

EAE

RANDOM EXERCISER
MD-11-DZKEC-A

EP-DZKEC-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA

This microfiche card contains a grid of frames. The frames are arranged in approximately 10 rows and 3 columns. Each frame contains a small table or data set, likely representing a random exercise. The text within the frames is too small to read clearly but appears to be organized in a structured format.

MAINDEC-11-DZKEC-A
P11

B01

.REM 2

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZKEC-A
PRODUCT NAME: EAE RANDOM EXERCIZER
DATE: AUGUST 1976
MAINTAINER: DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) BY DIGITAL EQUIPMENT CORPORATION 1970, 1976

MAINDEC-11-DZKEC-A
P11

MAINDEC-11-DZKEC-A
DZKECA.P11

MACY11 27(732) 03-NOV-76 15:35 PAGE -

001

108

109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160

5. OPERATING PROCEDURE
IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

- 1) TYPE CONTROL G (<+G>): THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:
 - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED)
IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.
 - B) IF A CONTROL U (<+U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

5.1 OPERATIONAL SWITCH SETTINGS

5.1.1 AT SA 200 .. ALL SWITCHES DOWN WILL TEST ALL FUNCTIONS OF THE EAE AND PRINT OUT ON ERRORS AND CONTINUE IN TEST. (BELL WILL RING AT COMPLETION OF 1000 PASS)

5.1.2 SWITCH SETTINGS ARE

- SW15 = 1 OR UP ... HALT ON ERROR
- SW14 = 1 OR UP ... SCOPE LOOP
- SW13 = 1 OR UP ... INHIBIT PRINTOUT
- SW12 = 1 OR UP ... INHIBIT TRACE TRAPPING
- SW11 = 1 OR UP ... INHIBIT ITERATION LOOP
- SW10 = 1 OR UP ... BELL ON ERROR
- 0 OR DOWN . BELL ON 1000 PASSES

161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214

5.2. SUBROUTINE ABSTRACTS

5.2.1 BEGIN SA 200

5.2.2 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUB-TEST AS IT IS BEING ENTERED. IF A SCOPE LOOP IS REQUESTED, IT WILL JUMP TO THE START OF THE SUBTEST THAT THE SCOPE LOOP IS REQUESTED FOR. IF SCOPE LOOP IS NOT REQUESTED, THERE WILL BE 100 ITERATIONS ON THAT SUBTEST BEFORE THE NEXT SUBTEST IS ENTERED. SWITCH 11 ON A 1 INHIBITS ITERATION OF SUBTESTS. SUPPORTS THE CONT-G ROUTINE

5.2.3 HLT

IS A ROUTINE THAT PRINTS-OUT AN ERROR MESSAGE. SEE 6.1. SUPPORTS THE CONT-G ROUTINE

5.2.4 TRTRAP

THIS ROUTINE WILL ALLOW THE TRACE BIT TRAP TO BE SET AT THE BEGINNING OF THE PROGRAM. WHEN SET IT CAUSES A TRAP AFTER EACH INSTRUCTION. THE FIRST INSTRUCTION EXECUTED UPON TRAPPING IS AN "RTI" WHICH RETURNS TO THE INTERRUPTED SEQUENCE OF INSTRUCTION. THIS SEQUENCE IS CONTINUED TILL THE END OF THE PROGRAM LOOP IS REACHED.

5.2.5 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS STARTING AT LOCATION 0, DESIGNED TO DETECT, AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS TO THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

THE PRINCIPAL OF THIS ROUTINE IS: THE VECTOR ENTERANCE CONTAINS THE ADDRESS OF THE TRAP HANDLING ROUTINE AND THE NEXT LOCATION CONTAINS THE ADDRESS TRAPPED TO DIVIDED BY 4. A MESSAGE WILL BE TYPED AS FOLLOWS:

TRAP TO LOCATION 4 FROM 3172 BY 177302

4 = ADDRESS TRAPPED TO
3172 = LOCATION THAT CAUSED THE TRAP
177302 = CONTENTS OF THAT LOCATION

115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

5.3 PROGRAM AND OR OPERATOR ACTION

5.3.1 LOADING AND STARTING AT 200 WITH ALL SWITCHES DOWN IS WORSE CASE TESTING. IF AN ERROR IS DETECTED HERE, THERE WILL BE A PRINTOUT. WHEN AN ERROR IS DETECTED AND IT IS NECESSARY TO SCOPE ON IT, PLACE SW15 UP TO HALT ON ERROR, HIT CONTINUE WITH SW14 UP TO LOOP ON ERROR, AND SW13 UP TO DELETE PRINTOUTS.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ERROR: 040000 000000 / 100000

SOFTWARE AC=000000 MQ=100000 SR=22
HARDWARE AC=000000 MQ=100000 SR=122

ERRORS OCCURED 75 TIMES OUT OF 100

THE LINE STARTING WITH ERROR IS THE OPERATION BEING PERFORMED WHEN THE FAILURE WAS DETECTED. THE 'SOFTWARE' LINE SHOWS THE CALCULATED RESULT AND THE 'HARDWARE' LINE IS THE RESULT BY THE HARDWARE EAE. THE LAST LINE REFLECTS THE NUMBER OF ERRORS OUT OF THE NUMBER OF TRIES. THIS LINE WILL ONLY BE TYPED IF THE ITERATIONS ARE ON.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

7.1 STARTING RESTRICTION

NONE

7.2 OPERATIONAL RESTRICTION

NONE

8. MISCELLANEOUS

THIS TEST IS ONLY AN EXERCIZER SO MAINDEC-11-DZKEB-A IS ASSUMED TO HAVE RUN ON THE EAE FIRST.

22067
22068
22069
22070
22071
22072
22073
22074
22075
22076
22077
22078
22079
22080
22081
22082
22083
22084
22085
22086
22087
22088
22089
22090
22091
22092
22093
22094
22095
22096
22097
22098
22099
22100
22101
22102
22103
22104
22105
22106
22107
22108
22109
22110
22111
22112
22113
22114
22115
22116
22117
22118
22119
22120
22121
22122
22123
22124
22125
22126
22127
22128
22129
22130
22131
22132
22133
22134
22135
22136
22137
22138
22139
22140
22141
22142
22143
22144
22145
22146
22147
22148
22149
22150
22151
22152
22153
22154
22155
22156
22157
22158
22159
22160
22161
22162
22163
22164
22165
22166
22167
22168
22169
22170
22171
22172
22173
22174
22175
22176
22177
22178
22179
22180
22181
22182
22183
22184
22185
22186
22187
22188
22189
22190
22191
22192
22193
22194
22195
22196
22197
22198
22199
22200

9.1 EXECUTION TIME

ABOUT 30 SECONDS WITH ALL SWITCHES DOWN

9. PROGRAM DESCRIPTION

THIS PROGRAM IS AN EXERCIZER OF THE MULTIPLY AND DIVIDE INSTRUCTIONS. IT FIRST GENERATES 2 RANDOM NUMBERS AND MULTIPLIES THEM TOGETHER. IT SAVES THIS ANSWER AND DIVIDES IT BY THE MULTIPLIER AND THE MULTIPLICAN SAVING BOTH ANSWERS AS IT GOES ALONG. THE ORIGINAL NUMBERS ARE HARDWARE MULTIPLIED BOTH WAYS AND COMPARED TO THE ORIGINAL ANSWER. THE TWO DIVISIONS ARE PERFORMED AND COMPARED TO THE ORIGINAL SOFTWARE ANSWERS. ONE MORE RANDOM NUMBER IS GENERATED AND THIS NUMBER IS DIVIDED INTO THE ORIGINAL TWO NUMBERS AND THE ANSWER CHECKED AGAINST THE HARDWARE ANSWERS. EACH HARDWARE MULTIPLY AND DIVIDE IS DONE 100 TIMES. IF SW11 IS UP, THEY WILL ONLY BE DONE ONCE. SW14 WILL CAUSE THE TEST THAT YOU ARE IN TO BE LOOPED UPON.

