MAINDEC 4

MULTIPLY AND DIVIDE TEST

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

COPY NO.

This manual contains proprietary information. It is provided to the customers of Digital Equipment Corporation to help them properly use and maintain DEC equipment. Revealing the contents to any person or organization for any other purpose is prohibited.

MAINDEC 4

MULTIPLY AND DIVIDE TEST

Abstract: Maindec 4 is a maintenance program designed to test the automatic multiply - divide and the instructions MUS and DIS (multiply step and divide step).

The method of testing is to obtain numbers from a random number generator and multiply or divide them.

The results are compared with those of a routine which simulates the hardware used. The program runs continuously using the typewriter as the indicator for all errors.

TABLE 1 TAPES REQUIRED

Maindec 4 (Multiply and Divide Test)

TABLE 2 SWITCHES

Switch	Setting	Function
SENSE Switch 1	1	Inhibits the testing of divide or DIS.
	0	No effect.
SENSE Switch 2	1	Inhibits the testing of multiply or MUS.
	0	No effect.
SENSE Switch 3	1	Inhibits all error printouts.
	0	No effect.
SENSE Switch 4	1	The random number generator does not advance.
	0	No effect.
TEST ADDRESS	0000	Starting address for automatic multiply and divide.
TEST ADDRESS	0001	Starting address for instructions MUS and DIS.

TABLE 3 LOAD SEQUENCE

1. Place Multiply - Divide Test in reader.
2. Depress READ IN.
3. The program will start automatically at address 0000. If MUS and DIS are to be tested, place TEST ADDRESS switches at 0001 and depress START.
4. Check to see that the MUS, MUL, DIS, and DIV switches are properly set.

TABLE 4 ERROR TYPEOUT FOR A DIS FAILURE

The first line typed informs the operator that the failure occurred on a DIS instruction. It also gives the contents of the AC, I/O, and memory prior to the execution of the DIS instruction.

The next line typed is what the AC, I/O, and MB should read at the completion of the DIS instruction.

The third line, typed in red, is the answer obtained from the DIS instruction.

TABLE 5 ERROR TYPEOUT FOR A MUS FAILURE

The first line typed informs the operator that the failure occurred on a MUS instruction. It also gives the contents of the AC, I/O, and memory prior to the execution of the MUS instruction.

The next line typed is what the AC, I/O and MB should read at the completion of the MUS instruction.

The third line, typed in red, is the answer obtained from the MUS instruction.

TABLE 6 ERROR TYPEOUT FOR A MULTIPLY FAILURE

The first line typed informs the operator that the failure occurred when multiply was used. It also gives the contents of the AC, I/O, and memory prior to the execution of the multiply.

This is followed by the simulator printout, which consists of the SRC, AC, I/O and MB. The first line printed occurs at the completion of TP-10 CY1. Subsequent lines give the contents of the AC, I/O and MB at the completion of MDP-5.

The last line typed is the contents of the registers at the completion of the multiply.

For a flow chart of automatic multiply, refer to Figure 7-5, page A 84 of the PDP-1 Maintenance Manual.

TABLE 7 ERROR TYPEOUT FOR A DIVIDE FAILURE

There are three possible causes for a divide failure. The first is where the division is good and the answer is correct but the program counter fails to increment.

In this case the error typeout consists of the contents of the AC, I/O and MB prior to the divide and the result of the divide.

The result is typed in red. The divide simulator is not typed.

The second case is that of bad divide. The term "bad divide" implies that the quotient would be larger than the AC. The division is not performed.

If the AC or I/O is altered by the divide, the typeout consists of the following:

1. The instruction performed (DIV).
2. The contents of the AC, I/O and MB prior to the execution of divide.
3. The first two steps and last step from the divide simulator.
4. The contents of the AC, I/O and MB at the completion of the divide instruction.
5. Those bits which differ from the original AC and I/O and therefore are in error.

The third case is where the answer from the hardware from a good division does not equal the answer from the simulator.

The typeout is as follows:

1. The instruction performed (DIV).
2. The contents of the AC, I/O and MB prior to the execution of divide.

The lines which follow this typeout tell the shift register to count the contents of the AC, I/O and MB.

Typeouts occur at the following times:

1. TP-10 CY1.
2. MDP-5 prior to the increment of SCR.
3. MDP-14

TABLE 7 ERROR TYPEOUT FOR A DIVIDE FAILURE (continued)

Refer to Figure 7-6, page A88 of the PDP-1 Maintenance Manual for a flow chart of the divide operation.

SUGGESTED APPLICATION

Because of time considerations, it is impossible to multiply and divide all possible combinations of numbers. Therefore, the numbers used are obtained from a random number generator.

