

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
2				*****
3	*			*
4	*			CU12 instruction tests
5	*			*
6	*			NOTE: This test is based the CLCL-et-al Test
7	*			modified to only test the Performance
8	*			of the CU12 instruction.
9	*			*
10	*			The MSG routine is from the Hercules Binary
11	*			Floating Point Validation Package by Stephen R. Orso
12	*			*
13	*			*****
14	*			** IMPORTANT! **
15	*			*****
16	*			*
17	*			This test uses the Hercules Diagnose X'008' interface
18	*			to display messages and thus your .tst runtest script
19	*			MUST contain a "DIAG8CMD ENABLE" statement within it!
20	*			*
21	*			James Wekel February 2024
22	*			*****
24				*****
25	*			*
26	*			CU12 Performance instruction tests
27	*			*
28	*			*****
29	*			*
30	*			This program ONLY tests the performance of the CU12
31	*			instructions.
32	*			*
33	*			Tests:
34	*			*
35	*			All tests are 'CU12 R0, R2'
36	*			*
37	*			1. CU12 with CC=0 - no crossed pages
38	*			source: 61 bytes (28 UTF8 Chars)
39	*			*
40	*			2. CU12 with CC=0 - source cross page
41	*			source: 61 bytes (28 UTF8 Chars)
42	*			*
43	*			3. CU12 with CC=0 - target cross page
44	*			source: 61 bytes (28 UTF8 Chars)
45	*			*
46	*			4. CU12 with CC=0 - both arguments crossed pages
47	*			source: 61 bytes (28 UTF8 Chars)
48	*			*
49	*			5. CU12 with CC=3 - both arguments crossed pages
50	*			source: 13,738 bytes only 4095+ processed
51	*			*
52	*			*
53	*			*****

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				55 **** 56 * 57 * Example Hercules Testcase: 58 * 59 *
				60 * *Testcase CU12-02-performance (Test CU12 instructions) 61 * mainsize 16 62 * numcpu 1 63 * sysclear 64 * archlvl z/Arch 65 * 66 * loadcore "\$(testpath)/CU12-02-performance.core" 0x0 67 * 68 * diag8cmd enable # (needed for messages to Hercules console) 69 * #r 408=ff # (enable timing tests) 70 * runtest 300 # (test duration, depends on host) 71 * diag8cmd disable # (reset back to default) 72 * 73 * *Done 74 * 75 * 76 ****
				78 **** 79 * Low Core Definitions 80 **** 81 *
00000000	00000000 00000D17	00000000	00000D17	82 CU122TST START 0 83 USING CU122TST, R0 Low core addressability
00000000 000001A0 000001A8	00000001 80000000 00000000 00000200	00000000	000001A0	85 ORG CU122TST+X'1A0' 86 DC X'0000000180000000' 87 DC AD(BEGIN)
000001B0 000001D0 000001D8	00020001 80000000 00000000 0000DEAD	000001B0	000001D0	89 ORG CU122TST+X'1D0' 90 DC X'0002000180000000' 91 DC AD(X' DEAD')
000001E0		000001E0	00000200	93 ORG CU122TST+X'200' Start of actual test program ..

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				95 **** 96 * The actual "CU122TST" program itself... 97 **** 98 * 99 * Architecture Mode: 370 100 * Register Usage: 101 * 102 * R0 (work) 103 * R1 (work) 104 * R2 (work) or MSG subroutine call 105 * R3 (work) 106 * R4 (work) 107 * R5-R7 (work) 108 * R8 First base register 109 * R9 Second base register 110 * R10-R12 (work) 111 * R13 CU12TEST Base (of current test) 112 * R14 Subroutine call 113 * R15 Secondary Subroutine call or work 114 * 115 ****	
00000200		00000200		117 USING BEGIN, R8	FIRST Base Register
00000200		00001200		118 USING BEGIN+4096, R9	SECOND Base Register
00000200	0580			120 BEGIN BALR R8, 0	Initialize FIRST base register
00000202	0680			121 BCTR R8, 0	Initialize FIRST base register
00000204	0680			122 BCTR R8, 0	Initialize FIRST base register
00000206	4190 8800		00000800	124 LA R9, 2048(, R8)	Initialize SECOND base register
0000020A	4190 9800		00000800	125 LA R9, 2048(, R9)	Initialize SECOND base register
				127 * 128 ** Run the performance tests... 129 *	
0000020E	45E0 8328		00000528	130 BAL R14, TEST91	Time CU12 instruction (speed test)