10. LISTING

FOLLOWING

11. FLOW CHART(S)

NA

2

355	000222	177314			LSH:	177314		
356	000224	177316			ASH:	177316		
357	000226	177570			SWR:	177570		
358	000230	177560			TKS:	177560		
359	000232	177562			TKB:	177562		
360	000234	177564			TFS:	177564		
361	000236	177566			TPB:	177566		
362		000060			TKV=	60		
363		000064			TPV=	64		
364		104000			SCOPE=	EMT		
365		104400			HLT=	TRAP		
366		000000			.AC=	%0		
367		000001			.MQ=	%1		
368		001000			=1000			
369	001000	012706	001000		BEGIN:	MOV	#BEGIN,%6	;SET PUSH DOWN LIST POINTER
370	001004	005067	003754			CLR	ERRORS	;ERROR COUNT
371	001010	005067	176762			CLR	177776	;CLEAR PSW, ENABLE INTERRUPT
372	001014	005003				CLR	%3	;ZERO PRINT REGISTER
373	001016	012767	000001	003706		MOV	#1,CNT	
374	001024	005067	003700			CLR	CINT	
375	001030	005703			64\$:	TST	%3	;WAIT FOR OLD PRINTING SEQUENCE
376	001032	001376				BNE	64\$; TO BE COMPLETED
377	001034	012703	004623			MOV	#HEAD,%3	;INITIALIZE PRINT REGISTER
378	001040	052777	000100	177166		BIS	#100,ATPS	
379	001046	005703			65\$:	TST	%3	;WAIT FOR OLD PRINTING SEQUENCE
380	001050	001376				BNE	65\$; TO BE COMPLETED
381	001052	012703	004655			MOV	#MAIN,%3	;INITIALIZE PRINT REGISTER
382	001056	052777	000100	177150		BIS	#100,ATPS	
383	001064	013746	000006		SJSWR:	MOV	@#6,-(SP)	;SAVE VECTORS
384	001070	013746	000004			MOV	@#4,-(SP)	
385	001074	012737	001114	000004		MOV	@64,@#4	;SET UP FOR TIMEOUT
386	001102	022777	177777	177116		CMP	#-1,@SWR	;REFERENCE HARDWARE SWITCH REGISTER
387	001110	001402				BEQ	65\$	
388	001112	000404				BR	66\$	
389	001114	022626			64\$:	CMP	(SP)+,(SP)+	;ADJUST STACK
390	001116	012767	000176	177102	65\$:	MOV	#SWREG,SWR	;POINT TO SOFTWARE SWITCH REG
391	001124	012637	000004		66\$:	MOV	(SP)+,@#4	;RESTORE VECTORS
392	001130	012637	000006			MOV	(SP)+,@#6	
393	001134	022767	000176	177064		CMP	#SWREG,SWR	;IS SWREG USED
394	001142	001002				BNE	67\$	
395	001144	004767	005250			JSR	PC,CNTLU	;ALLOW SWREG TO BE LOADED
396								

```

397
398
399
400
401
402
403
404 001150 012767 123456 003546 67$: MOV #123456,LONUM ;INITALIZE LONUM FOR RANDOM
405 001156 012767 176543 003542 MOV #176543,HINUM ;INITALIZE HINUM FOR RANDOM
406
407 001164 004767 005150 GO: JSR %7,CKSWR
408 001170 032777 010000 177030 BIT #10000,2SWR ;BIT 12=1 INHIBIT TRACE TRAP
409 001176 001004 SNE 1$ ;BRANCH IF NO
410 001200 052767 000020 176570 BIS #20,177776 ;SET PSW T BIT IF YES
411 001206 000402 BR 2$
412 001210 005067 176562 1$: CLR 177776 ;CLEAR PSW, NO TRAP TRACE
413 001214 012704 004734 2$: MOV #AREA,%4 ;SET POINTER TO ANSWER AREA
414 001220 004767 002626 JSR %7,RANDOM ;GET RANDOM NUMBERS
415 001224 016701 003474 MOV LONUM,.MQ ;LOAD MULTIPLIER (SOFTWARE)
416 001230 016702 003472 MOV HINUM,%2 ;LOAD MULTIPLICAN (SOFTWARE)
417 001234 004767 004570 JSR %7,MUL1 ;MULTIPLY
418 001240 010124 MOV .MQ,(4)+ ;SAVE MQ
419 001242 010024 MOV .AC,(4)+ ;SAVE AC
420 001244 016724 004032 MOV SRI,(4)+ ;SAVE SR
421
422 001250 016702 003450 MOV LONUM,%2 ;LOAD DIVISOR
423 001254 004767 003776 JSR %7,DIVI ;DIVIDE BY MULTIPLIER (SOFTWARE)
424 001260 010024 MOV .AC,(4)+ ;SAVE AC
425 001262 010124 MOV .MQ,(4)+ ;SAVE MQ
426 001264 016724 003762 MOV SRI,(4)+ ;SAVE SR
427
428 001270 005767 003430 TST LONUM
429 001274 001411 BEQ OK1
430 001276 005700 TST .AC
431 001300 001402 BEQ 3$
432 001302 004767 002670 JSR %7,BAD
433 001306 020167 003414 3$: CMP .MQ,HINUM
434 001312 001402 BEQ C^:
435 001314 004767 00265^ JSR %7,BAD
436
437 001320 012704 004734 OK1: MOV #AREA,%4 ;RESET POINTER
438 001324 012401 MOV (4)+,.MQ ;RESTORE MQ
439 001326 012400 MOV (4)+.AC ;RESTORE AC
440 001330 062704 000010 ADD #10,%4 ;INCREMENT POINTER
441
442 001334 016702 003366 MOV HINUM,%2 ;LOAD DIVISOR
443 001340 004767 003712 JSR %7,DIVI ;DIVIDE BY MULTIPLICAN (SOFTWARE)
444 001344 010024 MOV .AC,(4)+ ;SAVE AC
445 001346 010124 MOV .MQ,(4)+ ;SAVE MQ
446 001350 016714 003676 MOV SRI,(4) ;SAVE SR
447 001354 005767 003346 TST HINUM
448 001360 001411 BEQ ITER
449 001362 005700 TST .AC
450 001364 001402 BEQ 1$
451 001366 004767 002604 JSR %7,BAD
452 001372 020167 003326 1$: CMP .MQ,LONUM

```

453	001376	001402			BEQ	ITER		
454	001400	004767	002572		JSR	%7,BAD		
455								
456	001404	005067	003320		ITER:	CLR	CINT	;SET UP COUNT
457	001410	104000				SCOPE		
458	001412	012704	004734			MOV	#AREA,%4	;RESTORE POINTER
459	001416	012401				MOV	(4)+,MQ	;RESTORE MQ
460	001420	012400				MOV	(4)+,AC	;RESTORE AC
461	001422	011467	003624			MOV	(4),SRI	;RESTORE SRI
462								
463	001426	016777	003272	176554		MOV	LONUM,AMQ	;LOAD MULTIPLIER (HARDWARE)
464	001434	016777	003266	176550		MOV	HINUM,AMUL	;LOAD MULTIPLICAN AND MULTIPLY
465	001442	020077	176540			CMP	.AC,AC	;CHECK HIGH ORDER PRODUCT
466	001446	001401				BEQ	1\$;SKIP IF OK
467	001450	104400				HLT		
468	001452	020177	176532		1\$:	CMP	.MQ,AMQ	;CHECK LOW ORDER PRODUCT
469	001456	001401				BEQ	2\$;SKIP IF OK
470	001460	104400				HLT		
471	001462	126777	003564	176526	2\$:	CMPB	SRI,ASR	;CHECK STATUS REGISTER
472	001470	001401				BEQ	3\$;SKIP IF GOOD
473	001472	104400				HLT		
474								
475	001474	104000			3\$:	SCOPE		
476	001476	012704	004734			MOV	#AREA,%4	;RESTORE POINTER
477	001502	012401				MOV	(4)+,MQ	;RESTORE MQ
478	001504	012400				MOV	(4)+,AC	;RESTORE AC
479	001506	011467	003540			MOV	(4),SRI	;RESTORE SRI
480								
481	001512	016777	003210	176470		MOV	HINUM,AMQ	;LOAD MULTIPLIER (HARDWARE)
482	001520	016777	003200	176464		MOV	LONUM,AMUL	;LOAD MULTIPLICAN AND MULTIPLY
483	001526	020077	176454			CMP	.AC,AC	;CHECK HIGH ORDER PRODUCT
484	001532	001401				BEQ	4\$;SKIP IF OK
485	001534	104402				HLT+2		
486	001536	020177	176446		4\$:	CMP	.MQ,AMQ	;CHECK LOW ORDER PRODUCT
487	001542	001401				BEQ	5\$;SKIP IF OK
488	001544	104402				HLT+2		
489	001546	126777	003500	176442	5\$:	CMPB	SRI,ASR	;CHECK STATUS REGISTER
490	001554	001401				BEQ	6\$;SKIP IF GOOD
491	001556	104402				HLT+2		
492	001560	104000			6\$:	SCOPE		
493	001562	012704	004734			MOV	#AREA,%4	;RESTORE POINTER
494	001566	012477	176416			MOV	(4)+,AMQ	;RESTORE MQ
495	001572	012477	176410			MOV	(4)+,AC	;RESTORE AC
496	001576	005724				TST	(4)+	;INCREMENT POINTER
497	001600	012400				MOV	(4)+,AC	;RESTORE AC
498	001602	012401				MOV	(4)+,MQ	;RESTORE MQ
499	001604	011467	003442			MOV	(4),SRI	;RESTORE SRI
500								
501	001610	016777	003110	176366		MOV	LONUM,ADIV	;DIVIDE BY MULTIPLIER (HARDWARE)
502	001616	004767	000414			JSR	%7,FLOW	
503	001622	000410				BR	.SRI	
504	001624	020077	176356			CMP	.AC,AC	;CHECK REMAINDER
505	001630	001401				BEQ	7\$;SKIP IF OK
506	001632	104401				HLT+1		
507	001634	020177	176350		7\$:	CMP	.MQ,AMQ	;CHECK QUOTENT
508	001640	001401				BEQ	.SRI	;SKIP IF OK

