digital

MAINDEC CHANGE NOTICE

205MCN

12-DØTA-1
CHANGE NO.

Sheet 1 of 1

AUTHOR	PROGRAM DATE	PRODUCT LINE	MAINDEC NUMBER
W. LaFlamme DATE EXT. 4/17/72 362Ø	2/18/72	PDP-12 Family of 8 84-07032	12-DØTA

PROGRAM NAME FPP-12 TRACE-EPM **DEVICE** FPP-12

ITEM	
Ø	New Program submitted to the Program Library. Tests the EPM version of FPP-12.
1. 2/13/3	PROBLEM: FPP-12 Trace-EPM Program will not operate correctly connected to an LA3ØS. The reason for this is that the 8-bit ASCII code for an altmode character is 233 on the LA3ØS versus 375 on a standard TTY device. FIX: Change the contents of location 362 from 2 to 144 octal.

MDCN

Change 362 0002 to 0144

Change 177 to 7043

ECO FPP12A #5 has been installed

Change FIELD Φ

0113 - 2350 [0000
2350 - 1027 [0406
2351 - 7004 [7041
2352 - 7200
2353 - 7010
2354 - 5755
2355 - 7212
7204 - 5513

IDENTIFICATION

PRODUCT CODE: MAINDEC 12-D0TA-D

PRODUCT NAME: FPP-12 TRACE-EPM

DATE: FEBRUARY 18, 1972

AUTHOR: BILL LA FLAMME

MAINTAINER: DIAGNOSTIC GROUP

1. ABSTRACT
    ****

THIS PROGRAM IS DESIGNED TO AID THE ENGINEER IN TROUBLE SHOOTING THE FPP-12 WITH EPM HARDWARE. THE PROGRAM CONSISTS ESSENTIALLY OF THREE MAJOR SEGMENTS:

1.1 TELETYPE MONITOR
     ================

THE ENGINEER CAN CONTROL THE OPERATION OF THE PROGRAM AND INTERROGATE THE SIMULATOR THROUGH THE MONITOR. THERE ARE SEVERAL 2 CHARACTER COMMANDS USED FOR CONTROL. (SEE COMMANDS.) THESE COMMANDS ALLOW THE USER TO INPUT HIS OWN FPP-12 PROGRAM AND RUN IT. THE FPP-12 PROGRAM CAN CONSIST OF ANY SET OF FPP-12 INSTRUCTIONS. THE PROGRAM WILL RUN IN ANY OF THE 3 MODES OF OPERATION (FLOATING POINT, DOUBLE PRECISION, EXTENDED PRECISION).

1.2 SIMULATOR
     =========

THE SIMULATOR IS THE HEART OF THE PROGRAM. ALL OF THE HARDWARE REGISTERS ARE SIMULATED. THE FPP-12 IS STARTED IN THE MAINTENANCE MODE AND STEPPED THROUGH AN FPP-12 PROGRAM ONE TIME STATE AT A TIME. SIMULTANEOUSLY THE PROGRAM SIMULATES THE HARDWARE OPERATION. BEFORE STEPPING TO THE NEXT TIME STATE, THE PROGRAM COMPARES THE CONTENTS OF THE HARDWARE REGISTERS WHICH CAN BE READ WITH AN IOT, WITH THE CONTENTS OF THE CORRESPONDING SOFTWARE REGISTER.

THE FPP-12 PROGRAM CAN BE ANY SET OF FPP-12 INSTRUCTIONS LOCATED ANYWHERE IN CORE OUTSIDE THE LIMITS OF THE TRACE PROGRAM. THE INSTRUCTION SET CAN BE ANY LENGTH. AN OPERAND TABLE CAN BE ANY LENGTH AND ALSO CAN BE LOCATED ANYWHERE IN CORE OUTSIDE THE PROGRAM LIMITS.

1.3 MINI ROUTINES
     =============

A SET OF SUBROUTINES USED BY THE SIMULATOR TO PERFORM THE HARDWARE FUNCTIONS, COMPARE REGISTER CONTENTS, AND REPORT ERRORS.

Complet
DP

2. REQUIREMENTS

2.1 EQUIPMENT

1) AN FPP-12 FLOATING POINT PROCESSOR
   WITH EPM HARDWARE.

2) A PDP-8 OR PDP-12 WITH AT LEAST 8K OF MEMORY

3) AN ASR33 OR ASR35 TELETYPE

2.2 STORAGE

THE PROGRAM USES ALL OF FIELD 0 AND ALL OF FIELD 1

2.3 PRELIMINARY PROGRAMS

ALL PDP-8 OR PDP-12 PROCESSOR AND MEMORY DIAGNOSTICS.

3. LOADING PROCEDURE

LOAD THE PROGRAM WITH THE BIN LOADER, DIAL LOADER OR PS-8 LOADER.

4. STARTING PROCEDURE

START THE PROGRAM IN 8 MODE AT LOCATION 0020 IN FIELD 0.

THE PROGRAM WILL ENTER THE TELETYPE MONITOR AND TYPE AN ASTERISK
(*). THE PROGRAM IS NOW WAITING FOR INPUT FROM THE TTY.

5. OPERATING INSTRUCTIONS

THERE ARE TWO SETS OF OPERATING INSTRUCTIONS, THE BASIC SET (5.1) WILL AUTOMATICALLY ATTEMPT TO DETECT A FAULTY REGISTER WHITHIN A SPECIFIC TIME STATE.

