

.TITLE EAE-P1
.EBREL
/EAE SET UP DIAGNOSTIC
/
/REV. DATE -- 3/22/71 -- (1)
/
/

010000 A TSTRUN=10000

00000 R 000000 A UODSW 0 /COMM, WORD WITH MONITOR.
00001 R 000000 A 0
00002 R 000000 A 0
00003 R 000000 A 0
00004 R 000050 R .DSA EAESER
00005 R 000032 R .DSA EAEINT
00006 R 050105 A .SIXBT "EAEPT1"
00007 R 202461 A
00010 R 000040 A 40 /MASK FOR CHAIN MODE
00011 R A .BLOCK 7

/
00020 R 000000 A SYSERR 0
00021 R 000000 A ERWC 0
00022 R 000000 A ERCODE 0
00023 R A .BLOCK 7
00032 R 000000 A EAEINT 0
00033 R 200032 R LAC EAEINT
00034 R 040050 R DAC EAESER
00035 R 140020 R DZM SYSERR
00036 R 140021 R DZM ERWC
00037 R 600127 R JMP SETUP

/AC, MQ, LINK, AND SC CONTENTS FOR TYPE OUTS.
00040 R 000000 A ACSTRT 0
00041 R 000000 A MQSTRT 0
00042 R 000000 A LKSTRT 0
00043 R 000000 A SCSTRT 0
00044 R 000000 A ACEND 0
00045 R 000000 A MQEND 0
00046 R 000000 A LKEND 0
00047 R 000000 A SCEND 0

/EAESER 0 /SERVICE AND EXIT ROUTINES.
00051 R 620052 R JMP* EXIT

/EXIT 0
00052 R 000000 A
00053 R 040020 R DAC SYSERR
00054 R 750000 A CLA
00055 R 620050 R JMP* EAESER
.EJECT

00056	R	000000	A	CRLF	0	
00057	R	204562	R	LAC		PUT.2
00060	R	740001	A	CMA		
00061	R	344564	R	TAD		(1)
00062	R	040021	R	DAC		ERWC
00063	R	740100	A	SMA		
00064	R	740040	A	HLT		
00065	R	344565	R	TAD		(10)
00066	R	741100	A	SPA		
00067	R	740040	A	HLT		
00070	R	777776	A	LAW		-2
00071	R	040020	R	DAC		SYSERR
00072	R	620056	R	JMP*		CRLF
/						
00073	R	000000	A	MONBRK	0	
00074	R	044557	R	DAC		ACSAV#
00075	R	641002	A	LACQ		
00076	R	044560	R	DAC		MQSAV#
00077	R	200020	R	LAC		SYSERR
00100	R	740200	A	SZA		
00101	R	600104	R	JMP		.+3
00102	R	777773	A	LAW		-5
00103	R	140021	R	DZM		ERWC
00104	R	100052	R	JMS		EXIT
00105	R	100116	R	JMS		HOLD
00106	R	204560	R	LAC		MQSAV
00107	R	652000	A	LMQ		
00110	R	204557	R	LAC		ACSAV
00111	R	703344	A	DBR		
00112	R	620073	R	JMP*		MONBRK
/						
00113	R	140021	R	TERMIN	DZM	ERWC
00114	R	777774	A	LAW		-4
00115	R	600053	R	JMP		EXIT+1
/						
00116	R	000000	A	HOLD	0	
00117	R	750004	A	HOLD.1	LAS	
00120	R	500010	R	AND		UODSW+10
00121	R	741200	A	SNA		
00122	R	620116	R	JMP*		HOLD
00123	R	777773	A	LAW		-5
00124	R	140021	R	DZM		ERWC
00125	R	100052	R	JMS		EXIT
00126	R	600117	R	JMP		HOLD.1
				.EJECT		

/DOES EAE = OR THE MQ TO AC READ 0'S
/MQ SHOULD BE ZERO FROM RESET KEY

00127 R 100116 R
00130 R 650000 A
00131 R 744000 A
00132 R 750000 A

SETUP JMS HOLD
CLQ
CLL
CLA

00133 R 754000 A
00134 R 040041 R
00135 R 040040 R
00136 R 640002 A
00137 R 040044 R
00140 R 741200 A
00141 R 600150 R
00142 R 100704 R
00143 R 000202 A
00144 R 600040 R
00145 R 600041 R
00146 R 600044 R
00147 R 000000 A

EAERMQ CLA+4000 /CLEAR LINK
DAC MQSTRT
DAC ACSTRT
EAE+2 /OR MQ 1'S TO AC
DAC ACEND
SNA
JMP NOPAC
JMS ERROR
202
ACSTRT+600000
MQSTRT+600000
ACEND+600000
0

/DOES EAE NOP CLEAR THE AC?

00150 R 754001 A
00151 R 040040 R
00152 R 501023 R
00153 R 640000 A
00154 R 040044 R
00155 R 740001 A
00156 R 741200 A
00157 R 600165 R
00160 R 100704 R
00161 R 000225 A
00162 R 600040 R
00163 R 600044 R
00164 R 000000 A

NOPAC CLC+4000 /CLEAR LINK
DAC ACSTRT /AC AT START
AND KALL7 /MAKE MB=1#S BEFORE
EAE
DAC ACEND /AC AT END
CMA
SNA /AC ALTERED
JMP EAECAC /NO
JMS ERROR
225
ACSTRT+600000 /TYPE CONTENTS OF
ACEND+600000 /TYPE CONTENTS OF
0
.EJECT

/DOES EAE AND CLR AC BIT CLR THE AC?

/

00165	R	754001	A	EAECAC	CLC+4000	/CLEAR LINK
00166	R	641000	A		EAE+1000	/SHOULD CLEAR AC
00167	R	040044	R		DAC ACEND	
00170	R	741200	A		SNA	
00171	R	600177	R		JMP EAECLO	
00172	R	100704	R		JMS ERROR	
00173	R	000246	A		246	
00174	R	600040	R		ACSTRT+600000	
00175	R	600044	R		ACEND+600000	
00176	R	000000	A		0	

/

/DOES CLQ CLEAR THE MQ

/

00177	R	754001	A	EAECLO	CLC+4000	
00200	R	040041	R		DAC MQSTRT	
00201	R	640004	A		EAE+4	/SET MQ TO 1'S
00202	R	750000	A		CLA	
00203	R	040040	R		DAC ACSTRT	
00204	R	650000	A		CLQ	/CLEAR THE MQ
00205	R	040044	R		DAC ACEND	
00206	R	750000	A		CLA	
00207	R	640002	A		EAE+2	/OR MQ 1'S TO AC
00210	R	040045	R		DAC MQEND	
00211	R	741200	A		SNA	/READ 0'S BACK?
00212	R	600222	R		JMP MQITAC	
00213	R	100704	R		JMS ERROR	
00214	R	000264	A		264	
00215	R	600040	R		ACSTRT+600000	
00216	R	600041	R		MQSTRT+600000	
00217	R	600044	R		ACEND+600000	
00220	R	600045	R		MQEND+600000	
00221	R	000000	A		0	

.EJECT

/DOES MQ COMPLIMENT FROM 0'S TO 1'S
/AND MQ 1'S TO AC

/

00222	R	754000	A	MQ1TAC	CLA+4000	
00223	R	040040	R		DAC ACSTRT	
00224	R	040041	R		DAC MQSTRT	
00225	R	650004	A		CLQ+4	/CLEAR THE MQ AND COMPLIMENT
00226	R	040044	R		DAC ACEND	
00227	R	750000	A		CLA	
00230	R	640002	A		EAE+2	/OR THE MQ TO AC
00231	R	040045	R		DAC MQEND	
00232	R	740001	A		CMA	
00233	R	741200	A		SNA	
00234	R	600244	R		JMP NOPAC1	
00235	R	100704	R		JMS ERROR	
00236	R	000313	A		313	
00237	R	600040	R		ACSTRT+600000	
00240	R	600041	R		MQSTRT+600000	
00241	R	600044	R		ACEND+600000	
00242	R	600045	R		MQEND+600000	
00243	R	000000	A		0	

/

/DOES EAE=NOP WITH MQ=1'S ALTER THE AC

/

00244	R	754000	A	NOPAC1	CLA+4000	
00245	R	040040	R		DAC ACSTRT	
00246	R	750001	A		CLC	
00247	R	040041	R		DAC MQSTRT	
00250	R	650004	A		CLQ+4	/SET MQ TO ONES
00251	R	501023	R		AND KALL7	/MAKE MB TO 1'S
00252	R	640000	A		EAE	/NOP
00253	R	040044	R		DAC ACEND	
00254	R	740001	A		CMA	
00255	R	741200	A		SNA	/ONES FROM MQ TO AC?
00256	R	600265	R		JMP NOPMQ	
00257	R	100704	R		JMS ERROR	
00260	R	000341	A		341	
00261	R	600040	R		ACSTRT+600000	
00262	R	600041	R		MQSTRT+600000	
00263	R	600044	R		ACEND+600000	
00264	R	000000	A		0	

.EJECT

/DOES EAE NOP WITH MQ=1'S ALTER THE MQ

```

/
NOPMQ CLA+4000
00265 R 754000 A CLQ 4 /SET MQ TO 1'S
00266 R 650004 A AND KALL7 /MAKE MB TO 1#S BEFORE
00267 R 501023 R EAE /NOP
00270 R 640000 A DAC ACEND
00271 R 040044 R CLA
00272 R 750000 A EAE+2
00273 R 640002 A DAC MQEND
00274 R 040045 R CMA
00275 R 740001 A SNA /MQ STILL 1'S?
00276 R 741200 A JMP NOPMQ1
00277 R 600307 R JMS ERROR
00300 R 100704 R 366
00301 R 000366 A ACSTRT+600000
00302 R 600040 R MQSTRT+600000
00303 R 600041 R ACEND+600000
00304 R 600044 R MQEND+600000
00305 R 600045 R 0
00306 R 000000 A

```

/DOES NOP WITH AC=1'S ALTER MQ

```

/
NOPMQ1 CLA+4000
00307 R 754000 A DAC MQSTRT
00310 R 040041 R CLQ
00311 R 650000 A CLC
00312 R 750001 A DAC ACSTRT
00313 R 040040 R AND KALL7 /MAKE MB TO 1S BEFORE
00314 R 501023 R EAE /NOP
00315 R 640000 A DAC ACEND
00316 R 040044 R LACQ /GET MQ TO AC
00317 R 641002 A DAC MQEND
00320 R 040045 R SNA /ANY 1'S IN MQ
00321 R 741200 A JMP NOPLNK
00322 R 600332 R JMS ERROR
00323 R 100704 R 414
00324 R 000414 A ACSTRT+600000
00325 R 600040 R MQSTRT+600000
00326 R 600041 R ACEND+600000
00327 R 600044 R MQEND+600000
00330 R 600045 R 0
00331 R 000000 A .EJECT

```

/DOES NOP ALTER THE LINK
/AC 0'S MQ 0'S, AC 1'S MQ 1'S

```

/
00332 R 650000 A      NOPLNK CLQ
00333 R 140040 R      DZM ACSTRT
00334 R 140041 R      DZM MQSTRT
00335 R 140042 R      DZM LKSTRT
00336 R 200042 R      LAC LKSTRT
00337 R 740020 A      RAR
00340 R 200040 R      LAC ACSTRT      /SET LINK FOR TEST
00341 R 501023 R      AND KALL7      /MAKE MB TO ONES BEFORE
00342 R 540000 A      EAE      /NOP
00343 R 750010 A      GLK
00344 R 040046 R      DAC LKEND
00345 R 540042 R      SAD LKSTRT      /LINK ALTERED?
00346 R 600356 R      JMP NOPL.1
00347 R 100704 R      JMS ERROR
00350 R 000443 A      443
00351 R 700042 R      LKSTRT+700000      /ZERO SUPPRESS CONTENTS
00352 R 600040 R      ACSTRT+600000
00353 R 600041 R      MQSTRT+600000
00354 R 700046 R      LKEND+700000
00355 R 000000 A      0
00356 R 200042 R      NOPL.1 LAC LKSTRT
00357 R 440042 R      ISZ LKSTRT
00360 R 741200 A      SNA      /CHECKED L=0 AND L=1?
00361 R 600336 R      JMP NOPLNK+4
00362 R 200040 R      LAC ACSTRT
00363 R 740200 A      SZA      /CHECKED FOR AC=1'S
00364 R 600373 R      JMP EAESLK      /YES
00365 R 650004 A      CLQ+4      /SET MQ TO 1'S
00366 R 750001 A      CLC
00367 R 040040 R      DAC ACSTRT      /AC START =1'S
00370 R 040041 R      DAC MQSTRT
00371 R 140042 R      DZM LKSTRT      /LINK START=0
00372 R 600336 R      JMP NOPLNK+4
      .EJECT

```

/LINK SET TO 1 AND TO ZERO?

```

/
EAESLK  DZM LKSTRT      /START LINK 0 TO 1
          DZM MQSTRT      /MQ 0'S
          CLR
          LAC BIT0          /400000
          DAC ACSTRT
          LAC LKSTRT      /SET LINK INITIAL
          RAR
          LAC ACSTRT
          EAE+20000        /AC BIT 0 TO LINK
          DAC ACEND
          GLK
          DAC LKEND
          RTR
          SAD ACSTRT      /LINK SAME AS START?
          SKP
          JMP .+4          /ERROR
          LAC ACEND
          SAD ACSTRT      /AC ALTERED?
          JMP EAESL1
          JMS ERROR
          533
          LKSTRT+700000
          ACSTRT+600000
          LKEND+700000
          ACEND+600000
          0
EAESL1  ISZ LKSTRT      /NEXT PASS LINK 1 TO ZERO
          LAC ACSTRT
          DZM ACSTRT
          SZA
          JMP EAESLK+5
          .EJECT

```

```

00373 R 140042 R
00374 R 140041 R
00375 R 650000 A
00376 R 200770 R
00377 R 040040 R
00400 R 200042 R
00401 R 740020 A
00402 R 200040 R
00403 R 660000 A
00404 R 040044 R
00405 R 750010 A
00406 R 040046 R
00407 R 742020 A
00410 R 540040 R
00411 R 741000 A
00412 R 600416 R
00413 R 200044 R
00414 R 540040 R
00415 R 500425 R
00416 R 100704 R
00417 R 000533 A
00420 R 700042 R
00421 R 600040 R
00422 R 700046 R
00423 R 600044 R
00424 R 000000 A
00425 R 440042 R
00426 R 200040 R
00427 R 140040 R
00430 R 740200 A
00431 R 600400 R

```


/DOES NOP ALTER MQ=0'S WITH L=1

/

00432 R 140040 R	NOPLK1 DZM ACSTRT	/START AC 0'S
00433 R 140041 R	DZM MQSTRT	/MQ 0'S
00434 R 650000 A	CLQ	
00435 R 201000 R	LAC BIT17	/1=LINK
00436 R 740042 R	DAC LKSTRT	
00437 R 744020 A	RAR+4000	/CLR LINK, SET LINK
00440 R 501023 R	AND KALL7	/MAKE MB TO ONES BEFORE
00441 R 640000 A	EAE	/NOP
00442 R 040044 R	DAC ACEND	
00443 R 750010 A	GLK	
00444 R 040046 R	DAC LKEND	
00445 R 641002 A	LACQ	
00446 R 040045 R	DAC MQEND	
00447 R 741200 A	SNA	/MQ STILL ZERO'S
00450 R 200044 R	LAC ACEND	
00451 R 751200 A	SNAICLA	/AC STILL ZERO'S
00452 R 200046 R	LAC LKEND	
00453 R 740200 A	SZA	/LINK STILL 1
00454 R 600466 R	JMP ACORMQ	
00455 R 100704 R	JMS ERROR	
00456 R 000577 A	577	
00457 R 700042 R	LKSTRT+700000	
00460 R 600040 R	ACSTRT+600000	
00461 R 600041 R	MQSTRT+600000	
00462 R 700046 R	LKEND+700000	
00463 R 600044 R	ACEND+600000	
00464 R 600045 R	MQEND+600000	
00465 R 000000 A	0	

.EJECT

/WILL AC TO MQ TO AC ALL PATTERNS
/WITH MQ INITIALLY = 0 AND LINK = 0
/

00466 R 140040 R	ACORMQ	DZM ACSTRT	/START AC = 0'S
00467 R 140041 R		DZM MQSTRT	/MQ ALWAYS 0'S
00470 R 754000 A		CLL!CLA	
00471 R 650000 A		CLQ	
00472 R 200040 R		LAC ACSTRT	/GET NEXT SET
00473 R 643000 A		EAE+3000	/AC TO MQ
00474 R 040044 R		DAC ACEND	
00475 R 641002 A		LACQ	/MQ TO AC
00476 R 040045 R		DAC MQEND	
00477 R 540040 R		SAD ACSTRT	/MQ TO AC SAME AS START?
00500 R 741000 A		SKP	
00501 R 600505 R		JMP .+4	
00502 R 200044 R		LAC ACEND	/YES, TRY AC
00503 R 741200 A		SNA	/AC SHOULD BE 0
00504 R 600514 R		JMP ACOR.1	
00505 R 100704 R		JMS ERROR	
00506 R 000641 A		641	
00507 R 500040 R		ACSTRT+600000	
00510 R 500041 R		MQSTRT+600000	
00511 R 600044 R		ACEND+600000	
00512 R 600045 R		MQEND+600000	
00513 R 000000 A		0	
00514 R 100073 R	ACOR.1	JMS MONBRK	
00515 R 440040 R		ISZ ACSTRT	/TO 777777?
00516 R 600470 R		JMP ACORMQ+2	
		.EJECT	