If the multiply - divide logic makes errors only on a particular combination of numbers, the random number generator may take numerous steps to produce those numbers.

It is therefore desirable to allow the program to run as long as is practical. When a periodical check of the equipment is made, use of the marginal check facility is recommended.

If a machine malfunction occurs and the cause does not become obvious from the typeout, the following procedure may be useful:

1. For divide: With the TEST WORD switches, place in memory a program which divides the same numbers that failed.
2. Unsolder the wire on 2L2-H (MDP-5), and run a clip lead from 2L2-H to 1J1-E (the output of the stop pulse amplifier). The stop pulse may now be used to single step through the divide timing chain. The results of each timing chain may now be checked against the divide simulator results.
3. For multiply: With the TEST WORD switches, place a program in memory that multiplies the same numbers that failed.
4. Unsolder the wire on 2L9-W (MDP-6), and run a clip lead from 2L9-W to 1J1-E (the output of the stop pulse amplifier). The stop pulse may now be used to single step the multiply. The results of each timing chain may be checked against the multiply simulator results.

PROGRAM DESCRIPTION

Maindec 4 is a maintenance program designed to detect computer malfunctions in the automatic multiply - divide and the MUS and DIS instructions.

If the program starts at address 0000, it uses the MUL and DIV simulators. If it starts at address 0001, it uses the MUS and DIS simulators.

The program looks at SENSE switches 4, 1, and 2. A SENSE switch on zero causes the execution of the appropriate routine.

The method of testing is to compare the results of the hardware with the results from a simulator. If the results differ, a second pass is made with the simulator typing the results of each timing chain.

The program runs continuously with no computer halts.