565	002114	062767	000001	002610		ADD	#1,CNT		;COJNT NUMBER OF PASSES
566	002122	032767	000777	002602		BIT	#777,CNT		;CHECK FOR MULTIPLES OF 1000
567	002130	001411				BEQ	3\$		
568	002132				2\$:				
569	002132	005703			64\$:	TST	%3		;WAIT FOR OLD PRINTING SEQUENCE
570	002134	001376				BNE	64\$		TO BE COMPLETED
571	002136	012703	004704			MOV	#EAEOK,%3		;INITIALIZE PRINT REGISTER
572	002142	052777	000100	176064		BIS	#100,%7PS		
573	002150	000167	177010			JMP	GO		;RERUN
574	002154	004767	004160		3\$:	JSR	%7,CKSWR		
575	002160	032777	002000	176040		BIT	#2000,%SWR		;BIT 10=1 BELL ON ERROR
576									;BIT 10=0 BELL ON PASS COMPLETE
577	002166	001361				BNE	2\$		
578	002170	112767	000007	002564		MOVB	#7,.TYPE		;TYPE A 7
579	002176	005703			65\$:	TST	%3		;WAIT FOR OLD PRINTING SEQUENCE
580	002200	001376				BNE	65\$		TO BE COMPLETED
581	002202	012703	004762			MOV	#.TYPE,%3		;INITIALIZE PRINT REGISTER
582	002206	052777	000100	176020		BIS	#100,%7PS		
583	002214	005703			66\$:	TST	%3		;WAIT FOR OLD PRINTING SEQUENCE
584	002216	001376				BNE	66\$		TO BE COMPLETED
585	002220	012703	004704			MOV	#EAEOK,%3		;INITIALIZE PRINT REGISTER
586	002224	052777	000100	176002		BIS	#100,%7PS		
587	002232	000167	176726			JMP	GO		;RERUN
588									
589									
590	002236	117767	175754	000034	FLOW:	MOVB	%SR,TEMP		
591	002244	142767	000377	000026		BICB	#77,TEMP		
592	002252	122767	000200	000020		CMPB	#200,TEMP		
593	002260	001406				BEQ	EQZ		
594	002262	122767	000100	000010		CMPB	#100,TEMP		
595	002270	001402				BEQ	EQZ		
596	002272	062716	000002			ADD	#2,(6)		
597	002276	000207			EQZ:	RTS	%7		
598									
599	002300	000000			TEMP:	0			
600	002302	005767	002422		.EMT:	TST	CINT		;CHECK FOR FIRST ONE
601	002306	001013				BNE	NXT1		;SKIP IF FIRST
602	002310	005267	002414			INC	CINT		;DECREASE THE COUNTER
603	002314	004767	004020			JSR	%7,CKSWR		;CHECK FOR CONT-G
604	002320	032777	040000	175700		BIT	#40000,%SWR		;BIT 14=1 LOOP ON TEST
605	002326	001142				BNE	OVR		
606	002330	011667	002424			MOV	(6),LAD		;SAVE LOOP ADDRESS
607	002334	000002				RTI			;RETURN
608	002336	004767	003776		NXT1:	JSR	%7,CKSWR		;CHECK FOR CONT-G
609	002342	032777	004000	175656		BIT	#4000,%SWR		;BIT 11=1 INHIBIT ITERATION LOOP
610	002350	001405				BEQ	KIT		
611	002352	005067	002400			CLR	ERCNT		
612	002356	005067	002346		CLCI:	CLR	CINT		
613	002362	000747				BR	.EMT		
614	002364	005267	002340		KIT:	INC	CINT		
615	002370	004767	003744			JSR	%7,CKSWR		;CHECK FOR CONT-G
616	002374	032777	040000	175624		BIT	#40000,%SWR		;BIT 14=1 LOOP ON TEST
617	002402	001114				BNE	OVR		
618	002404	005767	002320			TST	CINT		
619	002410	100404				BMI	TST1		
620	002412	022767	000100	002310		CMP	#100,CINT		;CHECK FOR LAST ONE


```

677 002716 032777 040000 175302 BIT #40000,JSWR ;BIT 14=1 LOOP ON TEST
678 002724 001402 BEQ 15
679 002726 000167 000662 JMP NOHEAD
680 002732 005767 002020 15: TST ERCNT ;CHECK FOR FIRST ERROR
681 002736 001402 BEQ 35 ;NO, GO OVER HEADING PRINTER
682 002740 000167 000650 25: JMP NOHEAD
683 002744 011667 000660 35: MOV (6),HLTAD ;SAVE ADDRESS OF HLT
684 002750 005267 002010 INC ERRORS ;COUNT THE NUMBER OF ERRORS
685 002754 004767 003360 JSR %7,CKSWR ;CHECK FOR CONT-G
686 002760 032777 020000 175240 BIT #20000,JSWR ;BIT 13=1 INHIBIT ERROR TYPEOUTS
687 002766 001364 BNE 25
688 002770 112767 000137 001764 MOVB #'+,..TYPE ;TYPE A '+
689 002776 005703 645: TST %3 ;WAIT FOR OLD PRINTING SEQUENCE
690 003000 001376 BNE 645 ; TO BE COMPLETED
691 003002 012703 004762 MOV #.TYPE,%3 ;INITIALIZE PRINT REGISTER
692 003006 052777 000100 175220 BIS #100,%TIPS
693 003014 016767 001712 002042 MOV CNT,PRINT1 ;TYPE CNT IN OCTAL
694 003022 004767 002062 JSR %7,PRINTS ;AND SUPPRESS LEADING ZERO'S
695 003026 005703 655: TST %3 ;WAIT FOR OLD PRINTING SEQUENCE
696 003030 001376 BNE 655 ; TO BE COMPLETED
697 003032 012703 003634 MOV #ER1,%3 ;INITIALIZE PRINT REGISTER
698 003036 052777 000100 175170 BIS #100,%TIPS
699 003044 011646 MOV (6),-(6) ;PUT ADDRESS OF INSTRUCTION ON STACK
700 003046 162716 000002 SUB #2,(6) ;GET THE INSTRUCTION
701 003052 032776 000001 000000 BIT #1,%2(6) ;TEST FOR MULTIPLY OR DIVIDE
702 003060 001050 BNE DIVP ;GOTO DIVIDE PRINT IF ON A 1
703 003062 032776 000002 000000 BIT #2,%2(6) ;TEST FOR WHICH MULTIPLY
704 003070 001422 BEQ SECO
705 003072 016767 001626 001764 MOV LONUM,PRINT1 ;TYPE LONUM IN OCTAL
706 003100 004767 001774 JSR %7,PRINTR ;TYPE LEADING ZERO'S
707 003104 005703 665: TST %3 ;WAIT FOR OLD PRINTING SEQUENCE
708 003106 001376 BNE 665 ; TO BE COMPLETED
709 003110 012703 003650 MOV #ER2,%3 ;INITIALIZE PRINT REGISTER
710 003114 052777 000100 175112 BIS #100,%TIPS
711 003122 016767 001600 001734 MOV HINUM,PRINT1 ;TYPE HINUM IN OCTAL
712 003130 004767 001744 JSR %7,PRINTR ;TYPE LEADING ZERO'S
713 003134 000506 BR PSOFT
714 003136 SECO:
715 003136 016767 001564 001720 MOV HINUM,PRINT1 ;TYPE HINUM IN OCTAL
716 003144 004767 001730 JSR %7,PRINTR ;TYPE LEADING ZERO'S
717 003150 005703 645: TST %3 ;WAIT FOR OLD PRINTING SEQUENCE
718 003152 001376 BNE 645 ; TO BE COMPLETED
719 003154 012703 003650 MOV #ER2,%3 ;INITIALIZE PRINT REGISTER
720 003160 052777 000100 175046 BIS #100,%TIPS
721 003166 016767 001532 001670 MOV LONUM,PRINT1 ;TYPE LONUM IN OCTAL
722 003174 004767 001700 JSR %7,PRINTR ;TYPE LEADING ZERO'S
723 003200 000464 BR PSOFT
724 003202 012704 004736 DIVP: MOV #AREA+2,%4 ;SET POINTER AT AC
725 003206 011467 001652 MOV (4),PRINT1 ;TYPE (4) IN OCTAL
726 003212 004767 001662 JSR %7,PRINTR ;TYPE LEADING ZERO'S
727 003216 112767 000040 001536 MOVB #40,..TYPE ;TYPE A 40
728 003224 005703 645: TST %3 ;WAIT FOR OLD PRINTING SEQUENCE
729 003226 001376 BNE 645 ; TO BE COMPLETED
730 003230 012703 004762 MOV #.TYPE,%3 ;INITIALIZE PRINT REGISTER
731 003234 052777 000100 174772 BIS #100,%TIPS
732 003242 014467 001616 MOV -(4),PRINT1 ;TYPE -(4) IN OCTAL