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				132 ****	*****
				133 *	Test for normal or unexpected test completion...
				134 ****	*****
00000212	95FF 8208	00000408	136	CLI TIMEOPT, X' FF'	Was this a timing run?
00000216	4770 8A58	00000C58	137	BNE EOJ	No, timing run; just go end normally
0000021A	9505 8200	00000400	139	CLI TESTNUM, X' 05'	Did we end on expected test?
0000021E	4770 8A70	00000C70	140	BNE FAILTEST	No?! Then FAIL the test!
00000222	9599 8201	00000401	142	CLI SUBTEST, X' 99'	Did we end on expected SUB-test?
00000226	4770 8A70	00000C70	143	BNE FAILTEST	No?! Then FAIL the test!
0000022A	47F0 8A58	00000C58	145	B EOJ	Yes, then normal completion!
				147 ****	*****
				148 *	Fixed test storage locations ...
				149 ****	*****
0000022E	0000022E	00000400	151	ORG BEGIN+X' 200'	
00000400			152		
00000400	99		153 TESTADDR	DS OD	Where test/subtest numbers will go
00000401	99		154 TESTNUM	DC X' 99'	Test number of active test
			155 SUBTEST	DC X' 99'	Active test sub-test number
00000408	00		157		
00000408	00		158 TIMEOPT	DS OD	Set to non-zero to run timing tests
00000410			160		
00000410	00000000 00000000		161 SAVE3T5	DC 4F' 0'	
00000420	00000000		162 SAVER2	DC F' 0'	
00000424	00000000		163 SAVER13	DC F' 0'	
00000428	00000428	00000528	165	ORG *+X' 100'	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
167				*****
168	*			Define come helpful macros to ensure our counts are correct
169				*****
171				MACRO
172				OVERONLY &NUM
173				LCLA &CTR
174	&CTR			SETA &NUM
175	. LOOP			ANOP
176	. *			
177	*			
178		LM	R0, R3, OPSPERF	Get CU12 operands
179	. *			
180	&CTR	SETA	&CTR- 1	
181		AIF	(&CTR GT 0). LOOP	
182		MEND		
184				MACRO
185				DOINSTR &NUM
186				LCLA &CTR
187	&CTR	SETA	&NUM	
188	. LOOP	ANOP		
189	. *			
190	*			
191		LM	R0, R3, OPSPERF	Load CU12 operands
192		CU12	R0, R2	Do CU12
193	. *			
194	&CTR	SETA	&CTR- 1	
195		AIF	(&CTR GT 0). LOOP	
196		MEND		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT		
				198 **** 199 * TEST91 200 ****	Time CU12 instruction (speed test)	
00000528	91FF 8208		00000408	202 TEST91	TM TIMEOPT, X' FF'	Is timing tests option enabled?
0000052C	078E			203	BZR R14	No, skip timing tests
0000052E	41D0 8B18		00000D18	205	LA R13, CU12CTL	Point R13 --> testing control table
00000532		00000000		206 USING CU12PERF, R13 207 *		What each table entry looks like
		00000532	00000001	208 TST91LOP EQU *		
00000532	50D0 8224		00000424	209 ST R13, SAVER13		save current pref table base
00000536	4360 D000		00000000	211 IC R6, TNUM		Set test number
0000053A	4260 8200		00000400	212 STC R6, TESTNUM 213 *		
				214 ** Initialize operand data (move source to testing address)		
				215 * 216 *		Source
0000053E	5800 D018		00000018	217 L R0, OP2WHERE		Where to move operand-2 data to
00000542	5810 D010		00000010	218 L R1, OP2LEN		How much of it there is
00000546	5820 D00C		0000000C	219 L R2, OP2DATA		Where op2 data is right now
0000054A	5830 D010		00000010	220 L R3, OP2LEN		How much of it there is
0000054E	OE02			221 MVCL R0, R2 222 223 *		
						set up CU12 args
00000550	5800 D014		00000014	224 L R0, OP1WHERE		Where to move operand-1 data to
00000554	5810 D008		00000008	225 L R1, OP1LEN		operand-1 length
00000558	5820 D018		00000018	226 L R2, OP2WHERE		Where to move operand-2 data to
0000055C	5830 D010		00000010	227 L R3, OP2LEN		operand-2 length
00000560	9003 8868		00000A68	228 STM R0, R3, OPSPERF 229 *		save for each CU12 execution

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				231 **** 232 * Next, time the overhead... 233 ****
00000564	5870 8A8C	00000C8C	235	L R7, NUMLOOPS
00000568	B205 8A90	00000C90	236	STCK BEGCLOCK
0000056C	9035 8210	00000410	237	STM R3, R5, SAVE3T5
00000570	0560		238	BALR R6, 0
			239 *	
			240	OVERONLY 2
			241+*	100 sets of overhead (first 2)
00000572	9803 8868	00000A68	242+	LM R0, R3, OPSPERF
00000576	9803 8868	00000A68	243+*	Get CU12 operands
			244+	Get CU12 operands
			246 * ETC.
			248	PRINT OFF
			442	PRINT ON
			444	
			445+*	OVERONLY 2
000006FA	9803 8868	00000A68	446+	LM R0, R3, OPSPERF
000006FE	9803 8868	00000A68	447+*	Get CU12 operands
			448+	LM R0, R3, OPSPERF
			449 *	Get CU12 operands
00000702	0676		450	BCTR R7, R6
00000704	B205 8A98	00000C98	451	STCK ENDCLOCK
00000708	45F0 8908	00000B08	452	BAL R15, CALCDUR
0000070C	D207 8AA8 8AA0	00000CA8	453	MVC OVERHEAD, DURATION