THE COMPLETE SET (5,2) ALLOWS THE ENGINEER TO USE THE TROUBLE SHOOTING CAPABILITIES OF THE PROGRAM ALONG WITH A SCOPE TO ISOLATE A FAILING COMPONENT.

5.1 BASIC OPERATING INSTRUCTIONS.

1) SET SR02=1
2) SET ALL OTHER SWITCHES = 0
3) TYPE "ALT MODE"

THE FPP-12 IS STARTED IN THE MAINTENANCE MODE RUNNING AN FPP-12 PROGRAM WHICH WAS LOADED WITH THE TRACE PROGRAM, THE PROGRAM WILL RUN INDEFINITELY UNTIL AN ERROR IS DETECTED, THE ERROR WILL BE TYPED AND THE PROGRAM WILL WAIT FOR A RESPONSE FROM THE TTY KEYBOARD.

5.2 COMPLETE OPERATING INSTRUCTIONS

ANY SET OF FPP-12 INSTRUCTIONS CAN BE RUN AND CHECKED BY THE TRACE PROGRAM. THEREFORE A SET OF TTY INPUT COMMANDS ARE AVAILABLE TO ENABLE THE USER TO INPUT AND RUN HIS OWN FPP-12 PROGRAM. COMMANDS ARE ALSO AVAILABLE TO CONTROL THE OPERATION OF THE PROGRAM FOR TROUBLE SHOOTING. DIRECTIONS FOR USING THESE COMMANDS START AT PARAGRAPH 6.

THERE ARE 13 COMMANDS WHICH CAN BE INPUT THROUGH THE TELETYPE. ALL COMMANDS ARE TWO CHARACTERS AND ARE PRECEDED BY A PERIOD (.). MOST COMMANDS REQUIRE ONE OR MORE OPERANDS. EACH OPERAND MUST BE TERMINATED BY A SPACE OR A CARRIAGE RETURN. A SPACE SEPERATES THE ARGUMENTS WITHIN AN ARGUMENT STRING AND A CARRAIGE RETURN TERMINATES THE ARGUMENT STRING. TYPING A RUBOUT WILL DELETE ALL DATA AFTER THE LAST TERMINATOR. ANYTIME THAT THE SIMULATOR IS RUNNING, TYPING ANY CHARACTER WILL TRANSFER CONTROL TO THE TTY MONITOR AT THE END OF THE PRESENT TIME STATE. AT THIS TIME THE STATUS OF THE SIMULATED FPP-12 CAN BE INTERROGATED.

5.2.1 INSERTING AN FPP-12 PROGRAM

THE ".AS" COMMAND ALLOWS THE USER TO TYPE IN FOUR DIGIT OCTAL WORDS IN SEQUENCE TO BE RUN AS AN FPP-12 PROGRAM. THE FPP-12 INSTRUCTIONS YOU WISH TO TEST MUST BE TYPED INTO CORE. IF NO LOCATION IS GIVEN TO THE AS COMMAND, (SEE COMMANDS,) THE DATA IS INSERTED STARTING AT 4000 IN FIELD 1.

THE ".OP" COMMAND IS THE SAME AS THE ".AS" COMMAND EXCEPT THAT DATA IS INSERTED STARTING AT 4100 IN FIELD 1. THIS IS NORMALLY USED TO INSERT OPERANDS AND/OR A BASE TABLE.

5.2.2 RUNNING AN FPP-12 PROGRAM

THE FPP-12 PROGRAM CAN BE RUN IN THE TRACE MODE IN WHICH EACH TIME STATE IS TRACED AND CHECKED BY USING THE ".RT" COMMAND. OR IT CAN BE RUN IN THE FAST MODE WITH THE ".RF" COMMAND IN WHICH THE FPP-12 IS STARTED IN NORMAL MODE AND THE PROGRAM WAITS FOR IT TO EXIT. IN THE FAST MODE THE FPP-12 CAN BE STOPPED BY THE ".EX" COMMAND. THE FPP-12 PROGRAM CAN BE STARTED IN THE DOUBLE PRECISION MODE BY SETTING THE COMMAND REGISTER WITH THE ".CM" COMMAND.

5.2.3 ALL OTHER COMMANDS AND THE SWITCH REGISTER CAN BE USED TO CONTROL THE OPERATION OF THE SIMULATOR.

6. TTY MONITOR
****

THE TTY MONITOR ALLOWS THE USER TO COMUNICATE WITH THE PROGRAM. HE CAN INPUT PPP-12 INSTRUCTIONS AND OPERANDS TO BE RUN AND TRACED. HE CAN RUN THE TRACE IN THE SINGLE STATE OR CONTINUOUS MODE. HE CAN INTERROGATE THE STATUS OF MEMORY OR ANY REGISTER AT ANY TIME.

DUE TO MEMORY CONSTRAINTS, VERY LITTLE ERROR CHECKING IS DONE IN THE MONITOR.

THERE ARE SIX SPECIAL CHARACTERS USED IN THE MONITOR TO TELL IT WHAT TO DO. THESE 6 CHARACTERS ARE :

RETURN	TERMINATOR
SPACE	TERMINATOR
PERIOD	COMMAND SWITCH
RUBOUT	DELETES CURRENT DATA ENTRY
ALT MODE	SPECIAL EXIT
CNTRL	EXIT TRACE PROGRAM (SEE COMMANDS 7.)

