

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

```
2 *****
3 *
4 *      CLCE  instruction tests
5 *
6 *      NOTE: This is a copy of the CLCL-et-al Test
7 *             modified to only test the CLCLE instruction.
8 *             Specifically, instuction
9 *
10 *      CLCL  R10,R12
11 *
12 *      was changed to
13 *
14 *      CLCLE R10,R12,0
15 *      BC    B'0001',*-4      not finished?
16 *
17 *
18 *      James Wekel August 2022
19 *****
20 *****
21 *
22 *  This program tests proper functioning of the CLCLE instructions.
23 *
24 *  PLEASE NOTE that the tests are very SIMPLE TESTS designed to catch
25 *  obvious coding errors.  None of the tests are thorough.  They are
26 *  NOT designed to test all aspects of any of the instructions.
27 *
28 *****
29 *
30 *  Example Hercules Testcase:
31 *
32 *
33 *      *Testcase CLCE-03-basic (Test CLCLE instructions)
34 *
35 *      archlvl      390
36 *      mainsize     3
37 *      numcpu        1
38 *      sysclear
39 *
40 *      loadcore      "$(testpath)/CLCLE-03-basic.core" 0x0
41 *
42 *      runtest       1
43 *      *Done
44 *
45 *****
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				47 PRINT OFF
				3428 PRINT ON
				3430 *****
				3431 * SATK prolog stuff...
				3432 *****
				3434 ARCHLVL ZARCH=NO,MNOTE=NO
				3436+\$AL OPSYN AL
				3437+\$ALR OPSYN ALR
				3438+\$B OPSYN B
				3439+\$BAS OPSYN BAS
				3440+\$BASR OPSYN BASR
				3441+\$BC OPSYN BC
				3442+\$BCTR OPSYN BCTR
				3443+\$BE OPSYN BE
				3444+\$BH OPSYN BH
				3445+\$BL OPSYN BL
				3446+\$BM OPSYN BM
				3447+\$BNE OPSYN BNE
				3448+\$BNH OPSYN BNH
				3449+\$BNL OPSYN BNL
				3450+\$BNM OPSYN BNM
				3451+\$BNO OPSYN BNO
				3452+\$BNP OPSYN BNP
				3453+\$BNZ OPSYN BNZ
				3454+\$BO OPSYN BO
				3455+\$BP OPSYN BP
				3456+\$BXLE OPSYN BXLE
				3457+\$BZ OPSYN BZ
				3458+\$CH OPSYN CH
				3459+\$L OPSYN L
				3460+\$LH OPSYN LH
				3461+\$LM OPSYN LM
				3462+\$LPSW OPSYN LPSW
				3463+\$LR OPSYN LR
				3464+\$LTR OPSYN LTR
				3465+\$NR OPSYN NR
				3466+\$SL OPSYN SL
				3467+\$SLR OPSYN SLR
				3468+\$SR OPSYN SR
				3469+\$ST OPSYN ST
				3470+\$STM OPSYN STM
				3471+\$X OPSYN X
				3472+\$AHI OPSYN AHI
				3473+\$B OPSYN J
				3474+\$BC OPSYN BRC
				3475+\$BE OPSYN JE
				3476+\$BH OPSYN JH
				3477+\$BL OPSYN JL
				3478+\$BM OPSYN JM
				3479+\$BNE OPSYN JNE
				3480+\$BNH OPSYN JNH
				3481+\$BNL OPSYN JNL
				3482+\$BNM OPSYN JNM
				3483+\$BNO OPSYN JNO

LOC	OBJECT CODE	ADDR1	ADDR2	STMT
				3492 *****
				3493 * Initiate the CLCLE03 CSECT in the CODE region
				3494 * with the location counter at 0
				3495 *****
				3497 CLCLE03 ASALOAD REGION=CODE
		00000000	00003000	3498+CLCLE03 START 0, CODE
00000000	000A0000	00000008		3500+ PSW 0,0,2,0,X'008' 64-bit Restart ISR Trap New PSW
00000008		00000008	00000058	3501+ ORG CLCLE03+X'058'
00000058	000A0000	00000018		3503+ PSW 0,0,2,0,X'018' 64-bit External ISR Trap New PSW
00000060	000A0000	00000020		3504+ PSW 0,0,2,0,X'020' 64-bit Supervisor Call ISR Trap New PSW
00000068	000A0000	00000028		3505+ PSW 0,0,2,0,X'028' 64-bit Program ISR Trap New PSW
00000070	000A0000	00000030		3506+ PSW 0,0,2,0,X'030' 64-bit Machine Check Trap New PSW
00000078	000A0000	00000038		3507+ PSW 0,0,2,0,X'038' 64-bit Input/Output Trap New PSW
00000080		00000080	00000200	3508+ ORG CLCLE03+512
				3510 *****
				3511 * Create IPL (restart) PSW
				3512 *****
				3514 ASAIPL IA-BEGIN
		00000000	00003000	3515+CLCLE03 CSECT
00000200		00000200	00000000	3516+ ORG CLCLE03
00000000	00080000	00000200		3517+ PSW 0,0,0,0,BEGIN,24
00000008		00000008	00000200	3518+ ORG CLCLE03+512 Reset CSECT to end of assigned storage area
		00000000	00003000	3519+CLCLE03 CSECT

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				3521 *****	
				3522 *	The actual "CLCLE03" program itself...