/WILL AC TO MQ TO AC ALL PATTERNS
/WITH MQ = LAST PATTERN AND LINK = 1

/

00517 R 140040 R	ACLMQ	DZM ACSTRT	/START AC 0'S
00520 R 140041 R		DZM MQSTRT	/MQ 0'S
00521 R 650000 A		CLQ	
00522 R 201000 R		LAC BIT17	/LINK 1
00523 R 040042 R		DAC LKSTRT	
00524 R 744002 A		STL	/SET LINK
00525 R 200040 R		LAC ACSTRT	/GET NEXT CONSTANT
00526 R 652000 A		LMQ	/MQ TO 0'S, AC 1'S TO MQ
00527 R 040044 R		DAC ACEND	/SAVE AC RESULT
00530 R 750010 A		GLK	
00531 R 040046 R		DAC LKEND	/SAVE LINK RESULT
00532 R 641002 A		LACQ	/GET MQ
00533 R 040045 R		DAC MQEND	
00534 R 540040 R		SAD ACSTRT	/MQ = AC AT START?
00535 R 741000 A		SKP	
00536 R 600545 R		JMP ACLMQE	/MQ ERROR
00537 R 200040 R		LAC LKEND	
00540 R 741200 A		SNA	/LINK=1 AT END?
00541 R 600545 R		JMP ACLMQE	/LINK ERROR
00542 R 200044 R		LAC ACEND	
00543 R 540040 R		SAD ACSTRT	/AC END = AC START?
00544 R 600564 R		JMP ACL.1	
00545 R 100704 R	ACLMQE	JMS ERROR	
00546 R 000704 A		704	
00547 R 700042 R		LKSTRT+700000	
00550 R 600040 R		ACSTRT+600000	
00551 R 600041 R		MQSTRT+600000	
00552 R 700042 R		LKSTRT+700000	
00553 R 600040 R		ACSTRT+600000	
00554 R 600040 R		ACSTRT+600000	
00555 R 000000 A		0	
00556 R 100704 R		JMS ERROR	
00557 R 000704 A		704	
00560 R 700046 R		LKEND+700000	
00561 R 600044 R		ACEND+600000	
00562 R 600045 R		MQEND+600000	
00563 R 000000 A		0	
00564 R 100073 R	ACL.1	JMS MONBRK	
00565 R 200045 R		LAC MQEND	
00566 R 040041 R		DAC MQSTRT	/NEW MQ START
00567 R 440040 R		ISZ ACSTRT	/TO 777777?
00570 R 600524 R		JMP ACLMQ+5	
		.EJECT	

/DOES THE MQ COMPLIMENT ALL PATTERNS

```

/
00571 R 140040 R   COMPMQ  DZM ACSTRT
00572 R 200040 R   LAC ACSTRT   /GET NEXT PATTERN
00573 R 040041 R   DAC MQSTRT
00574 R 672000 A   LMQ+20000   /AC TO MQ, AC0 TO L
00575 R 640004 A   CMQ   /-MQ
00576 R 040044 R   DAC ACEND   /SAVE AC RESULT
00577 R 641002 A   LACQ   /GET MQ
00600 R 040045 R   DAC MQEND
00601 R 740001 A   CMA   //MQ
00602 R 540040 R   SAD ACSTRT  /-MQ = AC START?
00603 R 200044 R   LAC ACEND
00604 R 540040 R   SAD ACSTRT  /ACEND = AC START?
00605 R 600015 R   JMP C.1
00606 R 100704 R   JMS ERROR
00607 R 000763 A   763
00610 R 600040 R   ACSTRT+600000
00611 R 600041 R   MQSTRT+600000
00612 R 600044 R   ACEND+600000
00613 R 600045 R   MQEND+600000
00614 R 000000 A   0
00615 R 100073 R   C.1   JMS     MONBRK
00616 R 440040 R   ISZ ACSTRT
00617 R 600572 R   JMP COMPMQ+1

```

/DOES AC TO MQ ALL 1'S WITH MQ=1'S

```

ACONEQ  CLC
00621 R 040041 R   DAC MQSTRT
00622 R 040040 R   DAC ACSTRT
00623 R 650004 A   CLQ+4   /SET MQ=1'S
00624 R 642000 A   EAE+2000 /AC 1'S TO MQ1'S
00625 R 040044 R   DAC ACEND
00626 R 641002 A   LACQ
00627 R 040045 R   DAC MQEND
00630 R 740001 A   CMA
00631 R 741200 A   SNA   /MQ STAY 1'S
00632 R 600642 R   JMP EAEABS
00633 R 100704 R   JMS ERROR
00634 R 001017 A   1017
00635 R 600040 R   ACSTRT+600000
00636 R 600041 R   MQSTRT+600000
00637 R 600044 R   ACEND+600000
00640 R 600045 R   MQEND+600000
00641 R 000000 A   0
      .EJECT

```

/DOES ABS GET ABSOLUTE AC
/AND NOT DISTURB LINK=1 OR 0

00642 R	140040 R	EAEABS	DZM ACSTRT	/START AC 0'S
00643 R	201000 R		LAC BIT17	/LINK 1
00644 R	040042 R		DAC LKSTRT	
00645 R	200042 R		LAC LKSTRT	
00646 A	740020 A		RAR	/SET LINK
00647 R	200040 R		LAC ACSTRT	/GET AC START
00650 R	644000 A		ABS	/ABSOLUTE AC
00651 R	040044 R		DAC ACEND	/SAVE RESULT
00652 R	750010 A		GLK	
00653 R	040046 R		DAC LKEND	
00654 R	540042 R		SAD LKSTRT	/LINK SAME?
00655 R	741000 A		SKP	/YES
00656 R	600664 R		JMP .+6	/ERROR, LINK CHANGED
00657 R	200040 R		LAC ACSTRT	
00660 R	741100 A		SPA	/AC POSITIVE AT START?
00661 R	740001 A		CMA	/NO, SHOULD BE POS. ABS
00662 R	540044 R		SAD ACEND	/RESULT AC OK?
00663 R	600673 R		JMP EAEA.1	/YES
00664 R	100704 R		JMS ERROR	/ABS ERROR LINK OR AC
00665 R	001045 A		1045	
00666 R	700042 R		LKSTRT+700000	
00667 R	600040 R		ACSTRT+600000	
00670 R	700046 R		LKEND+700000	
00671 R	600044 R		ACEND+600000	
00672 R	000000 A		0	
00673 R	100073 R	EAEA.1	JMS MONBRK	
00674 R	440040 R		ISZ ACSTRT	
00675 R	741000 A		SKP	
00676 R	601024 R		JMP SCTST1	
00677 R	200042 R		LAC LKSTRT	
00700 R	740001 A		CMA	
00701 R	501000 R		AND BIT17	
00702 R	040042 R		DAC LKSTRT	
00703 R	600645 R		JMP EAEABS+3	
			.EJECT	

/EAE ERROR TYPEOUT ROUTINE (GENERAL PURPOSE)

```

/
00704 R 600704 R ERROR JMP .
00705 R 100725 R JMS RESPUT
00706 R 220704 R LAC* ERROR
00707 R 100732 R ERROR1 JMS PUT
00710 R 440704 R ISZ ERROR
00711 R 220704 R LAC* ERROR
00712 R 741200 A SNA
00713 R 600720 R JMP ERROR2
00714 R 504566 R AND (77777)
00715 R 044563 R DAC TEMP#
00716 R 224563 R LAC* TEMP
00717 R 600707 R JMP ERROR1
/
00720 R 100056 R ERROR2 JMS CRLF
00721 R 100073 R JMS MONBRK
00722 R 440704 R ISZ ERROR
00723 R 703344 A DBR
00724 R 620704 R JMP* ERROR

```

/RESET ERROR PUT AWAY ROUTINE.

```

/
00725 R 000000 A RESPUT 0
00726 R 204567 R LAC (ERCODE)
00727 R 044561 R DAC PUT.1#
00730 R 144562 R DZM PUT.2#
00731 R 620725 R JMP* RESPUT

```

```

/
00732 R 000000 A PUT 0
00733 R 064561 R DAC* PUT.1
00734 R 444561 R ISZ PUT.1
00735 R 444562 R ISZ PUT.2
00736 R 620732 R JMP* PUT

```

```

/
00737 R 000000 A SVCHAR 0
00740 R 000000 A 0
00741 R 000000 A 0
00742 R 000000 A 0
00743 R 000000 A 0
00744 R 000000 A 0
00745 R 000007 A SEVEN 7
00746 R 000077 A SEVSEV 77
00747 R 000076 A SEVSIX 76
00750 R 777756 A K18 777756
00751 R 000060 A SIXTY 60
00752 R 000070 A SEVENTY 70
00753 R 000074 A SEVN4 74
00754 R 000041 A FOUR1 41
00755 R 000037 A THREE7 37
00756 R 000061 A SIXONE 61
00757 R 000017 A ONESEV 17
00760 R 000071 A SEVONE 71

```

PAGE 15 EAAPT1 SRC EAE-P1

00761 R 000075 A SEVFIV 75
 .EJECT

00762	R	000003	A	THREE	3
00763	R	000045	A	FOUR5	45
00764	R	000044	A	FOUR4	44
00765	R	000034	A	THREE4	34
00766	R	000056	A	FIVE6	56
00767	R	252525	A	COMBIT	252525

/
/BIT AND NO BIT CONSTANTS

00770	R	400000	A	BIT0	400000
00771	R	200000	A	BIT1	200000
00772	R	100000	A	BIT2	100000
00773	R	000040	A	BIT12	40
00774	R	000020	A	BIT13	20

/

00775	R	000010	A	BIT14	10
-------	---	--------	---	-------	----

00776	R	000004	A	BIT15	4
00777	R	000002	A	BIT16	2
01000	R	000001	A	BIT17	1
01001	R	377777	A	NBIT0	377777
01002	R	577777	A	NBIT1	577777
01003	R	677777	A	NBIT2	677777
01004	R	737777	A	NBIT3	737777
01005	R	757777	A	NBIT4	757777
01006	R	767777	A	NBIT5	767777
01007	R	773777	A	NBIT6	773777
01010	R	775777	A	NBIT7	775777
01011	R	776777	A	NBIT8	776777
01012	R	777377	A	NBIT9	777377
01013	R	777577	A	NBIT10	777577
01014	R	777677	A	NBIT11	777677
01015	R	777737	A	NBIT12	777737
01016	R	777757	A	NBIT13	777757
01017	R	777767	A	NBIT14	777767
01020	R	777773	A	NBIT15	777773
01021	R	777775	A	NBIT16	777775
01022	R	777776	A	NBIT17	777776
01023	R	777777	A	KALL7	777777

.EJECT


```

/SHIFT COUNTER AND
/AC MQ SHIFT TEST
/
/SHIFT COUNTER TEST
/UTILIZES NORMALIZE INSTRUCTION
/WITH NO SHIFT TO DATA TEST S.C
/

```

```

01024 R 100073 R SGTST1 JMS MONBRK
01025 R 200746 R LAC SEVSEV
01026 R 040043 R DAC SCSTRT
01027 R 200771 R LAC BIT1 /200000 ALREADY NORMALIZED
01030 R 640400 A NORM-44 /SET SC TO -1 (77)
01031 R 641001 A LACS /SC TO AC
01032 R 040047 R DAC SCEND
01033 R 741200 A SNA /READ SC=0'S TO AC?
01034 R 601042 R JMP NOPSC /YES, CONTINUE
01035 R 100704 R JMS ERROR
01036 R 002200 A 2200
01037 R 700043 R SCSTRT+700000
01040 R 700047 R SCEND+700000
01041 R 000000 A 0

```

```

/
/DOES EAE NOP ALTER THE SC
/

```

```

01042 R 140043 R NOPSC DZM SCSTRT
01043 R 501023 R AND KALL7 /MAKE MB ONES BEFORE
01044 R 640000 A EAE /NOP=
01045 R 641001 A LACS /GET SC TO AC
01046 R 040047 R DAC SCEND
01047 R 741200 A SNA /SC STILL ZERO'S
01050 R 601056 R JMP SCT076
01051 R 100704 R JMS ERROR
01052 R 002225 A 2225
01053 R 700043 R SCSTRT+700000
01054 R 700047 R SCEND+700000
01055 R 000000 A 0

```

```

/
/DOES SC SET TO 76 AND +1 TO 77
SCT076

```

```

01056 R 200047 R LAC SCEND
01057 R 040043 R DAC SCSTRT
01060 R 201000 R LAC BIT17
01061 R 040041 R DAC MQSTRT
01062 R 200771 R LAC BIT1
01063 R 040040 R DAC ACSTRT
01064 R 640401 A NORM-43
01065 R 040044 R DAC ACEND
01066 R 641001 A LACS
01067 R 040047 R DAC SCEND
01070 R 540746 R SAD SEVSEV
01071 R 601073 R JMP .+2
01072 R 101300 R JMS SCERR
.EJECT

```

/DOES SC SET TO 74 AND +1 TO 75

01073 R 200047 R
01074 R 040043 R
01075 R 200762 R
01076 R 040041 R
01077 R 200771 R
01100 R 540403 A
01101 R 040044 R
01102 R 641001 A
01103 R 040047 R
01104 R 540761 R
01105 R 601107 R
01106 R 101300 R

SCT074 LAC SCEND
DAC SCSTRT
LAC THREE
DAC MQSTRT
LAC BIT1
NORM=41
DAC ACEND
LACS
DAC SCEND
SAD SEVFIV
JMP .+2
JMS SCERR

/SC TO 74+1 TO 75
/SAVE FOR ERROR TYPE

/DOES SC SET TO 70 AND +1 TO 71

01107 R 200047 R
01110 R 040043 R
01111 R 200745 R
01112 R 040041 R
01113 R 200771 R
01114 R 640407 A
01115 R 040044 R
01116 R 641001 A
01117 R 040047 R
01120 R 540760 R
01121 R 601123 R
01122 R 101300 R

SCT070 LAC SCEND
DAC SCSTRT
LAC SEVEN
DAC MQSTRT
LAC BIT1
NORM=35
DAC ACEND
LACS
DAC SCEND
SAD SEVONE
JMP .+2
JMS SCERR

//, SC TO 70 AND +1 TO 71
/SAVE FOR ERROR TYPE

/WILL SC SET TO 60 AND +1 TO 61

01123 R 200047 R
01124 R 040043 R
01125 R 200757 R
01126 R 040041 R
01127 R 200771 R
01130 R 640417 A
01131 R 040044 R
01132 R 641001 A
01133 R 040047 R
01134 R 540756 R
01135 R 601137 R
01136 R 101300 R

SCT060 LAC SCEND
DAC SCSTRT
LAC ONESEV
DAC MQSTRT
LAC BIT1
NORM=25
DAC ACEND
LACS
DAC SCEND
SAD SIXONE
JMP .+2
JMS SCERR
.EJECT

/NORM 17

/SET SC TO 60 AND +1 TO 61
/SAVE FOR ERROR TYPE

/REA

/WILL SC SET TO 40 AND +1 TO 41

```

/
01137 R 200047 R SCT040 LAC SCEND
01140 R 040043 R DAC SCSTRT
01141 R 200755 R LAC THREE7 /NORM 37
01142 R 040041 R DAC MQSTRT
01143 R 200771 R LAC BIT1 /20000 ALREADY NORMALIZED
01144 R 640437 A NORM-5 /SET SC TO 40 AND +1 TO 41
01145 R 040044 R DAC ACEND
01146 R 641001 A LACS /GET SC TO AC
01147 R 040047 R DAC SCEND /SAVE FOR ERROR TYPE
01150 R 540754 R SAD FOUR1 /READ 41 FROM SC TO AC
01151 R 601153 R JMP .+2 /YES
01152 R 101300 R JMS SCERR

```

/
/WILL SC SET TO 0 AND +1 TO 1

```

/
01153 R 200047 R SCT000 LAC SCEND
01154 R 040043 R DAC SCSTRT
01155 R 200746 R LAC SEVSEV /NORM 77
01156 R 040041 R DAC MQSTRT
01157 R 200771 R LAC BIT1
01160 R 640477 A NORM +33 /SC TO 00 +1 TO 01
01161 R 040044 R DAC ACEND
01162 R 641001 A LACS
01163 R 040047 R DAC SCEND
01164 R 541000 R SAD BIT17 /SC READ 01?
01165 R 601167 R JMP .+2 /YES
01166 R 101300 R JMS SCERR

```

/
/WILL SC SET TO 01 AND +1 TO 02

```

/
01167 R 200047 R SCT001 LAC SCEND
01170 R 040043 R DAC SCSTRT
01171 R 200747 R LAC SEVSIX /NORM 76
01172 R 040041 R DAC MQSTRT
01173 R 200771 R LAC BIT1
01174 R 640476 A NORM 32 /SET SC TO 1 +1 TO 2
01175 R 040044 R DAC ACEND
01176 R 641001 A LACS
01177 R 040047 R DAC SCEND
01200 R 540777 R SAD BIT16
01201 R 601203 R JMP .+2
01202 R 101300 R JMS SCERR
.EJECT