THERE ARE A NUMBER OF COMMANDS WHICH ALLOW THE USER TO CONTROL THE PROGRAM (SEE "COMMANDS" 7.). MOST OF THE COMMANDS REQUIRE ONE OR MORE ARGUMENTS. ARGUMENTS WILL BE GROUPS OF 1 - 4 OCTAL DIGITS. EACH COMMAND IS SEPERATED FROM ITS ARGUMENT BY A SPACE. EACH ARGUMENT IS SEPERATED BY A SPACE, AND A LINE IS TERMINATED WITH A CARRAIGE RETURN.
THIS MEANS THAT EVERY COMMAND THAT ACCEPTS AN ARGUMENT MUST HAVE AT LEAST TWO TERMINATOR CHARACTERS. ONE TO TERMINATE THE COMMAND AND ONE TO TERMINATE EACH ARGUMENT. THIS HOLDS TRUE EVEN IF NO ARGUMENT IS ENTERED. IN THE CASE OF NO ARGUMENT BEING ENTERED, THE SECOND TERMINATOR TELLS THE COMMAND ROUTINE TO CHECK TO SEE IF AN ARGUMENT WAS INPUT. THE OPERATION OCCURS IN THE FOLLOWING ORDER.

PERIOD TELLS THE MONITOR THAT THE TWO CHARACTERS PRECEDING THE NEXT TERMINATOR ARE TO BE TAKEN AS A COMMAND.

TWO CHARACTER COMMAND AND A TERMINATOR TELLS THE MONITOR TO DECODE THE COMMAND AND TRANSFER TO THE COMMAND ROUTINE. THE COMMAND ROUTINE THEN PERFORMS THE PROPER FUNCTION. IF AN ARGUMENT IS NEEDED, THE COMMAND ROUTINE TRANSFERS CONTROL BACK TO THE MONITOR. THE MONITOR REMEMBERS WHICH COMMAND IS BEING EXECUTED.

THE NEXT TERMINATOR TRANSFERS CONTROL BACK TO THE COMMAND ROUTINE. THIS ALSO PASSES ON THE ARGUMENT IF ANY. EACH TIME THE PROGRAM TRANSFERES OUT OF THE MONITOR ONLY ONE ARGUMENT IS PASSED ON.

WHEN A COMMAND THAT USES A FIXED NUMBER OF ARGUMENTS (0 OR 1) IS FINISHED, THE PROGRAM TYPES AN ASTERISK (*) AND RETURNS CONTROL TO THE MONITOR.

6.1 COMMAND FORMATS

THE FOLLOWING IS THE FORMAT FOR EACH COMMAND, AFTER THE ARGUMENT, IN PARENTHESIS, IS THE NUMBER IF ARGUMENTS THAT THE COMMAND ACCEPTS, A "C" AS THE NUMBER OF ARGUMENTS INDICATES THAT THE COMMAND WILL ACCEPT ARGUMENTS CONTINUOUSLY UNTIL ANOTHER COMMAND IS INPUT, FOR THE MEANING OF EACH COMMAND AND A DETAILED DESCRIPTION SEE PARAGRAPHS 7 AND 8.

.AS XXXX	(C)	ASSEMBLE
.OP XXXX	(C)	LOAD OPERAND
.TY XX	(1 - 14)	TYPE REGISTER
.EL XX	(1 - 14)	ERROR LIST
.SA ADDR	(1)	SET STEP ADDRESS
.RA ADDR	(1)	RESET STEP ADDRESS
.EA ADDR	(1)	EXIT ADDRESS
.CL	(0)	CLEAR STEP SWITCH
.RT ADDR	(1)	RUN TRACE
.RT E ADDR	(2)	RUN TRACE IN EPM MODE
.RF ADDR	(1)	RUN FAST
.RF E ADDR	(2)	RUN FAST IN EPM MODE
.EX	(0)	EXIT THE FPP-12
.SH XX	(1)	SET SPECIAL SHIFT

7. COMMANDS

7.1 SUMMARY (SEE DESCRIPTION (8.) FOR MORE DETAILS)

7.1.1 INPUT AND OUTPUT

.AS	ASSEMBLE	FPP-12 INSTRUCTIONS AND/OR OPERANDS ARE INPUT IN OCTAL FROM THE TTY AND STORED SEQUENTIALLY IN MEMORY, EACH TIME A CARRIAGE RETURN IS INPUT THE PROGRAM TYPES THE NEXT MEMORY LOCATION, IF AN ## IS TYPED (# = OCTAL FIELD DESIGNATOR) FOLLOWED BY A SPACE AND 4 OCTAL DIGITS, THE LOCATION COUNTER IS CHANGED TO THIS FIELD AND ADDRESS, IF NO LOCATION IS INPUT, THE DATA IS STORED STARTING AT 4000 IN FIELD 1.
.OP	OPERAND	SAME AS THE .AS COMMAND EXCEPT THAT THE LOCATION COUNTER IS SET TO 4100 IN FIELD 1 AND CANNOT BE CHANGED BY THE USER.
.TY	TYPE	REQUESTED DATA IS TYPED FOR THE OPERATORS USE, THE DATA REQUESTS ARE:
	PQ	SIMULATED Q REGISTER
	FQ	FPP-12 Q REGISTER
	A	SIMULATED A REGISTER
	B	SIMULATED B REGISTER
	MQ	SIMULATED MQ REGISTER
	AC	SIMULATED FAC
	PC	SIMULATED FPC
	IR	SIMULATED INSTRUCTION REGISTER
	OP	SIMULATED OP ADDRESS
	AD	SIMULATED ADDRS REGISTER
	ST	SIMULATED STATUS REGISTER
	PS	SIMULATED TIME STATE REGISTER (SEE DESCRIPTION ON NEXT PAGE)
	M	MODE