				3523 *****	
				3524 *	
				3525 *	Architecture Mode: 390
				3526 *	Addressing Mode: 31-bit
				3527 *	Register Usage:
				3528 *	
				3529 *	R0 (work)
				3530 *	R1
				3531 *	R2 First base register
				3532 *	R3
				3533 *	R4
				3534 *	R5-R7 (work)
				3535 *	R8
				3536 *	R9 Second base register
				3537 *	R10-R13 (work)
				3538 *	R14 Subroutine call
				3539 *	R15 Secondary Subroutine call or work
				3540 *	
				3541 *****	
00000200		00000000		3543	USING ASA,R0 Low core addressability
00000200		00000200		3544	USING BEGIN,R2 FIRST Base Register
00000200		00001200		3545	USING BEGIN+4096,R9 SECOND Base Register
00000200	0520			3547 BEGIN	BALR R2,0 Initalize FIRST base register
00000202	0620			3548	BCTR R2,0 Initalize FIRST base register
00000204	0620			3549	BCTR R2,0 Initalize FIRST base register
00000206	4190 2800		00000800	3551	LA R9,2048(,R2) Initalize SECOND base register
0000020A	4190 9800		00000800	3552	LA R9,2048(,R9) Initalize SECOND base register
				3554 *	
				3555 **	Run the tests...
				3556 *	
0000020E	45E0 202A		0000022A	3557	BAL R14,TEST01 Test CLCLE instruction
				3558 *	
00000212	45E0 2134		00000334	3559	BAL R14,TEST91 Test CLCLE page fault handling

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT		
					3561	*****	
					3562	* Test for normal or unexpected test completion...	
					3563	*****	
00000216	9591	9FFE		000021FE	3565	CLI TESTNUM,X'91'	Did we end on expected test?
0000021A	4770	2298		00000498	3566	BNE FAILTEST	No?! Then FAIL the test!
0000021E	9510	9FFF		000021FF	3568	CLI SUBTEST,X'10'	Did we end on expected SUB-test?
00000222	4770	2298		00000498	3569	BNE FAILTEST	No?! Then FAIL the test!
00000226	47F0	228A		0000048A	3571	B EOJ	Yes, then normal completion!

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				3573 *****	
				3574 * TEST01 Test CLCLE instruction	
				3575 *****	
0000022A	9201 9FFE		000021FE	3577 TEST01 MVI TESTNUM,X'01'	
				3578 *	
				3579 ** Initialize test parameters...	
				3580 *	
0000022E	9856 2364		00000564	3581 LM R5,R6,CLCL4 CLCL4 test Op1 address and length	
00000232	1E56			3582 ALR R5,R6 Point past last byte	
00000234	0650			3583 BCTR R5,0 Backup to last byte	
00000236	92FF 5000		00000000	3584 MVI 0(R5),X'FF' Force unequal compare (op1 high)	
				3585 *	
0000023A	9856 2384		00000584	3586 LM R5,R6,CLCLOP1 (same thing for CLCLOP1 test)	
0000023E	1E56			3587 ALR R5,R6 "	
00000240	0650			3588 BCTR R5,0 "	
00000242	92FF 5000		00000000	3589 MVI 0(R5),X'FF' "	
				3590 *	
00000246	9856 237C		0000057C	3591 LM R5,R6,CLCL8+8 CLCL8 test ==> OP2 <==	
0000024A	1E56			3592 ALR R5,R6	
0000024C	0650			3593 BCTR R5,0	
0000024E	92FF 5000		00000000	3594 MVI 0(R5),X'FF' ==> OPERAND-2 high (OP1 LOW) <==	
				3595 *	
				3596 ** Neither cross (one byte)	
				3597 *	
00000252	9201 9FFF		000021FF	3598 MVI SUBTEST,X'01'	
00000256	98AD 2304		00000504	3599 LM R10,R13,CLCL1	
0000025A	A9AC 0000		00000000	3600 CLCLE R10,R12,0	
0000025E	4710 205A		0000025A	3601 BC B'0001',*-4 not finished?	