```

733	003246	004767	001626		JSR	%7,PRINTR	:TYPE LEADING ZERO'S
734	003250	005703		65\$:	TST	%3	:WAIT FOR OLD PRINTING SEQUENCE
735	003254	001376			BNE	65\$: TO BE COMPLETED
736	003256	012703	003654		MOV	#ER2A,%3	:INITIALIZE PRINT REGISTER
737	003262	052777	000100	174744	BIS	#100,%TPS	
738	003270	032776	000004	000000	BIT	#4,%(6)	
739	003276	001406			SEQ	DIVZ	
740	003300	016767	001434	001556	MOV	AREA+4,PRINT1	:TYPE AREA+4 IN OCTAL
741	003306	004767	001566		JSR	%7,PRINTR	:TYPE LEADING ZERO'S
742	003312	000417			BR	PSOFT	
743	003314	032776	000002	000000	DIVZ:	BIT	#2,%(6)
744	003322	001006			BNE	SECI	:TEST FOR WHICH DIVIDE
745	003324	016767	001374	001532	MOV	LONUM,PRINT1	:TYPE LONUM IN OCTAL
746	003332	004767	001542		JSR	%7,PRINTR	:TYPE LEADING ZERO'S
747	003336	000405			BR	PSOFT	
748	003340				SECI:		
749	003340	016767	001362	001516	MOV	HINUM,PRINT1	:TYPE HINUM IN OCTAL
750	003346	004767	001526		JSR	%7,PRINTR	:TYPE LEADING ZERO'S
751	003352				PSOFT:		
752	003352	005703		64\$:	TST	%3	:WAIT FOR OLD PRINTING SEQUENCE
753	003354	001376			BNE	64\$: TO BE COMPLETED
754	003356	012703	003660		MOV	#ER3,%3	:INITIALIZE PRINT REGISTER
755	003362	052777	000100	174644	BIS	#100,%TPS	
756	003370	010067	001470		MOV	.AC,PRINT1	:TYPE .AC IN OCTAL
757	003374	004767	001500		JSR	%7,PRINTR	:TYPE LEADING ZERO'S
758	003400	005703		65\$:	TST	%3	:WAIT FOR OLD PRINTING SEQUENCE
759	003402	001376			BNE	65\$: TO BE COMPLETED
760	003404	012703	003714		MOV	#ER4,%3	:INITIALIZE PRINT REGISTER
761	003410	052777	000100	174616	BIS	#100,%TPS	
762	003416	010167	001442		MOV	.MQ,PRINT1	:TYPE .MQ IN OCTAL
763	003422	004767	001452		JSR	%7,PRINTR	:TYPE LEADING ZERO'S
764	003426	005703		66\$:	TST	%3	:WAIT FOR OLD PRINTING SEQUENCE
765	003430	001376			BNE	66\$: TO BE COMPLETED
766	003432	012703	003724		MOV	#ER5,%3	:INITIALIZE PRINT REGISTER
767	003436	052777	000100	174570	BIS	#100,%TPS	
768	003444	016767	001602	001412	MOV	SRI,PRINT1	:TYPE SRI IN OCTAL
769	003452	004767	001432		JSR	%7,PRINTS	:AND SUPRESS LEADING ZERO'S
770	003456	005703		67\$:	TST	%3	:WAIT FOR OLD PRINTING SEQUENCE
771	003460	001376			BNE	67\$: TO BE COMPLETED
772	003462	012703	003702		MOV	#ER3B,%3	:INITIALIZE PRINT REGISTER
773	003466	052777	000100	174540	BIS	#100,%TPS	
774	003474	005703		68\$:	TST	%3	:WAIT FOR OLD PRINTING SEQUENCE
775	003476	001376			BNE	68\$: TO BE COMPLETED
776	003500	012703	003672		MOV	#ER3A,%3	:INITIALIZE PRINT REGISTER
777	003504	052777	000100	174522	BIS	#100,%TPS	
778	003512	017767	174470	001344	MOV	JAC,PRINT1	:TYPE JAC IN OCTAL
779	003520	004767	001354		JSR	%7,PRINTR	:TYPE LEADING ZERO'S
780	003524	005703		69\$:	TST	%3	:WAIT FOR OLD PRINTING SEQUENCE
781	003526	001376			BNE	69\$: TO BE COMPLETED
782	003530	012703	003714		MOV	#ER4,%3	:INITIALIZE PRINT REGISTER
783	003534	052777	000100	174472	BIS	#100,%TPS	
784	003542	017767	174442	001314	MOV	JMQ,PRINT1	:TYPE JMQ IN OCTAL
785	003550	004767	001324		JSR	%7,PRINTR	:TYPE LEADING ZERO'S
786	003554	005703		70\$:	TST	%3	:WAIT FOR OLD PRINTING SEQUENCE
787	003556	001376			BNE	70\$: TO BE COMPLETED
788	003560	012703	003724		MOV	#ER5,%3	:INITIALIZE PRINT REGISTER

789	003564	052777	000100	174442	BIS	#100,STPS		
790	003572	117767	174420	000032	MOV B	JSR, SER		
791	003600	016767	000026	001256	MOV	SER, PRINT1		:TYPE SER IN OCTAL
792	003606	004767	001276		JSR	%7, PRINTS		:AND SUPPRESS LEADING ZERO'S
793	003612	005726			TST	(6)+		:RESTORE POINTER
794	003614	021667	000010		NOHEAD: CMP	(6), HLTAD		:CHECK FOR SAME HLT
795	003620	001002			SNE	18		
796	003622	005267	001130		INC	ERCNT		
797	003626	000002			15:	RTI		
798	003630	000000			HLTAD:	0		
799	003632	000000			SER:	0		
800	003634	057537	051105	047522	ER1:	.ASCII	"++ERROR: 8"	
801	003642	035122	020040	023040				
802	003650	054040	023040		ER2:	.ASCII	" X 8"	
803	003654	027440	023040		ER2A:	.ASCII	" / 8"	
804	003660	057537	047523	052106	ER3:	.ASCII	"++SOFTWARE"	
805	003666	040527	042522					
806	003672	020040	020040	041501	ER3A:	.ASCII	" AC=8"	
807	003700	023075						
808	003702	044137	051101	053504	ER3B:	.ASCII	"++HARDWARE8"	
809	003710	051101	023105					
810	003714	020040	020040	050515	ER4:	.ASCII	" MQ=8"	
811	003722	023075						
812	003724	020040	020040	051123	ER5:	.ASCII	" SR=8"	
813	003732	023075						
814	003734	057537	051105	047522	ER6:	.ASCII	"++ERRORS OCCURRED 8"	
815	003742	051522	047440	041503				
816	003750	051125	042522	020104				
817	003756	046						
818	003757	040	044524	042515	ER7:	.ASCII	" TIMES OUT OF 8"	
819	003764	020123	052517	020124				
820	003772	043117	023040					
821	003776	051537	043117	053524	ERNG:	.ASCII	"++SOFTWARE DIVIDE DOESN'T MATCH REAL ANSWER+8"	
822	004004	051101	020105	044504				
823	004012	044526	042504	042040				
824	004020	042517	047123	052047				
825	004026	046440	052101	044103				
826	004034	051040	040505	020114				
827	004042	047101	053523	051105				
828	004050	023137						
829								
830	004052	010046			RANDOM:	.EVEN		
831	004054	010146			MOV	.AC, -(6)		:SAVE R0
832	004056	010246			MOV	.MQ, -(6)		:SAVE R1
833	004060	010446			MOV	%2, -(6)		:SAVE R2
834	004062	016700	000636		MOV	%4, -(6)		:SAVE R4
835	004066	016701	000634		MOV	LONUM, .AC		:SET R0 WITH LOW
836	004072	012704	177771		MOV	HINUM, .MQ		:SET R1 WITH HIGH
837	004076	005002			MOV	8-7, %4		:SET SHIFT COUNT
838	004100	006300			CLR	%2		
839	004102	006101			SHIFT:	ASL	.AC	:SHIFT R0 LEFT AND
840	004104	006102			ROL	.MQ		:ROTATE CARRY INTO R1 AND
841	004106	005204			ROL	%2		:ROTATE CARRY INTO R2
842	004110	001373			INC	%4		:CHECK FOR DONE
843	004112	066702	000606		BNE	SHIFT		:CONTINUE SHIFT LOOP
844	004116	005501			ADD	LONUM, %2		:ADD NUMBER TO MAKE X 129
					ADC	.MQ		:PROPOGATE CARRY