FS	FPP-12 TIME STATE REGISTER (SEE DESCRIPTION BELOW)
AP	THE APT LOCATED IN THE PROGRAM, THIS IS THE APT WHICH IS USED IF NO OTHER IS SELECTED
X	SIMULATED INDEX REGISTERS
SH	SIMULATED SHIFT REGISTER
AL	ALL OF ABOVE
F AAAA N	N NO. OF WORDS STARTING AT FIELD F ADDRESS AAAA.
CNTRL D	RETURN TO DIAL
CNTRL P	RETURN TO PS8 MONITOR
CNTRL C	RETURN TO DTA MONITOR

TIME STATE REGISTER

AC BIT	FUNCTION
00	MOST SIGNIFICANT BIT OF THE TIME STATE
01	BIT 1 OF THE TIME STATE
02	BIT 2 OF THE TIME STATE
03	BIT 3 OF THE TIME STATE
04	CNR DEPOSIT FLOP (1) H
05	CNR FETCH FLOP (1) H
06	CNR EXECUTE FLOP (1) H
07	CNR EXIT FLOP (1) H
08	CNR INIATE FLOP (1) H
09	CNR PROCESS FLOP (1) H
10	TMSC SPECIAL STATE (1) H
11	TMSC EXECUTE (1) H

7.1.2 PROGRAM CONTROL

,SA	SET SINGLE STATE ADDR	4 DIGIT OCTAL ADDRESS IS INPUT FROM TTY, THE PROGRAM WILL ENTER THE SINGLE STATE MODE WHEN THIS ADDRESS IS ENCOUNTERED IN THE STEP ROUTINE
,RA	RESET SINGLE STATE ADDR	4 DIGIT OCTAL ADDRESS IS INPUT FROM THE TTY, THE PROGRAM WILL EXIT THE SINGLE STATE MODE WHEN THIS ADDRESS IS ENCOUNTERED IN THE STEP ROUTINE
,EA	EXIT ADDRESS	4 DIGIT OCTAL ADDRESS IS INPUT FROM THE TTY, WHEN THIS ADDRESS IS ENCOUNTERED IN THE STEP ROUTINE, THE PROGRAM WILL CLEAR THE FPP-12 WITH AN "FPICL" IOT AND GO TO INITIATE TO RESTART THE FPP-12 AND THE SIMULATOR
,CL	CLEAR SINGLE STATE SWITCH	CLEARS THE PROGRAM SINGLE STATE SWITCH WHICH WAS SET BY THE ,SA COMMAND.
,RT	RUN TRACE	4 DIGIT ADDRESS OF THE APT IS INPUT FROM THE TTY AND THE FPP-12 AND SIMULATOR ARE STARTED AT THIS ADDRESS, IF THE APT ADDRESS IS 0000 THE PROGRAM APT AND INDEX REGISTERS WILL BE USED TO RUN THE FPP-12 PROGRAM INPUT VIA THE ,AS COMMAND, IF THE USER SPECIFIES HIS OWN APT ADDRESS, THE FIELD BITS OF THE APT ADDRESS MUST BE SET IN THE LAST DIGIT OF THE ,CM COMMAND,
,RT E	START EPM	IF THE FIRST ARGUMENT OF THE ,RT COMMAND IS AN "E", THE FPP-12 WILL BE STARTED IN THE EPM MODE, AND THE PROGRAM WILL WAIT FOR A SECOND ARGUMENT CONSISTING OF THE APT ADDRESS (,RT E 0000)
,RF	RUN FAST	SAME AS RT EXCEPT THAT THE SIMULATOR IS NOT USED AND THE FPP-12 RUNS IN ITS NORMAL MODE (NOT MAINT)
,CM	SET COMMAND REGISTER	4 DIGITS ARE INPUT FROM THE TTY AND LOADED INTO THE PROGRAM COMMAND REGISTER, THIS WORD IS USED WITH THE "FPCOM" IOT TO START THE FPP-12, IF AN ALTERNATE APT ADDRESS IS USED WITH THE ,RT OR ,RF COMMANDS THE FIELD BITS OF THE APT ADDRESS MUST INPUT AS THE LAST DIGIT OF THE ,CM COMMAND,

.EX	EXIT FPP-12	A CPU FORCED EXIT IS ISSUED TO THE FPP-12. THE CURRENT INSTRUCTION IS FINISHED AND THE FPP-12 EXITS STORING THE APT.
.SH	SET SPECIAL SHIFT	2 DIGITS ARE LOADED FROM THE TTY INTO A SPECIAL SHIFT COUNTER IN THE PROGRAM, AND A SPECIAL SHIFT FLAG IS SET. IF NO DIGITS ARE INPUT BEFORE THE SECOND TERMINATOR, THE SPECIAL SHIFT COUNTER WILL BE SET TO Ø AND THE SPECIAL SHIFT FLAG WILL BE RESET. THIS SPECIAL SHIFT COUNT IS USED IN THE MULTIPLY AND DIVIDE TO ALTER THE NORMAL CYCLE OF THESE INSTRUCTIONS IN TIME STATE 2 ONLY.