00000262	4770 2298		00000498	3602 BNE FAILTEST	
00000266	4150 23A4		000005A4	3603 LA R5,ECLCL1	
0000026A	45F0 227A		0000047A	3604 BAL R15,ENDCLCL	
				3605 *	
				3606 ** Neither cross (two bytes)	
				3607 *	
0000026E	9202 9FFF		000021FF	3608 MVI SUBTEST,X'02'	
00000272	98AD 2314		00000514	3609 LM R10,R13,CLCL2	
00000276	A9AC 0000		00000000	3610 CLCLE R10,R12,0	
0000027A	4710 2076		00000276	3611 BC B'0001',*-4 not finished?	
0000027E	4770 2298		00000498	3612 BNE FAILTEST	
00000282	4150 23B4		000005B4	3613 LA R5,ECLCL2	
00000286	45F0 227A		0000047A	3614 BAL R15,ENDCLCL	
				3615 *	
				3616 ** Neither cross (four bytes)	
				3617 ** (inequality on last byte of op1)	
				3618 *	
0000028A	9204 9FFF		000021FF	3619 MVI SUBTEST,X'04'	
0000028E	98AD 2364		00000564	3620 LM R10,R13,CLCL4	
00000292	A9AC 0000		00000000	3621 CLCLE R10,R12,0	
00000296	4710 2092		00000292	3622 BC B'0001',*-4 not finished?	
0000029A	47D0 2298		00000498	3623 BNH FAILTEST (see INIT; CLCL4: op1 > op2)	
0000029E	4150 2404		00000604	3624 LA R5,ECLCL4	
000002A2	45F0 227A		0000047A	3625 BAL R15,ENDCLCL	

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT
					3627 *
					3628 ** Neither cross (eight bytes)
					3629 ** (inequality on last byte of op2)
					3630 *
000002A6	9208	9FFF		000021FF	3631 MVI SUBTEST,X'08'
000002AA	98AD	2374		00000574	3632 LM R10,R13,CLCL8
000002AE	A9AC	0000		00000000	3633 CLCLE R10,R12,0
000002B2	4710	20AE		000002AE	3634 BC B'0001',*-4 not finished?
000002B6	47B0	2298		00000498	3635 BNL FAILTEST (see INIT; CLCL8: op1 < op2)
000002BA	4150	2414		00000614	3636 LA R5,ECLCL8
000002BE	45F0	227A		0000047A	3637 BAL R15,ENDCLCL
					3638 *
					3639 ** Neither cross (1K bytes)
					3640 *
000002C2	9200	9FFF		000021FF	3641 MVI SUBTEST,X'00'
000002C6	98AD	2334		00000534	3642 LM R10,R13,CLCL1K
000002CA	A9AC	0000		00000000	3643 CLCLE R10,R12,0
000002CE	4710	20CA		000002CA	3644 BC B'0001',*-4 not finished?
000002D2	4770	2298		00000498	3645 BNE FAILTEST
000002D6	4150	23D4		000005D4	3646 LA R5,ECLCL1K
000002DA	45F0	227A		0000047A	3647 BAL R15,ENDCLCL
					3648 *
					3649 ** Both cross
					3650 *
000002DE	9222	9FFF		000021FF	3651 MVI SUBTEST,X'22'
000002E2	98AD	2344		00000544	3652 LM R10,R13,CLCLBOTH
000002E6	A9AC	0000		00000000	3653 CLCLE R10,R12,0
000002EA	4710	20E6		000002E6	3654 BC B'0001',*-4 not finished?
000002EE	4770	2298		00000498	3655 BNE FAILTEST
000002F2	4150	23E4		000005E4	3656 LA R5,ECLCLBTH
000002F6	45F0	227A		0000047A	3657 BAL R15,ENDCLCL
					3658 *
					3659 ** Only op1 crosses
					3660 ** (inequality on last byte of op1)
					3661 *
000002FA	9210	9FFF		000021FF	3662 MVI SUBTEST,X'10'
000002FE	98AD	2384		00000584	3663 LM R10,R13,CLCLOP1
00000302	A9AC	0000		00000000	3664 CLCLE R10,R12,0
00000306	4710	2102		00000302	3665 BC B'0001',*-4 not finished?