```

/WILL SC SET TO 03 AND +1 TO 04

```

/
01203 R 200047 R SCT003 LAC SCEND
01204 R 040043 R DAC SCSTRT
01205 R 200753 R LAC SEVNA /NORM 74
01206 R 040041 R DAC MQSTRT
01207 R 200771 R LAC BIT1
01210 R 640474 A NORM +30 /SET SC TO 3 +1 TO 4
01211 R 040044 R DAC ACEND
01212 R 641001 A LACS
01213 R 040047 R DAC SCEND
01214 R 540776 R SAD BIT15 /SC TO AC =4?
01215 R 601217 R JMP .+2 /YES
01216 R 101300 R JMS SCERR

```

/WILL SC SET TO 07 AND +1 TO 10

```

/
01217 R 200047 R SCT007 LAC SCEND
01220 R 040043 R DAC SCSTRT
01221 R 200752 R LAC SEVNTY /NORM 70
01222 R 040041 R DAC MQSTRT
01223 R 200771 R LAC BIT1
01224 R 640470 A NORM +24 /SET SC TO 7 +1 TO 10
01225 R 040044 R DAC ACEND
01226 R 641001 A LACS
01227 R 040047 R DAC SCEND
01230 R 540775 R SAD BIT14 /SC TO AC = 10?
01231 R 601233 R JMP .+2 /YES
01232 R 101300 R JMS SCERR

```

/WILL SC SET TO 17 AND +1 TO 20

```

/
01233 R 200047 R SCT017 LAC SCEND
01234 R 040043 R DAC SCSTRT
01235 R 200751 R LAC SIXTY /NORM 60
01236 R 040041 R DAC MQSTRT
01237 R 200771 R LAC BIT1
01240 R 640460 A NORM +14 /SC TO 17+1 TO 20
01241 R 040047 R DAC SCEND
01242 R 641001 A LACS
01243 R 040047 R DAC SCEND
01244 R 540774 R SAD BIT13 /SC TO AC = 20?
01245 R 601247 R JMP .+2 /YES
01246 R 101300 R JMS SCERR
.EJECT

```

/WILL SC SET TO 37 AND +1 TO 40

/

01247	R	200047	R	SCT037	LAC SCEND	
01250	R	200043	R		LAC SCSTRT	
01251	R	200773	R		LAC BIT12	/NORM 40
01252	R	040041	R		DAC MQSTRT	
01253	R	200771	R		LAC BIT1	
01254	R	640440	A		NORM-4	/SET SC TO 37 +1 TO 40
01255	R	040044	R		DAC ACEND	
01256	R	641001	A		LACS	
01257	R	040047	R		DAC SCEND	
01260	R	540773	R		SAD BIT12	/SC TO AC = 40?
01261	R	601263	R		JMP .+2	/YES
01262	R	101300	R		JMS SCERR	

/

/WILL SC SET TO 77 AND +1 TO 00

/

01263	R	200047	R	SCT077	LAC SCEND	
01264	R	040043	R		DAC SCSTRT	
01265	R	200746	R		LAC SEVSEV	/NORM 0
01266	R	040041	R		DAC MQSTRT	
01267	R	200771	R		LAC BIT1	
01270	R	640400	A		NORM-44	/SET SC TO 77 AND +1 TO 0
01271	R	040044	R		DAC ACEND	
01272	R	641001	A		LACS	/GET SC TO AC
01273	R	040047	R		DAC SCEND	
01274	R	741200	A		SNA	/SC TO AC = 00?
01275	R	601277	R		JMP .+2	/YES
01276	R	101300	R		JMS SCERR	
01277	R	601323	R		JMP NOPSC1	
					.EJECT	

```

01300 R 601300 R      SCERR    JMP .
01301 R 200041 R                 LAC M0STRT      /GET SC OF NORM
01302 R 740001 A                 CMA
01303 R 500746 R                 AND SEVSEV      /SHOULD SET SC TO
01304 R 040045 R                 DAC M0END
01305 R 341000 R                 TAD BIT17
01306 R 500746 R                 AND SEVSEV      /SC SHOULD +1 TO
01307 R 040046 R                 DAC LKEND
01310 R 100704 R                 JMS ERROR            /TYPE OUT
01311 R 002520 A                 2520
01312 R 700043 R                 SCSTRT+700000      /SC AT START
01313 R 600040 R                 ACSTRT+600000      /AC AT START
01314 R 700041 R                 M0STRT+700000      /SC PORTION OF NORM
01315 R 700045 R                 M0END+700000      /SHOULD SET SC TO
01316 R 700046 R                 LKEND+700000      /SC SHOULD +1 TO
01317 R 700047 R                 SCEND+700000      /SC TO AC EQUALED
01320 R 600044 R                 ACEND+600000      /AC AFTER NORM
01321 R 000000 A                 0
01322 R 621300 R                 JMP* SCERR

```

/

/DOES EAE NOP ALTER SC = 77

```

01323 R 200746 R      NOPSC1    LAC SEVSEV
01324 R 040043 R                 DAC SCSTRT
01325 R 200771 R                 LAC BIT1
01326 R 640401 A                 NORM=43            /SET SC TO 77
01327 R 501023 R                 AND KALL7          /MAKE MB TO ONES BEFORE
01330 R 640077 A                 EAE+77            /NOP SHOULD NOT ALTER SC
01331 R 641001 A                 LACS            /GET SC TO AC
01332 R 040047 R                 DAC SCEND
01333 R 540043 R                 SAD SCSTRT      /SC TO AC = 77?
01334 R 601342 R                 JMP ALSZER
01335 R 100704 R                 JMS ERROR
01336 R 002555 A                 2555
01337 R 700043 R                 SCSTRT+700000
01340 R 700047 R                 SCEND+700000
01341 R 000000 A                 0
                     .EJECT

```

/SHIFT TESTS
/ALS - ACCUMULATOR LEFT SHIFT
/DOES ALS AC = 0'S ALTER THE AC?

01342 R 100073 R
01343 R 140040 R
01344 R 140041 R
01345 R 140042 R
01346 R 140043 R
01347 R 651000 A
01350 R 744000 A
01351 R 640700 A
01352 R 040044 R
01353 R 750010 A
01354 R 040046 R
01355 R 641001 A
01356 R 040047 R
01357 R 200044 R
01360 R 741200 A
01361 R 741000 A
01362 R 101702 R

ALSZER JMS MONBRK
DZM ACSTRT
DZM MQSTRT
DZM LKSTRT
DZM SCSTRT
CLQ+1000
CLL
ALS
DAC ACEND
GLK
DAC LKEND
LACS
DAC SCEND
LAC ACEND
SNA
SKP
JMS ALSERR

/CLEAR AC - MQ AND LINK

/DOES ALS 01 AC = 0'S OK

01363 R 201000 R
01364 R 040043 R
01365 R 140040 R
01366 R 140041 R
01367 R 140042 R
01370 R 650000 A
01371 R 641000 A
01372 R 744000 A
01373 R 640701 A
01374 R 040044 R
01375 R 750010 A
01376 R 040046 R
01377 R 641001 A
01400 R 040047 R
01401 R 200044 R
01402 R 741200 A
01403 R 741000 A
01404 R 101702 R
01405 R 200041 R
01406 R 652000 A
01407 R 200041 R
01410 R 740200 A
01411 R 601416 R
01412 R 750001 A
01413 R 040041 R
01414 R 640004 A
01415 R 601371 R

ALS01 LAC BIT17
DAC SCSTRT
DZM ACSTRT
DZM MQSTRT
DZM LKSTRT
CLQ
EAE+1000
CLL
ALS 01
DAC ACEND
GLK
DAC LKEND
LACS
DAC SCEND
LAC ACEND
SNA
SKP
JMS ALSERR
LAC MQSTRT
LMQ
LAC MQSTRT
SZA
JMP .+5
CLC
DAC MQSTRT
EAE+4
JMP ALS01+6
.EJECT

/ALS 01

/AC 0'S TO START

/MQ 0'S

/LINK IS ZERO

/SHIFT AC LEFT 1

/LINK FOR TYPEOUTS

/SC FOR TYPEOUTS

/2ND PASS MQ = 1'S

/LINK TO AC 17
 /BIT = 0 L=0, BIT =0 L=1, BIT = 1 L = 0, BIT = 1 L = 1
 /

01416 R 140040 R
 01417 R 140041 R
 01420 R 140042 R
 01421 R 650000 A
 01422 R 200042 R
 01423 R 740020 A
 01424 R 200040 R
 01425 R 640701 A
 01426 R 040044 R
 01427 R 750010 A
 01430 R 040040 R
 01431 R 641001 A
 01432 R 040047 R
 01433 R 200040 R
 01434 R 740010 A
 01435 R 340042 R
 01436 R 540044 R
 01437 R 741000 A
 01440 R 101702 R
 01441 R 200042 R
 01442 R 440042 R
 01443 R 741200 A
 01444 R 601421 R
 01445 R 140042 R
 01446 R 200040 R
 01447 R 440040 R
 01450 R 741200 A
 01451 R 601421 R

ALSLNK DZM ACSTRT /START AC 0'S
 DZM MQSTRT
 DZM LKSTRT /LINK START 0
 CLQ
 LAC LKSTRT
 RAR /LINK = 0 OR 1
 LAC ACSTRT
 ALS 01
 DAC ACEND
 GLK
 DAC LKEND
 LACS
 DAC SCEND
 LAC ACSTRT
 RAL
 TAD LKSTRT
 SAD ACEND
 SKP
 JMS ALSERR
 LAC LKSTRT
 ISZ LKSTRT
 SNA /2ND PASS L=1
 JMP ALSLNK+3
 DZM LKSTRT
 LAC ACSTRT
 ISZ ACSTRT
 SNA /3RD AND 4TH PASS AC=1
 JMP ALSLNK+3
 .EJECT

/DOES ALS ALTER THE LINK = 1 OR 0

/

01452 R 140041 R	LNKALS	DZM MQSTRT	/MQ ALWAYS = 0
01453 R 140040 R		DZM ACSTRT	/START AC=0
01454 R 140042 R		DZM LKSTRT	/LINK START 0
01455 R 201000 R		LAC BIT17	
01456 R 040043 R		DAC SCSTRT	/SC = 01
01457 R 650000 A		CLQ	
01460 R 200042 R		LAC LKSTRT	
01461 R 740020 A		RAR	/LINK = 1 0 R 0
01462 R 200040 R		LAC ACSTRT	/AC = 0 OR 400000
01463 R 640701 A	KALS01	ALS 01	
01464 R 040044 R		DAC ACEND	/SAVE AC RESULT
01465 R 641001 A		LACS	
01466 R 040047 R		DAC SCEND	/SAVE SC RESULT
01467 R 750010 A		GLK	
01470 R 040046 R		DAC LKEND	
01471 R 540042 R		SAD LKSTRT	/LINK SAME AS STRT?
01472 R 741000 A		SKP	/YES
01473 R 101702 R		JMS ALSERR	
01474 R 200042 R		LAC LKSTRT	
01475 R 440042 R		ISZ LKSTRT	
01476 R 741200 A		SNA	
01477 R 601457 R		JMP LNKALS+5	/2ND AND 4TH PAS L = 1
01500 R 200040 R		LAC ACSTRT	
01501 R 200770 R		LAC BIT0	
01502 R 540040 R		SAD ACSTRT	/AC 0 ALREADY = 1
01503 R 601507 R		JMP .+4	
01504 R 140042 R		DZM LKSTRT	/3RD AND 4TH PASS
01505 R 040040 R		DAC ACSTRT	/AC=400000
01506 R 601457 R		JMP LNKALS+5	
		.EJECT	

/DOES ALS ALTER THE MQ

/

01507 R 140041 R
 01510 R 140040 R
 01511 R 140042 R
 01512 R 201000 R
 01513 R 040043 R
 01514 R 200042 R
 01515 R 740020 A
 01516 R 200041 R
 01517 R 652000 A
 01520 R 200040 R
 01521 R 640701 A
 01522 R 040044 R
 01523 R 750010 A
 01524 R 040046 R
 01525 R 641001 A
 01526 R 040047 R
 01527 R 641002 A
 01530 R 040045 R
 01531 R 540041 R
 01532 R 601543 R
 01533 R 100704 R
 01534 R 002700 A
 01535 R 700042 R
 01536 R 600040 R
 01537 R 700046 R
 01540 R 600044 R
 01541 R 600045 R
 01542 R 000000 A
 01543 R 200042 R
 01544 R 440042 R
 01545 R 741200 A
 01546 R 601514 R
 01547 R 140042 R
 01550 R 200040 R
 01551 R 740001 A
 01552 R 040040 R
 01553 R 740200 A
 01554 R 601514 R
 01555 R 200041 R
 01556 R 740001 A
 01557 R 040041 R
 01560 R 740200 A
 01561 R 601514 R

ALSMQT DZM MQSTRT
 DZM ACSTRT
 DZM LKSTRT
 LAC BIT17
 DAC SCSTRT
 LAC LKSTRT
 RAR
 LAC MQSTRT
 LMQ
 LAC ACSTRT
 ALS 01
 DAC ACEND
 GLK
 DAC LKEND
 LACS
 DAC SCEND
 LACQ
 DAC MQEND
 SAD MQSTRT
 JMP ALS.1
 JMS ERROR
 2760
 LKSTRT+700000
 ACSTRT+600000
 LKEND+700000
 ACEND+600000
 MQEND+600000
 0
 ALS.1 LAC LKSTRT
 ISZ LKSTRT
 SNA
 JMP ALSMQT+5
 DZM LKSTRT
 LAC ACSTRT
 CMA
 DAC ACSTRT
 SZA
 JMP ALSMQT+5
 LAC MQSTRT
 CMA
 DAC MQSTRT
 SZA
 JMP ALSMQT+5
 .EJECT

/1ST PASSES MQ = 0'S

/ALS 01 PLACE

/L=1 OR 0

/MQ = 0'S OR 1'S

/AC = 0'S OR 1'S

/MQ SAME AS START?

/YES

/EVERY OTHER PASS L = 1

/NEXT PASS L = 0

/AC=0'S, 1'S, 0'S, 1'S

/MQ = 0'S 4 PASSES 1'S 4 PASSES

/MQSTRT BACK TO 0'S?

/NO, TEST M1 = 1'S

/WILL AC0 GO TO LINK PROPERLY
/IMMEDIATELY FOLLOWING AN ALS LEFT SHIFT
/0-0,0-1,1=0,1-1

01562 R 140042 R
01563 R 140041 R
01564 R 200770 R
01565 R 040040 R
01566 R 201000 R
01567 R 040043 R
01570 R 200042 R
01571 R 740020 A
01572 R 200040 R
01573 R 640701 A
01574 R 660000 A
01575 R 040044 R
01576 R 750010 A
01577 R 040046 R
01600 R 540041 R
01601 R 601612 R
01602 R 100704 R
01603 R 003042 A
01604 R 700043 R
01605 R 700042 R
01606 R 600040 R
01607 R 700046 R
01610 R 600044 R
01611 R 000000 A
01612 R 200042 R
01613 R 440042 R
01614 R 741200 A
01615 R 601570 R
01616 R 140042 R
01617 R 200771 R
01620 R 540040 R
01621 R 601625 R
01622 R 040040 R
01623 R 440041 R
01624 R 601570 R

SGNSHF DZM LKSTRT
DZM MQSTRT
LAC BIT0
DAC ACSTRT
LAC BIT17
DAC SCSTRT
LAC LKSTRT
RAR
LAC ACSTRT
ALS 01
EAE+20000
DAC ACEND
GLK
DAC LKEND
SAD MQSTRT
JMP NSNERR
JMS ERROR
3042
SCSTRT+700000
LKSTRT+700000
ACSTRT+600000
LKEND+700000
ACEND+600000
0
NSNERR LAC LKSTRT
ISZ LKSTRT
SNA
JMP SGNSHF+6
DZM LKSTRT
LAC BIT1
SAD ACSTRT
JMP HSALS
DAC ACSTRT
ISZ MQSTRT
JMP SGNSHF+6
.EJECT

/LK TO 0 FIRST
/TO COMPARE LINK ONLY
/FIRST AC0=1 GOES TO 0
/SHIFT=1 0PLACE
/MAKE L=START
/AC0=1 GOES TO 0 OR = 0 GOES TO 1
/GET SIGN OF AC
/SAVE FOR TYPEOUTS
/SAVE FOR TYPEOUTS
/L=CORRECT RESULT
/YES
/THIS PASS L=1
/TESTED SIGN=1
/YES

/WILL ALS SHIFT 1 TO 18 PLACES?
/1ST PASS BIT 2ND PASS NO BIT

01625 R 140041 R
01626 R 201000 R
01627 R 040040 R
01630 R 140042 R
01631 R 101722 R
01632 R 200750 R
01633 R 044547 R
01634 R 201463 R
01635 R 041645 R
01636 R 201000 R
01637 R 040043 R
01640 R 200042 R
01641 R 740020 A
01642 R 200041 R
01643 R 652000 A
01644 R 200040 R
01645 R 640701 A
01646 R 040044 R
01647 R 750010 A
01650 R 040046 R
01651 R 641001 A
01652 R 040047 R
01653 R 740200 A
01654 R 601662 R
01655 R 200046 R
01656 R 540042 R
01657 R 200044 R
01660 R 561753 R
01661 R 601663 R
01662 R 101702 R
01663 R 200041 R
01664 R 740001 A
01665 R 040041 R
01666 R 441645 R
01667 R 441753 R
01670 R 440043 R
01671 R 444547 R
01672 R 601640 R
01673 R 440042 R
01674 R 200040 R
01675 R 740001 A
01676 R 040040 R
01677 R 741100 A
01700 R 601631 R
01701 R 601777 R

HSALS DZM MQSTRT
LAC BIT17
DAC ACSTRT
DZM LKSTRT
JMS SIMALS
LAC K18
DAC A10
LAC KALS01
DAC HSALSE
LAC BIT17
DAC SCSTRT
HSALS LAC LKSTRT
RAR
LAC MQSTRT
LMQ
LAC ACSTRT
HSALSE ALS 01
DAC ACEND
GLK
DAC LKEND
LACS
DAC SCEND
SZA
JMP .+6
LAC LKEND
SAD LKSTRT
LAC ACEND
SAD* SALS RP
JMP .+2
JMS ALSERR
LAC MQSTRT
CMA
DAC MQSTRT
ISZ HSALSE
ISZ SALS RP
ISZ SCSTRT
ISZ A10
JMP HSALS L
ISZ LKSTRT
LAC ACSTRT
CMA
DAC ACSTRT
SPA
JMP HSALS+4
JMP LLSTS1
.EJECT

/MQ ALTERNATES
/FROM 1'S TO 0'S

/1 TO 18 PLACES

/SC GO TO ZERO?