8. DESCRIPTION OF COMMANDS

8.1 DATA INPUT AND OUTPUT

SINCE IT IS NECESSARY TO INPUT FPP-12 INSTRUCTIONS AND OPERANDS TO BE RUN, AND TO INTERROGATE THE STATUS OF THE SIMULATED REGISTERS, THE FOLLOWING TTY CONTROL COMMANDS ARE DEFINED. ANY TIME THAT THE TRACE PROGRAM IS RUNNING TYPING A CARRIAGE RETURN WILL TRANSFER CONTROL TO THE TTY MONITOR. TYPING "ALT MODE" WILL RETURN CONTROL TO THE TRACE PROGRAM.

8.1.1 ,AS

USED TO INPUT FPP-12 INSTRUCTIONS WRITTEN BY THE USER TO BE RUN BY THE TRACE PROGRAM. ANY NUMBER OF INSTRUCTIONS CAN BE INPUT TO BE RUN IN SEQUENCE AS AN FPP-12 PROGRAM. THE INSTRUCTIONS MUST BE CONVERTED INTO OCTAL CODE BY THE USER AND MUST BE INPUT IN OCTAL. CARE MUST BE TAKEN TO INPUT CORRECT FIELD DESIGNATORS IN THE COMMAND, THIS IS ESPECIALLY TRUE WITH BITS 9-11 OF DOUBLE WORD COMMANDS. INPUT DATA IS NORMALLY STORED SEQUENTIALLY STARTING AT 4000 IN FIELD 1, HOWEVER THE USER MAY CHANGE THE STORAGE LOCATION AT ANY TIME TYPING AN ASTERISK FOLLOWED BY A FIELD DESIGNATOR, A SPACE, AND 4 DIGIT ABSOLUTE ADDRESS (I.E. *2 3000). EACH TIME A CARRIAGE RETURN IS TYPED THE PROGRAM RESPONDS WITH THE NEXT ADDRESS TO BE STORED INTO. SINCE THE TTY MONITOR IS USED TO INPUT DATA IT IS NOT NECESSARY TO END THE ,AS COMMAND. ONCE THIS ROUTINE IS ENTERED IT WILL CONTINUE TO ACCEPT AND STORE DATA UNTIL A NEW COMMAND IS INPUT.

TWO EXTRA FEATURES OF THE ,AS COMMAND ALLOW THE USER TO INPUT THE INITIAL SETTINGS OF THE FAC AND INDEX REGISTERS. WHILE IN THE ,AS ROUTINE TYPING "AC XXXX XXXX XXXX" WILL CAUSE THE PROGRAM TO STORE THE THREE OCTAL ARGUMENTS INTO THE FAC PORTION OF THE APT. TYPING "X# XXXX" (# = THE INDEX REGISTER NUMBER) WILL CAUSE THE PROGRAM TO STORE THE OCTAL ARGUMENT INTO THE SPECIFIED INDEX REGISTER LOCATED IN THE PROGRAM. AFTER THE ARGUMENTS ARE INPUT AND STORED BY THE PROGRAM THE NORMAL PORTION OF THE ,AS ROUTINE CONTINUES. BOTH OF THESE FEATURES ASSUME THAT THE USER WILL START THE PROGRAM WITHOUT USING AN AUXILIARY APT ADDRESS (SEE ,RT AND ,RF).

8.1.2 ,OP

USED TO INPUT OPERANDS FOR USE WITH THE FPP-12 INSTRUCTIONS INPUT VIA THE ,AS COMMAND. OCTAL DATA IS INPUT FROM THE TTY AND STORED SEQUENTIALLY STARTING AT 4100 IN FIELD 1. THE STARTING ADDRESS CANNOT BE MODIFIED BY THE USER. LIKE THE ,AS COMMAND THIS ROUTINE HAS NO WAY TO END. THE ROUTINE WILL CONTINUE TO INPUT AND STORE DATA UNTILL A NEW COMMAND IS TYPED IN.

8,1,3 ,TY THE PROGRAM CAN TYPE OUT THE CONTENTS OF ANY MEMORY LOCATION(S), THIS IS NORMALLY USED TO INTERROGATE THE CONTENTS OF SIMULATED REGISTERS OR OPERANDS, (SEE "7,1,1 SUMMARY" FOR ARGUMENT LIST,), ALL TYPEOUTS EXCEPT FO, FS, AP ARE THE CONTENTS OF THE SIMULATED REGISTER AT THE TIME OF THE TYPEOUT, THIS IS THE INFORMATION THAT SHOULD BE CONTAINED IN THE FPP-12 HARDWARE REGISTER AT THE TIME,
THE FORMAT OF THE REQUEST IS:

,TY A B PO FO ETC

UP TO 18 ARGUMENTS CAN BE ENTERED IN ONE ARGUMENT STRING, EACH ARGUMENT IS SEPERATED BY A SPACE AND THE STRING IS TERMINATED BY A CARRAIGE RETURN,

A SCOPE CAN BE USED TO FIND OUT IF THE HARDWARE REGISTER IS CORRECT OR IN ERROR, IF IT IS KNOWN THAT AN ERROR OCCURS IN A SPECIFIC TIME STATE, THE FPP-12 AND TRACE PROGRAM CAN BE STOPPED BEFORE ENTERING THAT TIME STATE (SEE ,SA), AT THIS TIME THE USER CAN TYPEOUT THE CONTENTS OF VARIOUS REGISTERS AND COMPARE THE FPP-12 TO THIS DATA TO DETERMINE THE CAUSE OF THE FAILURE,

THE VARIABLE TYPEOUT (F AAAA N) CAN BE USED TO CHECK FPP-12 INSTRUCTIONS AND OPERANDS INPUT BY THE USER BEFORE RUNNING THEM, IF THE VARIABLE TYPE OUT IS USED, THE ARGUMENTS MUST BE THE FIRST 3 ARGUMENTS AND NO OTHER ARGUMENTS WILL BE PROCESSED