0000030A	47D0	2298		00000498	3666 BNH FAILTEST (see INIT; CLCLOP1: op1 > op2)
0000030E	4150	2424		00000624	3667 LA R5,ECLCLOP1
00000312	45F0	227A		0000047A	3668 BAL R15,ENDCLCL
					3669 *
					3670 ** Only op2 crosses
					3671 *
00000316	9220	9FFF		000021FF	3672 MVI SUBTEST,X'20'
0000031A	98AD	2354		00000554	3673 LM R10,R13,CLCLOP2
0000031E	A9AC	0000		00000000	3674 CLCLE R10,R12,0
00000322	4710	211E		0000031E	3675 BC B'0001',*-4 not finished?
00000326	4770	2298		00000498	3676 BNE FAILTEST
0000032A	4150	23F4		000005F4	3677 LA R5,ECLCLOP2
0000032E	45F0	227A		0000047A	3678 BAL R15,ENDCLCL
					3679 *
00000332	07FE				3680 BR R14

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				3682 *****	
				3683 * TEST91	Test CLCLE page fault handling
				3684 *****	
00000334	9291 9FFE		000021FE	3686 TEST91 MVI TESTNUM,X'91'	
00000338	9200 9FFF		000021FF	3687 MVI SUBTEST,X'00'	
				3688 *	
				3689 ** First, make sure we start clean!	
				3690 *	
0000033C	98AD 2394		00000594	3691 LM R10,R13,CLCLPF	Retrieve CLCLE PF test parameters
00000340	0EAC			3692 MVCL R10,R12	(forces full comparison)
				3693 *	
				3694 ** Initialize Dynamic Address Translation tables...	
				3695 *	
00000342	58A0 22A8		000004A8	3696 L R10,=A(SEGTABLS)	Segment Tables Origin
00000346	41B0 0020		00000020	3697 LA R11,NUMPGTBS	Number of Segment Table Entries
0000034A	58C0 22AC		000004AC	3698 L R12,=A(PAGETABS)	Page Tables Origin
0000034E	1F00			3699 SLR R0,R0	First Page Frame Address
00000350	4160 0004		00000004	3700 LA R6,4	Size of one table entry
00000354	5870 22B0		000004B0	3701 L R7,=A(PAGE)	Size of one Page Frame
00000358	50C0 A000		00000000	3703 SEGLLOOP ST R12,0(,R10)	Seg Table Entry <= Page Table Origin
0000035C	960F A003		00000003	3704 OI 3(R10),X'0F'	Seg Table Entry <= Page Table Length
00000360	1EA6			3705 ALR R10,R6	Bump to next Segment Table Entry
00000362	41D0 0010		00000010	3707 LA R13,16	Page Table Entries per Page Table
00000366	5000 C000		00000000	3708 PAGELLOOP ST R0,0(,R12)	Page Table Entry = Page Frame Address
0000036A	1E07			3709 ALR R0,R7	Increment to next Page Frame Address
0000036C	1EC6			3710 ALR R12,R6	Bump to next Page Table Entry
0000036E	46D0 2166		00000366	3711 BCT R13,PAGELLOOP	Loop until Page table is complete
00000372	46B0 2158		00000358	3713 BCT R11,SEGLLOOP	Loop until all
				3714 * Segment Table Entries built	
				3715 *	
				3716 ** Update desired page table entry to cause page fault	
				3717 *	
00000376	98AD 2394		00000594	3718 LM R10,R13,CLCLPF	Retrieve CLCLE PF test parameters
0000037A	185A			3719 LR R5,R10	R5 --> Operand-1
0000037C	5E50 22B4		000004B4	3720 AL R5,=A(PFPGBYTS)	R5 --> Operand-1 Page Fault address
00000380	1865			3721 LR R6,R5	R6 --> Address where PF should occur
00000382	8850 000C		0000000C	3722 SRL R5,12	R5 = Page Frame number
00000386	8950 0002		00000002	3723 SLL R5,2	R5 = Page Table Entry number
0000038A	9204 9FFF		000021FF	3725 MVI SUBTEST,X'04'	
0000038E	5E50 22AC		000004AC	3726 AL R5,=A(PAGETABS)	R5 --> Page Table Entry
00000392	9604 5002		00000002	3727 OI 2(R5),X'04'	Mark this page invalid

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	
					3729 *	
					3730 **	Install program check routine to catch the page fault
					3731 *	
00000396	9202	9FFF		000021FF	3732	MVI SUBTEST,X'02'
0000039A	D207	21D8 0068	000003D8	00000068	3733	MVC SVPGMNEW,PGMNPSW Save original Program New PSW
000003A0	4100	21E8		000003E8	3734	LA R0,MYPGMNEW Point to temporary Pgm New routine
000003A4	5000	006C		0000006C	3735	ST R0,PGMNPSW+4 Point Program New PSW to our routine
000003A8	9208	0069		00000069	3736	MVI PGMNPSW+1,X'08' Make it a non-disabled-wait PSW!
