6	47D0 8CCE		00000ECE	1090 BNH MSGOK		Yes, continue
00000ECA	4100 005F		0000005F	1091 LA R0,L'MSGMSG		No, set to maximum
00000ECE	1820			1093 MSGOK LR R2,R0		Copy length to work register
00000ED0	0620			1094 BCTR R2,0		Minus-1 for execute
00000ED2	4420 8CFC		00000EFC	1095 EX R2,MSGMVC		Copy message to O/P buffer
00000ED6	4120 200A		0000000A	1097 LA R2,1+L'MSGCMD(,R2)		Calculate true command length
00000EDA	4110 8D02		00000F02	1098 LA R1,MSGCMD		Point to true command
00000EDE	83120008			1100 DC X'83',X'12',X'0008'		Issue Hercules Diagnose X'008'
00000EE2	4780 8CE8		00000EE8	1101 BZ MSGRET		Return if successful
00000EE6	0000			1102 DC H'0'		CRASH for debugging purposes
00000EE8	9802 8CF0		00000EF0	1104 MSGRET LM R0,R2,MSGSAVE		Restore registers
00000EEC	07F2			1105 BR R2		Return to caller
00000EF0	00000000 00000000		00000F0B	1107 MSGSAVE DC 3F'0' 00000000 1108 MSGMVC MVC MSGMSG(0),0(R1)		Registers save area Executed instruction
00000EFC	D200 8D0B 1000					
00000F02	D4E2C7D5 D6C8405C			1110 MSGCMD DC C'MSGNOH * '		*** HERCULES MESSAGE COMMAND ***
00000F0B	40404040 40404040			1111 MSGMSG DC CL95' '		The message text to be displayed