ANY TIME THE PROGRAM IS TYPING OUT, IT CAN BE STOPPED BY TYPING A RETURN ON THE KEYBOARD,
THE PROGRAM WILL THEN RETURN CONTROL TO THE TTY MONITOR,

8,1,3,1 ,EL ANY TIME AN ERROR IS DETECTED, THE TY ROUTINE IS ENTERED AFTER THE ERROR TYPEOUT (SEE "ERRORS" 10,) WITH A PRESET ARGUMENT STRING, THIS ARGUMENT STRING IS ENTERED WITH THE ,EL (ERROR LIST) COMMAND, WHEN THE PROGRAM IS LOADED THE ERROR LIST CONTAINS THE FPC AND INSTRUCTION REGISTER, THIS CAN BE CHANGED TO ANY ARGUMENT STRING THAT CAN BE ENTERED IN THE TY COMMAND EXCEPT THE VARIABLE TYPEOUT FORMAT,

8,1,4 CNTRL D THE PROGRAM WILL READ THE DIAL MONITOR IN FROM TAPE UNIT 0 AND TRANSFER CONTROL TO THE DIAL MONITOR,

8,1,5 CNTRL P THE PROGRAM JUMPS TO 7600 IN FIELD 0, IF THIS AREA OF MEMORY HAS NOT BEEN ALTERED, CONTROL WILL BE TRANSFERED TO PS8,

8,1,6 CNTRL C THE PROGRAM JUMPS TO 7700 IN FIELD 0, IF THIS AREA OF MEMORY HAS NOT BEEN ALTERED, CONTROL WILL BE TRANSFERED TO THE DTA MONITOR,

8,2 SINGLE STATE CONTROL

EVERY TIME STATE IS ENTERED THROUGH A SUBROUTINE CALLED "STEP", THE DEFINED INSTRUCTION "FSTEP" GOES TO STEP, IT IS IN THE "STEP" ROUTINE THAT MOST ERROR ARE DETECTED, THE "STEP" ROUTINE ALSO CHECKS VARIOUS INFORMATION INPUT BY THE USER TO CONTROL THE OPERATION OF THE PROGRAM, IN THE SINGLE STATE MODE THE PROGRAM WILL TRANSFER CONTROL TO THE TTY MONITOR EACH TIME IT ENTERS THE "STEP" SUBROUTINE, THIS IS DONE AFTER CHECKING FOR ERRORS, BUT BEFORE STEPPING THE PPP-12 TO THE NEXT TIME STATE, THE FOLLOWING COMMANDS ARE USED TO INPUT THE INFORMATION USED IN THE "STEP" ROUTINE, THE ADDRESS REFERRED TO IS THE ADDRESS IN WHICH THE "FSTEP" INSTRUCTION CALLING THE STEP ROUTINE IS LOCATED,

8,2,1 ,SA EACH TIME THE "STEP" ROUTINE IS ENTERED, THE ADDRESS INPUT AS THE ARGUMENT IN THE ,SA COMMAND IS COMPARED WITH THE ADDRESS IF THE "FSTEP" INSTRUCTION, IF THERE IS A MATCH, THE PROGRAM SETS THE PROGRAM SINGLE STATE SWITCH, THE PROGRAM WILL THEN BE IN THE SINGLE STATE MODE, THE PROGRAM SINGLE STATE SWITCH PERFORMS THE SAME FUNCTION AS SWITCH 3 ON THE CONSOLE, (SEE SWITCH OPTIONS 9.) THE PROGRAM WILL STEP THROUGH ONE TIME STATE EACH TIME AN "ALT MODE" IS TYPED, BEFORE TRANSFERRING TO THE TTY MONITOR THE PROGRAM WILL TYPE THE LOCATION+1 OF THE "FSTEP" INSTRUCTION, THE USER CAN THEN LOOK AT THE LISTING TO FIND WHICH TIME STATE IS ABOUT TO BE ENTERED, IF THE ,SA ARGUMENT IS 0000 THERE WILL NEVER BE A MATCH SO THE THE PROGRAM WILL NOT ENTER THE SINGLE STATE MODE,

8,2,2 ,RA EACH TIME THE "STEP" ROUTINE IS REENTERED FROM THE TTY MONITOR IN THE SINGLE STATE MODE, THE ADDRESS OF THE "FSTEP" INSTRUCTION IS COMPARED WITH THE ,RA ARGUMENT, IF THERE IS A MATCH, THE PROGRAM SINGLE STATE SWITCH IS RESET AND THE PROGRAM CONTINUES, BY USING THE ,SA AND ,RA COMMANDS TOGETHER, THE PROGRAM CAN BE RUN IN THE SINGLE STATE MODE FOR A FEW TIME STATES AND IN THE CONTINUOUS MODE FOR ALL EXCEPT THOSE TIME STATES,

8,2,3 ,CL THE PROGRAM SINGLE STATE SWITCH IS CLEARED BY THIS COMMAND. THE SET SINGLE STATE ADDRESS (,SA) AND RESET SINGLE STATE ADDRESS (,RA) ARE NOT AFFECTED. THIS COMMAND ALLOWS THE USER TO ENTER THE SINGLE STATE MODE AT A GIVEN TIME STATE WITH THE ,SA COMMAND AND STEP THROUGH A VARIABLE NUMBER OF TIME STATES, THEN CLEAR THE SINGLE STATE SWITCH TO RUN CONTINUOUSLY UNTILL THE TIME STATE IS REACHED AGAIN.