					3737 *	
					3738 **	Run the test: should cause a page fault
					3739 *	
000003AC	920F	9FFF		000021FF	3740	MVI SUBTEST,X'0F'
000003B0	B700	22BC		000004BC	3741	LCTL R0,R0,CRLREG0 Switch to DAT mode
000003B4	B711	22C0		000004C0	3742	LCTL R1,R1,CTLREG1 Switch to DAT mode
000003B8	8200	21E0		000003E0	3743	LPSW DATONPSW Switch to DAT mode
000003BC	4700	21BC		000003BC	3744	BEGDATON NOP * (pad)
000003C0	4700	21C0		000003C0	3745	NOP * (pad)
000003C4	B20D	0000		00000000	3746	PTLB , Purge Translation Lookaside Buffer
000003C8	A9AC	0000		00000000	3747	PFINSADR CLCLE R10,R12,0 Page Fault should occur on this instr
000003CC	4710	21C8		000003C8	3748	BC B'0001',*-4 not finished?
000003D0					3749	CNOP 0,8 (align to doubleword)
000003D0	00000000	00000000			3750	LOGICERR DC D'0' We should never reach here!
000003D8	00000000	00000000			3751	SVPGMNEW DC D'0' Original Program New PSW
000003E0	04080000	000003BC			3752	DATONPSW DC XL4'04080000',A(BEGDATON) Enable DAT PSW
					3753 *	
					3754 **	Temporary Program New routine:
					3755 **	Restore original Program New PSW
					3756 *	
000003E8	D207	0068 21D8	00000068	000003D8	3757	MYPGMNEW MVC PGMNPSW,SVPGMNEW Restore original Program New PSW
					3758 *	
					3759 **	Verify Program Check occurred on expected instruction
					3760 *	
000003EE	9268	9FFF		000021FF	3761	MVI SUBTEST,X'68'
000003F2	D503	22B8 002C	000004B8	0000002C	3762	CLC =A(PFINSADR),PGMOPSW+4 Program Check where expected?
000003F8	4770	2298		00000498	3763	BNE FAILTEST No?! Something is VERY WRONG!
					3764 *	
					3765 **	Verify Program Check was indeed a page fault
					3766 *	
000003FC	9211	9FFF		000021FF	3767	MVI SUBTEST,X'11'
00000400	9511	008F		0000008F	3768	CLI PGMICODE+1,X'11' Verify it's a Page Fault interrupt
00000404	4770	2298		00000498	3769	BNE FAILTEST If not then something is VERY WRONG!

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT	
					3771 *	
					3772 **	Verify Page Fault occurred on expected Page
					3773 *	
00000408	9205	9FFF		000021FF	3774	MVI SUBTEST,X'05'
0000040C	5800	0090		00000090	3775	L R0,PGMTRX
00000410	8800	000C		0000000C	3776	SRL R0,12
00000414	8900	000C		0000000C	3777	SLL R0,12
						Get where Page Fault occurred
00000418	8860	000C		0000000C	3779	SRL R6,12
0000041C	8960	000C		0000000C	3780	SLL R6,12
						Where Page Fault is expected
00000420	1506				3782	CLR R0,R6
00000422	4770	2298		00000498	3783	BNE FAILTEST
					3784 *	Page Fault occur on expected Page?
					3785 **	No? Then something is very wrong!
					3786 *	Verify CLCLE instruction registers were updated as expected
00000426	9206	9FFF		000021FF	3787	MVI SUBTEST,X'06'
0000042A	55A0	2394		00000594	3788	CL R10,CLCLPF
0000042E	47D0	2298		00000498	3789	BNH FAILTEST
00000432	55C0	239C		0000059C	3790	CL R12,CLCLPF+4+4
00000436	47D0	2298		00000498	3791	BNH FAILTEST
						(op1 greater than starting value?)
0000043A	9207	9FFF		000021FF	3793	MVI SUBTEST,X'07'
0000043E	15BD				3794	CLR R11,R13
00000440	4770	2298		00000498	3795	BNE FAILTEST
00000444	55B0	2398		00000598	3796	CL R11,CLCLPF+4
00000448	47B0	2298		00000498	3797	BNL FAILTEST
0000044C	55D0	23A0		000005A0	3798	CL R13,CLCLPF+4+4+4
00000450	47B0	2298		00000498	3799	BNL FAILTEST
						(op2 greater than starting value?)