LOC	OBJECT CODE	ADDR1	ADDR2	STMT			
				455 ****			
				456 * Now do the actual timing run...			
				457 ****			
00000712	5870 8A8C		00000C8C	459 L R7, NUMLOOPS			
00000716	B205 8A90		00000C90	460 STCK BEGCLOCK			
0000071A	0560			461 BALR R6, 0			
				462 *		100 sets of instructions	
				463 DOINSTR 2		(first 2)	
0000071C	9803 8868		00000A68	464+ LM R0, R3, OPSPERF		Load CU12 operands	
00000720	B2A7 0002			465+ CU12 R0, R2		Do CU12	
00000724	9803 8868		00000A68	466+ LM R0, R3, OPSPERF		Load CU12 operands	
00000728	B2A7 0002			467+ CU12 R0, R2		Do CU12	
				468+ LM R0, R3, OPSPERF			
				469+ CU12 R0, R2			
				471 * ETC.			
				473 PRINT OFF			
				763 PRINT ON			
				765 DOINSTR 2		(last 2)	
00000A2C	9803 8868		00000A68	766+ LM R0, R3, OPSPERF		Load CU12 operands	
00000A30	B2A7 0002			767+ CU12 R0, R2		Do CU12	
00000A34	9803 8868		00000A68	768+ LM R0, R3, OPSPERF		Load CU12 operands	
00000A38	B2A7 0002			769+ CU12 R0, R2		Do CU12	
				770+ LM R0, R3, OPSPERF			
				771+ CU12 R0, R2			
00000A3C	0676			773 BCTR R7, R6			
00000A3E	B205 8A98		00000C98	774 STCK END CLOCK			
00000A42	9835 8210		00000410	776 LM R3, R5, SAVE3T5			
00000A46	D204 8AE9 8A80	00000CE9	00000C80	777 MVC PRTLINE+33(5), =CL5' CU12'			
00000A4C	45F0 8888		00000A88	778 BAL R15, RPT SPEED			
				779 *			
				780 * more performance tests?			
00000A50	58D0 8224		00000424	781 * L R13, SAVER13		restore perf table base	
00000A54	41D0 D028		00000028	782 LA R13, CU12NEXT		Go on to next table entry	
00000A58	D503 8A74 D000	00000C74	00000000	783 CLC =F' 0', 0(R13)		End of table?	
00000A5E	4770 8332		00000532	784 BNE TST91LOP		No, loop...	
00000A62	07FE			785 BR R14		Return to caller or FAILTEST	
00000A68	00000000 00000000			786			
				788 OPSPERF DS 4D		Performance test R0-R3	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT			
				790 ****	*****	*****	*****
				791 * RPT SPEED	RPT SPEED	Report instruction speed	
				792 ****	*****	*****	*****
00000A88	50F0 88F0		00000AF0	794 RPT SPEED	ST	R15, RPTSAVE	Save return address
00000A8C	5050 88F4		00000AF4	795 *	ST	R5, RPTSVR5	Save R5
00000A90	45F0 8908		00000B08	796 *	BAL	R15, CALCDUR	Calculate duration
00000A94	4150 8AA8		00000CA8	797 *	LA	R5, OVERHEAD	Subtract overhead
00000A98	4160 8AA0		00000CA0	799 800	LA	R6, DURATION	From raw timing
00000A9C	4170 8AA0		00000CA0	801	LA	R7, DURATION	Yielding true instruction timing
00000AA0	45F0 895C		00000B5C	802 *	BAL	R15, SUBDWORD	Do it
00000AA4	98AB 8AA0		00000CA0	803 *	LM	R10, R11, DURATION	Convert to... ... microseconds
00000AA8	8CA0 000C		0000000C	804	SRDL	R10, 12	
00000AAC	4EA0 8AB0		00000CB0	805	CVD	R10, TICKSAAA	convert HIGH part to decimal
00000AB0	4EB0 8AB8		00000CB8	806 *	CVD	R11, TICKSBBC	convert LOW part to decimal
00000AB4	F877 8AC0 8AB0	00000CC0	00000CB0	807	ZAP	TICKSTOT, TICKSAAA	Calculate...
00000ABA	FC75 8AC0 8A85	00000CC0	00000C85	808	MP	TICKSTOT, =P' 4294967296'	... decimal...
00000AC0	FA77 8AC0 8AB8	00000CC0	00000CB8	809 *	AP	TICKSTOT, TICKSBBC	... microseconds
00000AC6	D20B 8AF3 8B0C	00000CF3	00000DOC	810			
00000ACC	DE0B 8AF3 8AC3	00000CF3	00000CC3	811	MVC	PRTLINE+43(L' EDIT), EDIT	(edit into... ... print line)
				812		PRTLINE+43(L' EDIT), TICKSTOT+3	
				813 *			
				814			
				815	ED		
				816			
				817 *			
				818 *		Use Hercules Diagnose for Message to console	
				819 *			
00000AD2	9002 88F8		00000AF8	820	STM	R0, R2, RPTDWSAV	save regs used by MSG
00000AD6	4100 0044		00000044	821	LA	R0, PRTLNG	message length
00000ADA	4110 8AC8		00000CC8	822	LA	R1, PRTLINE	message address
00000ADE	4520 8990		00000B90	823	BAL	R2, MSG	call Hercules console MSG display
00000AE2	9802 88F8		00000AF8	824	LM	R0, R2, RPTDWSAV	restore regs
00000AE6	5850 88F4		00000AF4	825	L	R5, RPTSVR5	Restore R5
00000AEA	58F0 88F0		00000AF0	826	L	R15, RPTSAVE	Restore return address
00000AEE	07FF			827	BR	R15	Return to caller
00000AF0	00000000			828			
00000AF4	00000000			829			
00000AF8	00000000 00000000			830 RPTSAVE	DC	F' 0'	R15 save area
				831 RPTSVR5	DC	F' 0'	R5 save area
				832 RPTDWSAV	DC	2D' 0'	R0-R2 save area for MSG call
				833 RPTDWSAV	DC		
				834 RPTDWSAV	DC		
				835 RPTDWSAV	DC		
				836 RPTDWSAV	DC		
				837 RPTDWSAV	DC		
				838 RPTDWSAV	DC		
				839 RPTDWSAV	DC		
				840 RPTDWSAV	DC		
				841 RPTDWSAV	DC		
				842 RPTDWSAV	DC		
				843 RPTDWSAV	DC		
				844 RPTDWSAV	DC		
				845 RPTDWSAV	DC		
				846 RPTDWSAV	DC		
				847 RPTDWSAV	DC		
				848 RPTDWSAV	DC		
				849 RPTDWSAV	DC		
				850 RPTDWSAV	DC		
				851 RPTDWSAV	DC		
				852 RPTDWSAV	DC		
				853 RPTDWSAV	DC		
				854 RPTDWSAV	DC		
				855 RPTDWSAV	DC		
				856 RPTDWSAV	DC		
				857 RPTDWSAV	DC		
				858 RPTDWSAV	DC		
				859 RPTDWSAV	DC		
				860 RPTDWSAV	DC		
				861 RPTDWSAV	DC		
				862 RPTDWSAV	DC		
				863 RPTDWSAV	DC		
				864 RPTDWSAV	DC		
				865 RPTDWSAV	DC		
				866 RPTDWSAV	DC		
				867 RPTDWSAV	DC		
				868 RPTDWSAV	DC		
				869 RPTDWSAV	DC		
				870 RPTDWSAV	DC		
				871 RPTDWSAV	DC		
				872 RPTDWSAV	DC		
				873 RPTDWSAV	DC		
				874 RPTDWSAV	DC		
				875 RPTDWSAV	DC		
				876 RPTDWSAV	DC		
				877 RPTDWSAV	DC		
				878 RPTDWSAV	DC		
				879 RPTDWSAV	DC		
				880 RPTDWSAV	DC		
				881 RPTDWSAV	DC		
				882 RPTDWSAV	DC		