845	004120	066701	000602		AUD	HINUM,.MQ		;ADD NUMBER TO MAKE X 129
846	004124	005502			ADC	%2		;PROPOGATE CARRY
847	004126	062700	001057		ADD	#1057,.AC		;ADD LOW CONSTANT
848	004132	005501			ADC	.MQ		;PROPOGATE CARRY
849	004134	005502			ADC	%2		;PROPOGATE CARRY
850	004136	062701	047401		ADD	#47401,.MQ		;ADD HIGH CONSTANT
851	004142	005502			ADC	%2		;PROPOGATE CARRY
852	004144	062702	000006		ADD	#6,%2		;ADD HIGHEST CONSTART
853	004150	060200			ADD	%2,.AC		;REPRIME RO WITH HIGHEST DIGIT
854	004152	005501			ADC	.MQ		;PROPOGATE CARRY
855	004154	010067	000544		MOV	.AC,LONUM		;SAVE RO
856	004160	010167	000542		MOV	.MQ,HINUM		;SAVE R1
857	004164	012604			MOV	(6)+,%4		;RESTORE R4
858	004166	012602			MOV	(6)+,%2		;RESTORE R2
859	004170	012601			MOV	(6)+,.MQ		;RESTORE R1
860	004172	012600			MOV	(6)+,.AC		;RESTORE RO
861	004174	000207			RTS	%7		;RETURN
862								
863								
864	004176				BAC:			
865	004176	005703			64\$:	TST	%3	;WAIT FOR OLD PRINTING SEQUENCE
866	004200	001376				BNE	64\$; TO BE COMPLETED
867	004202	012703	003776		MOV	#ERNG,%3		;INITIALIZE PRINT REGISTER
868	004206	052777	000100	174020	BIS	#100,%TPS		
869	004214	005703			65\$:	TST	%3	;WAIT FOR OLD PRINTING SEQUENCE
870	004216	001376				BNE	65\$; TO BE COMPLETED
871	004220	012703	004536		MOV	#CON,%3		;INITIALIZE PRINT REGISTER
872	004224	052777	000100	174002	BIS	#100,%TPS		
873	004232	005703			66\$:	TST	%3	;WAIT FOR THE PRINTER TO FINISH
874	004234	001376				BNE	66\$	
875	004236	000000			HALT			
876	004240	004767	002074		JSR	%7,CKSWR		;CHECK FOR CONT-G
877	004244	000207			RTS	%7		
878	004246	013767	177776	000444	.TRAP:	MOV	#177776.TAD	;GET TRAP ADDRESS + 2
879	004254	022706	000400		CMP	#400,%6		;CHECK FOR LIST OVERFLOW
880	004260	100412			BMI	1\$		
881	004262	005703			64\$:	TST	%3	;WAIT FOR OLD PRINTING SEQUENCE
882	004264	001376				BNE	64\$; TO BE COMPLETED
883	004266	012703	004570		MOV	#LSTERR,%3		;INITIALIZE PRINT REGISTER
884	004272	052777	000100	173734	BIS	#100,%TPS		
885	004300	000000			HALT			;HALT HERE MEANS PUSH DOWN LIST OVERFLOW
886	004302	004767	002032		JSR	%7,CKSWR		;CHECK FOR CONT-G
887	004306	000241			1\$:	CLC		
888	004310	006167	000404		ROL	TAD		;ROTATE TWICE LEFT
889	004314	006167	000400		ROL	TAD		
890	004320	011667	000376		MOV	(6),TLOC		;GET RETURN ADDRESS
891	004324	162767	000002	000370	SUB	#2,TLOC		;ADDRESS OF CAUSE
892	004332	005703			65\$:	TST	%3	;WAIT FOR OLD PRINTING SEQUENCE
893	004334	001376				BNE	65\$; TO BE COMPLETED
894	004336	012703	004476		MOV	#TMES1,%3		;INITIALIZE PRINT REGISTER
895	004342	052777	000100	173664	BIS	#100,%TPS		
896	004350	016767	000344	000506	MOV	TAD,PRINT1		;TYPE TAD IN OCTAL
897	004356	004767	000526		JSR	%7,PRINTS		;AND SUPPRESS LEADING ZERO'S
898	004362	005703			66\$:	TST	%3	;WAIT FOR OLD PRINTING SEQUENCE
899	004364	001376				BNE	66\$; TO BE COMPLETED
900	004366	012703	004522		MOV	#TMES2,%3		;INITIALIZE PRINT REGISTER

901	004372	052777	000100	173634	BIS	#100, @TPS	
902	004400	016767	000316	000456	MOV	TLOC, PRINT1	;TYPE TLOC IN OCTAL
903	004406	004767	000466		JSR	%7, PRINTR	;TYPE LEADING ZERO'S
904	004412	005703			TST	%3	;WAIT FOR OLD PRINTING SEQUENCE
905	004414	001376			BNE	67\$	TO BE COMPLETED
906	004416	012703	004531		MOV	#TMES3, %3	;INITIALIZE PRINT REGISTER
907	004422	052777	000100	173634	BIS	#100, @TPS	
908	004430	017767	000266	000426	MOV	@TLOC, PRINT1	;TYPE @TLOC IN OCTAL
909	004436	004767	000436		JSR	%7, PRINTR	;TYPE LEADING ZERO'S
910	004442	005703			TST	%3	;WAIT FOR OLD PRINTING SEQUENCE
911	004444	001376			BNE	68\$	TO BE COMPLETED
912	004446	012703	004536		MOV	#CON, %3	;INITIALIZE PRINT REGISTER
913	004452	052777	000100	173554	BIS	#100, @TPS	
914	004460	005703			TST	%3	;WAIT FOR THE PRINTER TO FINISH
915	004462	001376			BNE	69\$	
916	004464	000000			HALT		;HALT
917	004466	004767	001646		JSR	%7, CKSWR	;CHECK FOR CONT-G
918	004472	000002			RTI		;RETURN
919							
920	004474	000002			.TRT:	RTI	
921	004476	057537	051124	050101	TMES1:	.ASCII	"←TRAP TO LOCATION 8"
922	004504	052040	020117	047514			
923	004512	040503	044524	047117			
924	004520	021040					
925	004522	043570	047522	020115	TMES2:	.ASCII	" FROM 8"
926	004530	046					
927	004531	040	054502	023040	TMES3:	.ASCII	" BY 8"
928	004536	044137	052111	041440	CON:	.ASCII	"←HIT CONTINUE TO PROCEED←8"
929	004544	047117	044524	052516			
930	004552	020105	047524	050040			
931	004560	047522	042503	042504			
932	004566	023137					
933	004570	050137	051525	020110	LSTERR:	.ASCII	"←PUSH DOWN LIST OVERFLOW ←8"
934	004576	047504	047127	046040			
935	004604	051511	020124	053117			
936	004612	051105	046106	053517			
937	004620	057440	046				
938	004623	137	045440	030505	HEAD:	.ASCII	"← KE11 RANDOM EXERCISER ←8"
939	004630	020061	040522	042116			
940	004636	046517	042440	042530			
941	004644	041522	051511	051105			
942	004652	057440	046				
943	004655	137	046440	044501	MAIN:	.ASCII	"← MAINDEC-11-DZKEC-A ←8"
944	004662	042116	041505	030455			
945	004670	026461	055104	042513			
946	004676	026503	020101	023137			
947	004704	020137	040505	020105	EAEOK:	.ASCII	"← EAE OK ←8"
948	004712	045517	057440	046			


```

993          ;*****
994          ;          OCTAL DUMP OF A WORD
995          ;*****
996
997
998 005064 000000 PRINT1: 0
999 005066 000000 000000 000C00 PRINT2: .WORD 0,0,0,0
1000 005074 000000
1001 005076 000 000 PRINT3: .BYTE 0,0
1002
1003 005100 112767 000001 177770 PRINTR: MOVB #1,PRINT3 ;SET ZERO FILL SWITCH
1004 005106 000402 BR PRINTCM
1005 005110 005067 177762 PRINTS: CLR PRINT3 ;SUPRESS LEADING ZERO'S
1006 005114 112767 177772 177755 PRINTCM: MOVB #-6,PRINT3+1 ;SET COUNT
1007 005122 010546 MOV %5,-(6) ;SAVE R5
1008 005124 012705 005066 MOV #PRINT2,%5 ;SET POINTER TO FIRST ASCII CHAR.
1009 005130 105015 CLRB (5) ;CLEAR FIRST BYTE
1010 005132 000407 BR PRINTF ;ROTATE FIRST BIT
1011 005134 105015 PRINTL: CLRB (5) ;CLEAR BYTE OF CHARACTER
1012 005136 006167 177722 ROL PRINT1 ;ROTATE BIT INTO C
1013 005142 106115 ROLB (5) ;PACK IT
1014 005144 006167 177714 ROL PRINT1 ;ROTATE BIT INTO C
1015 005150 106115 ROLB (5) ;PACK IT
1016 005152 006167 177706 PRINTF: ROL PRINT1 ;ROTATE BIT INTO C
1017 005156 106115 ROLB (5) ;PACK IT
1018 005160 105715 TSTB (5)
1019 005162 001402 BEQ 1$
1020 005164 105267 177706 INCB PRINT3
1021 005170 105767 177702 1$: TSTB PRINT3 ;CHECK FILL SWITCH
1022 005174 001402 BEQ 2$
1023 005176 152725 000060 BISB #'0,(5)+ ;MAKE INTO ASCII CHAR
1024 005202 105267 177671 2$: INCB PRINT3+1
1025 005206 001352 BNE PRINTL ;REPEAT
1026 005210 022705 005066 CMP #PRINT2,%5
1027 005214 001002 BNE 3$
1028 005216 112725 000060 MOVB #'0,(5)+
1029 005222 105015 3$: CLRB (5)
1030 005224 005703 64$: TST %3 ;WAIT FOR OLD PRINTING SEQUENCE
1031 005226 001376 BNE 64$ ; TO BE COMPLETED
1032 005230 012703 005066 MOV #PRINT2,%3 ;INITIALIZE PRINT REGISTER
1033 005234 052777 000100 172772 BIS #100,ATPS
1034 005242 012605 MOV (6)+,%5 ;RESTORE R5
1035 005244 000207 RTS %7