00000454	9208	9FFF		000021FF	3801	MVI SUBTEST,X'08'
00000458	55A0	2434		00000634	3802	CL R10,ECLCLPF
0000045C	47B0	2298		00000498	3803	BNL FAILTEST
						(stop before end?)
00000460	9209	9FFF		000021FF	3805	MVI SUBTEST,X'09'
00000464	15A6				3806	CLR R10,R6
00000466	4720	2298		00000498	3807	BH FAILTEST
						(stop at or before expected page?)
0000046A	9210	9FFF		000021FF	3809	MVI SUBTEST,X'10'
0000046E	187A				3810	LR R7,R10
00000470	1E7B				3811	ALR R7,R11
00000472	1576				3812	CLR R7,R6
00000474	47D0	2298		00000498	3813	BNH FAILTEST
						(op1 stopped address)
						(add remaining length)
						(would remainder reach PF page?)
00000478	07FE				3815	BR R14
						Success!

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT				
					3817	*****			
					3818	* Verify CLCLE ending register values			
					3819	* R10-R12 = actual ending values, R5 --> expected ending values			
					3820	*****			
0000047A	90AD	2444		00000644	3822	ENDCLCL STM R10,R13,CLCLEND	Save actual ending register values		
0000047E	D50F	5000 2444	00000000	00000644	3823	CLC 0(4*4,R5),CLCLEND	Do they have the expected values?		
00000484	4770	2298		00000498	3824	BNE FAILTEST	If not then the test has failed		
00000488	07FF				3825	BR R15	Otherwise return to caller		

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	
				3843 *****	
				3844 * Working Storage	
				3845 *****	
000004A8				3847 LTORG ,	Literals pool
000004A8	00003000			3848 =A(SEGTABLS)	
000004AC	00003080			3849 =A(PAGETABS)	
000004B0	00001000			3850 =A(PAGE)	
000004B4	00005000			3851 =A(PFPGBYTS)	
000004B8	000003C8			3852 =A(PFINSADR)	
		00000400	00000001	3854 K EQU 1024	One KB
		00001000	00000001	3855 PAGE EQU (4*K)	Size of one page
		00010000	00000001	3856 K64 EQU (64*K)	64 KB
		00100000	00000001	3857 MB EQU (K*K)	1 MB
		000021FE	00000001	3859 TESTADDR EQU (2*PAGE+X'200'-2)	Where test/subtest numbers will go
		00200000	00000001	3861 MAINSIZE EQU (2*MB)	Minimum required storage size
		00000020	00000001	3862 NUMPGTBS EQU ((MAINSIZE+K64-1)/K64)	Number of Page Tables needed
		00000002	00000001	3863 NUMSEGTB EQU ((NUMPGTBS*4)/(16*4))	Number of Segment Tables
		00003000	00000001	3864 SEGTABLS EQU (3*PAGE)	Segment Tables Origin
		00003080	00000001	3865 PAGETABS EQU (SEGTABLS+(NUMPGTBS*4))	Page Tables Origin
000004BC	00B00060			3866 CRLREG0 DC 0A(0),XL4'00B00060'	Control Register 0
000004C0	00003002			3867 CTLREG1 DC A(SEGTABLS+NUMSEGTB)	Control Register 1

LOC	OBJECT	CODE	ADDR1	ADDR2	STMT
					3870 *****
					3871 * CLCLE Test Parameters: A(operand-1),A(operand-2)
					3872 *****
000004C4	00010000	00110000			3874 CLC1 DC A(1*K64),A(MB+(1*K64)) both equal
000004CC	00010000	00110000			3875 CLC2 DC A(1*K64),A(MB+(1*K64)) both equal
000004D4	0000FFF4	0010FFDE			3876 CLCBOTH DC A(1*K64-12),A(MB+(1*K64)-34) both equal
000004DC	00010000	0010FFDE			3877 CLCOP2 DC A(1*K64),A(MB+(1*K64)-34) both equal
000004E4	00020000	00120000			3879 CLC4 DC A(2*K64),A(MB+(2*K64)) op1 HIGH
000004EC	00030000	00130000			3880 CLC8 DC A(3*K64),A(MB+(3*K64)) op1 LOW!