/WAS LINK ALTERED

/RESULT OF SHIFT OK?

/EVEN PASSES MQ = 777777

/INCREMENT COUNT
/ADVANCE RESULT POINTER
/FOR TYPEOUTS SC+1
/SHIFT 18 TIMES?

/NO BIT PASS L = 1

/2ND PASS AC STRT=777776

/MADE 2 PASSES?
/NO, SHIFT NO BIT

/ALS INSTRUCTION
/COMMON ERROR TYPEOUT

01702	R	601702	R	ALSERR	JMP .
01703	R	100704	R		JMS ERROR
01704	R	003175	A		3175
01705	R	500042	R		LKSTRT+500000
01706	R	600040	R		ACSTRT+600000
01707	R	600041	R		MQSTRT+600000
01710	R	700043	R		SCSTRT+700000
01711	R	000000	A		0
01712	R	100704	R		JMS ERROR
01713	R	003175	A		3175
01714	R	500046	R		LKEND+500000
01715	R	600044	R		ACEND+600000
01716	R	600045	R		MQEND+600000
01717	R	700047	R		SCEND+700000
01720	R	000000	A		0
01721	R	621702	R		JMP* ALSERR

/SIMULATE ALS OPERATION
/STORES SHIFTS 1 TO 18 PLACES

01722	R	601722	R	SIMALS	JMP .
01723	R	204570	R		LAC (RESULT-1
01724	R	044556	R		DAC A17
01725	R	044554	R		DAC A15
01726	R	341000	R		TAD BIT17
01727	R	041753	R		DAC SALSRP
01730	R	200750	R		LAC K18
01731	R	044555	R		DAC A16
01732	R	200042	R		LAC LKSTRT
01733	R	740020	A		RAR
01734	R	200040	R		LAC ACSTRT
01735	R	740010	A		RAL
01736	R	444556	R		ISZ A17
01737	R	064556	R		DAC* A17
01740	R	444555	R		ISZ A16
01741	R	200042	R		LAC LKSTRT
01742	R	740020	A		RAR
01743	R	444554	R		ISZ A15
01744	R	224554	R		LAC* A15
01745	R	740010	A		RAL
01746	R	444556	R		ISZ A17
01747	R	064556	R		DAC* A17
01750	R	444555	R		ISZ A16
01751	R	601741	R		JMP .-10
01752	R	621722	R		JMP* SIMALS
01753	R	000000	A	SALSRP	0
01754	R	000000	A	RESULT	0

/SET RESULT POINTER TO START

/
.BLOCK 22
.EJECT

/RESERVE 17 SHIFT LOCATIONS.

/LONG LEFT SHIFT

/

/

/LLS 01 ALL ZERO'S

/

01777	R	140040	R	LLSTS1	DZM ACSTRT	
02000	R	100073	R		JMS	MONBRK
02001	R	140041	R		DZM MQSTRT	
02002	R	140042	R		DZM LKSTRT	
02003	R	201000	R		LAC BIT17	
02004	R	040043	R		DAC SCSTRT	
02005	R	650000	A		CLQ	/START SCOPE LOOP
02006	R	754000	A		CLA!CLL	/CLR AC AND LINK
02007	R	640601	A		LLS 01	
02010	R	040044	R		DAC ACEND	
02011	R	750010	A		GLK	
02012	R	040046	R		DAC LKEND	
02013	R	641001	A		LACS	
02014	R	040047	R		DAC SCEND	
02015	R	641002	A		LACQ	
02016	R	040045	R		DAC MQEND	
02017	R	741200	A		SNA	/MQ STILL 0'S?
02020	R	200044	R		LAC ACEND	
02021	R	741200	A		SNA	/AC STILL 0'S?
02022	R	200046	R		LAC LKEND	
02023	R	741200	A		SNA	/LINK STILL 0'S?
02024	R	200047	R		LAC SCEND	
02025	R	741200	A		SNA	/SC GO TO ZERO?
02026	R	602030	R		JMP .+2	
02027	R	102417	R		JMS LLSERR	
					.EJECT	

/DOES LINK GO TO MQ17 ON AN LLS
/0-0, 1-0, 0-1, 1-1

02030 R 140041 R
02031 R 140040 R
02032 R 140042 R
02033 R 201000 R
02034 R 040043 R
02035 R 200041 R
02036 R 652000 A
02037 R 200042 R
02040 R 740020 A
02041 R 200040 R
02042 R 640601 A
02043 R 040044 R
02044 R 750010 A
02045 R 040046 R
02046 R 641001 A
02047 R 040047 R
02050 R 641002 A
02051 R 040045 R
02052 R 200042 R
02053 R 740020 A
02054 R 200041 R
02055 R 740010 A
02056 R 540045 R
02057 R 741000 A
02060 R 102417 R
02061 R 200042 R
02062 R 440042 R
02063 R 741200 A
02064 R 602035 R
02065 R 140042 R
02066 R 200041 R
02067 R 440041 R
02070 R 741200 A
02071 R 602035 R

LLSTS2. DZM MQSTRT
DZM ACSTRT
DZM LKSTRT
LAC BIT17
DAC SCSTRT
LAC MQSTRT
LMQ
LAC LKSTRT
RAR
LAC ACSTRT
LLS 01
DAC ACEND
GLK
DAC LKEND
LACS
DAC SCEND
LACQ
DAC MQEND
LAC LKSTRT
RAR
LAC MQSTRT
RAL
SAD MQEND
SKP
JMS LLSERR
LAC LKSTRT
ISZ LKSTRT
SNA
JMP LLSTS2+5
DZM LKSTRT
LAC MQSTRT
ISZ MQSTRT
SNA
JMP LLSTS2+5
.EJECT

/LLS 01

/2 PASSES = 0 START SCOPE LOOP

/2 PASSES = 1 (MQ17)

/L=1 EVERY 2ND PASS

/AC ALWAYS = 0

/SAVE RESULTS

/2ND OR 4TH PASS?

/NEXT PASS L = 0

/MADE WITH MQ17=1?

/DOES LINK NOT GO TO AC17 ON AN LLS

/DOES MQ0 GO TO AC17 ON AN LLS

/

```

02072 R 140040 R LLSACT DZM ACSTRT
02073 R 140041 R DZM MQSTRT
02074 R 140042 R DZM LKSTRT
02075 R 201000 R LAC BIT17
02076 R 040043 R DAC SCSTRT
02077 R 200041 R LAC MQSTRT /START SCOPE LOOP
02100 R 652000 A LMQ
02101 R 200042 R LAC LKSTRT
02102 R 740020 A RAR /L=0, 1, 0, 1
02103 R 200040 R LAC ACSTRT /AC=0, 0, 1, 1
02104 R 640601 A LLS 01
02105 R 040044 R DAC ACEND
02106 R 750010 A GLK
02107 R 040046 R DAC LKEND
02110 R 641001 A LACS
02111 R 040047 R DAC SCEND /SAVE SC FOR TYPEOUT
02112 R 641002 A LACQ
02113 R 040045 R DAC MQEND /MQ FOR TYPEOUT
02114 R 540042 R SAD LKSTRT /LINK TO MQ17?
02115 R 741000 A SKP /YES, OK
02116 R 602125 R JMP .+7 /MQ ERROR
02117 R 200041 R LAC MQSTRT
02120 R 740010 A RAL
02121 R 200040 R LAC ACSTRT
02122 R 740010 A RAL
02123 R 540044 R SAD ACEND /AC0 SHOULD BE = MQ0
02124 R 741000 A SKP
02125 R 102417 R JMS LLSERR
02126 R 200042 R LAC LKSTRT
02127 R 440042 R ISZ LKSTRT /L=0,1,0, 1,0, 1, 0, 1
02130 R 741200 A SNA
02131 R 602077 R JMP LLSACT+5
02132 R 140042 R DZM LKSTRT
02133 R 200040 R LAC ACSTRT
02134 R 440040 R ISZ ACSTRT /AC0 = 0, 0, 1, 1, 0, 0, 1, 1
02135 R 741200 A SNA
02136 R 602077 R JMP LLSACT+5
02137 R 140040 R DZM ACSTRT
02140 R 200770 R LAC BIT0
02141 R 540041 R SAD MQSTRT /TESTED MQ0 = 1?
02142 R 602145 R JMP .+3 /YES
02143 R 040041 R DAC MQSTRT /MQ0 = 0, 4 PASSES
02144 R 602077 R JMP LLSACT+5 /=1, 4 PASSES
.EJECT

```


/WILL EACH BIT OF THE MQ SHIFT TO THE NEXT
 /1-0, AND 0-1 LEFT
 /

```

02145 R 201000 R          LLSTS3  LAC BIT17          /START MQ 17 TO MQ 16
02146 R 040041 R          DAC MQSTRT
02147 R 040043 R          DAC SCSTRT
02150 R 140042 R          DZM LKSTRT
02151 R 140040 R          DZM ACSTRT
02152 R 200041 R          LAC MQSTRT          /START SCOPE LOOP
02153 R 652000 A          LMQ
02154 R 754000 A          CLA!CLL          /AC AND L ALWAYS 0'S
02155 R 640601 A          LLS 01
02156 R 040044 R          DAC ACEND
02157 R 750010 A          GLK
02160 R 040046 R          DAC LKEND          /FOR TYPEDUTS
02161 R 641001 A          LACS
02162 R 040047 R          DAC SCEND          /FOR TYPEDUTS
02163 R 641002 A          LACQ
02164 R 040045 R          DAC MQEND
02165 R 200041 R          LAC MQSTRT
02166 R 740010 A          RAL
02167 R 540045 R          SAD MQEND
02170 R 741000 A          SKP
02171 R 602176 R          JMP .+5
02172 R 200040 R          LAC ACSTRT
02173 R 740010 A          RAL
02174 R 540044 R          SAD ACEND
02175 R 602177 R          JMP .+2
02176 R 102417 R          JMS LLSERR
02177 R 200041 R          LAC MQSTRT          /SET UP NEXT MQ BIT
02200 R 744010 A          CLL!RAL
02201 R 040041 R          DAC MQSTRT
02202 R 741400 A          SZL          /TESTED MQ0 = 1
02203 R 602206 R          JMP .+3
02204 R 100073 R          JMS MONBRK
02205 R 602152 R          JMP LLSTS3+5
                                .EJECT

```

/WILL EACH BIT OF THE MQ SHIFT TO THE NEXT
/1-1, 0-1, 1-0 LEFT

```

/
02206 R 201022 R LLSTS4 LAC NBIT17 /START 777776
02207 R 040041 R DAC MQSTRT
02210 R 740001 A CMA
02211 R 040043 R DAC SCSTRT /LLS 01
02212 R 040042 R DAC LKSTRT /LINK ALWAYS = 1
02213 R 750001 A CLC
02214 R 040040 R DAC ACSTRT /AC = 1'S ALL
02215 R 200041 R LAC MQSTRT /START SCOPE LOOP
02216 R 652000 A LMQ
02217 R 754003 A STLCLC
02220 R 640001 A LLS 01
02221 R 040044 R DAC ACEND
02222 R 750010 A GLK
02223 R 040046 R DAC LKEND /L, FOR TYPEOUT
02224 R 641001 A LACS
02225 R 040047 R DAC SCEND /SC FOR TYPEOUT
02226 R 641002 A LACQ
02227 R 040045 R DAC MQEND
02230 R 200041 R LAC MQSTRT /SIMULATE LLS
02231 R 744002 A STL /TO GET
02232 R 740010 A RAL /COMPARE CONSTANT
02233 R 540045 R SAD MQEND /MQ SHIFT OK?
02234 R 741000 A SKP /YES
02235 R 602242 R JMP .+5
02236 R 200040 R LAC ACSTRT
02237 R 740010 A RAL
02240 R 540044 R SAD ACEND /AC SHIFT OK?
02241 R 602243 R JMP .+2
02242 R 102417 R JMS LLSERR
02243 R 744002 A STL
02244 R 200041 R LAC MQSTRT
02245 R 740010 A RAL
02246 R 040041 R DAC MQSTRT
02247 R 740400 A SNL /TESTED MQ0 = 0
02250 R 602253 R JMP .+3
02251 R 100073 R JMS MONBRK
02252 R 602215 R JMP LLSTS4+7
.EJECT

```

/WILL MQ AC SHIFT A 1 BIT 1 TO 44 PLACES

/USES LLS SIGNED

/

02253	R	140040	R	LLSTSS	DZM ACSTRT	/AC START ZEROS
02254	R	201000	R		LAC BIT17	
02255	R	040043	R		DAC SCSTRT	/SC INCREMENTED TO 44
02256	R	040041	R		DAC MQSTRT	/MQ START BIT 17 = 1
02257	R	040042	R		DAC LKSTRT	
02260	R	043314	R		DAC MQCOMK	
02261	R	143313	R		DZM ACCOMK	
02262	R	202440	R		LAC KLLSS1	
02263	R	042275	R		DAC LLSSEX	/RESET SHIFT TO 1
02264	R	203314	R	LLSSL1	LAC MQCOMK	
02265	R	744010	A		CLL!RAL	
02266	R	043314	R		DAC MQCOMK	
02267	R	203313	R		LAC ACCOMK	
02270	R	740010	A		RAL	
02271	R	043313	R		DAC ACCOMK	/START SCOPE LOOP
02272	R	200041	R		LAC MQSTRT	
02273	R	652000	A		LMQ	
02274	R	754002	A		STL!CLA	
02275	R	660601	A	LLSSEX	LLSS 01	/SC = 1 TO 44
02276	R	040044	R		DAC ACEND	
02277	R	641001	A		LACS	
02300	R	040047	R		DAC SCEND	
02301	R	750010	A		GLK	
02302	R	040046	R		DAC LKEND	
02303	R	641002	A		LACQ	
02304	R	040045	R		DAC MQEND	
02305	R	543314	R		SAD MQCOMK	
02306	R	741000	A		SKP	
02307	R	602313	R		JMP .+4	
02310	R	203313	R		LAC ACCOMK	
02311	R	540044	R		SAD ACEND	
02312	R	741000	A		SKP	
02313	R	602321	R		JMP .+6	
02314	R	200046	R		LAC LKEND	
02315	R	741200	A		SNA	/LINK GO TO 0
02316	R	200047	R		LAC SCEND	
02317	R	741200	A		SNA	/SC END = 0
02320	R	602322	R		JMP .+2	
02321	R	102441	R		JMS LLSSER	/END SCOPE LOOP
02322	R	442275	R		ISZ LLSSEX	
02323	R	440043	R		ISZ SCSTRT	
02324	R	200043	R		LAC SCSTRT	
02325	R	240763	R		XOR FOURS	
02326	R	741200	A		SNA	
02327	R	602332	R		JMP .+3	
02330	R	100073	R		JMS MONBRK	
02331	R	602264	R		JMP LLSSL1	
					.EJECT	

```

/WILL MQ AC SHIFT A NO BIT 1 TO 44 PLACES
02332 R 140042 R LLSTS6 DZM LKSTRT
02333 R 201000 R LAC BIT17
02334 R 040043 R DAC SCSTRT
02335 R 740001 A CMA
02336 R 040041 R DAC MQSTRT
02337 R 043314 R UAC MQCOMK
02340 R 750001 A CLC
02341 R 040040 R DAC ACSTRT
02342 R 043313 R DAC ACCOMK
02343 R 202440 R LAC KLLSS1
02344 R 042357 R DAC LLSSX2
02345 R 203314 R LLSSL2 LAC MQCOMK /FORM AC
02346 R 744002 A STL
02347 R 740010 A RAL /AND MQ
02350 R 043314 R DAC MQCOMK /COMPARE CONSTANTS
02351 R 203313 R LAC ACCOMK
02352 R 740010 A RAL
02353 R 043313 R DAC ACCOMK
02354 R 200041 R LAC MQSTRT /SET UP SHIFT START SCOPE LOOP
02355 R 652000 A LMQ
02356 R 754001 A CLLCLC
02357 R 660601 A LLSSX2 LLSS 01 /SC=1 TO 44 PLACES
02360 R 040044 R DAC ACEND
02361 R 641001 A LACS
02362 R 040047 R DAC SCEND /GET SC FOR TEST 4
02363 R 750010 A GLK
02364 R 040046 R DAC LKEND /LINK SHOULD BE 1
02365 R 641002 A LACQ
02366 R 040045 R DAC MQEND
02367 R 543314 R SAU MQCOMK /MQ SHIFT OK?
02370 R 741000 A SKP
02371 R 602375 R JMP .+4
02372 R 203313 R LAC ACCOMK
02373 R 540044 R SAU ACEND /AC SHIFT OK?
02374 R 741000 A SKP
02375 R 602401 R JMP .+4
02376 R 200046 R LAC LKEND
02377 R 541000 R SAU BIT17 /LINK SET TO 1?
02400 R 741000 A SKP
02401 R 602405 R JMP .+4
02402 R 200047 R LAC SCEND
02403 R 741200 A SNA /SC GO TO 0?
02404 R 741000 A SKP
02405 R 102441 R JMS LLSSER
02406 R 442357 R ISZ LLSSX2 /ADVANCE TO NEXT SHIFT
02407 R 440043 R ISZ SCSTRT
02410 R 200043 R LAC SCSTRT
02411 R 240763 R XOR FOURS /SHIFTED 44 PLACES?
02412 R 741200 A SNA
02413 R 602410 R JMP .+3
02414 R 100073 R JMS MONBRK
02415 R 602345 R JMP LLSSL2