				883 RPTDWSAV	DC		
				884 RPTDWSAV	DC		
				885 RPTDWSAV	DC		
				886 RPTDWSAV	DC		
				887 RPTDWSAV	DC		
				888 RPTDWSAV	DC		
				889 RPTDWSAV	DC		
				890 RPTDWSAV	DC		
				891 RPTDWSAV	DC		
				892 RPTDWSAV	DC		
				893 RPTDWSAV	DC		
				894 RPTDWSAV	DC		
				895 RPTDWSAV	DC		
				896 RPTDWSAV	DC		
				897 RPTDWSAV	DC		
				898 RPTDWSAV	DC		
				899 RPTDWSAV	DC		
				900 RPTDWSAV	DC		
				901 RPTDWSAV	DC		
				902 RPTDWSAV	DC		
				903 RPTDWSAV	DC		
				904 RPTDWSAV	DC		
				905 RPTDWSAV	DC		
				906 RPTDWSAV	DC		
				907 RPTDWSAV	DC		
				908 RPTDWSAV	DC		
				909 RPTDWSAV	DC		
				910 RPTDWSAV	DC		
				911 RPTDWSAV	DC		
				912 RPTDWSAV	DC		
				913 RPTDWSAV	DC		
				914 RPTDWSAV	DC		
				915 RPTDWSAV	DC		
				916 RPTDWSAV	DC		
				917 RPTDWSAV	DC		
				918 RPTDWSAV	DC		
				919 RPTDWSAV	DC		
				920 RPTDWSAV	DC		
				921 RPTDWSAV	DC		
				922 RPTDWSAV	DC		
				923 RPTDWSAV	DC		
				924 RPTDWSAV	DC		
				925 RPTDWSAV	DC		
				926 RPTDWSAV	DC		
				927 RPTDWSAV	DC		
				928 RPTDWSAV	DC		
				929 RPTDWSAV	DC		
				930 RPTDWSAV	DC		
				931 RPTDWSAV	DC		
				932 RPTDWSAV	DC		
				933 RPTDWSAV	DC		
				934 RPTDWSAV	DC		
				935 RPTDWSAV	DC		
				936 RPTDWSAV	DC		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT			
				835 ****	*****	*****	*****
				836 *	CALCDUR	Cal culate DURATION	
				837 ****	*****	*****	*****
00000B08	50F0 894C	00000B4C	839 CALCDUR	ST STM	R15, CALCRET R5, R7, CALCWORK	Save return address Save work registers	
00000B0C	9057 8950	00000B50	840 *	LM SRDL SLDL STM	R6, R7, BEGCLOCK R6, 6 R6, 6 R6, R7, BEGCLOCK	Remove CPU number from clock value "	
00000B10	9867 8A90	00000C90	842				
00000B14	8C60 0006	00000006	843	SRDL	R6, 6	"	
00000B18	8D60 0006	00000006	844	SLDL	R6, 6	"	
00000B1C	9067 8A90	00000C90	845	STM	R6, R7, BEGCLOCK	"	
			846 *				
00000B20	9867 8A98	00000C98	847	LM	R6, R7, ENDCLOCK	Remove CPU number from clock value	
00000B24	8C60 0006	00000006	848	SRDL	R6, 6	"	
00000B28	8D60 0006	00000006	849	SLDL	R6, 6	"	
00000B2C	9067 8A98	00000C98	850	STM	R6, R7, ENDCLOCK	"	
			851 *				
00000B30	4150 8A90	00000C90	852	LA	R5, BEGCLOCK	Starting time	
00000B34	4160 8A98	00000C98	853	LA	R6, ENDCLOCK	Endi ng time	
00000B38	4170 8AA0	00000CA0	854	LA	R7, DURATION	Difference	
00000B3C	45F0 895C	00000B5C	855	BAL	R15, SUBDWORD	Cal culate duration	
			856 *				
00000B40	9857 8950	00000B50	857	LM	R5, R7, CALCWORK	Restore work registers	
00000B44	58F0 894C	00000B4C	858	L	R15, CALCRET	Restore return address	
00000B48	07FF		859	BR	R15	Return to caller	
00000B4C	00000000		861 CALCRET	DC	F' 0'	R15 save area	
00000B50	00000000 00000000		862 CALCWORK	DC	3F' 0'	R5-R7 save area	
			864 ****	*****	*****	*****	
			865 *	SUBDWORD	Subtract two doubl ewords		
			866 *	R5 --> subtrahend, R6 --> minuend, R7 --> result			
			867 ****	*****	*****	*****	
00000B5C	9014 8980	00000B80	869 SUBDWORD	STM	R1, R4, SUBDWSAV	Save registers	
			870 *				
00000B60	9812 5000	00000000	871	LM	R1, R2, 0(R5)	Subtrahend (value to subtract)	
00000B64	9834 6000	00000000	872	LM	R3, R4, 0(R6)	Minuend (what to subtract FROM)	
00000B68	1F42		873	SLR	R4, R2	Subtract LOW part	
00000B6A	47B0 8972	00000B72	874	BNM	*+4+4	(branch if no borrow)	
00000B6E	5F30 8A78	00000C78	875	SL	R3, =F' 1'	(otherwise do borrow)	
00000B72	1F31		876	SLR	R3, R1	Subtract HIGH part	
00000B74	9034 7000	00000000	877	STM	R3, R4, 0(R7)	Store results	
			878 *				
00000B78	9814 8980	00000B80	879	LM	R1, R4, SUBDWSAV	Restore registers	
00000B7C	07FF		880	BR	R15	Return to caller	
00000B80	00000000 00000000		882 SUBDWSAV	DC	2D' 0'	R1-R4 save area	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT			
				884 **** 885 * Issue HERCULES MESSAGE pointed to by R1, length in R0 886 * R2 = return address 887 ****			
00000B90	4900 8A7C		00000C7C	889 MSG	CH	R0, =H' 0'	Do we even HAVE a message?
00000B94	07D2			890	BNHR	R2	No, ignore
00000B96	9002 89C8		00000BC8	892	STM	R0, R2, MSGSAVE	Save registers
00000B9A	4900 8A7E		00000C7E	894	CH	R0, =AL2(L' MSGMSG)	Message length within limits?
00000B9E	47D0 89A6		00000BA6	895	BNH	MSGOK	Yes, continue
00000BA2	4100 005F		0000005F	896	LA	R0, L' MSGMSG	No, set to maximum
00000BA6	1820		00000BD4	898 MSGOK	LR	R2, R0	Copy length to work register
00000BA8	0620			899	BCTR	R2, 0	Minus-1 for execute
00000BAA	4420 89D4			900	EX	R2, MSGMVC	Copy message to O/P buffer
00000BAE	4120 200A		0000000A	902	LA	R2, 1+L' MSGCMD(, R2)	Calculate true command length
00000BB2	4110 89DA		00000BDA	903	LA	R1, MSGCMD	Point to true command
00000BB6	83120008		00000BC0	905	DC	X' 83' , X' 12' , X' 0008'	Issue Hercules Diagnose X' 008'
00000BBA	4780 89C0			906	BZ	MSGRET	Return if successful
00000BBE	0000			907	DC	H' 0'	CRASH for debugging purposes
00000BC0	9802 89C8		00000BC8	909 MSGRET	LM	R0, R2, MSGSAVE	Restore registers
00000BC4	07F2			910	BR	R2	Return to caller
00000BC8	00000000 00000000		00000BE3	912 MSGSAVE	DC	3F' 0'	Registers save area
00000BD4	D200 89E3 1000		00000000	913 MSGMVC	M/C	MSGMSG(0) , 0(R1)	Executed instruction
00000BDA	D4E2C7D5 D6C8405C			915 MSGCMD	DC	C' MSGNOH * '	*** HERCULES MESSAGE COMMAND ***
00000BE3	40404040 40404040			916 MSGMSG	DC	CL95' '	The message text to be displayed