000004F4	00040000	00140000			3881 CLC256 DC A(4*K64),A(MB+(4*K64)) op1 HIGH
000004FC	0004FFF4	00150000			3882 CLCOP1 DC A(5*K64-12),A(MB+(5*K64)) op1 HIGH

00000000 00003000 3884 CLCLE03 CSECT ,


```
3938 ****
3939 *      Fixed storage locations
3940 ****
```

000021FE	00	3944	TESTNUM	DC	X'00'	Test number of active test
000021FF	00	3945	SUBTEST	DC	X'00'	Active test sub-test number

```
00003000  00                                3949 DATTABS  DC    X'00'      Segment and Page Tables will go here...
```


SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES											
ASA	4	000000	512	3959	3543											
ASBEGIN	U	000000	1	3960	3965	4007	4043	4052	4070	4077	4083	4087	4091	4097	4114	
ASEND	U	000200	1	4113	4114											
ASLENGTH	U	000200	1	4114												
BCEXTCOD	H	00001A	2	3977												
BCIOCOD	H	00003A	2	3985												
BCMCKCOD	H	000032	2	3983												
BCPGMCOD	H	00002A	2	3981												
BCSVCCOD	H	000022	2	3979												
BEGDATON	I	0003BC	4	3744	3752											
BEGIN	I	000200	2	3547	3517	3544	3545									
CAW	F	000048	4	3989												
CAWADDR	R	000049	3	3992												
CAWKEY	X	000048	1	3990												
CAWSUSP	U	000008	1	3991												
CHANID	F	0000A8	4	4044												
CLC1	A	0004C4	4	3874												
CLC2	A	0004CC	4	3875												
CLC256	A	0004F4	4	3881												
CLC4	A	0004E4	4	3879												
CLC8	A	0004EC	4	3880												
CLCBOTH	A	0004D4	4	3876												
CLCL1	A	000504	4	3890	3599											
CLCL1K	A	000534	4	3896	3642											
CLCL2	A	000514	4	3892	3609											
CLCL256	A	000524	4	3894												
CLCL4	A	000564	4	3902	3581	3620										
CLCL8	A	000574	4	3904	3591	3632										
CLCLBOTH	A	000544	4	3898	3652											
CLCLE03	J	000000	12289	3498	3501	3508	3516	3518	3942	3947						
CLCLEND	F	000644	4	3934	3822	3823										
CLCLOP1	A	000584	4	3906	3586	3663										
CLCLOP2	A	000554	4	3900	3673											
CLCLPF	A	000594	4	3908	3691	3718	3788	3790	3796	3798						
CLCOP1	A	0004FC	4	3882												
CLCOP2	A	0004DC	4	3877												
CODE	2	000000	12289	3498												
CPUID	U	00031B	1	4116												
CRLREG0	A	0004BC	4	3866	3741											
CSW	F	000040	8	3988												
CTLREG1	A	0004C0	4	3867	3742											
DATONPSW	X	0003E0	4	3752	3743											
DATTABS	X	003000	1	3949												
DWAT0008	3	000490	8	3836	3835											
DWAT0009	3	0004A0	8	3841	3840											
ECLCL1	A	0005A4	4	3914	3603											
ECLCL1K	A	0005D4	4	3920	3646											
ECLCL2	A	0005B4	4	3916	3613											
ECLCL256	A	0005C4	4	3918												
ECLCL4	A	000604	4	3926	3624											
ECLCL8	A	000614	4	3928	3636											
ECLCLBTH	A	0005E4	4	3922	3656											
ECLCLOP1	A	000624	4	3930	3667											
ECLCLOP2	A	0005F4	4	3924	3677											
ECLCLPF	A	000634	4	3932	3802											
ENDCLCL	I	00047A	4	3822	3604	3614	3625	3637	3647	3657	3668	3678				

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES																
PCFETO	A	0000C4	4	4056																	
PERACCID	X	0000A1	1	4034																	
PERADDR	F	000098	4	4031																	
PERCODE	X	000096	1	4028																	
PERCODMK	U	0000F0	1	4029																	
PFINSADR	I	0003C8	4	3747	3762																
PFPAGE	U	000005	1	3935	3936																
PFPGBYTS	U	005000	1	3936	3720																
PGMACCID	X	0000A0	1	4033																	
PGMDXC	F	000090	4	4023																	
PGMICODE	H	00008E	2	4022	3768																
PGMIID	F	00008C	4	4018																	