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				1113 **** 1114 * Normal completion or Abnormal termination PSWs 1115 ****	
00000F70	00020001 80000000			1117 EOJPSW DC 0D'0',X'0002000180000000',AD(0)	
00000F80	B2B2 8D70	00000F70	1119 EOJ	LPSWE EOJPSW	Normal completion
00000F88	00020001 80000000			1121 FAILPSW DC 0D'0',X'0002000180000000',AD(X'BAD')	
00000F98	B2B2 8D88	00000F88	1123 FAILTEST LPSWE FAILPSW		Abnormal termination
				1125 **** 1126 * Working Storage 1127 ****	
00000F9C		1129	LTORG ,		Literals pool
00000F9C	00000000	1130	=F'0'		
00000FA0	00000001	1131	=F'1'		
00000FA4	0000	1132	=H'0'		
00000FA6	005F	1133	=AL2(L'MSGMSG)		
00000FA8	C3E4E2C5 40	1134	=CL5'CUSE'		
00000FAD	04294967 296C	1135	=P'4294967296'		
		00000400 00000001	1137 K EQU 1024		One KB
		00001000 00000001	1138 PAGE EQU (4*K)		Size of one page
		00008000 00000001	1139 K32 EQU (32*K)		32 KB
		00010000 00000001	1140 K64 EQU (64*K)		64 KB
		00100000 00000001	1141 MB EQU (K*K)		1 MB
00000FB4	00002710		1143 NUMLOOPS DC F'10000'		10,000 * 100 = 1,000,000
00000FB8	BBBBBBBB BBBBCCCC		1145 BEGCLOCK DC 0D'0',8X'BB'		Begin
00000FC0	EEEEEEEE EEEEEE		1146 ENDCLOCK DC 0D'0',8X'EE'		End
00000FC8	DDDDDDDD DDDDDDDD		1147 DURATION DC 0D'0',8X'DD'		Diff
00000FD0	FFFFFF FFNNNN		1148 OVERHEAD DC 0D'0',8X'FF'		Overhead
00000FD8	00000000 0000000C		1150 TICKSAAA DC PL8'0'		Clock ticks high part
00000FE0	00000000 0000000C		1151 TICKSBBB DC PL8'0'		Clock ticks low part
00000FE8	00000000 0000000C		1152 TICKSTOT DC PL8'0'		Total clock ticks
00000FF0	40404040 40404040		1154 PRTLINE DC C' 1,000,000 iterations of XXXXX'		
00001016	40A39696 9240F9F9		1155 PRTLNG DC C' took 999,999,999 microseconds'		
00001034	40202020 6B202020	00000044 00000001	1156 EDIT DC *-PRTLINE		X'402020206B2020206B202120'
		1157	EDIT DC		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				1159 **** 1160 * CUSETEST DSECT 1161 ****	
00000000 00				1163 CUSETEST DSECT , 1164 TNUM DC X'00' 1165 DC XL3'00'	CUSE table number
00000001 000000					
00000004 00				1167 SSLEN DC AL1(0)	CUSE - SS length
00000005 00				1168 PAD DC X'00'	CUSE - Pad byte
00000006 00				1169 SS1LAST DC X'00'	First-Operand SS last byte
00000007 00				1170 SS2LAST DC X'00'	Second-Operand SS last byte
00000008 00000000				1172 SS1ADDR DC A(0)	First-Operand SS Address
0000000C 00000000				1173 SS1LEN DC A(0)	First-Operand SS length
00000010 00000000				1174 SS2ADDR DC A(0)	Second-Operand SS Address
00000014 00000000				1175 SS2LEN DC A(0)	Second-Operand SS length
00000018 00000001				1177 OPSWHERE EQU *	
00000018 00000000				1178 OP1WHERE DC A(0)	Where Operand-1 data should be placed
0000001C 00000000				1179 OP1LEN DC F'0'	CUSE - First-Operand Length
00000020 00000000				1180 OP2WHERE DC A(0)	Where Operand-2 data should be placed
00000024 00000000				1181 OP2LEN DC F'0'	CUSE - Second-Operand Length
				1182	
00000028 00000000				1184 FAILMASK DC A(0)	Failure Branch on Condition mask
0000002C 00000000				1186 *	Ending register values
00000030 00000000				1187 ENDOP1 DC A(0)	Operand 1 address
00000034 00000000				1188 DC A(0)	Operand 1 length
00000038 00000000				1189 ENDOP2 DC A(0)	Operand 2 address
				1190 DC A(0)	Operand 2 length
0000003C 00000001				1192 CUSENEXT EQU *	Start of next table entry...
AABBCCDD 00000001				1194 REG2PATT EQU X'AABBCCDD'	Polluted Register pattern
000000DD 00000001				1195 REG2LOW EQU X'DD'	(last byte above)

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				1197 **** 1198 * CUSE Performe Test data... 1199 ****
00001040		00000000 0000E9AF	1201	CUSE2TST CSECT , 1202 CUSEPERF DC 0A(0)
				Start of table
				1204 **** 1205 * performance test data 1206 ****
				1208 * Cross page bounday - operand-1 and operand-2
00001040			1210	PTE6 DS 0F
00001040	E6		1211	DC X'E6'
00001041	000000		1212	DC XL3'00'
			1213 *	Test Num
00001044	01		1214	DC AL1(1)
00001045	00		1215	DC X'00'
00001046	77		1216	DC X'77'
00001047	77		1217	DC X'77'
			1218 *	Source
00001048	000031B0 00000020		1219	DC A(COP1A),A(032)
00001050	00009DB0 00000020		1220	DC A(COP2A),A(032)
			1221 *	Target
00001058	00B2FFA0 00000200		1222	DC A(11*MB+(6*K32)-96),A(512)
00001060	00C2FF80 00000200		1223	DC A(12*MB+(6*K32)-128),A(512)
			1224 *	Op-1 & length Op-2 & length
00001068	00000006		1225	DC A(6) not CC0 or CC3
			1226 *	Fail mask Ending register values
0000106C	00B30180 00000020		1227	DC A(11*MB+(6*K32)+(512-32)-96),A(032)
00001074	00C30160 00000020		1228	DC A(12*MB+(6*K32)+(512-32)-128),A(032)
				OP-1 OP-2
0000107C			1230	PTE1 DS 0F
0000107C	E1		1231	DC X'E1'
0000107D	000000		1232	DC XL3'00'
			1233 *	Test Num
00001080	04		1234	DC AL1(4)
00001081	00		1235	DC X'00'
00001082	EE		1236	DC X'EE'
00001083	EE		1237	DC X'EE'
			1238 *	Source
00001084	000031B0 00000004		1239	DC A(COP1A),A(004)
0000108C	00009DB0 00000004		1240	DC A(COP2A),A(004)
			1241 *	Target
00001094	00B07FC1 00000200		1242	DC A(11*MB+(1*K32)-63),A(512)
0000109C	00C07FC8 00000200		1243	DC A(12*MB+(1*K32)-56),A(512)
			1244 *	Op-1 & length Op-2 & length
000010A4	00000007		1245	DC A(7) CC0
			1246 *	Fail mask Ending register values
000010A8	00B081BD 00000004		1247	DC A(11*MB+(1*K32)-63+(512-4)),A(004)
000010B0	00C081C4 00000004		1248	DC A(12*MB+(1*K32)-56+(512-4)),A(004)
000010B8			1250	PTE2 DS 0F