8,2,4 ,EA EACH TIME THE "STEP" ROUTINE IS ENTERED, THE ADDRESS OF THE "FSTEP" INSTRUCTION IS COMPARED WITH THE ,EA ARGUMENT. IF THERE IS A MATCH, THE PROGRAM ISSUES AN "FPICL" IOT TO THE FPP-12 AND GOES TO INIATE TO RESTART THE FPP-12 AND THE SIMULATOR. THIS COMMAND IS USEFULL IF AN ERROR OCCURS IN THE MIDDLE OF A MAJOR STATE, THE REMAINING TIME STATES AND INSTRUCTIONS CAN BE BYPASSED.
IF THE EA ADDRESS IS USED, THE EPM MODE IS RESET, THEREFORE IF THE EPM MODE IS BEING TESTED, THE FIRST INSTRUCTION OF THE FPP-12 PROGRAM SHOULD BE A "START E" INSTRUCTION (0050).

8.3 TRACE CONTROLS

THE FOLLOWING COMMANDS ARE USED TO SETUP, START AND STOP THE TRACE SIMULATION OF AN FPP-12 PROGRAM.

8.3.1 .CM

LOADS THE COMMAND REGISTER WITH ONE 12 BIT WORD (4 OCTAL DIGITS). THIS WORD IS ENTERED AS THE ARGUMENT OF THE .CM COMMAND. THE COMMAND REGISTER WILL BE USED WITH A "FPCOM" IOT WHEN STARTING THE FPP-12 AND THE SIMULATOR. IF AN AUXILIARY ADDRESS IS USED WITH THE .RT OR .RF COMMANDS, THE FIELD BITS OF THE APT ADDRESS MUST BE ENTERED AS THE LAST DIGIT OF THE .CM COMMAND.

8.3.2 .SH

A MAINTENANCE IOT IN THE FPP-12 ALLOWS A USER TO LOAD THE SHIFT COUNTER UNDER PROGRAM CONTROL. THIS IS ONLY USEFULL IN A MULTIPLY OR DIVIDE INSTRUCTION. TYPING .SH WITH AN ARGUMENT CONSISTING OF A TWO OCTAL DIGIT NUMBER WILL LOAD A SPECIAL SHIFT REGISTER WITH THE ARGUMENT AND SET A SPECIAL SHIFT FLAG. WHEN TIME STATE 2 OF A MULTIPLY OR DIVIDE IS ENTERED, THE PROGRAM LOOKS AT HE SPECIAL SHIFT FLAG. IF THE FLAG IS SET, THE SPECIAL SHIFT REGISTER IS LOADED INTO THE HARDWARE AND SIMULATED SHIFT COUNTER.

TYPING .SH FOLLOWED BY TWO TERMINATOR CHARACTERS WITH NO ARGUMENT WILL RESET THE SPECIAL SHIFT FLAG AND ZERO THE SPECIAL SHIFT REGISTER.

8,3,3 ,RT THIS COMMAND TRANSFERES CONTROL TO THE
TRACE SIMULATOR, THIS WILL START THE FPP-12
IN MAINTENANCE MODE AND TRACE EACH TIME STATE,

THE PROGRAM HAS THE ABILITY TO RUN A FPP-12
PROGRAM LOCATED ANY WHERE IN MEMORY OUTSIDE
THE LIMITS OF THE TRACE PROGRAM, THE APT AND
INDEX REGISTERS CAN ALSO BE LOCATED ANY WHERE
IN MEMORY EXCEPT LOCATION 0000 OF ANY MEMORY
FIELD, WITHIN THE TRACE PROGRAM IS AN APT
WHICH IS SET UP AUTOMATICALLY TO RUN A FPP-12
PROGRAM LOCATED AT 4000 IN FIELD 1, THE BASE
POINTER IS SET TO 4100 IN FIELD 1, THE INDEX
POINTER IS SET TO A SET OF INDEX REGISTERS
LOCATED WITHIN THE TRACE PROGRAM, THIS APT
AND INDEX TABLE IS FOR THE CONVIENENCE OF THE
USER SINCE THE ,AS AND ,OP COMMANDS STORE
THIER DATA IN FIELD 1 AT LOCATIONS 4000 AND 4100
RESPECTIVLY, THIS APT AND INDEX TABLE
WILL BE USED IF THE USER SPECIFIES ADDRESS
0000 AS THE APT ADDRESS, THE USER MAY SPECIFY
THE ADDRESS OF HIS OWN APT BY TYPING ",RT ADDR",
THE AUXILIARY APT AND INDEX REGISTERS MAY BE
SETUP BY USING THE ,AS COMMAND, THESE MAY ALSO
BE PART OF A REAL FPP-12 PROGRAM LOCATED IN
FIELD 2 OR ABOVE WHICH WAS PREVIOUSLY LOADED
BY SOME OTHER MEANS,
IF THE ARGUMENT ENTERED IS AN "E", THE FPP-12
WILL BE STARTED IN THE EPM MODE AND THE PROGRAM
WILL WAIT FOR A SECOND ARGUMENT WHICH WILL BE
THE APT ADDRESS, (,RT E 0000)
IF THE ARGUMENT IS NOT AN "E" THEN ONLY ONE
ARGUMENT WILL BE ACCEPTED,