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST

Location	Contents	Mnemonic Code	Remarks
0000	600532	jmp ess	
0001	600517	jmp dss	
0002	0 opt,	0	
0003	260023	dap opx	
0004	710006	setup aaā,6	
0005	241046		
0006	200002	lac opt	
0007	661007	ral 7	
0010	240002	dac opt	
0011	21000	and (7	
0012	650100	sza i	
0013	700020	law char r0	
0014	663777	swap	
0015	663777		
0016	730003	tyo	
0017	461046	count aaa,opt 4	
0020	600006		
0021	221001	lio (36	
0022	730003	tyo	
0023	600023 opx,	jmp .	
0024	260065 et1,	dap et3	
0025	201047	lac src̄	
0026	671077	rar 77	
0027	640005	szf 5	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0030	600053	jmp eta	
0031	764000	pri2	
0032	663007		
0033	663777		
0034	663777		
0035	650100		
0036	700020		
0037	663777		
0040	663777		
0041	730003		
0042	764000	pri2	
0043	663007		
0044	663777		
0045	663777		
0046	650100		
0047	700020		
0050	663777		
0051	663777		
0052	730003		
0053	221001 eta,	lio (36	
0054	730003	tyo	
0055	201050	lac ac̄	
0056	170002	jda opt	
0057	201051	lac io̅	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0060	170002	jda opt	
0061	201052	lac me$\overline{m}$	
0062	170002	jda opt	
0063	221002	return	
0064	730003		
0065	600065 et3,	jmp .	
0066	260103 ran,	dap rn	/random number
0067	221053	lio ra$\overline{1}$	
0070	201054	lac ra$\overline{2}$	
0071	673771	rcr 771	
0072	61053	xor ra1	
0073	221054	lio ra2	
0074	241054	dac ra2	
0075	321053	dio ra1	
0076	663007	rcl 7	
0077	401054	add ra2	
0100	61053	xor ra1	
0101	241052	dac me$\overline{m}$	
0102	241055	dac me$\overline{1}$	
0103	600103 rn,	jmp .	
0104	0 a,	0	
0105	260200	dap ret	
0106	200104	lac a	
0107	341056	dzm sm$\overline{b}$	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0110	341057	dzm srm̄	
0111	341047	dzm src̄	
0112	640200	spa	/if minus
0113	761000	cma	/complement
0114	663777	swap	
0115	663777		
0116	200104	lac a	
0117	640200	spa	
0120	441057	idx srm	
0121	441047	idx src	
0122	760005	clf 5	
0123	321051	dio iō	/psuedo in/out
0124	201052	lac mem̄	
0125	640200	spa	
0126	441056	idx smb	
0127	201052	lac mēm	
0130	640200	spa	
0131	761000	cma	
0132	241052	dac mem	
0133	341050	dzm ac̄	/psuedo ac
0134	640002	szf 2	
0135	620024	jsp et1	
0136	221051 b,	lio io	
0137	672001	rir 1	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0140	652000	spi i	/test io 17
0141	600145	jmp . 4	
0142	201050	lac ac	
0143	401052	add mem	
0144	241050	dac ac	/mb pad and carry
0145	441047	idx src	/increment scr
0146	201050	lac ac	
0147	221051	lio io	
0150	677001	scr 1	/multiply shift
0151	21003	and (377777	
0152	241050	dac ac	
0153	321051	dio io	
0154	640002	szf 2	
0155	620024	jsp et1	
0156	201047	lac src	
0157	521004	sas (22	
0160	600136	jmp b	/scr $\overline{22}$
0161	441047	idx src	
0162	760015	stf 5	
0163	201051	lac io	
0164	41050	ior ac	
0165	650100	sza i	
0166	600173	jmp ret-5	
0167	201057	lac srm	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0170	61056	xor smb̄	
0171	640100	sza	/test for sign
0172	600201	jmp c	
0173	640002	szf 2	
0174	620024	jsp et1	
0175	201050	lac ac	
0176	221051	lio io	
0177	760005	clf 5	
0200	600200 ret,	jmp .	
0201	201050 c,	lac ac	
0202	761000	cma	
0203	241050	dac ac	
0204	201051	lac io	
0205	761000	cma	
0206	241051	dac io	
0207	600173	jmp ret-5	
0210	0 d,	0	
0211	321051	dio iō	
0212	260405	dap dre	/divide
0213	760005	clf 5	
0214	760004	clf 4	
0215	341056	dzm smb̄	
0216	341057	dzm srm̄	
0217	341047	dzm sr̄c	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0220	201052	lac mēm	
0221	640200	spa	/if mb minus set smb
0222	441056	idx smb	
0223	201052	lac mem	
0224	640400	sma	
0225	761000	cma	/comp mb if minus
0226	241052	dac mem	
0227	200210	lac d	
0230	640200	spa	/if ac minus set srm
0231	441057	idx sr̄m	
0232	200210	lac d	
0233	241050	dac ac̄	
0234	200210	lac d	
0235	640400	sma	/if ac minus
0236	600244	jmp . 6	/comp ac io
0237	761000	cma	
0240	241050	dac ac̄	
0241	201051	lac io	
0242	761000	cma	
0243	241051	dac io	
0244	640002	szf 2	
0245	620024	jsp et1	
0246	600266	jmp e	
0247	441047 f,	idx src	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0250	201052	lac mem	
0251	221050	lio ac	
0252	652000	spi i	
0253	761000	cma	
0254	241052	dac mem	/+ ac comp mb
0255	201050	lac ac	
0256	221051	lio io	
0257	61005	xor (400000	
0260	663001	rcl 1	
0261	241050	dac ac	
0262	321051	dio io	/divide shift
0263	672001	rir 1	
0264	652000	spi i	/if io 17, zero
0265	441050	idx ac	/increment ac
0266	201050 e,	lac ac	/ones comp add
0267	401052	add mem	/=pad, carry or clear
0270	241050	dac ac	
0271	201052	lac mem	
0272	640200	spa	
0273	761000	cma	/comp mb, if minus
0274	241052	dac mem	
0275	640002	szf 2	
0276	620024	jsp et1	
0277	201047	lac src	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0300	501004	sad (22	
0301	600310	jmp g	
0302	640100	sza	
0303	600247	jmp f	/f = mdp-1 routine
0304	201050	lac ac	
0305	640200	spa	
0306	600247	jmp f	
0307	600310	jmp g	
0310	441047 g,	idx src	
0311	201050	lac ac	
0312	401052	add mem	
0313	241050	dac ac	
0314	700002	law 2	
0315	21047	and src	
0316	650100	sza 1	
0317	600406	jmp eee	
0320	700020	law 20	
0321	21047	and src	
0322	650100	sza i	
0323	600330	jmp . 5	
0324	201050	lac ac	
0325	675001	sar 1	/shift mdp-9
0326	21003	and (377777	/make ac positave
0327	241050	dac ac	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0330	201050 h,	lac ac	
0331	650100	sza i	
0332	600341	jmp j	
0333	201057	lac srm	
0334	650100	sza i	
0335	600341	jmp j	
0336	201050	lac ac	
0337	761000	cma	
0340	241050	dac ac	
0341	201051 j,	lac io	
0342	650100	sza i	
0343	600357	jmp k	
0344	700020	law 20	
0345	21047	and src	
0346	650100	sza i	
0347	600357	jmp k	
0350	201057	lac srm	/spq level
0351	61056	xor smb	
0352	650100	sza i	
0353	600357	jmp k	
0354	201051	lac io	
0355	761000	cma	/if(io not +0)·scr 1
0356	241051	dac io	/spq ,comp io
0357	700020 k,	law 20	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0360	21047	and src	
0361	640100	sza	
0362	600371	jmp 1	
0363	201057	lac srm	
0364	650100	sza i	
0365	600371	jmp 1	
0366	201051	lac io	/if scr 0=0 •srm =1
0367	761000	cma	/comp io
0370	241051	dac io	
0371	221051 1,	lio io	
0372	440405	idx dre	
0373	321052	dio mem	
0374	201050	lac ac	
0375	663777	swap	
0376	663777		
0377	241050	dac ac	
0400	321051	dio io	
0401	760015	stf 5	
0402	640002	szf 2	
0403	620024	jsp et1	
0404	760005	clf 5	
0405	600405 dre,	jmp .	
0406	221055 eee,	move me1,mem	
0407	321052		