LOC	OBJECT CODE	ADDR1	ADDR2	STMT		
000010B8	E2			1251 DC X'E2'		Test Num
000010B9	000000			1252 DC XL3'00'		
				1253 *		
000010BC	08			1254 DC AL1(8)		SS Length
000010BD	00			1255 DC X'00'		Pad Byte
000010BE	77			1256 DC X'77'		First-Operand SS last byte
000010BF	77			1257 DC X'77'		Second-Operand SS last byte
				1258 *		Source
000010C0	000031B0 00000008			1259 DC A(COP1A),A(008)		Op-1 SS & length
000010C8	00009DB0 00000008			1260 DC A(COP2A),A(008)		OP-2 SS & length
				1261 *		Target
000010D0	00B0FFA0 00000200			1262 DC A(11*MB+(2*K32)-96),A(512)		Op-1 & length
000010D8	00C0FF80 00000200			1263 DC A(12*MB+(2*K32)-128),A(512)		Op-2 & length
				1264 *		
000010E0	00000007			1265 DC A(7) CC0		Fail mask
				1266 *		Ending register values
000010E4	00B10198 00000008			1267 DC A(11*MB+(2*K32)+(512-8)-96),A(008)		OP-1
000010EC	00C10178 00000008			1268 DC A(12*MB+(2*K32)+(512-8)-128),A(008)		OP-2
000010F4				1270 PTF2 DS 0F		
000010F4	F2			1271 DC X'F2'		Test Num
000010F5	000000			1272 DC XL3'00'		
				1273 *		
000010F8	20			1274 DC AL1(32)		SS Length
000010F9	00			1275 DC X'00'		Pad Byte
000010FA	77			1276 DC X'77'		First-Operand SS last byte
000010FB	77			1277 DC X'77'		Second-Operand SS last byte
				1278 *		Source
000010FC	000031B0 00000020			1279 DC A(COP1A),A(032)		Op-1 SS & length
00001104	00009DB0 00000020			1280 DC A(COP2A),A(032)		OP-2 SS & length
				1281 *		Target
0000110C	00D0FFA0 00000200			1282 DC A(13*MB+(2*K32)-96),A(512)		Op-1 & length
00001114	00E0FF80 00000200			1283 DC A(14*MB+(2*K32)-128),A(512)		Op-2 & length
				1284 *		
0000111C	00000007			1285 DC A(7) CC0		Fail mask
				1286 *		Ending register values
00001120	00D10180 00000020			1287 DC A(13*MB+(2*K32)+(512-32)-96),A(032)		OP-1
00001128	00E10160 00000020			1288 DC A(14*MB+(2*K32)+(512-32)-128),A(032)		OP-2
00001130				1290 PTE7 DS 0F		
00001130	E7			1291 DC X'E7'		Test Num
00001131	000000			1292 DC XL3'00'		
				1293 *		
00001134	04			1294 DC AL1(4)		SS Length
00001135	00			1295 DC X'00'		Pad Byte
00001136	77			1296 DC X'77'		First-Operand SS last byte
00001137	77			1297 DC X'77'		Second-Operand SS last byte
				1298 *		Source
00001138	000079B0 00000020			1299 DC A(COP1C),A(032)		Op-1 SS & length
00001140	0000E5B0 00000020			1300 DC A(COP2C),A(032)		OP-2 SS & length
				1301 *		Target
00001148	00B37FA0 00000200			1302 DC A(11*MB+(7*K32)-96),A(512)		Op-1 & length
00001150	00C37F80 00000200			1303 DC A(12*MB+(7*K32)-128),A(512)		Op-2 & length
				1304 *		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT			
00001158	00000006			1305 1306 *	DC A(6) not CC0 or CC3	Fail mask	
						Ending register values	