```

```

1036                                     ;:*****
1037                                     ;:
1038                                     ;:   DIVIDE SUBROUTINE
1039                                     ;:   RO,RI CONTAIN AC,MQ RESPECTIVELY
1040                                     ;:*****
1041
1042
1043 005246 000000      Z:      .WORD  0
1044 005250 000000      COUNT: .WORD  0
1045 005252 000000      SRI:    .WORD  0
1046 005254 000000      SINE:   .WORD  0
1047
1048
1049 005256 012767 177760 177764 DIVI:  MOV    #-16.,COUNT  ;SET UP STEP COUNTER
1050 005264 005067 177762          CLR    SRI
1051 005270 005067 177760          CLR    SINE
1052 005274 005700          TST    .AC
1053 005276 100002          BPL    STEP1
1054 005300 105167 177750          COMB   SINE
1055 005304 005067 177736      STEP1:  CLR    Z          ;INITIALIZE Z INDICATORS
1056 005310 000241          CLC
1057 005312 005700          TST    .AC          ;CHECK REMAINDER SIGN
1058 005314 100002          BPL    1$
1059 005316 105167 177725          COMB   Z+1
1060 005322 005702          1$:    TST    %2
1061 005324 100002          BPL    2$          ;CHECK DIVISOR SIGN
1062 005326 105167 177715          COMB   Z+1
1063 005332 105767 177711          2$:    TSTB   Z+1          ;-1 IF DIFFERENT, 0 IF SAME
1064 005336 100403          BMI    3$
1065 005340 005267 177702          INC    Z          ;SET Z=1 IF SAME
1066 005344 000261          SEC          ;SET C TO A 1
1067 005346 006101          3$:    ROL    .MQ          ;ROTATE AC AND MQ ONCE LEFT STEP2
1068 005350 006100          ROL    .AC
1069 005352 105767 177670          TSTB   Z          ;STEP3
1070 005356 001406          BEQ    4$
1071 005360 160200          SUB    %2,.AC          ;SUBTRACT IF Z = 1
1072 005362 106167 177667          ROL    SINE+1
1073 005366 105167 177663          CO.#B  SINE+1
1074 005372 000403          BR     5$
1075 005374 060200          4$:    ADD    %2,.AC          ;ADD IF Z = 0
1076 005376 106167 177653          ROLB   SINE+1
1077 005402 022767 177760 177640 5$:    CMP    #-16.,COUNT  ;CHECK FOR FIRST TIME THROUGH STEP4
1078 005410 001014          BNE    STEP6
1079 005412 005700          STEP6: TST    .AC
1080 005414 001412          BEQ    STEP8
1081 005416 106067 177633          RORB   SINE+1
1082 005422 103404          BCS    1$          ;IF C AND DIVIDEND ARE
1083 005424 105767 177624          TSTB   SINE
1084 005430 100431          BMI    OFLW          ;
1085 005432 000403          BR     STEP8          ;   DIFFERENT, AN OVERFLOW
1086 005434 105767 177614          1$:    TSTB   SINE          ;   HAS OCCURRED
1087 005440 100025          BPL    OFLW
1088

```

1089	005442	005267	177602	STEP8:	INC	COUNT	
1090	005446	001402			BEQ	1\$	
1091	005450	000167	177630		JMP	STEP1	
1092							
1093	005454	005700		1\$:	TST	.AC	;BEGINNING OF CORPECTION ROUTINE
1094	005456	001031			BNE	STEP9	;IF THE REMAINDER IS ZERO,
1095	005460	000261		STEP10:	SEC		;SHIFT THE QUOTIENT ONCE
1096	005462	006101			ROL	.MQ	;FILLING WITH "1"
1097	005464	116705	177564	STP9:	MOVB	SINE,%5	
1098	005470	005702			TST	%2	
1099	005472	100001			BPL	1\$	
1100	005474	005105			COM	%5	
1101	005476	005705		1\$:	TST	%5	
1102	005500	100403			BMI	2\$	
1103	005502	005701			TST	.MQ	
1104	005504	100403			BMI	OFLW	
1105	005506	000474			BR	DONE	
1106	005510	005701		2\$:	TST	.MQ	
1107	005512	003472		3\$:	BLE	DONE	
1108	005514	105767	177534	OFLW:	TSTB	SINE	
1109	005520	100004			BPL	1\$	
1110	005522	052767	000100 177522		BIS	#100,SR1	
1111	005530	000470			BR	DONE!	
1112	005532	052767	000200 177512	1\$:	BIS	#200,SR1	
1113	005540	000464			BR	DONE!	
1114							
1115							
1116	005542	010267	177502	STEP9:	MOV	%2,COUNT	
1117	005546	100402			BMI	1\$	
1118	005550	005467	177474		NEG	COUNT	
1119	005554	020067	177470	1\$:	CMP	.AC,COUNT	;IF THE DIVISOR WAS (+) NEGATE IT
1120	005560	001416			BEQ	STEP11	;IF THE SAME AS REMAINDER, GO
1121							;TO STEP 11
1122	005562	105067	177461		CLRB	Z+1	
1123	005566	005700			TST	.AC	;COMPARE THE SIGNS OF THE
1124	005570	100002			BPL	2\$;REMAINDER AND THE
1125	005572	105167	177451		COMB	Z+1	;DIVIDEND. IF THEY
1126	005576	105767	177452	2\$:	TSTB	SINE	;ARE THE SAME, GO TO
1127	005602	100002			BPL	3\$;STEP 10
1128	005604	105167	177437		COMB	Z+1	
1129	005610	105767	177433	3\$:	TSTB	Z+1	
1130	005614	001721			BEQ	STEP10	

1131	005616	005067	177424	STEP11:	CLR	Z	
1132	005622	005700			TST	.AC	
1133	005624	100002			BPL	1\$;SET Z=1 IF REMAINDER AND
1134	005626	105167	177415		COMB	Z+1	; THE DIVISOR HAVE THE
1135	005632	005702		1\$:	TST	%2	; SAME SIGN
1136	005634	100002			BPL	2\$	
1137	005636	105167	177405		COMB	Z+1	
1138	005642	105767	177401	2\$:	TSTB	Z+1	
1139	005646	001003			BNE	3\$	
1140	005650	005267	177372		INC	Z	
1141	005654	005201			INC	.MQ	;ADD Z TO QUOTIENT
1142	005656	000241		3\$:	CLC		
1143	005660	006101			ROL	.MQ	;SHIFT QUOTIENT LEFT ONCE
1144							; FILLING WITH "0"
1145	005662	105767	177360		TSTB	Z	
1146	005666	001402			BEQ	4\$	
1147	005670	160200			SUB	%2, .AC	;SUBTRACT DIVISOR FROM REMAINDER
1148	005672	000674			BR	STP9	
1149	005674	060200		4\$:	ADD	%2, .AC	;ADD DIVISOR TO REMAINDER
1150	005676	000672			BR	STP9	
1151							
1152	005700	005701		DONE:	TST	.MQ	
1153	005702	100003			BPL	DONE1	
1154	005704	052767	000300 177340		BIS	#300, SR1	
1155	005712	005700		DONE1:	TST	.AC	;TEST FOR AC=0
1156	005714	001010			BNE	XX	
1157	005716	005701			TST	.MQ	;TEST FOR MQ=0
1158	005720	001003			BNE	1\$	
1159	005722	052767	000004 177322		BIS	#4, SR1	;SET BIT 2 IF AC=MQ=0
1160	005730	052767	000020 177314	1\$:	BIS	#20, SR1	;SET BIT 4 IF AC=0
1161	005736	005701		XX:	TST	.MQ	;TEST FOR MQ=0
1162	005740	001003			BNE	1\$	
1163	005742	052767	000010 177302		BIS	#10, SR1	;SET BIT 3 IF MQ=0
1164	005750	005701		1\$:	TST	.MQ	;GET SIGN OF MQ
1165	005752	100007			BPL	ORV	
1166	005754	022700	177777		CMP	#-1, .AC	;TEST FOR SINGLE PRECISION
1167	005760	001011			BNE	NXT	
1168	005762	052767	000042 177262		BIS	#42, SR1	;SET BITS 1 AND 5 IF SINGLE
1169	005770	000405			BR	NXT	; PRECISION NEGATIVE NUMBER
1170	005772	005700		ORV:	TST	.AC	
1171	005774	001003			BNE	NXT	
1172	005776	052767	000002 177246		BIS	#2, SR1	;SET BIT 1 IF SINGLE PRECISION
1173							; POSITIVE NUMBER
1174	006004	022700	177777	NXT:	CMP	#-1, .AC	
1175	006010	001003			BNE	1\$	
1176	006012	052767	000040 177232		BIS	#40, SR1	
1177	006020	000207		1\$:	RTS	%7	