```

PAGE 37 EAAPT1 SRC EAE-P1

02416 R 602464 R

JMP LRSTS1
.EJECT

/COMMON ERROR TYPEOUT LLS

/

02417	R	602417	R	LLSERR	JMP .
02420	R	100704	R		JMS ERROR
02421	R	003745	A		3745
02422	R	700043	R		SCSTRT+700000
02423	R	402417	R		LLSERR+400000
02424	R	500042	R		LKSTRT+500000
02425	R	600040	R		ACSTRT+600000
02426	R	600041	R		MQSTRT+600000
02427	R	000000	A		0
02430	R	100704	R		JMS ERROR
02431	R	003745	A		3745
02432	R	500046	R		LKEND+500000
02433	R	600044	R		ACEND+600000
02434	R	600045	R		MQEND+600000
02435	R	700047	R		SCEND+700000
02436	R	000000	A		0
02437	R	622417	R		JMP* LLSERR
02440	R	660601	A	KLLSS1	LLSS 01

/TO SET UP LONG LEFT SHIFTS

/

/COMMON ERROR TYPEOUT

/LLS SIGNED

/

02441	R	602441	R	LLSSER	JMP .
02442	R	100704	R		JMS ERROR
02443	R	003770	A		3770
02444	R	700043	R		SCSTRT+700000
02445	R	402441	R		LLSSER+400000
02446	R	500042	R		LKSTRT+500000
02447	R	600040	R		ACSTRT+600000
02450	R	600041	R		MQSTRT+600000
02451	R	000000	A		0
02452	R	100704	R		JMS ERROR
02453	R	003770	A		3770
02454	R	603313	R		ACCOMK+600000
02455	R	603314	R		MQCOMK+600000
02456	R	500046	R		LKEND+500000
02457	R	600044	R		ACEND+600000
02460	R	600045	R		MQEND+600000
02461	R	700047	R		SCEND+700000
02462	R	000000	A		0
02463	R	622441	R		JMP* LLSSER
					.EJECT

/LONG RIGHT SHIFT
/LRS 01 AC, MQ AND L = 0'S

```

/
02464 R 140040 R      LRSTS1  DZM ACSTRT      /SET INITIAL CONDITIONS
02465 R 140041 R      DZM MQSTRT
02466 R 140042 R      DZM LKSTRT
02467 R 201000 R      LAC BIT17
02470 R 040043 R      DAC SCSTRT
02471 R 650000 A      CLQ          /START SCOPE LOOP
02472 R 754000 A      CLAI CLL
02473 R 640501 A      LRS 01
02474 R 040044 R      DAC ACEND
02475 R 750010 A      GLK
02476 R 040046 R      DAC LKEND
02477 R 641001 A      LACS
02500 R 040047 R      DAC SCEND
02501 R 641002 A      LACQ
02502 R 040045 R      DAC MQEND
02503 R 741200 A      SNA          /MQ SHOULD BE 0
02504 R 200044 R      LAC ACEND
02505 R 741200 A      SNA          /AC=0?
02506 R 200047 R      LAC SCEND
02507 R 741200 A      SNA          /SC GO TO 0?
02510 R 200046 R      LAC LKEND
02511 R 741200 A      SNA          /LINK STILL 0?
02512 R 741000 A      SKP
02513 R 103247 R      JMS LRSERR
                        .EJECT

```

/DOES LINK GO TO AC 0 ON AN LRS

/0-0, 1-0, 0-1, 1-1

02514	R	140041	R	LRSTS2	DZM	MQSTRT	
02515	R	140040	R		DZM	ACSTRT	
02516	R	140042	R		DZM	LKSTRT	
02517	R	201000	R		LAC	BIT17	
02520	R	040043	R		DAC	SCSTRT	
02521	R	200042	R		LAC	LKSTRT	/START SCOPE LOOP
02522	R	740020	A		RAR		
02523	R	650000	A		CLQ		
02524	R	200040	R		LAC	ACSTRT	/SET UP COMPLETE
02525	R	640501	A		LRS	01	
02526	R	040044	R		DAC	ACEND	/SAVE RESULTS
02527	R	641002	A		LACQ		
02530	R	040045	R		DAC	MQEND	
02531	R	641001	A		LACS		
02532	R	040047	R		DAC	SCEND	
02533	R	750010	A		GLK		
02534	R	040046	R		DAC	LKEND	/LINK SHOULD NOT CHANGE
02535	R	540042	R		SAD	LKSTRT	
02536	R	741000	A		SKP		
02537	R	602551	R		JMP	+.12	
02540	R	200045	R		LAC	MQEND	
02541	R	740200	A		SZA		
02542	R	602551	R		JMP	+.7	
02543	R	200042	R		LAC	LKSTRT	
02544	R	740020	A		RAR		
02545	R	200040	R		LAC	ACSTRT	
02546	R	740020	A		RAR		
02547	R	540044	R		SAD	ACEND	
02550	R	741000	A		SKP		
02551	R	103247	R		JMS	LRSEERR	
02552	R	200042	R		LAC	LKSTRT	
02553	R	440042	R		ISZ	LKSTRT	
02554	R	741200	A		SNA		
02555	R	602521	R		JMP	LRSTS2+5	
02556	R	140042	R		OZM	LKSTRT	
02557	R	200770	R		LAC	BIT0	
02560	R	540040	R		SAD	ACSTRT	
02561	R	602564	R		JMP	+.3	
02562	R	040040	R		DAC	ACSTRT	
02563	R	602521	R		JMP	LRSTS2+5	
					.EJECT		

/DOES AC17 GO TO MQW ON AN LRS

/0-0, 1-0, 0-1, AND 1-1

```

02564 R 140042 R LRSTS3 DZM LKSTRT /LINK ALWAYS 0
02565 R 140041 R DZM MQSTRT
02566 R 140040 R DZM ACSTRT
02567 R 201000 R LAC BIT17 /SHIFT OF 1
02570 R 040043 R DAC SCSTRT
02571 R 744000 A CLL
02572 R 200041 R LAC MQSTRT /SET MQ
02573 R 652000 A LMQ
02574 R 200040 R LAC ACSTRT
02575 R 640501 A LRS 01
02576 R 040044 R DAC ACEND
02577 R 750010 A GLK
02600 R 040046 R DAC LKEND
02601 R 641001 A LACS
02602 R 040047 R DAC SCEND
02603 R 641002 A LACQ
02604 R 040045 R DAC MQEND
02605 R 200040 R LAC ACSTRT /GENERATE MQ
02606 R 740020 A RAR /COMPARE
02607 R 200041 R LAC MQSTRT /CONSTANT
02610 R 740020 A RAR
02611 R 540045 R SAD MQEND /AC17 TO MQW OK?
02612 R 741000 A SKP
02613 R 602616 R JMP .+3
02614 R 200044 R LAC ACEND
02615 R 740200 A SZA /AC GO TO 0?
02616 R 602622 R JMP .+4
02617 R 200046 R LAC LKEND
02620 R 741200 A SNA
02621 R 741000 A SKP
02622 R 103247 R JMS LRSERR
02623 R 200040 R LAC ACSTRT
02624 R 440040 R ISZ ACSTRT
02625 R 741200 A SNA
02626 R 602571 R JMP LRSTS3+5
02627 R 200770 R LAC BIT0
02630 R 140040 R DZM ACSTRT
02631 R 540041 R SAD MQSTRT
02632 R 602635 R JMP .+3
02633 R 040041 R DAC MQSTRT
02634 R 602571 R JMP LRSTS3+5
.EJECT

```

/DOES AC17 NOT GO TO LINK ON AN LRS

02635	R	140042	R	LRSTS4	DZM LKSTRT	
02636	R	140040	R		DZM ACSTRT	
02637	R	140041	R		DZM MQSTRT	/MQ ALWAYS ZERO
02640	R	201000	R		LAC BIT17	
02641	R	040043	R		DAC SCSTRT	/SHIFT OF 1
02642	R	650000	A		CLQ	
02643	R	200042	R		LAC LKSTRT	
02644	R	740020	A		RAR	/SET LINK INITIAL 0 OR 1
02645	R	200040	R		LAC ACSTRT	/AC=1 OR 0
02646	R	640501	A		LRS 01	
02647	R	040044	R		DAC ACEND	
02650	R	641001	A		LACS	
02651	R	040047	R		DAC SCEND	
02652	R	641002	A		LACQ	
02653	R	040045	R		DAC MQEND	
02654	R	750010	A		GLK	
02655	R	040046	R		DAC LKEND	
02656	R	540042	R		SAD LKSTRT	/WAS LINK ALTERED
02657	R	741000	A		SKP	
02660	R	103247	R		JMS LRSERR	
02661	R	200042	R		LAC LKSTRT	
02662	R	440042	R		ISZ LKSTRT	
02663	R	741200	A		SNA	/TESTED L=1?
02664	R	602642	R		JMP LRSTS4+5	/NO
02665	R	140042	R		DZM LKSTRT	
02666	R	200040	R		LAC ACSTRT	
02667	R	440040	R		ISZ ACSTRT	
02670	R	741200	A		SNA	/TESTED AC 17=1
02671	R	602642	R		JMP LRSTS4+5	
					.EJECT	

/WILL AC MQ SHIFT A 1 BIT EACH POSITION RIGHT

/

```

02672 R 140042 R LRS1S5 DZM LKSTRT
02673 R 140041 R DZM MQSTRT
02674 R 143314 R DZM MQCOMK
02675 R 201000 R LAC BIT17
02676 R 040043 R DAC SCSTRT
02677 R 200770 R LAC BIT0
02700 R 040040 R DAC ACSTRT
02701 R 043313 R DAC ACCOMK
02702 R 203313 R LAC ACCOMK
02703 R 744020 A CLL!RAR
02704 R 043313 R DAC ACCOMK
02705 R 203314 R LAC MQCOMK
02706 R 740020 A RAR
02707 R 043314 R DAC MQCOMK
02710 R 200041 R LRST5L LAC MQSTRT
02711 R 652000 A LMQ
02712 R 744000 A CLL
02713 R 200040 R LAC ACSTRT
02714 R 640501 A LRS 01
02715 R 040044 R DAC ACEND
02716 R 750010 A GLK
02717 R 040046 R DAC LKEND
02720 R 641001 A LACS
02721 R 040047 R DAC SCEND
02722 R 641002 A LACQ
02723 R 040045 R DAC MQEND
02724 R 543314 R SAD MQCOMK
02725 R 741000 A SKP
02726 R 602732 R JMP .+4
02727 R 203313 R LAC ACCOMK
02730 R 540044 R SAD ACEND
02731 R 741000 A SKP
02732 R 103247 R JMS LRSERR
02733 R 200040 R LAC ACSTRT
02734 R 744020 A CLL!RAR
02735 R 040040 R DAC ACSTRT
02736 R 200041 R LAC MQSTRT
02737 R 740020 A RAR
02740 R 040041 R DAC MQSTRT
02741 R 741400 A SZL
02742 R 602745 R JMP .+3
02743 R 100073 R JMS MONBRK
02744 R 602702 R JMP LRST5L-6
.EJECT

```

/GENERATE COMPARE
/CONSTANTS

/MQ SHIFT OK?

/AC SHIFT OK?

/WILL AC-MQ SHIFT A NO BIT 1 POSITION
/RIGHT FROM EACH BIT
LRSTS6 LAC BIT17

02745 R 201000 R
02746 R 040043 R
02747 R 040042 R
02750 R 201001 R
02751 R 040040 R
02752 R 043313 R
02753 R 750001 A
02754 R 040041 R
02755 R 043314 R
02756 R 203313 R
02757 R 744002 A
02760 R 740020 A
02761 R 043313 R
02762 R 203314 R
02763 R 740020 A
02764 R 043314 R
02765 R 200041 R
02766 R 652000 A
02767 R 744002 A
02770 R 200040 R
02771 R 640501 A
02772 R 040044 R
02773 R 641001 A
02774 R 040047 R
02775 R 750010 A
02776 R 040046 R
02777 R 641002 A
03000 R 040045 R
03001 R 543314 R
03002 R 741000 A
03003 R 603007 R
03004 R 203313 R
03005 R 540044 R
03006 R 741000 A
03007 R 103247 R
03010 R 200040 R
03011 R 744002 A
03012 R 740020 A
03013 R 040040 R
03014 R 200041 R
03015 R 740020 A
03016 R 040041 R
03017 R 740400 A
03020 R 603023 R
03021 R 100073 R
03022 R 602756 R

DAC SCSTRT
DAC LKSTRT
LAC NBIT0
DAC ACSTRT
DAC ACCOMK
CLC
DAC MQSTRT
DAC MQCOMK
LAC ACCOMK
STL
RAR
DAC ACCOMK
LAC MQCOMK
RAR
DAC MQCOMK
LRST6L LAC MQSTRT
LMQ
STL
LAC ACSTRT
LRS 01
DAC ACEND
LACS
DAC SCEND
GLK
DAC LKEND
LACQ
DAC MQEND
SAD MQCOMK
SKP
JMP .+4
LAC ACCOMK
SAD ACEND
SKP
JMS LRSERR
LAC ACSTRT
STL
RAR
DAC ACSTRT
LAC MQSTRT
RAR
DAC MQSTRT
SNL
JMP .+3
JMS MUNBRK
JMP LRST6L-7
.EJECT

/377777

/GENERATE NEXT

/SET OF
/AC MQ COMPARE
/CONSTANTS

/SET UP LRS

/FOR TYPEOUTS

/FOR TYPEOUTS

/MQ SHIFT OK?

/AC SHIFT OK?