0000115C	00B3817D	00000023		1307	DC A(11*MB+(7*K32)+(512-32)-96-3),A(032+3)	OP-1	
00001164	00C3815D	00000023		1308	DC A(12*MB+(7*K32)+(512-32)-128-3),A(032+3)	OP-2	
0000116C				1310 PTF4	DS 0F		
0000116C	F4			1311	DC X'F4'	Test Num	
0000116D	000000			1312	DC XL3'00'		
				1313 *			
00001170	20			1314	DC AL1(32)	SS Length	
00001171	00			1315	DC X'00'	Pad Byte	
00001172	77			1316	DC X'77'	First-Operand SS last byte	
00001173	77			1317	DC X'77'	Second-Operand SS last byte	
				1318 *		Source	
00001174	000031B0	00000020		1319	DC A(COP1A),A(032)	Op-1 SS & length	
0000117C	00009DB0	00000020		1320	DC A(COP2A),A(032)	OP-2 SS & length	
				1321 *		Target	
00001184	00D1FFA0	00000F80		1322	DC A(13*MB+(4*K32)-96),A(4096-128)	Op-1 & length	
0000118C	00E1FF80	00000F80		1323	DC A(14*MB+(4*K32)-128),A(4096-128)	Op-2 & length	
				1324 *			
00001194	00000006			1325	DC A(6) not CC0 or CC3	Fail mask	
				1326 *		Ending register values	
00001198	00D20F00	00000020		1327	DC A(13*MB+(4*K32)+(4096-128-32)-96),A(032)	OP-1	
000011A0	00E20EE0	00000020		1328	DC A(14*MB+(4*K32)+(4096-128-32)-128),A(032)	OP-2	
000011A8	00000000			1330	DC A(0)	end of table	
000011AC	00000000			1331	DC A(0)	end of table	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				1333 **** 1334 * CUSE Operand-1 scan data... 1335 ****
000011B0				1337 DS 0F
000011B0	98765432	98765432		1338 DC 2048XL4 '98765432' 1339 COP1A DC 256XL4 '111111F0'
000031B0	111111F0	111111F0		
000035B0				1341 DS 0F
000035B0	98765432	98765432		1342 DC 2048XL4 '98765432' 1343 COP1B DC 256XL4 '40404040'
000055B0	40404040	40404040		
000059B0				1345 DS 0F
000059B0	11223344	11223344		1346 DC 2048XL4 '11223344' 1347 COP1C DC 256XL4 '40404040'
000079B0	40404040	40404040		
				1349 **** 1350 * CUSE Operand-2 scan data 1351 ****
00007DB0				1353 DS 0F
00007DB0	89ABCDEF	89ABCDEF		1354 DC 2048XL4 '89ABCDEF' 1355 COP2A DC 256XL4 '111111F0'
00009DB0	111111F0	111111F0		
0000A1B0				1357 DS 0F
0000A1B0	89ABCDEF	89ABCDEF		1358 DC 2048XL4 '89ABCDEF' 1359 COP2B DC 256XL4 '40404040'
0000C1B0	40404040	40404040		
0000C5B0				1361 DS 0F
0000C5B0	FF223344	FF223344		1362 DC 2048XL4 'FF223344' 1363 COP2C DC 256XL4 '40404040'
0000E5B0	40404040	40404040		
				1365 **** 1366 * Register equates 1367 ****
				00000000 00000001 1369 R0 EQU 0 00000001 00000001 1370 R1 EQU 1 00000002 00000001 1371 R2 EQU 2 00000003 00000001 1372 R3 EQU 3 00000004 00000001 1373 R4 EQU 4 00000005 00000001 1374 R5 EQU 5 00000006 00000001 1375 R6 EQU 6 00000007 00000001 1376 R7 EQU 7 00000008 00000001 1377 R8 EQU 8 00000009 00000001 1378 R9 EQU 9 0000000A 00000001 1379 R10 EQU 10 0000000B 00000001 1380 R11 EQU 11 0000000C 00000001 1381 R12 EQU 12