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				918 ****	*****
				919 *	Normal completion or Abnormal termination PSWs
				920 ****	*****
00000C48	00020001 80000000			922 EOJPSW DC OD' 0' , X' 0002000180000000' , AD(0)	
00000C58	B2B2 8A48	00000C48	924 EOJ LPSWE EOJPSW		Normal completion
00000C60	00020001 80000000			926 FAILPSW DC OD' 0' , X' 0002000180000000' , AD(X' BAD')	
00000C70	B2B2 8A60	00000C60	928 FAILTEST LPSWE FAILPSW		Abnormal termination

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				930 ****	*****
				931 * Working Storage	
				932 ****	*****
00000C74				934 LTORG ,	Literals pool
00000C74	00000000			935 =F' 0'	
00000C78	00000001			936 =F' 1'	
00000C7C	0000			937 =H' 0'	
00000C7E	005F			938 =AL2(L' MSGMSG)	
00000C80	C3E4F1F2 40			939 =CL5' CU12'	
00000C85	04294967 296C			940 =P' 4294967296'	
		00000400	00000001	942 K EQU 1024	One KB
		00001000	00000001	943 PAGE EQU (4*K)	Size of one page
		00004000	00000001	944 K16 EQU (16*K)	16 KB
		00008000	00000001	945 K32 EQU (32*K)	32 KB
		00010000	00000001	946 K64 EQU (64*K)	64 KB
		00100000	00000001	947 MB EQU (K*K)	1 MB
00000C8C	00002710			949 NUMLOOPS DC F' 10000'	10,000 * 100 = 1,000,000
00000C90	BBBBBBBB BBBBCCCC			951 BEGCLOCK DC OD' 0' , 8X' BB'	Begin
00000C98	EEEEEEEE EEEEEE			952 ENDCLOCK DC OD' 0' , 8X' EE'	End
00000CA0	DDDDDDDD DDDDDDDD			953 DURATION DC OD' 0' , 8X' DD'	Diff
00000CA8	FFFFFFFF FFFFFFFF			954 OVERHEAD DC OD' 0' , 8X' FF'	Overhead
00000CB0	00000000 0000000C			956 TICKSAAA DC PL8' 0'	Clock ticks high part
00000CB8	00000000 0000000C			957 TICKSBBB DC PL8' 0'	Clock ticks low part
00000CC0	00000000 0000000C			958 TICKSTOT DC PL8' 0'	Total clock ticks
00000CC8	40404040 40404040			960 PRTLINE DC C' 1,000,000 iterations of XXXXX'	
00000CEE	40A39696 9240F9F9			961 DC C' took 999,999,999 microseconds'	
00000DOC	40202020 6B202020	00000044	00000001	962 PRTLNG EQU *- PRTLINE	
				963 EDIT DC X' 402020206B2020206B202120'	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				967 ****	
				968 * CU12TEST DSECT	
				969 ****	
00000000 00				971 CU12PERF DSECT ,	
00000001 00				972 TNUM DC X'00'	CU12 test number
00000002 00				973 DC X'00'	
00000003 00				974 DC X'00'	
				975 MB DC X'00'	MB byte stored into CU12 instruction
00000004 00000000				977	
00000008 00000000				978 OP1DATA DC A(0)	Pointer to Operand 1 - result
0000000C 00000000				979 OP1LEN DC F'0'	length - result
00000010 00000000				980 OP2DATA DC A(0)	Pointer to Operand-2 data - source
				981 OP2LEN DC F'0'	length - source
00000014 00000000		00000014	00000001	983 OPSWHERE EQU *	
00000018 00000000				984 OP1WHERE DC A(0)	result - Where should be placed
				985 OP2WHERE DC A(0)	source - Where should be placed
0000001C 00000000				987 FAILMASK DC A(0)	Failure Branch on Condition mask
00000020 00000000				989 *	
00000024 00000000				990 ENDLN1 DC A(0)	Ending register values
				991 ENDLN2 DC A(0)	target length
				992	source length
00000028 00000001		00000001		994 CU12NEXT EQU *	Start of next table entry...

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
		00000D18	00004425	997 CU12TST CSECT ,
00000D18				999 ****CU12 Testing Control tables (ref: CU12TEST DSECT) 1000 * CU12 Testing Control tables (ref: CU12TEST DSECT) 1001 ***** 1002 PRINT DATA 1003 CU12CTL DC 0A(0) start of table 1004 ***** 1005 * tests with CC=0 MB=0 1006 *****
00000D18				1008 CCOT1 DS OF 1009 DC X'01' 1010 DC X'00', X'00' 1011 DC X'00' 1012 *
00000D1C	00000E38 00000044			1013 DC A(UTF16A), A(UTF16AED-UTF16A) 1014 DC A(UTF8A), A(UTF8AEND-UTF8A)
00000D24	00000DF0 0000003D			target - 0p1 & length Source - 0p2 & length
00000D2C	00400000			1015 1016 DC A(4*MB+(0*K16)) 1017 DC A(2*MB+(0*K16))
00000D30	00200000			target source
00000D34	00000007			1018 * 1019 DC A(7) 1020 DC A(0) 1021 DC A(0)
00000D38	00000000			FailCC - not CC0 Result - target len
00000D3C	00000000			Result - source len
00000D40				1023 CCOT2 DS OF 1024 DC X'02' 1025 DC X'00', X'00' 1026 DC X'00' 1027 *
00000D44	00000E38 00000044			1028 DC A(UTF16A), A(UTF16AED-UTF16A) 1029 DC A(UTF8A), A(UTF8AEND-UTF8A)
00000D4C	00000DF0 0000003D			target - 0p1 & length Source - 0p2 & length
00000D54	0010C000			1030 1031 DC A(1*MB+(3*K16)) 1032 DC A(2*MB+(5*K16)-23)
00000D58	00213FE9			target source
00000D5C	00000007			1033 * 1034 DC A(7) 1035 DC A(0) 1036 DC A(0)
00000D60	00000000			FailCC - not CC0 Result - target len
00000D64	00000000			Result - source len
00000D68				1038 CCOT3 DS OF 1039 DC X'03' 1040 DC X'00', X'00' 1041 DC X'00' 1042 *
00000D6C	00000E38 00000044			1043 DC A(UTF16A), A(UTF16AED-UTF16A) 1044 DC A(UTF8A), A(UTF8AEND-UTF8A)
00000D74	00000DF0 0000003D			target - 0p1 & length Source - 0p2 & length
00000D7C	0011BFE9			1045 1046 DC A(1*MB+(7*K16)-23) 1047 DC A(2*MB+(9*K16))
00000D80	00224000			target source
1048 *				