/SHIFTED TILL MQ17=0

/WILL AC MQ SHIFT A 1 BIT
 /RIGHT TO 44 PLACES
 /

03023	R	140041	R	LRSTS7	DZM MQSTRT	
03024	R	140042	R		DZM LKSTRT	
03025	R	201000	R		LAC BIT17	
03026	R	040043	R		DAC SCSTRT	
03027	R	200770	R		LAC BIT0	
03030	R	040040	R		DAC ACSTRT	
03031	R	043313	R		DAC ACCOMK	
03032	R	143314	R		DZM MQCOMK	
03033	R	202771	R		LAC LRST6L+4	/LRS 01
03034	R	043047	R		DAC LRST7E	/FOR EXECUTE
03035	R	203313	R	LRST7L	LAC ACCOMK	
03036	R	744020	A		CLL!RAR	/GENERATE AC/MQ
03037	R	043313	R		DAC ACCOMK	/COMPARE CONSTANTS
03040	R	203314	R		LAC MQCOMK	
03041	R	740020	A		RAR	
03042	R	043314	R		DAC MQCOMK	
03043	R	200041	R		LAC MQSTRT	/SET UP LRS
03044	R	652000	A		LMQ	
03045	R	744000	A		CLL	
03046	R	200040	R		LAC ACSTRT	
03047	R	640501	A	LRST7E	LRS 01	/1 TO 44 PLACES
03050	R	040044	R		DAC ACEND	
03051	R	750010	A		GLK	
03052	R	040046	R		DAC LKEND	
03053	R	641001	A		LACS	
03054	R	040047	R		DAC SCEND	
03055	R	641002	A		LACQ	
03056	R	040045	R		DAC MQEND	
03057	R	543314	R		SAD MQCOMK	/MQ SHIFT OK?
03060	R	741000	A		SKP	
03061	R	603065	R		JMP .+4	
03062	R	203313	R		LAC ACCOMK	
03063	R	540044	R		SAD ACEND	/AC END OK?
03064	R	741000	A		SKP	
03065	R	103270	R		JMS LRSER1	
03066	R	443047	R		ISZ LRST7E	/INCREMENT SHIFT COUNT
03067	R	440043	R		ISZ SCSTRT	/FOR TYPEOUTS
03070	R	200763	R		LAC FOUR5	
03071	R	540043	R		SAD SCSTRT	/SHIFTED 44 PLACES?
03072	R	603075	R		JMP .+3	/YES
03073	R	100073	R		JMS MONBRK	
03074	R	603035	R		JMP LRST7L	
					.EJECT	

/WILL AC MQ SHIFT A NO BIT RIGHT
/1 TO 44 PLACES

/

03075 R 750001 A	LRSTS8	CLC	
03076 R 040041 R		DAC MQSTRT	/MQ START = 1'S
03077 R 043314 R		DAC MQCOMK	
03100 R 201001 R		LAC NBIT0	/AC START BIT 0=0
03101 R 040040 R		DAC ACSTRT	
03102 R 043313 R		DAC ACCOMK	
03103 R 201000 R		LAC BIT17	
03104 R 040043 R		DAC SCSTRT	
03105 R 040042 R		DAC LKSTRT	
03106 R 202771 R		LAC LRST6L+4	/LRS 01
03107 R 043123 R		DAC LRST8E	/FOR EXECUTE
03110 R 203313 R	LRST8L	LAC ACCOMK	/GENERATE
03111 R 744002 A		STL	
03112 R 740020 A		RAR	/NEXT
03113 R 043313 R		DAC ACCOMK	/COMPARE CONSTANTS
03114 R 203314 R		LAC MQCOMK	
03115 R 740020 A		RAR	
03116 R 043314 R		DAC MQCOMK	
03117 R 200041 R		LAC MQSTRT	/SET UP LRS
03120 R 652000 A		LMQ	
03121 R 200040 R		LAC ACSTRT	
03122 R 744002 A		STL	
03123 R 640501 A	LRST8E	LRS 01	/1 TO 44 PLACES
03124 R 040044 R		DAC ACEND	
03125 R 750010 A		GLK	
03126 R 040046 R		DAC LKEND	
03127 R 641001 A		LACS	
03130 R 040047 R		DAC SCEND	
03131 R 641002 A		LACQ	
03132 R 040045 R		DAC MQEND	
03133 R 543314 R		SAD MQCOMK	/MQ SHIFT OK?
03134 R 741000 A		SKP	
03135 R 603141 R		JMP .+4	
03136 R 203313 R		LAC ACCOMK	
03137 R 540044 R		SAD ACEND	/AC SHIFT OK?
03140 R 741000 A		SKP	
03141 R 103270 R		JMS LRSER1	
03142 R 443123 R		ISZ LRST8E	/ADVANCE SHIFT
03143 R 440043 R		ISZ SCSTRT	/COUNT
03144 R 200763 R		LAC FOURS	
03145 R 540043 R		SAD SCSTRT	/SHIFTED 44 PLACES
03146 R 741000 A		SKP	
03147 R 603110 R		JMP LRST8L	
		.EJECT	

/WILL MQ SHIFT LEFT 1
/EVERY COMBINATION OF BITS

03150	R	140041	R	LLSSEQ	DZM MQSTRT	/AC AND MQ WILL
03151	R	140040	R		DZM ACSTRT	/ALWAYS BE =
03152	R	201000	R		LAC BIT17	
03153	R	040043	R		DAC SCSTRT	/SHIFT IS ALWAYS 1
03154	R	140042	R		DZM LKSTRT	
03155	R	200041	R		LAC MQSTRT	
03156	R	660000	A		EAE+20000	
03157	R	740010	A		RAL	
03160	R	043313	R		DAC ACCOMK	/AC SHOULD
03161	R	043314	R		DAC MQCOMK	/=MQ
03162	R	200041	R		LAC MQSTRT	
03163	R	652000	A		LMQ	
03164	R	660601	A		LLSS 01	
03165	R	040044	R		DAC ACEND	
03166	R	641001	A		LACS	
03167	R	040047	R		DAC SCEND	
03170	R	750010	A		GLK	
03171	R	040046	R		DAC LKEND	
03172	R	641002	A		LACQ	
03173	R	040045	R		DAC MQEND	/MQ AND
03174	R	540044	R		SAD ACEND	/AC SHIFT OK
03175	R	741000	A		SKP	
03176	R	102441	R		JMS LLSSER	
03177	R	440040	R		ISZ ACSTRT	
03200	R	740000	A		NOP	
03201	R	440041	R		ISZ MQSTRT	
03202	R	741000	A		SKP	
03203	R	603206	R		JMP .+3	
03204	R	100073	R		JMS NONBRK	
03205	R	603155	R		JMP LLSSEQ+5	
					.EJECT	

```

/WILL MQ SHIFT RIGHT 1 EVERY
/COMBINATION OF BITS
03206 R 140040 R LRSSEQ DZM ACSTRT /AC AND MQ
03207 R 140041 R DZM MQSTRT /ALWAYS =
03210 R 201000 R LAC BIT17
03211 R 040043 R DAC SCSTRT /ALWAYS SHIFT OF 1
03212 R 200040 R LAC ACSTRT
03213 R 501000 R AND BIT17
03214 R 040042 R DAC LKSTRT /LINK = AC 17
03215 R 740020 A RAR /SO THAT AC WILL = MQ
03216 R 200041 R LAC MQSTRT
03217 R 740020 A RAR
03220 R 043314 R DAC MQCOMK /AC AND MQ
03221 R 043313 R DAC ACCOMK /SHOULD BE =
03222 R 740010 A RAL
03223 R 652000 A LMQ
03224 R 640501 A LRS 01
03225 R 040044 R DAC ACEND
03226 R 641001 A LACS
03227 R 040047 R DAC SCEND
03230 R 750010 A GLK
03231 R 040046 R DAC LKEND
03232 R 641002 A LACQ
03233 R 040045 R DAC MQEND
03234 R 540044 R SAD ACEND /AC AND MQ R 1 OK
03235 R 741000 A SKP
03236 R 103270 R JMS LRSER1
03237 R 440040 R ISZ ACSTRT
03240 R 740000 A NOP
03241 R 440041 R ISZ MQSTRT /ALL COMBINATIONS
03242 R 741000 A SKP
03243 R 603246 R JMP .+3
03244 R 100073 R JMS MONBRK
03245 R 603212 R JMP LRSSEQ+4
03246 R 603316 R JMP RANSHF
.EJECT

```


/LRS COMMON ERROR TYPEOUT

/SHIFT OF 1

03247	R	603247	R	LRSEERR	JMP .
03250	R	100704	R		JMS ERROR
03251	R	004630	A		4630
03252	R	700043	R		SCSTRT+700000
03253	R	403247	R		LRSEERR+400000
03254	R	500042	R		LKSTRT+500000
03255	R	600040	R		ACSTRT+600000
03256	R	600041	R		MQSTRT+600000
03257	R	000000	A		0
03260	R	100704	R		JMS ERROR
03261	R	004630	A		4630
03262	R	500046	R		LKEND+500000
03263	R	600044	R		ACEND+600000
03264	R	600045	R		MQEND+600000
03265	R	700047	R		SCEND+700000
03266	R	000000	A		0
03267	R	623247	R		JMP* LRSEERR

/

/LRS COMMON ERROR TYPEOUT

/SHIFTS OF MORE THAN 1

03270	R	603270	R	LRSER1	JMP .
03271	R	100704	R		JMS ERROR
03272	R	004652	A		4652
03273	R	700043	R		SCSTRT+700000
03274	R	403270	R		LRSER1+400000
03275	R	500042	R		LKSTRT+500000
03276	R	600040	R		ACSTRT+600000
03277	R	600041	R		MQSTRT+600000
03300	R	000000	A		0
03301	R	100704	R		JMS ERROR
03302	R	004652	A		4652
03303	R	603313	R		ACCOMK+600000
03304	R	603314	R		MQCOMK+600000
03305	R	500046	R		LKEND+500000
03306	R	600044	R		ACEND+600000
03307	R	600045	R		MQEND+600000
03310	R	700047	R		SCEND+700000
03311	R	000000	A		0
03312	R	623270	R		JMP* LRSER1
03313	R	000000	A	ACCOMK	0
03314	R	000000	A	MQCOMK	0
03315	R	000000	A	SCCOMK	0

.EJECT

/RANDOM DATA SHIFTS
/NORMALIZE TEST
/
/START RANDOM DATA SHIFTS
/LEFT 0 TO 44 PLACES

03316 R 201006 R
03317 R 044040 R
03320 R 104016 R
03321 R 040041 R
03322 R 104016 R
03323 R 044044 R
03324 R 204571 R
03325 R 044547 R
03326 R 341000 R
03327 R 044550 R
03330 R 200041 R
03331 R 652000 A
03332 R 444547 R
03333 R 064547 R
03334 R 204044 R
03335 R 040040 R
03336 R 660601 A
03337 R 444547 R
03340 R 064547 R
03341 R 641002 A
03342 R 444547 R
03343 R 064547 R
03344 R 444550 R
03345 R 224550 R
03346 R 444550 R
03347 R 640601 A
03350 R 444547 R
03351 R 064547 R
03352 R 641002 A
03353 R 444547 R
03354 R 064547 R
03355 R 204572 R
03356 R 544547 R
03357 R 741000 A
03360 R 603344 R
03361 R 750010 A
03362 R 040042 R
03363 R 140043 R
03364 R 204042 R
03365 R 043403 R
03366 R 204573 R
03367 R 044547 R

RANSHF LAC NBIT5
DAC PASSK
JMS RANGEN
DAC MQSTRT
JMS RANGEN
DAC SHFBUF
LAC (SHFBUF
DAC A10
TAD BIT17
DAC A11
LAC MQSTRT
LMQ
ISZ A10
DAC* A10
LAC SHFBUF
DAC ACSTRT
LLSS 01
ISZ A10
DAC* A10
LACQ
ISZ A10
DAC* A10
SETLLS ISZ A11
LAC* A11
ISZ A11
LLS 01
ISZ A10
DAC* A10
LACQ
ISZ A10
DAC* A10
LAC (SHFBUF+111
SAD A10
SKP
JMP SETLLS
GLK
DAC LKSTRT
DZM SCSTRT
LAC KLLSS
DAC LRANEX
LAC (SHFBUF-1
DAC A10
.EJECT

/GENERATE AC START

/SHIFTED 44 PLACES?

```

03370 R 444547 R L RANLP ISZ A10
03371 R 224547 R LAC* A10
03372 R 043313 R DAC ACCOMK
03373 R 444547 R ISZ A10
03374 R 224547 R LAC* A10
03375 R 043314 R DAC MQCOMK
03376 R 200041 R LAC MQSTRT
03377 R 652000 A LMQ
03400 R 200042 R LAC LKSTRT
03401 R 740020 A RAR
03402 R 200040 R LAC ACSTRT
03403 R 660600 A L RANEX LLSS
03404 R 040044 R DAC ACEND
03405 R 750010 A GLK
03406 R 040046 R DAC LKEND
03407 R 641001 A LACS
03410 R 040047 R DAC SCEND
03411 R 641002 A LACQ
03412 R 040045 R DAC MQEND
03413 R 543314 R SAD MQCOMK
03414 R 741000 A SKP
03415 R 603421 R JMP .+4
03416 R 203313 R LAC ACCOMK
03417 R 540044 R SAD ACEND
03420 R 741000 A SKP
03421 R 102441 R JMS LLSSER
03422 R 443403 R ISZ L RANEX
03423 R 440043 R ISZ SCSTRT
03424 R 200763 R LAC FOURS
03425 R 540043 R SAD SCSTRT
03426 R 741000 A SKP
03427 R 603370 R JMP L RANLP
.EJECT

```

/0 TO 44 PLACES

/MQ = PREDICTED?

/AC END = PREDICTED?

/SHIFTED 44 PLACES?

```

/RANDOM DATA RIGHT 0 TO 44 PLACES
03430 R 201000 R RANRIT LAC NBITS
03431 R 044040 R DAC PASSK
03432 R 104016 R JMS RANGEN /GENERATE MQ START
03433 R 040041 R DAC MQSTRT
03434 R 104016 R JMS RANGEN /GENERATE ACSTRT
03435 R 040040 R DAC ACSTRT
03436 R 204573 R LAC (SHFBUF-1
03437 R 044547 R DAC A10
03440 R 044550 R DAC A11
03441 R 200040 R LAC ACSTRT
03442 R 444547 R ISZ A10
03443 R 064547 R DAC* A10
03444 R 200041 R LAC MQSTRT
03445 R 444547 R ISZ A10
03446 R 064547 R DAC* A10
03447 R 652000 A LMQ
03450 R 744000 A CLL
03451 R 444550 R SETLRS ISZ A11
03452 R 224550 R LAC* A11
03453 R 444550 R ISZ A11 /GENERATE AC MQ
03454 R 640501 A LRS 01
03455 R 444547 R ISZ A10
03456 R 064547 R DAC* A10
03457 R 641002 A LACQ
03460 R 444547 R ISZ A10
03461 R 064547 R DAC* A10
03462 R 204572 R LAC (SHFBUF+111
03463 R 544547 R SAD A10
03464 R 741000 A SKP
03465 R 603451 R JMP SETLRS
03466 R 204041 R LAC KLRS
03467 R 043506 R DAC RRANEX
03470 R 140042 R DZM LKSTRT
03471 R 140043 R DZM SCSTRT
03472 R 204573 R LAC (SHFBUF-1
03473 R 044547 R DAC A10
03474 R 444547 R RRANLP ISZ A10
03475 R 224547 R LAC* A10
03476 R 043313 R DAC ACCOMK
03477 R 444547 R ISZ A10
03500 R 224547 R LAC* A10
03501 R 043314 R DAC MQCOMK
03502 R 200041 R LAC MQSTRT
03503 R 652000 A LMQ
03504 R 200040 R LAC ACSTRT
03505 R 744000 A CLL
.EJECT

```

```

03506 R 640500 A RRANEX LRS 0 /0 TO 44 PLACES
03507 R 040044 R DAC ACEND
03510 R 750010 A GLK
03511 R 040046 R DAC LKEND
03512 R 641001 A LACS
03513 R 040047 R DAC SCEND
03514 R 641002 A LACQ
03515 R 040045 R DAC MQEND
03516 R 543314 R SAD MQCOMK
03517 R 741000 A SKP
03520 R 603524 R JMP .+4
03521 R 203313 R LAC ACCOMK
03522 R 540044 R SAD ACEND
03523 R 741000 A SKP
03524 R 103270 R JMS LRSER1
03525 R 443506 R ISZ RRANEX
03526 R 440043 R ISZ SCSTRT
03527 R 200763 R LAC FOUR5
03530 R 540043 R SAD SCSTRT
03531 R 741000 A SKP
03532 R 603474 R JMP RRANLP

```

```

/
/RANDOM DATA SEQUENCED
/

```

```

03533 R 201006 R RANSEQ LAC NBIT5
03534 R 044040 R DAC PASSK
03535 R 104016 R JMS RANGEN
03536 R 040040 R DAC ACSTRT
03537 R 661000 A EAE!21000 /GET AC SIGN CLR AC
03540 R 750010 A GLK
03541 R 043744 R DAC SVSIGN
03542 R 742020 A RTR
03543 R 043745 R DAC SVSIGN+1
03544 R 104016 R JMS RANGEN
03545 R 501022 R AND NBIT17 /MAKE MQ17=AC0
03546 R 343744 R TAD SVSIGN
03547 R 040041 R DAC MQSTRT
03550 R 201022 R LAC NBIT17
03551 R 043746 R DAC SVMASK
03552 R 201001 R LAC NBIT0
03553 R 043747 R DAC SVMASK+1
03554 R 140043 R DZM SCSTRT
03555 R 140042 R DZM LKSTRT
03556 R 200041 R RANSQ0 LAC MQSTRT /SEQUENCE 0
03557 R 652000 A LMQ
03560 R 744000 A CLL
03561 R 200040 R LAC ACSTRT
03562 R 660601 A LLSS 1
03563 R 640502 A LRS 2
03564 R 660602 A LLSS 2
03565 R 640501 A LRS 1
03566 R 103750 R JMS SEQCOM
03567 R 103715 R JMS NXTSEQ

```

PAGE 54

EAEPT1 SRC

EAE-P1

.EJECT

/SEQUENCE 1
/RIGHT 2, L4, R4, L2

```

03570 R 200041 R RANSQ1 LAC MQSTRT /SEQUENCE 1 R2, L4, R4, L2
03571 R 744000 A CLL
03572 R 652000 A LMQ /SET UP
03573 R 200040 R LAC ACSTRT
03574 R 660502 A LRSS+2
03575 R 660604 A LLSS+4
03576 R 640504 A LRS+4
03577 R 660602 A LLSS+2
03600 R 103750 R JMS SEQCOM /COMPARE RESULTS
03601 R 103715 R JMS NXTSEQ

```

/LEFT 3, RIGHT 6, LEFT 6, RIGHT 3
/SEQUENCE 2

```

03602 R 200041 R RANSQ2 LAC MQSTRT
03603 R 652000 A LMQ
03604 R 744000 A CLL
03605 R 200040 R LAC ACSTRT
03606 R 660603 A LLSS+3
03607 R 640506 A LRS+6
03610 R 660606 A LLSS+6
03611 R 640503 A LRS+3
03612 R 103750 R JMS SEQCOM
03613 R 103715 R JMS NXTSEQ

```

/SEQUENCE 3
/RIGHT 4, LEFT 8, RIGHT 8, LEFT 4

```

03614 R 200041 R RANSQ3 LAC MQSTRT
03615 R 744000 A CLL
03616 R 652000 A LMQ
03617 R 200040 R LAC ACSTRT
03620 R 660504 A LRSS+4
03621 R 660610 A LLSS+10
03622 R 640510 A LRS+10
03623 R 660604 A LLSS+4
03624 R 103750 R JMS SEQCOM
03625 R 103715 R JMS NXTSEQ