LOC	OBJECT CODE	ADDR1	ADDR2	STMT		
	0000000D	00000001	1382	R13	EQU	13
	0000000E	00000001	1383	R14	EQU	14
	0000000F	00000001	1384	R15	EQU	15

1386 END

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES								
BEGCLOCK	D	00000FB8	8	1145	233	557	1037	1040	1047				
BEGIN	I	00000200	2	98	65	95	96						
CALCDUR	I	00000E30	4	1034	549	992							
CALCRET	F	00000E74	4	1056	1034	1053							
CALCWORK	F	00000E78	4	1057	1035	1052							
COP1A	X	000031B0	4	1339	1219	1239	1259	1279	1319				
COP1B	X	000055B0	4	1343									
COP1C	X	000079B0	4	1347	1299								
COP2A	X	00009DB0	4	1355	1220	1240	1260	1280	1320				
COP2B	X	0000C1B0	4	1359									
COP2C	X	0000E5B0	4	1363	1300								
CUSE2TST	J	00000000	59824	60	63	67	71	130	61				
CUSENEXT	J	0000003C	1	1192	980								
CUSEPERF	A	00001040	4	1202	153								
CUSETEST	4	00000000	60	1163	154								
DURATION	D	00000FC8	8	1147	550	995	996	999	1049				
EDIT	X	00001034	12	1157	1009	1010							
ENDCLOCK	D	00000FC0	8	1146	548	971	1042	1045	1048				
ENDOP1	A	0000002C	4	1187									
ENDOP2	A	00000034	4	1189									
EOJ	I	00000F80	4	1119	116	124							
EOJPSW	D	00000F70	8	1117	1119								
FAILMASK	A	00000028	4	1184									
FAILPSW	D	00000F88	8	1121	1123								
FAILTEST	I	00000F98	4	1123	119	122							
IMAGE	I	00000000	59824	0									
K	U	00000400	1	1137	1138	1139	1140	1141					
K32	U	00008000	1	1139	1222	1223	1227	1228	1242	1243	1247	1248	1262
					1283	1287	1288	1302	1303	1307	1308	1322	1323
K64	U	00010000	1	1140									
MB	U	00100000	1	1141	1222	1223	1227	1228	1242	1243	1247	1248	1262
					1283	1287	1288	1302	1303	1307	1308	1322	1323
MSG	I	00000EB8	4	1084	1018								
MSGCMD	C	00000F02	9	1110	1097	1098							
MSGMSG	C	00000F0B	95	1111	1091	1108	1089						
MSGMVC	I	00000EFC	6	1108	1095								
MSGOK	I	00000ECE	2	1093	1090								
MSGRET	I	00000EE8	4	1104	1101								
MSGSAVE	F	00000EF0	4	1107	1087	1104							
NUMLOOPS	F	00000FB4	4	1143	232	556							
OP1LEN	F	0000001C	4	1179	167	171							
OP1WHERE	A	00000018	4	1178	166								
OP2LEN	F	00000024	4	1181	180	184							
OP2WHERE	A	00000020	4	1180	179								
OPSWHERE	U	00000018	1	1177	239	242	250	253	256	259	262	265	268
					283	286	289	292	295	298	301	304	307
					322	325	328	331	334	337	340	343	346
					361	364	367	370	373	376	379	382	385
					400	403	406	409	412	415	418	421	424
					439	442	445	448	451	454	457	460	463
					478	481	484	487	490	493	496	499	502
					517	520	523	526	529	532	535	541	544
					583	587	591	595	599	603	607	611	615
					635	639	643	647	651	655	659	663	667
					687	691	695	699	703	707	711	715	719
					739	743	747	751	755	759	763	767	771