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0410	221054	move ra2,io	
0411	321051		
0412	201053	lac ra1	
0413	241050	dac ac	
0414	600401	jmp dre-4	
0415	260437 bb,	dap bb2	
0416	760002	clf 2	/mul inst
0417	221053	move ra1,ac	
0420	321050		
0421	0 fm4,	0	
0422	201050	lac ac̄	
0423	541052	mul mem	
0424	241060	dac ac1̄	
0425	321061	dio io1̄	
0426	201050	lac ac	
0427	0 fm2,	0	
0430	521060	sas ac1	
0431	760012	stf 2	
0432	663777	swap	
0433	663777		
0434	521061	sas io1	
0435	760012	stf 2	
0436	650002	szf i 2	
0437	600437 bb2,	jmp .	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0440	640030	szs 30	
0441	600437	jmp bb2	
0442	221055	move me1,mem	
0443	321052		
0444	0 fm5,	0	
0445	221053	move ra1,ac	
0446	321050		
0447	201006	lac (flexo mul	
0450	640005	szf 5	
0451	201007	lac (flexo mus	
0452	170567	jda inf	/print ac,io,mem
0453	201050	lac ac	
0454	0 fm3,	0	
0455	620760	jsp ans	/print mach ans
0456	600437	jmp bb2	
0457	221010 dd,	load ra1,257431	
0460	321053		
0461	221011	load ra2,671713	
0462	321054		
0463	221002	return	
0464	730003		
0465	650040	szs i 40	
0466	620066	jsp ran	
0467	650010	szs i 10	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0470	620703	jsp cc	/divide
0471	650020	szs i 20	
0472	620415	jsp bb	/multiply
0473	600465	jmp .-6	
0474	260516 sd,	dap sd2	/simulate dis
0475	201050	lac ac̄	
0476	221051	lio io̅	
0477	61005	xor (400000	
0500	663001	rcl 1	/dis shift
0501	321051	dio io	
0502	672001	rir 1	
0503	642000	spi	/if io 17 a one
0504	761000	cma	/comp AC
0505	652000	spi i	/add 1 to AC
0506	401012	add (1	/if io17 is zero
0507	401052	add mēm	/mb pad ,carry
0510	642000	spi	
0511	761000	cma	/if io 17 a one
0512	401013	add (777777	/comp ac
0513	241050	dac ac	/make -0 = +0
0514	640002	szf 2	
0515	620024	jsp et1	
0516	600516 sd2,	jmp .	
0517	760015 dss,	stf 5	/start for dis,mus

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0520	221014	load fd2,jsp sd	
0521	320722		
0522	320754	dio fd3	/setup for dis
0523	221015	load fm2,jsp sm	
0524	320427		
0525	320454	dio fm3	/setup for mus
0526	221016	load fm4,jsp ss2	
0527	320421		
0530	320444	dio fm5	
0531	600457	jmp dd	/start program
0532	760007 ess,	clf 7	/start for mul,div
0533	221017	load fd2,jda d	
0534	320722		
0535	320754	dio fd3	
0536	221020	load fm2,jda a	
0537	320427		
0540	320454	dio fm3	/setup for mul,div
0541	221021	load fm4,dzm io	
0542	320421		
0543	320444	dio fm5	
0544	600457	jmp dd	
0545	260550 ss2,	dap . 3	
0546	221054	move ra2,io	
0547	321051		