LOC	OBJECT CODE	ADDR1	ADDR2	STMT			
00000D84	00000007			1049	DC	A(7)	Fail CC - not CCO
00000D88	00000000			1050	DC	A(0)	Result - target len
00000D8C	00000000			1051	DC	A(0)	Result - source len
00000D90				1053 CCOT4	DS	OF	
00000D90	04			1054	DC	X'04'	Test Num
00000D91	0000			1055	DC	X'00', X'00'	
00000D93	00			1056	DC	X'00'	MB
				1057 *			
00000D94	00000E38 00000044			1058	DC	A(UTF16A), A(UTF16AED-UTF16A)	target - 0p1 & length
00000D9C	00000DF0 0000003D			1059	DC	A(UTF8A), A(UTF8AEND-UTF8A)	Source - 0p2 & length
				1060			
00000DA4	0012BFE9			1061	DC	A(1*MB+(11*K16)-23)	target
00000DA8	00233FE9			1062	DC	A(2*MB+(13*K16)-23)	source
				1063 *			
00000DAC	00000007			1064	DC	A(7)	Fail CC - not CCO
00000DB0	00000000			1065	DC	A(0)	Result - target len
00000DB4	00000000			1066	DC	A(0)	Result - source len
				1068 *	DC	A(0)	end of table
				1069 *	DC	A(0)	end of table
				1070 *	DC	A(0)	end of table
				1071			
				1072			
00000DB8				1073 CCOT5	DS	OF	
00000DB8	05			1074	DC	X'05'	Test Num
00000DB9	0000			1075	DC	X'00', X'00'	
00000DBB	00			1076	DC	X'00'	MB
				1077 *			
00000DBC	00000E38 000FC000			1078	DC	A(UTF16A), A(1*MB-16*K)	target - 0p1 & length
00000DC4	00000E7C 000035AA			1079	DC	A(UTF8B), A(UTF8BEND-UTF8B)	Source - 0p2 & length
				1080			
00000DCC	0031FF01			1081	DC	A(3*MB+(8*K16)-(255))	target
00000DD0	002FFF01			1082	DC	A(3*MB+(0*K16)-(255))	source
				1083 *			
00000DD4	00000007			1084	DC	A(7)	Fail CC - not CCO
00000DD8	00000000			1085	DC	A(0)	Result - target len
00000DDC	00000000			1086	DC	A(0)	Result - source len
				1088	DC	A(0)	end of table
00000DE0	00000000			1089	DC	A(0)	end of table
00000DE4	00000000			1090	DC	A(0)	end of table

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				1092 ****= 1093 * CU12 UTF-8 tests 1094 ****=
00000DEC	0000003D		1096	UTF8ALN DC A(UTF8AEND-UTF8A)
00000DF0			1097	UTF8A DS OH
00000DF0	00		1098	DC XL1' 00' first UTF-8 1 Byte character
00000DF1	31		1099	DC XL1' 31' 1
00000DF2	39		1100	DC XL1' 39' 9
00000DF3	40		1101	DC XL1' 40' @
00000DF4	41		1102	DC XL1' 41' A
00000DF5	42		1103	DC XL1' 42' B
00000DF6	7F		1104	DC XL1' 7F' last UTF-8 1 Byte character
00000DF7	C280		1106	DC XL2' C280' first UTF-8 2 Byte character
00000DF9	C380		1107	DC XL2' C380' c3 80 LATIN CAPITAL LETTER A WITH GRAVE
00000DFB	C3B8		1108	DC XL2' C3B8' c3 b8 LATIN SMALL LETTER O WITH STROKE
00000DFD	D09C		1109	DC XL2' D09C' D0 9C Đœ Cyrillic Capital Letter Em
00000DFF	DFBF		1110	DC XL2' DFBF' last UTF-8 2 Byte character DF BF Đć
00000E01	43		1112	DC XL1' 43' C
00000E02	EOA080		1114	DC XL3' EOA080' first UTF-8 3 Byte character
			1115 *	E0 A0 80 à € Samaritan Letter Alaf
00000E05	EOA18D		1116	DC XL3' EOA18D' E0 A1 8D à ā Mandaic Letter An
00000E08	EA9FBD		1117	DC XL3' EA9FBD' EA 9F BD êY½ Latin Epigraphic Inverted M
00000E0B	EFBF87		1118	DC XL3' EFbf87' EF BF 87 i᷑ Hal fwidht Hangul Letter E
00000E0E	EFBFBF		1119	DC XL3' EFBFBF' last UTF-8 3 Byte character EF BF BF
00000E11	44		1121	DC XL1' 44' D
00000E12	F0908080		1123	DC XL4' F0908080' first UTF-8 4 Byte character
			1124 *	F0 90 80 80 ð•€€ Linear B Syllable B008 A
00000E16	F0908487		1125	DC XL4' F0908487' F0 90 84 87 ð•„‡ Aegean Number One
00000E1A	F09294B5		1126	DC XL4' F09294B5' F0 92 94 B5 Cuneiform Sign She Plus Sar
00000E1E	F09082B8		1127	DC XL4' F09082B8' F0 90 82 B8 ð•, Linear B Ideogram B177
00000E22	F096AB83		1128	DC XL4' F096AB83' F0 96 A8 83 ð-„ƒ Bamum Letter Phase-f Ka
00000E26	F0989A9F		1129	DC XL4' F0989A9F' last UTF-8 4 Byte character
00000E2A	45		1131	DC XL1' 45' E
00000E2B	4E		1132	DC XL1' 4E' N
00000E2C	44		1133	DC XL1' 44' D
00000E2D			1134	UTF8AEND DS 0X
			1135	
			1137	****=
			1138 *	CU12 UTF-12 Result
			1139	****=
00000E2D	E4C6E3F3 F27A		1140	DC C' UFT32: '
00000E34	00000044		1141	UTF16ALN DC A(UTF16AED-UTF16A)
00000E38			1142	UTF16A DC 0X
00000E38	0000		1143	DC X' 0000'
00000E3A	0031		1144	DC X' 0031'
00000E3C	0039		1145	DC X' 0039'
00000E3E	0040		1146	DC X' 0040'
00000E40	0041		1147	DC X' 0041'