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES	791	795	799	803	807	811	815	819	823	827	831	835	839
						843	847	851	855	859	863	867	871	875	879	883	887	891
						895	899	903	907	911	915	919	923	927	931	935	939	943
						947	951	955	962	966								
OVERHEAD	D	00000FD0	8	1148	550	994												
PAD	X	00000005	1	1168	193													
PAGE	U	00001000	1	1138														
PRTLINE	C	00000FF0	38	1154	1156	974	1009	1010	1017									
PRTLNG	U	00000044	1	1156	1016													
PTE1	F	0000107C	4	1230														
PTE2	F	000010B8	4	1250														
PTE6	F	00001040	4	1210														
PTE7	F	00001130	4	1290														
PTF2	F	000010F4	4	1270														
PTF4	F	0000116C	4	1310														
R0	U	00000000	1	1369	61	192	1015	1016	1019	1084	1087	1089	1091	1093	1104			
R1	U	00000001	1	1370	193	1017	1064	1066	1071	1074	1098	1108						
R10	U	0000000A	1	1379	168	169	170	172	181	182	183	185	235	547	558	970	999	
					1000	1002												
R11	U	0000000B	1	1380	171	184	999	1003										
R12	U	0000000C	1	1381														
R13	U	0000000D	1	1382														
R14	U	0000000E	1	1383	109	151	983											
R15	U	0000000F	1	1384	549	975	989	992	997	1022	1023	1034	1050	1053	1054	1075		
R2	U	00000002	1	1371	166	172	174	175	234	239	242	250	253	256	259	262	265	
					268	271	274	277	280	283	286	289	292	295	298	301	304	
					307	310	313	316	319	322	325	328	331	334	337	340	343	
					346	349	352	355	358	361	364	367	370	373	376	379	382	
					385	388	391	394	397	400	403	406	409	412	415	418	421	
					424	427	430	433	436	439	442	445	448	451	454	457	460	
					463	466	469	472	475	478	481	484	487	490	493	496	499	
					502	505	508	511	514	517	520	523	526	529	532	535	541	
					544	562	563	566	567	575	576	579	580	583	584	587	588	
					591	592	595	596	599	600	603	604	607	608	611	612	615	
					616	619	620	623	624	627	628	631	632	635	636	639	640	
					643	644	647	648	651	652	655	656	659	660	663	664	667	
					668	671	672	675	676	679	680	683	684	687	688	691	692	
					695	696	699	700	703	704	707	708	711	712	715	716	719	
					720	723	724	727	728	731	732	735	736	739	740	743	744	
					747	748	751	752	755	756	759	760	763	764	767	768	771	
					772	775	776	779	780	783	784	787	788	791	792	795	796	
					799	800	803	804	807	808	811	812	815	816	819	820	823	
					824	827	828	831	832	835	836	839	840	843	844	847	848	
					851	852	855	856	859	860	863	864	867	868	871	872	875	
					876	879	880	883	884	887	888	891	892	895	896	899	900	
					903	904	907	908	911	912	915	916	919	920	923	924	927	
					928	931	932	935	936	939	940	943	944	947	948	951	952	
					955	956	962	963	966	967	973	1015	1018	1019	1066	1068	1085	
R3	U	00000003	1	1372	167	169	1067	1070	1071	1072								
R4	U	00000004	1	1373	179	185	187	188	563	567	576	580	584	588	592	596	600	
					604	608	612	616	620	624	628	632	636	640	644	648	652	
					656	660	664	668	672	676	680	684	688	692	696	700	704	
					708	712	716	720	724	728	732	736	740	744	748	752	756	
					760	764	768	772	776	780	784	788	792	796	800	804	808	
					812	816	820	824	828	832	836	840	844	848	852	856	860	