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0550	600550	jmp .	/setup io for mus
0551	260566 sm,	dap sm9	/simulate mus
0552	201050	lac ac	
0553	221051	lio io	
0554	672001	rir 1	
0555	642000	spi	
0556	401052	add mem	/io 17 a one,add
0557	662001	ril 1	
0560	673001	rcr 1	
0561	21003	and (377777	
0562	241050	dac ac	
0563	321051	dio io	
0564	640002	szf 2	
0565	620024	jsp et1	
0566	600566 sm9,	jmp .	
0567	0 inf,	0	
0570	260702	dap ing	
0571	200567	lac .-2	
0572	663077	type	
0573	730003		
0574	663077		
0575	730003		
0576	663077		
0577	730003		

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0600	764000	space	
0601	730003		
0602	201022	lac (flexo ac	
0603	663077	type	
0604	730003		
0605	663077		
0606	730003		
0607	663077		
0610	730003		
0611	201023	lac (flexo =	
0612	663077	type	
0613	730003		
0614	663077		
0615	730003		
0616	663077		
0617	730003		
0620	764000	space	
0621	730003		
0622	201050	lac ac	
0623	170002	jda opt	/print contents AC
0624	201024	lac (flexo io	
0625	663077	type	
0626	730003		
0627	663077		

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0630	730003		
0631	663077		
0632	730003		
0633	764000	space	
0634	730003		
0635	201023	lac (flexo =	
0636	663077	type	
0637	730003		
0640	663077		
0641	730003		
0642	663077		
0643	730003		
0644	764000	space	
0645	730003		
0646	201051	lac io	
0647	170002	jda opt	/print contents IO
0650	764000		
→ΛV→	<ΛD→→→	SPACE	
0651	730003		
0652	201025	lac (flexo mem	
0653	663077	type	
0654	730003		
0655	663077		
0656	730003		

Location	Contents	Mnemonic Code	Remarks
0657	663077		
0660	730003		
0661	764000	space	
0662	730003		
0663	201023	lac (flexo =	
0664	663077	type	
0665	730003		
0666	663077		
0667	730003		
0670	663077		
0671	730003		
0672	764000	space	
0673	730003		
0674	201052	lac mem	
0675	170002	jda opt	
0676	221002	repeat 2,return	
0677	730003		
0700	221002		
0701	730003		
0702	600702 ing,	jmp .	
0703	260757 cc,	dap cc2	
0704	221052	move mem,me1̄	
0705	321055		
0706	760002	clf 2	
0707	201053	lac ra1	

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0710	760003	clf 3	
0711	241050	dac ac	
0712	221054	lio ra2	
0713	321051	dio io	
0714	561052	div mem	
0715	760013	stf 3	
0716	241060	dac ac1̄	
0717	321061	dio io1̄	
0720	201050	lac ac	/to div simulator
0721	221051	lio io	
0722	0 fd2,	0	
0723	760003	clf 3	
0724	201050	lac ac	/does div = div simulator
0725	521060	sas ac1	
0726	760012	stf 2	
0727	201051	lac io	
0730	521061	sas io1	
0731	760012	stf 2	
0732	640003	szf 3	
0733	600736	jmp . 3	
0734	650002	szf i 2	
0735	600757	jmp cc2	/no error
0736	640030	szs 30	
0737	600757	jmp cc2	/reject error ss 3

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0740	221055	move me1,mem	
0741	321052		
0742	221053	move ra1,ac	
0743	321050		
0744	221054	move ra2,io	
0745	321051		
0746	201026	lac (flexo div	
0747	640005	szf 5	
0750	201027	lac (flexo dis	
0751	170567	jda inf	
0752	201050	lac ac	/print from simulator
0753	221051	lio io	
0754	0 fd3,	0	
0755	760000	nop	
0756	620760	jsp ans	
0757	600757 cc2,	jmp .	
0760	260777 ans,	dap an	
0761	201030	lac (flexo ans	
0762	221002	return	
0763	730003		
0764	221031	lio (35	
0765	730003	tyo	
0766	221060	move ac1,ac	
0767	321050		

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
0770	221061	move io1,io	
0771	321051		
0772	221055	move me1,mem	
0773	321052		
0774	170567	jda inf	
0775	221032	lio (34	
0776	730003	tyo	
0777	600777 an,	jmp .	
1000	7 constants		
1001	36		
1002	77		
1003	377777		
1004	22		
1005	400000		
1006	442443		
1007	442422		
1010	257431		
1011	671713		
1012	1		
1013	777777		
1014	620474		
1015	620551		
1016	620545		
1017	170210		

PROGRAM LISTING

MAINDEC 4 - MULTIPLY AND DIVIDE TEST (continued)

Location	Contents	Mnemonic Code	Remarks
1020	170104		
1021	341051		
1022	616300		
1023	743372		
1024	7146		
1025	446544		
1026	647125		
1027	647122		
1030	614522		
1031	35		
1032	34		