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
00000E42	0042		1148	DC	X' 0042'
00000E44	007F		1149	DC	X' 007F'
00000E46	0080		1150	DC	X' 0080'
00000E48	00C0		1151	DC	X' 00C0'
00000E4A	00F8		1152	DC	X' 00F8'
00000E4C	041C		1153	DC	X' 041C'
00000E4E	07FF		1154	DC	X' 07FF'
00000E50	0043		1155	DC	X' 0043'
00000E52	0800		1156	DC	X' 0800'
00000E54	084D		1157	DC	X' 084D'
00000E56	A7FD		1158	DC	X' A7FD'
00000E58	FFC7		1159	DC	X' FFC7'
00000E5A	FFFF		1160	DC	X' FFFF'
00000E5C	0044		1161	DC	X' 0044'
		1162 *			utf16 pairs
00000E5E	D800		1163	DC	X' D800'
00000E60	DC00		1164	DC	X' DC00'
00000E62	D800		1165	DC	X' D800'
00000E64	DD07		1166	DC	X' DD07'
00000E66	D809		1167	DC	X' D809'
00000E68	DD35		1168	DC	X' DD35'
00000E6A	D800		1169	DC	X' D800'
00000E6C	DCB8		1170	DC	X' DCB8'
00000E6E	D81A		1171	DC	X' D81A'
00000E70	DEC3		1172	DC	X' DEC3'
00000E72	D821		1173	DC	X' D821'
00000E74	DE9F		1174	DC	X' DE9F'
		1175			
00000E76	0045		1176	DC	X' 0045'
00000E78	004E		1177	DC	X' 004E'
00000E7A	0044		1178	DC	X' 0044'
00000E7C		1179	UTF16AED	DS	0X

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				1181 **** 1182 * UTF-8 LONG LONG LONG String (CC=3 result) 1183 ****
00000E7C				1185 UTF8B DS OF
00000E7C	0A			1186 DC x'0a'
00000E7D	3C			1187 DC x'3c'
00000E7E	21			1188 DC x'21'
00000E7F	44			1189 DC x'44'
00000E80	4F			1190 DC x'4f'
00000E81	43			1191 DC x'43'
00000E82	54			1192 DC x'54'
00000E83	59			1193 DC x'59'
00000E84	50			1194 DC x'50' 1195 * ETC.
				1197 PRINT OFF 14917 PRINT ON 14918 * ETC. 14919
0000441B	79			14920 DC x'79'
0000441C	3E			14921 DC x'3e'
0000441D	0A			14922 DC x'0a'
0000441E	3C			14923 DC x'3c'
0000441F	2F			14924 DC x'2f'
00004420	68			14925 DC x'68'
00004421	74			14926 DC x'74'
00004422	6D			14927 DC x'6d'
00004423	6C			14928 DC x'6c'
00004424	3E			14929 DC x'3e'
00004425	0A			14930 DC x'0a'
00004426				14931 UTF8BEND DS 0C
				14933 **** 14934 * Register equates 14935 ****
		00000000	00000001	14937 R0 EQU 0
		00000001	00000001	14938 R1 EQU 1
		00000002	00000001	14939 R2 EQU 2
		00000003	00000001	14940 R3 EQU 3
		00000004	00000001	14941 R4 EQU 4
		00000005	00000001	14942 R5 EQU 5
		00000006	00000001	14943 R6 EQU 6
		00000007	00000001	14944 R7 EQU 7
		00000008	00000001	14945 R8 EQU 8
		00000009	00000001	14946 R9 EQU 9
		0000000A	00000001	14947 R10 EQU 10
		0000000B	00000001	14948 R11 EQU 11
		0000000C	00000001	14949 R12 EQU 12
		0000000D	00000001	14950 R13 EQU 13