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES	864	868	872	876	880	884	888	892	896	900	904	908	912
						916	920	924	928	932	936	940	944	948	952	956	963	967
R5	U	00000005	1	1374	1064 1067 1068 1072 1074 180 182 234 239 242 250 253 256 259 262 265 268 271 274 277 280 283 286 289 292 295 298 301 304 307 310 313 316 319 322 325 328 331 334 337 340 343 346 349 352 355 358 361 364 367 370 373 376 379 382 385 388 391 394 397 400 403 406 409 412 415 418 421 424 427 430 433 436 439 442 445 448 451 454 457 460 463 466 469 472 475 478 481 484 487 490 493 496 499 502 505 508 511 514 517 520 523 526 529 532 535 541 544 562 566 575 579 583 587 591 595 599 603 607 611 615 619 623 627 631 635 639 643 647 651 655 659 663 667 671 675 679 683 687 691 695 699 703 707 711 715 719 723 727 731 735 739 743 747 751 755 759 763 767 771 775 779 783 787 791 795 799 803 807 811 815 819 823 827 831 835 839 843 847 851 855 859 863 867 871 875 879 883 887 891 895 899 903 907 911 915 919 923 927 931 935 939 943 947 951 955 962 966 973 990 994 1021 1035 1047 1052 1066 R6 U 00000006 1 1375 153 154 157 979 980 981 995 1037 1038 1039 1040 1042 1043 1044 1045 1048 1067 R7 U 00000007 1 1376 159 160 232 547 556 970 990 996 1021 1035 1037 1040 1042 1045 1049 1052 1072 R8 U 00000008 1 1377 95 98 99 100 102 R9 U 00000009 1 1378 96 102 103 REG2LOW U 000000DD 1 1195 REG2PATT U AABBCDDD 1 1194 RPTDWSAV D 00000E20 8 1028 1015 1019 RPTSAVE F 00000E10 4 1025 989 1022 RPTSPEED I 00000DA6 4 989 975 RPTSVR5T7 F 00000E14 4 1026 990 1021 SAVE2T5 F 00000410 4 140 234 973 SAVER2 F 00000420 4 141 SAVER6 F 00000424 4 142 157 979 SS1ADDR A 00000008 4 1172 168 SS1LAST X 00000006 1 1169 175 SS1LEN A 0000000C 4 1173 170 SS2ADDR A 00000010 4 1174 181 SS2LAST X 00000007 1 1170 188 SS2LEN A 00000014 4 1175 183 SSLEN R 00000004 1 1167 192 SUBDWORD I 00000E84 4 1064 997 1050 SUBDWSAV D 00000EA8 8 1077 1064 1074 SUBTEST X 00000401 1 134 121 TEST91 I 00000528 4 150 109 TESTADDR D 00000400 8 132 TESTNUM X 00000400 1 133 118 160 TICKSAAA P 00000FD8 8 1150 1002 1005 TICKSBBB P 00000FE0 8 1151 1003 1007 TICKSTOT P 00000FE8 8 1152 1005 1006 1007 1010 TIMEOPT X 00000408 1 137 115 150 TNUM X 00000000 1 1164 159 TST91LOP U 00000532 1 156 982 =AL2(L'MSGMSG) R 00000FA6 2 1133 1089 =CL5'CUSE' C 00000FA8 5 1134 974													

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
=F'0'	F	00000F9C	4	1130	981
=F'1'	F	00000FA0	4	1131	1070
=H'0'	H	00000FA4	2	1132	1084
=P'4294967296'	P	00000FAD	6	1135	1006

MACRO DEFN REFERENCES

DOINSTR	214	560	573	960
OVERONLY	200	237	248	539

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

Entry: 0

Image	IMAGE	59824	0000-E9AF	0000-E9AF
Region		59824	0000-E9AF	0000-E9AF
CSECT	CUSE2TST	59824	0000-E9AF	0000-E9AF

STMT

FILE NAME

```
1      /devstor/dev/tests/CUSE-02-performance.asm
```

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