1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224

006022 000000
006024 000000
006026 000000
006030 005067 17777C
006034 005067 177212
006040 005067 177762
006044 005000
006046 005701
006050 100002
006052 005167 177750
006056 005702
006060 100002
006062 005167 177740
006066 012767 177760 177726
006074 022701 100000
006100 091011
006102 005702
006104 100023
006106 010200
006110 005400
006112 000241
006114 005001
006116 006000
006120 006001
006122 000451
006124 022702 100000
006130 001011
006132 005701
006134 100007
006136 010100
006140 005400
006142 000241
006144 005001
006146 006000
006150 006001
006152 000435
006154 105067 177645
006160 132767 000001 177636

MULTIPLY SUBROUTINE
R0,R1 CONTAINS AC,MQ RESPECTIVELY

CONT: .WORD 0
ONE: .WORD 0
CHEK: .WORD 0
MUL1: CLR ONE ;CLEAR Z STEP1
CLR SR1
CLR CHEK
CLR .AC ;CLEAR AC
TST .MQ
BPL 1\$
COM CHEK
1\$: TST %2
BPL 2\$
COM CHEK
2\$: MOV #-16,CONT ;INITIALIZE COUNT
CMP #100000,.MQ
BNE CNXT
TST %2
BPL STEP2
MOV %2,.AC
NEG .AC
CLC
CLR .MQ
ROR .AC
ROR .MQ
BR DONE2
CMP #100000,%2
BNE STEP2
TST .MQ
BPL STEP2
MOV .MQ,.AC
NEG .AC
CLC
CLR .MQ
ROR .AC
ROR .MQ
BR DONE2
STEP2: CLRB ONE+1
BITS #1,ONE

1225	006166	001402			BEQ	1S	
1226	006170	105167	177631		COMB	ONE+1	;COMP ONE+1 IF Z=1
1227	006174	032701	000001	1S:	BIT	#1,.MQ	
1228	006200	001402			BEQ	2S	
1229	006202	105167	177617		COMB	ONE+1	;COMP ONE+1 IF R1 IS ODD
1230	006206	105767	177613	2S:	TSTB	ONE+1	;IF THE LOW ORDER BIT IS THE SAME
1231	006212	001406			BEQ	STEP4	;AS Z, GO TO STEP4
1232	006214	032701	000001	STEP3:	BIT	#1,.MQ	
1233	006220	001002			BNE	1S	
1234	006222	060200			ADD	%2.AC	;ADD MULTIPLICAND IF MQ IS EVEN
1235	006224	000401			BR	STEP4	
1236	006226	160200		1S:	SUB	%2.AC	;SUBTRACT MULTIPLICAND IF MQ IS ODD
1237	006230	006200		STEP4:	ASR	.AC	
1238	006232	006001			ROR	.MQ	
1239	006234	106167	177564		ROLB	ONE	
1240							
1241	006240	005267	177556	STEP5:	INC	CONT	
1242	006244	001343			BNE	STEP2	;GO TO STEP2 IF NOT 16TH PASS
1243	006246	005700		DONE2:	TST	.AC	
1244	006250	001416			BEQ	D02	
1245	006252	005767	177550		TST	CHEK	
1246	006256	100406			BMI	D0N1	
1247	005260	005700			TST	.AC	
1248	006262	100011			BPL	D02	
1249	005264	005401			NEG	.MQ	
1250	006266	005500			ADC	.AC	
1251	006270	005400			NEG	.AC	
1252	006272	000405			BR	D02	
1253	006274	005700		D0N1:	TST	.AC	
1254	006276	100403			BMI	D02	
1255	006300	005401			NEG	.MQ	
1256	006302	005500			ADC	.AC	
1257	006304	005400			NEG	.AC	
1258	006306	005700		D02:	TST	.AC	
1259	006310	100200			BPL	DONE1	
1260	006312	052767	000300 176732		BIS	#300,SR1	
1261	006320	000167	177366		JMP	DONE1	
1262							
1263							

CHECK SWITCH REGISTER ROUTINE. CHECKS FOR 1G TO ALLOW CHANGING
OF LOC.175.

1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319

006324 000000
006326 000000
006330 000000
006332 000000
006334 000000
006336 000000

006340 022767 000176 171660
006346 001157
006350 105777 171654
006354 100154
006356 017767 171650 177746
006364 042767 177600 177740
006372 022767 000007 177732
006400 001142
006402 005703
006404 001376
006406 012703 006762
006412 052777 000100 171614
006420
006422 005703
006424 001376
006426 012703 006766
006430 052777 000100 171576
006436 017767 171564 176420
006444 004767 176430
006450 005703
006452 001376
006454 012703 006775
006460 052777 000100 171546
006466 005337 006324
006472 005067 177626
006476 012767 000007 177622
006504 004767 000200
006510 042767 177600 177614
006516 122767 000025 177606
006524 001001
006526 000734
006530 122767 000015 177574
006536 001017
006540 012767 000200 177562
006546 005703
006550 001376
006552 012703 007011
006556 052777 000100 171450
006564 022767 000007 177534
006572 001041
006574 000444
006576 122767 000060 177526

JEMPST: .WORD 0
CONTT: .WORD 00
RDSW: .WORD 00
TIB: .WORD 00
TOB: .WORD 00
TOG: .WORD 0

SWR: CMP #SWREG, SWR
BNE OUT
TSTB #TKS
BPL OUT
MOV #TKB, TIB
BIC #177600, TIB
CMP #7, TIB
BNE OUT
64S: TST %3
BNE 64S
MOV #SCNTG, %3
BIS #100, #TPS

CNTLU: 64S: TST %3
BNE 64S
MOV #SMSWR, %3
BIS #100, #TPS
MOV #SWR, PRINT1
JSR %7, PRINTR
65S: TST %3
BNE 65S
MOV #SMNEW, %3
BIS #100, #TPS
CLR #TEMPST
\$READ: CLR TEMPST
MOV #7, CONTT
1S: JSR %7, TTIN
BIC #177600, TIB
CMPB #7, TIB
BNE 2S
3S: BR CNTLU
2S: CMPB #15, TIB
BNE 4S
MOV #200, RDSW
64S: TST %3
BNE 64S
MOV #SCRLF, %3
BIS #100, #TPS
CMP #7, CONTT
PNE 7S
8S: BR OUT
4S: CMPB #60, TIB

: SOFTWARE SWITCH REGISTER PRESENT
: NO, GET OUT
: YES, WAIT FOR
: READY, GET CHARACTER
: AND STRIP OFF
: THE GARBAGE
: IS IT A <1G>

: WAIT FOR OLD PRINTING SEQUENCE
: TO BE COMPLETED
: INITIALIZE PRINT REGISTER

: WAIT FOR OLD PRINTING SEQUENCE
: TO BE COMPLETED
: INITIALIZE PRINT REGISTER

: TYPE #SWR IN OCTAL
: TYPE LEADING ZERO'S
: WAIT FOR OLD PRINTING SEQUENCE
: TO BE COMPLETED
: INITIALIZE PRINT REGISTER