LOC	OBJECT CODE	ADDR1	ADDR2	STMT		
	0000000E	00000001	14951 R14	EQU	14	
	0000000F	00000001	14952 R15	EQU	15	

14954 ENR

CU12-02-performance (Test CU12 instructions)											12 Feb 2024	12:47:12	Page	22		
	SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES										
					497	500	503	506	509	512	515	518	521	524	527	
					530	533	536	539	542	545	548	551	554	557	560	
					563	566	569	572	575	578	581	584	587	590	593	
					596	599	602	605	608	611	614	617	620	623	626	
					629	632	635	638	641	644	647	650	653	656	659	
					662	665	668	671	674	677	680	683	686	689	692	
					695	698	701	704	707	710	713	716	719	722	725	
					728	731	734	737	740	743	746	749	752	755	758	
					761	767	770									
OPSWHERE	U	000014	1	983												
OVERHEAD	D	000CA8	8	954	453	799										
PAGE	U	001000	1	943												
PRTLINE	C	000CC8	38	960	962	777	814	815	822							
PRTLNG	U	000044	1	962	821											
R0	U	000000	1	14937	83	217	221	224	228	242	244	251	253	255	257	
					259	261	263	265	267	269	271	273	275	277	279	
					281	283	285	287	289	291	293	295	297	299	301	
					303	305	307	309	311	313	315	317	319	321	323	
					325	327	329	331	333	335	337	339	341	343	345	
					347	349	351	353	355	357	359	361	363	365	367	
					369	371	373	375	377	379	381	383	385	387	389	
					391	393	395	397	399	401	403	405	407	409	411	
					413	415	417	419	421	423	425	427	429	431	433	
					435	437	439	441	446	448	465	466	468	469	476	
					477	479	480	482	483	485	486	488	489	491	492	
					494	495	497	498	500	501	503	504	506	507	509	
					510	512	513	515	516	518	519	521	522	524	525	
					527	528	530	531	533	534	536	537	539	540	542	
					543	545	546	548	549	551	552	554	555	557	558	
					560	561	563	564	566	567	569	570	572	573	575	
					576	578	579	581	582	584	585	587	588	590	591	
					593	594	596	597	599	600	602	603	605	606	608	
					609	611	612	614	615	617	618	620	621	623	624	
					626	627	629	630	632	633	635	636	638	639	641	
					642	644	645	647	648	650	651	653	654	656	657	
					659	660	662	663	665	666	668	669	671	672	674	
					675	677	678	680	681	683	684	686	687	689	690	
					692	693	695	696	698	699	701	702	704	705	707	
					708	710	711	713	714	716	717	719	720	722	723	
					725	726	728	729	731	732	734	735	737	738	740	
					741	743	744	746	747	749	750	752	753	755	756	
					758	759	761	762	767	768	770	771	820	821	824	
	R1	U	000001	1	14938	218	225	822	869	871	876	879	903	913		
	R10	U	00000A	1	14947	804	805	807								
	R11	U	00000B	1	14948	804	808									
	R12	U	00000C	1	14949											
	R13	U	00000D	1	14950	205	206	209	782	783	784					
	R14	U	00000E	1	14951	130	203	786								
	R15	U	00000F	1	14952	452	778	794	797	802	827	828	839	855	858	859
	R2	U	000002	1	14939	219	221	226	466	469	477	480	483	486	489	492
					495	498	501	504	507	510	513	516	519	522	525	
					528	531	534	537	540	543	546	549	552	555	558	
					561	564	567	570	573	576	579	582	585	588	591	
					594	597	600	603	606	609	612	615	618	621	624	

ASMA Ver. 0.7.0	CU12-02-performance (Test CU12 instructions)								12 Feb 2024 12:47:12				Page	23	
	SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES									
					627	630	633	636	639	642	645	648	651	654	657
					660	663	666	669	672	675	678	681	684	687	690
					693	696	699	702	705	708	711	714	717	720	723
					726	729	732	735	738	741	744	747	750	753	756
					759	762	768	771	820	823	824	871	873	890	892
R3	U	000003	1	14940	220	227	228	237	242	244	251	253	255	257	259
					261	263	265	267	269	271	273	275	277	279	281
					283	285	287	289	291	293	295	297	299	301	303
					305	307	309	311	313	315	317	319	321	323	325
					327	329	331	333	335	337	339	341	343	345	347
					349	351	353	355	357	359	361	363	365	367	369
					371	373	375	377	379	381	383	385	387	389	391
					393	395	397	399	401	403	405	407	409	411	413
					415	417	419	421	423	425	427	429	431	433	435
					437	439	441	446	448	465	468	476	479	482	485
					488	491	494	497	500	503	506	509	512	515	518
					521	524	527	530	533	536	539	542	545	548	551
					554	557	560	563	566	569	572	575	578	581	584
					587	590	593	596	599	602	605	608	611	614	617
					620	623	626	629	632	635	638	641	644	647	650
					653	656	659	662	665	668	671	674	677	680	683
					686	689	692	695	698	701	704	707	710	713	716
R4	U	000004	1	14941	869	872	873	877	879	879	879	879	879	879	879
R5	U	000005	1	14942	237	776	795	799	826	840	852	857	871	871	871
R6	U	000006	1	14943	211	212	238	450	461	773	800	842	843	844	845
R7	U	000007	1	14944	235	450	459	773	801	840	842	845	847	850	854
R8	U	000008	1	14945	117	120	121	122	122	124					
R9	U	000009	1	14946	118	124	124	125							
RPTDWSAV	D	000AF8	8	833	820	824									
RPTSAVE	F	000AF0	4	830	794	827									
RPTSPEED	I	000A88	4	794	778										
RPTSVR5	F	000AF4	4	831	795	826									
SAVE3T5	F	000410	4	161	237	776									
SAVER13	F	000424	4	163	209	782									
SAVER2	F	000420	4	162											
SUBDWORD	I	000B5C	4	869	802	855									
SUBDWSAV	D	000B80	8	882	869	879									
SUBTEST	X	000401	1	155	142										
TEST91	I	000528	4	202	130										
TESTADDR	D	000400	8	153											
TESTNUM	X	000400	1	154	139	212									
TICKSAAA	P	000CB0	8	956	807	810									
TICKSBBB	P	000CB8	8	957	808	812									
TICKSTOT	P	000CC0	8	958	810	811	812	812	815						
TIMEOPT	X	000408	1	158	136	202									
TNUM	X	000000	1	972	211										
TST91LOP	U	000532	1	208	785										
UTF16A	X	000E38	1	1142	1013	1028	1043	1058	1078	1141					
UTF16AED	X	000E7C	1	1179	1013	1028	1043	1058	1141						
UTF16ALN	A	000E34	4	1141											
UTF8A	H	000DF0	2	1097	1014	1029	1044	1059	1096						

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES
UTF8AEND	X	000E2D	1	1134	1014 1029 1044 1059 1096
UTF8ALN	A	000DEC	4	1096	
UTF8B	F	000E7C	4	1185	1079
UTF8BEND	C	004426	1	14931	1079
=AL2(L' MSGMSG)	R	000C7E	2	938	894
=CL5' CU12'	C	000C80	5	939	777
=F' 0'	F	000C74	4	935	784
=F' 1'	F	000C78	4	936	875
=H' 0'	H	000C7C	2	937	889
=P' 4294967296'	P	000C85	6	940	811

MACRO DEFN REFERENCES

DOINSTR	185	463	474	765
OVERONLY	172	240	249	444

DESC	SYMBOL	SIZE	POS	ADDR
Entry: 0				
Image	IMAGE	17446	0000-4425	0000-4425
Region		17446	0000-4425	0000-4425
CSECT	CU12TST	3352	0000-0D17	0000-0D17
CSECT	CU12TST	14094	0D18-4425	0D18-4425

STM **FILE NAME**

1 /devstor/dev/tests/. /CU12-02-performance.asm

** NO ERRORS FOUND **