```

/SEQUENCE 4 LEFT 5, RIGHT 10, LEFT 10, RIGHT 5

```

03626 R 200041 R RANSQ4 LAC MQSTRT
03627 R 744000 A CLL
03630 R 652000 A LMQ
03631 R 200040 R LAC ACSTRT
03632 R 660605 A LLSS+5
03633 R 640512 A LRS+12
03634 R 660612 A LLSS+12
03635 R 640505 A LRS+5
03636 R 103750 R JMS SEQCOM
03637 R 103715 R JMS NXTSEQ
.EJECT

```

/SEQUENCE 5 RIGHT 6, LEFT 12, RIGHT 12, LEFT 6

03640	R	200041	R	RANSQ5	LAC	MQSTRT
03641	R	652000	A		LMQ	
03642	R	744000	A		CLL	
03643	R	200040	R		LAC	ACSTRT
03644	R	660506	A		LRSS+6	
03645	R	660614	A		LLSS+14	
03646	R	640514	A		LRS+14	
03647	R	660606	A		LLSS+6	
03650	R	103750	R		JMS	SEQCOM
03651	R	103715	R		JMS	NXTSEQ

/SEQUENCE 6 LEFT 7 RIGHT 14, LEFT 14, RIGHT 7

03652	R	200041	R	RANSQ6	LAC	MQSTRT
03653	R	652000	A		LMQ	
03654	R	744000	A		CLL	
03655	R	200040	R		LAC	ACSTRT
03656	R	660607	A		LLSS+7	
03657	R	640516	A		LRS+16	
03660	R	660616	A		LLSS+16	
03661	R	640507	A		LRS+7	
03662	R	103750	R		JMS	SEQCOM
03663	R	103715	R		JMS	NXTSEQ

/SEQUENCE 7 RIGHT 8, LEFT 16, RIGHT 16, LEFT 8

03664	R	200041	R	RANSQ7	LAC	MQSTRT
03665	R	652000	A		LMQ	
03666	R	200040	R		LAC	ACSTRT
03667	R	744000	A		CLL	
03670	R	660510	A		LRSS+10	
03671	R	660620	A		LLSS+20	
03672	R	640520	A		LRS+20	
03673	R	660610	A		LLSS+10	
03674	R	103750	R		JMS	SEQCOM
03675	R	103715	R		JMS	NXTSEQ

/SEQUENCE 8 LEFT 9, RIGHT 18, LEFT 18, RIGHT 9

03676	R	200041	R	RANSQ8	LAC	MQSTRT
03677	R	652000	A		LMQ	
03700	R	200040	R		LAC	ACSTRT
03701	R	744000	A		CLL	
03702	R	660611	A		LLSS+11	
03703	R	640522	A		LRS+22	
03704	R	660622	A		LLSS+22	
03705	R	640511	A		LRS+11	
03706	R	103750	R		JMS	SEQCOM
03707	R	444040	R		ISZ	PASSK
03710	R	741000	A		SKP	
03711	R	603714	R		JMP	.+3
03712	R	100073	R		JMS	MONBRK
03713	R	603535	R		JMP	RANSEQ+2
03714	R	604157	R		JMP	NRMLZE
						.EJECT

/SET AC SIGN INTO NEXT AC

/AND MQ BITS

/

03715	R	603715	R	NXTSEQ	JMP	.	
03716	R	203744	R		LAC	SVSIGN	
03717	R	744010	A		CLL	!RAL	
03720	R	043744	R		DAC	SVSIGN	/TO FILL MQ
03721	R	203745	R		LAC	SVSIGN+1	
03722	R	744020	A		CLL	!RAR	
03723	R	043745	R		DAC	SVSIGN+1	
03724	R	203746	R		LAC	SVMASK	
03725	R	744002	A		STL		
03726	R	740010	A		RAL		
03727	R	043746	R		DAC	SVMASK	
03730	R	500041	R		AND	MQSTRT	/CLR MQ BIT
03731	R	343744	R		TAD	SVSIGN	/MAKE MQ = AC 0
03732	R	040041	R		DAC	MQSTRT	
03733	R	203747	R		LAC	SVMASK+1	
03734	R	744002	A		STL		
03735	R	740020	A		RAR		
03736	R	043747	R		DAC	SVMASK+1	
03737	R	500040	R		AND	ACSTRT	/CLR AC BIT
03740	R	343745	R		TAD	SVSIGN+1	/MAKE ACX = AC 0
03741	R	040040	R		DAC	ACSTRT	
03742	R	440043	R		ISZ	SCSTRT	/INDICATE NEXT SEQUENCE
03743	R	623715	R		JMP*	NXTSEQ	
03744	R	000000	A	SVSIGN	0		
03745	R	000000	A		0		
03746	R	000000	A	SVMASK	0		
03747	R	000000	A		0		
					.EJECT		

/RANDOM DATA SEQUENCED
/COMMON COMPARE AND ERROR TYPE

```

/
SEQCOM  JMP      .
03750 R 603750 R      DAC ACEND
03751 R 040044 R      GLK
03752 R 750010 A      DAC LKEND
03753 R 040046 R      LACS
03754 R 641001 A      DAC SCEND
03755 R 040047 R      LACQ
03756 R 641002 A      DAC MQEND
03757 R 040045 R      SAD MQSTRT      /MQ SAME AS START
03760 R 540041 R      SKP
03761 R 741000 A      JMP      .+3      /ERROR MQ
03762 R 603765 R      LAC SCEND
03763 R 200047 R      SZA
03764 R 740200 A      JMP      .+4      /ERROR SC
03765 R 603771 R      LAC ACSTRT
03766 R 200040 R      SAD ACEND
03767 R 540044 R      SKP
03770 R 741000 A      JMP      .+5      /ERROR AC
03771 R 603776 R      EAE!21000      /GET AC SIGN CLR AC
03772 R 661000 A      GLK
03773 R 750010 A      SAD LKEND      /LINK END = AC SIGN?
03774 R 540046 R      JMP* SEQCOM      /ALL OK = EXIT
03775 R 623750 R      JMS ERROR
03776 R 100704 R      5500
03777 R 005500 A      SCSTRT+700000
04000 R 700043 R      SEQCOM+400000
04001 R 403750 R      LKSTRT+500000
04002 R 500042 R      ACSTRT+600000
04003 R 600040 R      MQSTRT+600000
04004 R 600041 R      0
04005 R 000000 A      JMS      ERROR
04006 R 100704 R      5500
04007 R 005500 A      LKEND+500000
04010 R 500046 R      ACEND+600000
04011 R 600044 R      MQEND+600000
04012 R 600045 R      SCEND+700000
04013 R 700047 R      0
04014 R 000000 A      JMP* SEQCOM      /ERROR EXIT
04015 R 623750 R      .EJECT

```

/RANDOM NUMBER GENERATOR
/18 BIT

04016	R	604016	R	RANGEN	JMP	.
04017	R	204036	R		LAC	RANNO
04020	R	744020	A		CLL	RAR
04021	R	741400	A		SZL	
04022	R	240770	R		XUR	BIT0
04023	R	244037	R		XOR	RANNO+1
04024	R	304037	R		ADD	RANNO+1
04025	R	044036	R		DAC	RANNO
04026	R	444043	R		ISZ	RANCNT
04027	R	604034	R		JMP	+.5
04030	R	364574	R		TAD*	(0)
04031	R	044036	R		DAC	RANNO
04032	R	777000	A		LAW	-1000
04033	R	044043	R		DAC	RANCNT
04034	R	204036	R		LAC	RANNO
04035	R	624016	R		JMP*	RANGEN
04036	R	736425	A	RANNO	736425	
04037	R	335671	A		335671	
04040	R	000000	A	PASSK	0	
04041	R	640500	A	KLRS	LRS	
04042	R	660600	A	KLLSS	LLSS	
04043	R	777000	A	RANCNT	-1000	
				/		
				/		
04044	R	000000	A	SHFBUF	0	
04045	R		A		.BLOCK	112
					.EJECT	

/NORMALIZE TEST
 /DOES NORMS GET AC 0 = 0 TO L
 /

```

04157 R 140041 R NRMLZE DZM MQSTRT
04160 R 140043 R DZM SCSTRT
04161 R 200771 R LAC BIT1
04162 R 040040 R DAC ACSTRT
04163 R 140042 R DZM LKSTRT
04164 R 200042 R LAC LKSTRT /START SCOPE LOOP
04165 R 740020 A RAR
04166 R 650000 A CLQ
04167 R 200040 R LAC ACSTRT
04170 R 660400 A NORMS=44 /SC = 0
04171 R 641002 A LACQ
04172 R 040045 R DAC MQEND /SAVE RESULTS
04173 R 641001 A LACS
04174 R 040047 R DAC SCEND
04175 R 750010 A GLK
04176 R 143315 R DZM SCCOMK
04177 R 040046 R DAC LKEND
04200 R 740200 A SZA /AC SIGN IS 0
04201 R 104525 R JMS NORMSE
04202 R 200042 R LAC LKSTRT
04203 R 440042 R ISZ LKSTRT
04204 R 741200 A SNA
04205 R 604164 R JMP NRMLZE+5

```

/

/DOES NORMS GET AC0=1 TO L

```

04206 R 750001 A NRMLZ1 CLC
04207 R 040041 R DAC MQSTRT
04210 R 140043 R DZM SCSTRT
04211 R 140042 R DZM LKSTRT
04212 R 750001 A CLC
04213 R 040040 R DAC ACSTRT
04214 R 200042 R LAC LKSTRT /START SCOPE LOOP
04215 R 740020 A RAR
04216 R 200040 R LAC ACSTRT
04217 R 650004 A CLQ+4 /SET MQ = 1'S
04220 R 660400 A NORMS=44
04221 R 040044 R DAC ACEND
04222 R 641002 A LACQ
04223 R 040045 R DAC MQEND
04224 R 641001 A LACS
04225 R 040047 R DAC SCEND
04226 R 750010 A GLK
04227 R 143315 R DZM SCCOMK
04230 R 040046 R DAC LKEND
04231 R 741200 A SNA
04232 R 104525 R JMS NORMSE
04233 R 200042 R LAC LKSTRT
04234 R 440042 R ISZ LKSTRT
04235 R 741200 A SNA
04236 R 604214 R JMP NRMLZ1+6

```

PAGE 61

EAEPT1 SRC

EAE-P1

.EJECT

/WILL NORM STOP SHIFT WITH
 /AC 0 AND AC 1 UNEQUAL? 01, 10

```

/
04237 R 140041 R NRMLZ2 DZM MQSTRT
04240 R 140042 R DZM LKSTRT
04241 R 200746 R LAC SEVSEV
04242 R 043315 R DAC SCCOMK
04243 R 200771 R LAC BIT1
04244 R 040040 R DAC ACSTRT
04245 R 201000 R LAC BIT17
04246 R 040043 R DAC SCSTRT
04247 R 200041 R LAC MQSTRT /START SCOPE LOOP
04250 R 652000 A LMQ
04251 R 744000 A CLL
04252 R 200040 R LAC ACSTRT /SET UP COMPLETE
04253 R 640401 A NORM=43 /SC = 1
04254 R 040044 R DAC ACEND
04255 R 641002 A LACQ
04256 R 040045 R DAC MQEND /SAVE RESULTS
04257 R 750010 A GLK
04260 R 040046 R DAC LKEND
04261 R 641001 A LACS
04262 R 040047 R DAC SCEND
04263 R 540746 R SAD SEVSEV /SC = -1?
04264 R 741000 A SKP
04265 R 104503 R JMS NORMER
04266 R 200040 R LAC ACSTRT
04267 R 740001 A CMA
04270 R 040040 R DAC ACSTRT
04271 R 200041 R LAC MQSTRT
04272 R 740001 A CMA
04273 R 040041 R DAC MQSTRT
04274 R 740200 A SZA
04275 R 604247 R JMP NRMLZ2+10
      .EJECT

```

/DOES NORM NOT STOP SHIFT

/ON AC 0 = AC1 00, 11,

/

04276 R 140041 R	NRMLZ3	DZM MQSTRT	
04277 R 140042 R		DZM LKSTRT	
04300 R 200777 R		LAC BIT16	/ NORMALIZE SC = 2
04301 R 040043 R		DAC SCSTRT	
04302 R 200746 R		LAC SEVSEV	
04303 R 043315 R		DAC SCCOMK	
04304 R 200772 R		LAC BIT2	
04305 R 040040 R		DAC ACSTRT	
04306 R 200041 R		LAC MQSTRT	/START SCOPE LOOP
04307 R 652000 A		LMQ	
04310 R 744000 A		CLL	
04311 R 200040 R		LAC ACSTRT	/COMPLETE SET UP
04312 R 660402 A		NORMS-42	/SC = 2
04313 R 040044 R		DAC ACEND	
04314 R 641001 A		LACS	
04315 R 040047 R		DAC SCEND	/SAVE RESULTS
04316 R 750010 A		GLK	
04317 R 040046 R		DAC LKEND	
04320 R 641002 A		LACQ	
04321 R 040045 R		DAC MQEND	
04322 R 741100 A		SPA	
04323 R 740001 A		CMA	/MQ = ALL 0'S OR ALL 1'S
04324 R 740200 A		SZA	
04325 R 604333 R		JMP .+6	/ERROR IN MQ
04326 R 200044 R		LAC ACEND	
04327 R 741100 A		SPA	/AC NEGATIVE?
04330 R 740001 A		CMA	/MAKE POSITIVE
04331 R 540771 R		SAD BIT1	/AC NORMALIZE CORRECT?
04332 R 741000 A		SKP	
04333 R 604337 R		JMP .+4	/AC IN ERROR
04334 R 200047 R		LAC SCEND	
04335 R 540746 R		SAD SEVSEV	/SC = -1?
04336 R 741000 A		SKP	
04337 R 104525 R		JMS NORMSE	
04340 R 200040 R		LAC ACSTRT	
04341 R 740001 A		CMA	
04342 R 040040 R		DAC ACSTRT	
04343 R 200041 R		LAC MQSTRT	
04344 R 740001 A		CMA	
04345 R 040041 R		DAC MQSTRT	
04346 R 740200 A		SZA	
04347 R 604306 R		JMP NRMLZ3+10	
		.EJECT	

/WILL NORMALIZE NORMALIZE A POSITIVE
 /NUMBER WITH A 1 FROM AC BIT 1 TO AC 17 WITH SC=44 AT START
 /AND A NEGATIVE NUMBER WITH A0 IN AC BIT 1
 /TO AC1
 /AC = MQ AT NORMS START.
 /AC & MQ SHOULD EQUAL
 /200000 OR 577777 AT END.

04350	R	140042	R	NRMLZ4	DZM LKSTRT	
04351	R	200764	R		LAC FOUR4	
04352	R	040043	R		DAC SCSTRT	
04353	R	200771	R		LAC BIT1	
04354	R	040040	R		DAC ACSTRT	
04355	R	040041	R		DAC MQSTRT	
04356	R	200765	R	NR4A	LAC THREE4	
04357	R	043315	R		DAC SCCOMK	/TO COMPARE SC
04360	R	200041	R	NR4B	LAC MQSTRT	/SCOPE LOOP START
04361	R	652000	A		LMQ	
04362	R	744000	A		CLL	
04363	R	200040	R		LAC ACSTRT	/SET UP COMPLETE
04364	R	660444	A		NORMS	/SC = 44
04365	R	040044	R		DAC ACEND	
04366	R	641002	A		LACQ	
04367	R	040045	R		DAC MQEND	/SAVE RESULTS
04370	R	750010	A		GLK	
04371	R	040046	R		DAC LKEND	
04372	R	641001	A		LACS	
04373	R	040047	R		DAC SCEND	
04374	R	543315	R		SAD SCCOMK	
04375	R	741000	A		SKP	
04376	R	604412	R		JMP NR4C	/SC, ERROR
04377	R	200044	R		LAC ACEND	
04400	R	741100	A		SPA	
04401	R	740001	A		CMA	
04402	R	540771	R		SAD BIT1	/AC SHOULD BE = 20000.
04403	R	741000	A		SKP	
04404	R	604412	R		JMP NR4C	
04405	R	200045	R		LAC MQEND	
04406	R	741100	A		SPA	
04407	R	740001	A		CMA	
04410	R	540771	R		SAD BIT1	/MQ SHOULD BE = 200000
04411	R	741000	A		SKP	
					.EJECT	

04412	R	104525	R	NR4C	JMS NORMSE	
04413	R	200041	R		LAC MQSTRT	
04414	R	652000	A		LMQ	
04415	R	744000	A		CLL	
04416	R	200040	R		LAC ACSTRT	/SHIFT AC & MQ
04417	R	443315	R		ISZ SCCOMK	/WHEN AC NOT EQUAL MQ
04420	R	660501	A		LRSS+1	
04421	R	040040	R		DAC ACSTRT	/CHANGE SIGNS
04422	R	641002	A		LACQ	
04423	R	040041	R		DAC MQSTRT	/AC AND MQ STILL EQUAL
04424	R	540040	R		SAD ACSTRT	/DO, AGAIN.
04425	R	604360	R		JMP NR4B	/DONE ALL NEGATIVES YET,
04426	R	740100	A		SMA	/YES, DO NEXT TEST.
04427	R	604434	R		JMP NRMLZ5	/2ND SERIES, POSITIVES DONE, DO NEGATIVES.
04430	R	201002	R		LAC NBIT1	/NEGATIVE NUMBERS
04431	R	040040	R		DAC ACSTRT	
04432	R	040041	R		DAC MQSTRT	
04433	R	604356	R		JMP NR4A	
					.EJECT	