PGMIILC	X	00008D	1	4020																	
PGMIILCM	U	00000C	1	4021																	
PGMNPSW	F	000068	8	4000	3733	3735	3736	3757													
PGMOPSW	F	000028	8	3972	3980	3762															
PGMTRX	F	000090	4	4024	3775																
PMCW1_0	X	000004	1	4122																	
PMCW1_8	X	000005	1	4125																	
PMCWB	U	000004	1	4157																	
PMCWCHP0	X	000010	1	4146																	
PMCWCHP1	X	000011	1	4147																	
PMCWCHP2	X	000012	1	4148																	
PMCWCHP3	X	000013	1	4149																	
PMCWCHP4	X	000014	1	4150																	
PMCWCHP5	X	000015	1	4151																	
PMCWCHP6	X	000016	1	4152																	
PMCWCHP7	X	000017	1	4153																	
PMCWDNUM	H	000006	2	4137																	
PMCWE	U	000080	1	4126																	
PMCWEXC	X	00001B	1	4156																	
PMCWIP	F	000000	4	4121																	
PMCWISCM	U	000038	1	4123																	
PMCWLM	U	000060	1	4127																	
PMCWLMG	U	000020	1	4128																	
PMCWLML	U	000040	1	4129																	
PMCWLP	X	000008	1	4139																	
PMCWLPUM	X	00000A	1	4141																	
PMCW	U	000004	1	4133																	
PMCWMBI	H	00000C	2	4143																	
PMCWMM	U	000018	1	4130																	
PMCWMMC	U	000008	1	4132																	
PMCWME	U	000010	1	4131																	
PMCW	X	00000F	1	4145																	
PMCW	X	00000B	1	4142																	
PMCW	X	000009	1	4140																	
PMCW	X	00000E	1	4144																	
PMCWRES1	X	000018	4	4154																	
PMCWRES2	X	000018	3	4155																	
PMCW	U	000001	1	4159																	
PMCW	U	000002	1	4134																	
PMCW	U	000001	1	4135																	
PMCW	U	000002	1	4158																	
R0	U	000000	1	4173	3543	3699	3708	3709	3734	3735	3741	3775	3776	3777	3782						
R1	U	000001	1	4174	3742																
R10	U	00000A	1	4183	3599	3600	3609	3610	3620	3621	3632	3633	3642	3643	3652	3653	3663	3664			

SYMBOL	TYPE	VALUE	LENGTH	DEFN	REFERENCES		
TESTNUM	X	0021FE	1	3944	3565	3577	3686
TIMER	F	000050	4	3995			
TTDES	F	000054	4	3996			
UA0	F	000010	8	3968			
UA1	F	00004C	4	3993			
UA2	F	0000A4	4	4038			
UA3	F	0000B4	4	4047			
UA4	X	0000B8	1	4048			
UA5	X	0000CC	8	4058			
UA6	X	0000EC	8	4064			
UA7	F	000118	8	4075			
UA8	X	000180	32	4104			
ZBRKADDR	A	000110	8	4074			
ZEMONCNT	F	00010C	4	4073			
ZEMONCTR	A	000100	8	4071			
ZEMONSIZ	F	000108	4	4072			
ZEXTNPSW	X	0001B0	16	4107			
ZEXTOPSW	X	000130	16	4099			
ZIONPSW	X	0001F0	16	4111			
ZIOOPSW	X	000170	16	4103			
ZMCKNPSW	X	0001E0	16	4110			
ZMCKOPSW	X	000160	16	4102			
ZMKFAILA	F	0000F8	8	4066			
ZMONCODE	F	0000B0	8	4041			
ZPGMNPSW	X	0001D0	16	4109			
ZPGMOPSW	X	000150	16	4101			
ZPGMTRX	F	0000A8	8	4040			
ZRSTNPSW	X	0001A0	16	4106			
ZRSTOPSW	X	000120	16	4098			
ZSASDISP	U	0011C0	1	4112			
ZSVCNPSW	X	0001C0	16	4108			
ZSVCOPSW	X	000140	16	4100			
=A(PAGE)	A	0004B0	4	3850	3701		
=A(PAGETABS)	A	0004AC	4	3849	3698	3726	
=A(PFINSADR)	A	0004B8	4	3852	3762		
=A(PFPGBYTS)	A	0004B4	4	3851	3720		
=A(SEGTABLS)	A	0004A8	4	3848	3696		

[illegible]

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

Entry: 0

Image	IMAGE	12289	0000-3000	0000-3000
Region	CODE	12289	0000-3000	0000-3000
CSECT	CLCLE03	12289	0000-3000	0000-3000

STMT	FILE NAME
1	c:\Users\Fish\Documents\Visual Studio 2008\Projects\MyProjects\ASMA-0\CLCLE-03-basic\CLCLE-03-basic.asm
2	C:\Users\Fish\Documents\Visual Studio 2008\Projects\Hercules_Git_Harold\SATK-0\srcasm\satk.mac

** NO ERRORS FOUND **