: GO READ A CHARACTER
: STRIP OFF GARBAGE
: IS IT A 1U?
: BRANCH IF NOT
: START OVER
: IS IT A <CR>?
: BRANCH IF NOT

: WAIT FOR OLD PRINTING SEQUENCE
: TO BE COMPLETED
: INITIALIZE PRINT REGISTER

: WAS %3 FIRST CHARACTER
: CHANGE SWR IF NOT FIRST ONE
: GET OUT

```

1320 006604 003004          BGT      55
1321 006606 122767 000067 177516  CMPB    #67,TIB
1322 006614 002010          BGE      65
1323          55:
1324 006616 005703          655:   TST      '3'
1325 006620 001376          BNE      655
1326 006622 012703 007005  MOV     #SQUEST,%3
1327 006626 052777 000100 171400  BIS     #100,@TPS
1328 006634 000734          BR       35
1329 006636 006367 177462          65:   ASL     TEMPST
1330 006642 006367 177456          ASL     TEMPST
1331 006646 006367 177452          ASL     TEMPST
1332 006652 142767 000060 177452  SICB   #60,TIB
1333 006660 156767 177446 177436  BISB   TIB,TEMPST
1334 006666 005367 177434          DEC     CONT
1335 006672 001751          BEQ     55
1336 006674 000703          BR      15
1337 006676 016777 177422 171322  75:   MOV     TEMPST,@SWR
1338 006704 000733          BR      85
1339 006706 000207          OUT:   RTS     PC
1340
1341          ::*****
1342          :      TTY READ SUBROUTINE*****
1343          :
1344          ::*****
1345
1346
1347
1348 006710 005077 171314          TTIN:  CLR     @TKS
1349 006714 005077 171312          CLR     @TKB
1350 006720 005067 177406          CLR     TIB
1351 006724 005277 171300          INC     @TKS
1352 006730 105777 171274          TTIN1: TSTB   @TKS
1353 006734 100375          BPL     TTIN1
1354 006736 017767 171270 177366  MOV     @TKB,TIB
1355 006744 105777 171264          TTIN2: TSTB   @TPS
1356 006750 100375          BPL     TTIN2
1357 006752 116777 177354 171256  MOVB   TIB,@TPB
1358 006760 000207          RTS     %7
1359
1360 006762 057137 023107          $CNTG: .ASCII "'-1G&"
1361 006766 051537 051127 020075  $MSWR: .ASCII "'-SWR= &"
1362 006774 046
1363 006775 040 047040 053505  $MNEW: .ASCII "' NEW= &"
1364 007002 020075 046
1365 007005 040 020077 046  $QUEST: .ASCII "' ? &"
1366 007011 137 023040  $CRLF: .ASCII "'- &"
1367
1368
1369 007014 000000          OFL:   0
1370 000001          .END
;FIRST CHAR FLAG

```


MAINDEC-11-DZKEC-A MACY11 27(732) 03-NOV-76 15:35 PAGE 34
DZKECA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

.TRT	004474	333	920*										
.TYPE	004762	578*	581	645*	648	669*	672	688*	691	727*	730	967*	

.SRDDE	10
.SRDOC	10
.SREAO	10
.SR2AZ	10
.SSAVE	10
.SSB2D	10
.SSB2O	10
.SSCOP	10
.SSIZE	10
.SSUPR	10
.STRAP	10
.STYPB	10
.STYPD	10
.STYPE	10
.STYPO	10
.S40CA	10
.117C	10

ADC	844	846	848	849	851	854	1250	1256							
ADD	440	518	565	596	843	845	847	850	852	853	1075	1149	1234		
ASL	838	1329	1330	1331											
ASR	1237														
BCC	1082														
BEQ	387	429	431	434	448	450	453	466	469	472	484	497	490	505	508
	511	527	530	533	555	558	561	567	593	595	610	623	626	668	678
	681	704	739	971	978	980	1019	1022	1070	1080	1090	1120	1130	1146	1225
	1228	1231	1244	1335											
BGE	1322														
BGT	1320														
BIC	974	1283	1305												
BICB	591	1332													
BIS	378	382	410	572	582	586	633	639	649	656	673	692	698	710	720
	731	737	755	761	767	773	777	783	789	868	872	884	895	901	907
	913	1033	1110	1112	1154	1159	1160	1163	1168	1172	1176	1260	1289	1294	1300
	1315	1327													
BISB	1023	1333													
BIT	408	566	575	604	609	616	625	666	677	686	701	703	738	743	1227
	1232														
BITB	1224														
BLE	1107														
BMI	619	880	1064	1084	1102	1104	1117	1246	1254						
BNE	376	380	394	409	570	577	580	584	601	605	617	631	637	647	654
	658	671	675	687	690	696	702	708	718	729	735	744	753	759	765
	771	775	781	787	795	842	866	870	874	882	893	899	905	911	915
	973	1025	1027	1031	1078	1094	1139	1156	1158	1162	1167	1171	1175	1202	1213
	1233	1242	1279	1285	1287	1292	1298	1307	1310	1313	1317	1325			
BPL	621	652	1053	1058	1061	1087	1099	1109	1124	1127	1133	1136	1153	1165	1195
	1198	1204	1215	1248	1259	1281	1353	1356							
BR	388	411	503	525	553	613	628	713	723	742	747	986	988	1004	1010
	1074	1085	1105	1111	1113	1148	1150	1169	1211	1222	1235	1252	1308	1318	1328
	1336	1338													
CLC	887	1056	1142	1207	1218										
CLR	370	371	372	374	412	456	546	611	612	627	640	644	837	975	1005
	1050	1051	1055	1131	1190	1191	1192	1193	1208	1219	1301	1302	1348	1349	1350
CLRB	1009	1011	1029	1122	1223										
CMP	386	389	393	433	452	465	468	483	486	504	507	526	529	554	557
	620	794	879	1026	1077	1119	1166	1174	1201	1212	1278	1284	1316		
CMPB	471	489	510	532	560	592	594	972	977	979	1306	1309	1319	1321	
COM	1100	1196	1199												
COMB	1054	1059	1062	1073	1125	1128	1134	1137	1226	1229					
DEC	641	1334													
EMT	364														
HALT	332	659	875	885	916										
INC	602	614	684	796	841	983	1065	1089	1140	1141	1241	1351			
INCB	1020	1024													
JMP	346	573	587	660	663	679	682	1091	1261						
JSR	395	407	414	417	423	432	435	443	451	454	502	524	541	544	552
	574	603	608	615	624	635	643	650	665	676	685	694	706	712	716
	722	726	733	741	746	750	757	763	769	779	785	792	876	886	897
	903	909	917	1296	1304										
MOV	369	373	377	381	383	384	385	390	391	392	404	405	413	415	416
	418	419	420	422	424	425	426	437	438	439	442	444	445	446	458
	459	460	461	463	464	476	477	478	479	481	482	493	494	495	497
	498	499	501	515	516	517	519	520	521	523	536	537	538	539	540

	542	543	548	549	550	551	571	581	585	606	632	634	638	642	648
	655	662	672	683	691	693	697	699	705	709	711	715	719	721	724
	725	730	732	736	740	745	749	754	756	760	762	766	768	772	776
	778	782	784	788	791	830	831	832	833	834	835	836	855	856	857
	858	859	860	867	871	878	883	890	894	896	900	902	906	908	912
	984	985	987	1007	1008	1032	1034	1049	1116	1200	1205	1216	1282	1288	1293
	1295	1299	1303	1311	1314	1326	1337	1354							
MOV B	578	590	645	669	688	727	790	981	1003	1006	1028	1097	1357		
NEG	1118	1206	1217	1249	1251	1255	1257								
ROL	839	840	888	889	1012	1014	1016	1067	1068	1096	1143				
ROL B	1013	1015	1017	1072	1076	1239									
ROR	1209	1210	1220	1221	1238										
ROR B	1081														
RTI	607	797	918	920	976	982									
RTS	597	861	877	1035	1177	1339	1358								
SEC	1066	1095													
SUB	700	891	1071	1147	1236										
TRAP	365														
TST	375	379	428	430	447	449	496	569	579	583	600	618	622	630	636
	646	651	653	657	661	670	674	680	689	695	707	717	728	734	752
	758	764	770	774	780	786	793	865	869	873	881	892	898	904	910
	914	1030	1052	1057	1060	1079	1093	1098	1101	1103	1106	1123	1132	1135	1152
	1155	1157	1161	1164	1170	1194	1197	1203	1214	1243	1245	1247	1253	1258	1286
	1291	1297	1312	1324											
TST B	970	1018	1021	1063	1069	1083	1086	1108	1126	1129	1138	1145	1230	1280	1352
	1355														
.ABS	330														
.ASCII	800	802	803	804	806	808	810	812	814	818	821	921	925	927	928
	933	938	943	947	1360	1361	1363	1365	1366						
.BYTE	590	1001													
.ENABL	1	299													
.ENC	1370														
.ENDC	302	307	315	329	398	400	994	996	1037	1040	1179	1182	1265	1268	1342
	1344														
.EVEN	829	949	991	1367											
.IF	300	303	314	327	397	399	993	995	1036	1039	1178	1181	1264	1267	1341
	1343														
.IFF	300	315	327	398	400	994	996	1037	1040	1179	1182	1265	1268	1342	1344
.IIF	302	307													
.LIST	1	299	300	301	302	327	328	329	332	368					
.MACR	368														
.MACRO	1	299													
.NLIST	1	299	300	301	302	327	328	329	332	368					
.REM	1														
.REPT	300	327	332												
.TITLE	302														
.WORD	950	951	952	953	954	955	956	965	966	967	968	999	1043	1044	1045
	1046	1186	1187	1188	1270	1271	1272	1273	1274	1275					

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

* ,DZKECA.SEG/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.CO,DZKECA.P11

L03

MAINDEC-11-DZKEC-A MACY11 27(732) 03-NOV-76 15:35 PAGE 41
DZKECA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

RUN-TIME: 24 23 2 SECONDS
RUN-TIME RATIO: 136/55=2.4
CORE USED: 33K (65 PAGES)