/WILL A COMPLEMENT BIT PATTERN NORMALIZE
/MQ = 252525 AND 525252 AC = 0'S OR 1'S

```

/
04434 R 140040 R NRMLZ5 DZM ACSTRT
04435 R 200767 R LAC COMBIT /252525 PATTERN
04436 R 040041 R DAC MQSTRT
04437 R 200764 R LAC FOUR4
04440 R 040043 R DAC SCSTRT
04441 R 140042 R DZM LKSTRT
04442 R 200041 R LAC MQSTRT /SCOPE LOOP START
04443 R 652000 A LMQ
04444 R 744000 A CLL
04445 R 200040 R LAC ACSTRT
04446 R 660444 A NORMS
04447 R 040044 R DAC ACEND
04450 R 641001 A LACS
04451 R 040047 R DAC SCEND
04452 R 750010 A GLK
04453 R 040046 R DAC LKEND
04454 R 641002 A LACQ
04455 R 040045 R DAC MQEND
04456 R 741100 A SPA
04457 R 740001 A CMA
04460 R 740200 A SZA
04461 R 604465 R JMP .+4
04462 R 200044 R LAC ACEND /ACEND SHOULD
04463 R 540041 R SAD MQSTRT /= MQSTRT
04464 R 741000 A SKP
04465 R 604472 R JMP .+5 /AC ERROR
04466 R 200766 R LAC FIVE6
04467 R 043315 R DAC SCCOMK
04470 R 540047 R SAD SCEND /SC INDICATE SHIFT 18
04471 R 741000 A SKP
04472 R 104525 R JMS NORMSE
04473 R 750001 A CLC
04474 R 040040 R DAC ACSTRT
04475 R 200041 R LAC MQSTRT
04476 R 740001 A CMA
04477 R 040041 R DAC MQSTRT
04500 R 741100 A SPA
04501 R 604442 R JMP NRMLZ5+6
04502 R 600113 R JMP TERNIN
.EJECT

```

/NORMALIZE ERROR TYPEOUTS

/

04503	R	604503	R	NORMER	JMP .	
04504	R	100704	R		JMS ERROR	
04505	R	006215	A		6215	
04506	R	700043	R		SCSTRT+700000	
04507	R	404503	R		NORMER+400000	/ERROR ADDRESS
04510	R	500042	R		LKSTRT+500000	
04511	R	600040	R		ACSTRT+600000	
04512	R	600041	R		MQSTRT+600000	
04513	R	000000	A		0	
04514	R	100704	R		JMS ERROR	
04515	R	006215	A		6215	
04516	R	500046	R		LKEND+500000	
04517	R	600044	R		ACEND+600000	
04520	R	600045	R		MQEND+600000	
04521	R	703315	R		SCCOMK+700000	
04522	R	700047	R		SCEND+700000	
04523	R	000000	A		0	
04524	R	624503	R		JMP* NORMER	

/

/NORMALIZE SIGNED ERROR TYPEOUTS

/

04525	R	604525	R	NORMSE	JMP .	
04526	R	100704	R		JMS ERROR	
04527	R	006243	A		6243	
04530	R	700043	R		SCSTRT+700000	
04531	R	404525	R		NORMSE+400000	
04532	R	500042	R		LKSTRT+500000	
04533	R	600040	R		ACSTRT+600000	
04534	R	600041	R		MQSTRT+600000	
04535	R	000000	A		0	
04536	R	100704	R		JMS ERROR	
04537	R	006243	A		6243	
04540	R	500046	R		LKEND+500000	
04541	R	600044	R		ACEND+600000	
04542	R	600045	R		MQEND+600000	
04543	R	703315	R		SCCOMK+700000	
04544	R	700047	R		SCEND+700000	
04545	R	000000	A		0	
04546	R	624525	R		JMP* NORMSE	

/

04547	R		A	AUTIND	.BLOCK 10	
					.EJECT	

004547 R A10=AUTIND
 004550 R A11=AUTIND+1
 004554 R A15=AUTIND+5
 004555 R A16=AUTIND+6
 004556 R A17=AUTIND+7

/
 644000 A ABS=644000
 640700 A ALS=640700
 650000 A CLQ=650000
 640004 A CMQ=640004
 641002 A LACQ=641002
 641001 A LACS=641001
 640600 A LLS=640600
 660600 A LLSS=660600
 652000 A LMQ=652000
 640500 A LRS=640500
 660500 A LRSS=660500
 640444 A NORM=640444
 660444 A NORMS=660444
 640002 A OMQ=640002

/
 000000 R .END UODSW

04564 R 000001 A *L
 04565 R 000010 A *L
 04566 R 077777 A *L
 04567 R 000022 R *L
 04570 R 001753 R *L
 04571 R 004044 R *L
 04572 R 004155 R *L
 04573 R 004043 R *L
 04574 R 000000 A *L

SIZE=04600 NO ERROR LINES

ABS	644000	A	ACCOMK	03313	R	ACEND	00044	R	ACLMQ	00517	R
ACLMQE	00545	R	ACL.1	00564	R	ACONEQ	00620	R	ACORMQ	00466	R
ACOR.1	00514	R	ACSAV	04557	R	ACSTRT	00040	R	ALS	640700	A
ALSERR	01702	R	ALSLNK	01416	R	ALSMQT	01507	R	ALSZER	01342	R
ALS.1	01543	R	ALS01	01363	R	AUTIND	04547	R	A10	004547	R
A11	004550	R	A15	004554	R	A16	004555	R	A17	004556	R
BIT0	00770	R	BIT1	00771	R	BIT12	00773	R	BIT13	00774	R
BIT14	00775	R	BIT15	00776	R	BIT16	00777	R	BIT17	01000	R
BIT2	00772	R	CLQ	650000	A	CMQ	640004	A	COMBIT	00767	R
COMPMQ	00571	R	CRLF	00056	R	C.1	00615	R	EAEABS	00642	R
EAEA.1	00673	R	EAECAC	00165	R	EAECLQ	00177	R	EAEINT	00032	R
EAERMQ	00133	R	EAESER	00050	R	EAESLK	00373	R	EAESL1	00425	R
ERCODE	00022	R	ERROR	00704	R	ERROR1	00707	R	ERROR2	00720	R
ERWC	00021	R	EXIT	00052	R	FIVE6	00766	R	FOUR1	00754	R
FOUR4	00764	R	FOURS	00763	R	HOLD	00116	R	HOLD.1	00117	R
HSALS	01625	R	HSALSE	01645	R	HSALSL	01640	R	KALL7	01023	R
KALS01	01463	R	KLLSS	04042	R	KLLSS1	02440	R	KLRS	04041	R
K18	00750	R	LACQ	641002	A	LACS	641001	A	LKEND	00046	R
LKSTRT	00042	R	LLS	640600	A	LLSACT	02072	R	LLSERR	02417	R
LLSS	660600	A	LLSSEQ	03150	R	LLSSER	02441	R	LLSSEX	02275	R
LLSSL1	02264	R	LLSSL2	02345	R	LLSSX2	02357	R	LLSTS1	01777	R
LLSTS2	02030	R	LLSTS3	02145	R	LLSTS4	02206	R	LLSTS5	02253	R
LLSTS6	02332	R	LMQ	652000	A	LNKALS	01452	R	LRANEX	03403	R
LRANLP	03370	R	LRS	640500	A	LRSEER	03247	R	LRSER1	03270	R
LRSS	660500	A	LRSSEQ	03200	R	LRSTS1	02464	R	LRSTS2	02514	R
LRSTS3	02564	R	LRSTS4	02635	R	LRSTS5	02672	R	LRSTS6	02745	R
LRSTS7	03023	R	LRSTS8	03075	R	LRST5L	02710	R	LRST6L	02765	R
LRST7E	03047	R	LRST7L	03035	R	LRST8E	03123	R	LRST8L	03110	R
MONBRK	00073	R	MQCOMK	03314	R	MQEND	00045	R	MQSAV	04560	R
MQSTRT	00041	R	MQ1TAC	00222	R	NBIT0	01001	R	NBIT1	01002	R
NBIT10	01013	R	NBIT11	01014	R	NBIT12	01015	R	NBIT13	01016	R
NBIT14	01017	R	NBIT15	01020	R	NBIT16	01021	R	NBIT17	01022	R
NBIT2	01003	R	NBIT3	01004	R	NBIT4	01005	R	NBIT5	01006	R
NBIT6	01007	R	NBIT7	01010	R	NBIT8	01011	R	NBIT9	01012	R
NOPAC	00150	R	NOPAC1	00244	R	NOPLK1	00432	R	NOPLNK	00332	R
NOPL.1	00356	R	NOPMQ	00265	R	NOPMQ1	00307	R	NOPSC	01042	R
NOPSC1	01323	R	NORM	640444	A	NORMER	04503	R	NORMS	660444	A
NORMSE	04525	R	NRMLZE	04157	R	NRMLZ1	04206	R	NRMLZ2	04237	R
NRMLZ3	04276	R	NRMLZ4	04350	R	NRMLZ5	04434	R	NR4A	04356	R
NR4B	04360	R	NR4C	04412	R	NSNERR	01612	R	NXTSEQ	03715	R
OMQ	640002	A	ONESEV	00757	R	PASSK	04040	R	PUT	00732	R
PUT.1	04561	R	PUT.2	04562	R	RANCNT	04043	R	RANGEN	04016	R
RANNO	04036	R	RANRIT	03430	R	RANSEQ	03533	R	RANSHF	03316	R
RANSQ0	03556	R	RANSQ1	03570	R	RANSQ2	03602	R	RANSQ3	03614	R
RANSQ4	03626	R	RANSQ5	03640	R	RANSQ6	03652	R	RANSQ7	03664	R
RANSQ8	03676	R	RESPUT	00725	R	RESULT	01754	R	RRANEX	03506	R
RRANLP	03474	R	SALSRP	01753	R	SCCOMK	03315	R	SCEND	00047	R
SCERR	01300	R	SCSTRT	00043	R	SCT000	01153	R	SCT001	01167	R
SCT003	01203	R	SCT007	01217	R	SCT017	01233	R	SCT037	01247	R
SCT040	01137	R	SCT060	01123	R	SCT070	01107	R	SCT074	01073	R
SCT076	01056	R	SCT077	01263	R	SCTST1	01024	R	SEQCOM	03750	R
SETLLS	03344	R	SETLRS	03451	R	SETUP	00127	R	SEVEN	00745	R
SEVFIV	00761	R	SEVNTY	00752	R	SEVN4	00753	R	SEVONE	00760	R

SEVSEV	00746 R	SEVSIX	00747 R	SGNSHF	01562 R	SHFBUF	04044 R
SIMALS	01722 R	SIXONE	00756 R	SIXTY	00751 R	SVCHAR	00737 R
SVMASK	03746 R	SVSIGN	03744 R	SYSERR	00020 R	TEMP	04563 R
TERMIN	00113 R	THREE	00762 R	THREE4	00765 R	THREE7	00755 R
TSTRUN	010000 A	UODSW	00000 R				

UODSW	00000 R	SYSERR	00020 R	ERWC	00021 R	ERCODE	00022 R
EAEINT	00032 R	ACSTRT	00040 R	MQSTRT	00041 R	LKSTRT	00042 R
SCSTRT	00043 R	ACEND	00044 R	MQEND	00045 R	LKEND	00046 R
SCEND	00047 R	EAESER	00050 R	EXIT	00052 R	CRLF	00056 R
MONBRK	00073 R	TERMIN	00113 R	HOLD	00116 R	HOLD.1	00117 R
SETUP	00127 R	EAERMQ	00133 R	NOPAC	00150 R	EAECAC	00165 R
EAECLQ	00177 R	MQ1TAC	00222 R	NOPAC1	00244 R	NOPMQ	00265 R
NOPMQ1	00307 R	NOPLNK	00332 R	NOPL.1	00356 R	EAESLK	00373 R
EAESL1	00425 R	NOPLK1	00432 R	ACORMQ	00466 R	ACOR.1	00514 R
ACLMQ	00517 R	ACLMQE	00545 R	ACL.1	00564 R	COMPMQ	00571 R
C.1	00615 R	ACONEQ	00620 R	EAEABS	00642 R	EAEA.1	00673 R
ERROR	00704 R	ERROR1	00707 R	ERROR2	00720 R	RESPUT	00725 R
PUT	00732 R	SVCHAR	00737 R	SEVEN	00745 R	SEVSEV	00746 R
SEVSIX	00747 R	K18	00750 R	SIXTY	00751 R	SEVNTY	00752 R
SEVN4	00753 R	FOUR1	00754 R	THREE7	00755 R	SIXONE	00756 R
ONESEV	00757 R	SEVONE	00760 R	SEVFIV	00761 R	THREE	00762 R
FOUR5	00763 R	FOUR4	00764 R	THREE4	00765 R	FIVE6	00766 R
COMBIT	00767 R	BIT0	00770 R	BIT1	00771 R	BIT2	00772 R
BIT12	00773 R	BIT13	00774 R	BIT14	00775 R	BIT15	00776 R
BIT16	00777 R	BIT17	01000 R	NBIT0	01001 R	NBIT1	01002 R
NBIT2	01003 R	NBIT3	01004 R	NBIT4	01005 R	NBIT5	01006 R
NBIT6	01007 R	NBIT7	01010 R	NBIT8	01011 R	NBIT9	01012 R
NBIT10	01013 R	NBIT11	01014 R	NBIT12	01015 R	NBIT13	01016 R
NBIT14	01017 R	NBIT15	01020 R	NBIT16	01021 R	NBIT17	01022 R
KALL7	01023 R	SCTST1	01024 R	NOPSC	01042 R	SCT076	01056 R
SCT074	01073 R	SCT070	01107 R	SCT060	01123 R	SCT040	01137 R
SCT000	01153 R	SCT001	01167 R	SCT003	01203 R	SCT007	01217 R
SCT017	01233 R	SCT037	01247 R	SCT077	01263 R	SCERR	01300 R
NOPSC1	01323 R	ALSZER	01342 R	ALS01	01363 R	ALSLNK	01416 R
LNKALS	01452 R	KALS01	01463 R	ALSMQT	01507 R	ALS.1	01543 R
SGNSHF	01562 R	NSNERR	01612 R	HSALS	01625 R	HSALSL	01640 R
HSALSE	01645 R	ALSERR	01702 R	SIMALS	01722 R	SALS SRP	01753 R
RESULT	01754 R	LLSTS1	01777 R	LLSTS2	02030 R	LLSACT	02072 R
LLSTS3	02145 R	LLSTS4	02206 R	LLSTS5	02253 R	LLSSL1	02264 R
LLSSEX	02275 R	LLSTS6	02332 R	LLSSL2	02345 R	LLSSX2	02357 R
LLSERR	02417 R	KLLSS1	02440 R	LLSSER	02441 R	LRSTS1	02464 R
LRSTS2	02514 R	LRSTS3	02564 R	LRSTS4	02635 R	LRSTS5	02672 R
LRST5L	02710 R	LRSTS6	02745 R	LRST6L	02765 R	LRSTS7	03023 R
LRST7L	03035 R	LRST7E	03047 R	LRSTS8	03075 R	LRST8L	03110 R
LRST8E	03123 R	LLSSEQ	03150 R	LRSEEQ	03206 R	LRSEER	03247 R
LRSER1	03270 R	ACCOMK	03313 R	MQCOMK	03314 R	SCCOMK	03315 R
RANSHF	03316 R	SETLLS	03344 R	LRANLP	03370 R	LRANEX	03403 R
RANRIT	03430 R	SETLRS	03451 R	RRANLP	03474 R	RRANEX	03506 R
RANSEQ	03533 R	RANSQ0	03556 R	RANSQ1	03570 R	RANSQ2	03602 R
RANSQ3	03614 R	RANSQ4	03626 R	RANSQ5	03640 R	RANSQ6	03652 R
RANSQ7	03664 R	RANSQ8	03676 R	NXTSEQ	03715 R	SVSIGN	03744 R
SVMASK	03746 R	SEQCOM	03750 R	RANGEN	04016 R	RANNO	04036 R
PASSK	04040 R	KLRS	04041 R	KLLSS	04042 R	RANCNT	04043 R
SHFBUF	04044 R	NRMLZE	04157 R	NRMLZ1	04206 R	NRMLZ2	04237 R
NRMLZ3	04276 R	NRMLZ4	04350 R	NR4A	04356 R	NR4B	04360 R
NR4C	04412 R	NRMLZ5	04434 R	NORMER	04503 R	NORMSE	04525 R
AUTIND	04547 R	A10	004547 R	A11	004550 R	A15	004554 R
A16	004555 R	A17	004556 R	ACSAV	04557 R	MQSAV	04560 R

PUT.1	04561 R	PUT.2	04562 R	TEMP	04563 R	TSTRUN	010000 A
OMQ	640002 A	CMQ	640004 A	NORM	640444 A	LRS	640500 A
LLS	640600 A	ALS	640700 A	LACS	641001 A	LACQ	641002 A
ABS	644000 A	CLQ	650000 A	LMQ	652000 A	NORMS	660444 A
LRSS	660500 A	LLSS	660600 A				