8,3,4 ,RF

THIS COMMAND STARTS THE FPP-12 IN THE NORMAL RUNNING MODE. NO ERROR CHECKING OR TRACING IS DONE. THE PRIMARY USE OF THIS COMMAND IS TO ALLOW THE FPP-12 TO RUN A PROGRAM AT ITS FASTEST SPEED FOR SCOPING. THE ARGUMENTS ARE THE SAME AS THOSE FOR ,RT (SEE 8,3,3). ONCE THE FPP-12 IS STARTED THE PROGRAM WAITS IN A LOOP FOR THE FPP-12 TO EXIT OR FOR AN INPUT FROM THE TTY KEYBOARD. THE USER CAN TYPE A CARRAIGE RETURN TO BRING CONTROL TO THE TTY MONITOR. AT THIS TIME THE FPP-12 IS STILL RUNNING BUT THE PROGRAM IS NOT CHECKING IT. TYPING A ,EX COMMAND AT THIS TIME WILL FORCE THE FPP-12 TO EXIT AND RETURN CONTROL TO THE TTY MONITOR.

8,3,5 ,EX

IF THE FPP-12 IS RUNNING, A "CPU FORCE EXIT" IOT IS ISSUED. SUFFICIENT MAINTENANCE IOTS ARE ISSUED TO ALLOW THE FPP-12 TO COMPLETE ITS CURRENT INSTRUCTION AND STORE THE APT, THUS PERFORMING A NORMAL EXIT.

9.1 SWITCH OPTIONS (USED ONLY IN TRACE MODE)

SR00 (1) INHIBIT ERROR HALT

SR01 (1) INHIBIT ERROR TYPEOUTS

SR02 (1) RESTART PROGRAM AFTER AN EXIT. USED IN BOTH TRACE AND FAST MODE.

SR03 (1) SINGLE STATE MODE. OVERRIDES THE .SA .RA .CL COMMANDS (SEE "SINGLE STATE CONTROLS" 8.2)

SR04 (1) ENTER STOP. STOPS THE PROGRAM EACH TIME A MAJOR STATE IS ENTERED. TYPE ALT MODE TO CONTINUE.

SR05 (1) TTY TRACE. TYPES EACH MAJOR STATE AS IT IS ENTERED.

SR06 (1) INHIBIT ERROR LIST TYPEOUT. ERROR LIST IS THE ARGUMENT STRING ENTERED WITH THE .EL COMMAND. THE ERROR LINE OF THE TYPEOUT WILL STILL BE TYPED IF SR01 = 0.

SR11 (1) OUTPUT MESSAGES ON THE LP08 OR LP12 LINE PRINTER INSTEAD OF THE TTY.

ERRORS

THE TRACE PROGRAM CAN DETECT 7 TYPES OF ERRORS IN THE FPP-12.

THERE IS ONE TYPEOUT IN THE ERROR ROUTINE WHICH MAY OR MAY NOT BE AN ERROR, THIS TYPEOUT OCCURS IF THE FPP-12 DID NOT CAUSE A SKIP AFTER A "FPST" IOT, AFTER THE TYPEOUT, CONTROL IS TRANSFERED TO THE TTY MONITOR, TYPING "ALT MODE" WILL CAUSE THE PROGRAM TO CONTINUE RIGHT AFTER THE "FPST" IOT, THIS TYPEOUT IS:

"FPP-12 DID NOT START"

ONE ERROR, DATA ERROR STORING THE APT, WHICH IS THE LEAST LIKELY TO OCCUR HAS ITS OWN MESSAGE FORMAT, THIS ERROR TYPES THE MESSAGE:

"ERROR SAVING APT IN EXIT"

AFTER TYPING THE MESSAGE, SINCE THE FPP-12 HAS FINISHED ITS EXIT, THE PROGRAM CONTINUES AS THOUGH NO ERROR OCCURED.

ALL OTHER ERRORS TYPE A MESSAGE WITH THE SAME FORMAT,

```
"ERROR XXXX * MAJOR STATE * REG"
 FPC    =FIELD ADDR
 IR     =INST
```

THE MEANING OF THIS MESSAGE IS:

ERROR	SINCE THE TTY IS USED FOR OTHER TYPEOUTS, THIS IS DISTINGUISHED AS AN ERROR MESSAGE.
XXXX	THE P,C, OR P,C,+1 OF THE INSTRUCTION THAT TRANSFERED TO THE ERROR CHECKING ROUTINE, BY LOOKING IN THE LISTING AT THE P,C, LOCATION (ALWAYS FIELD Ø), THE USER CAN DETERMINE WHICH TIME STATE FAILED, IF THE P,C, POINTS TO A LOCATION OF AN "FSTEP" INSTRUCTION, THE ERROR WAS GENERATED IN THE PREVIOUS TIME STATE.
MAJOR STATE	THE MAJOR STATE IN WHICH THE ERROR WAS DETECTED, THIS IS OFTEN THE FAILING INSTRUCTION,
REG	THE REGISTER FOUND TO BE IN ERROR, I,E, O REG,
FPC IR	THESE 2 LINES ARE THE REQUESTS ENTERED INTO THE ERROR LIST (SEE ,EL COMMAND 8,1,3,1), THE ERROR LIST CONTAINS THE FPC AND IR IF NO LIST WAS ENTERED BY THE USER, THIS ERROR LIST CAN BE UP TO 18 LINES, SETTING SRØ6 = 1 WILL INHIBIT THIS PORTION OF THE TYPEOUT, IF SRØ1 = 1 THE ENTIRE TYPEOUT WILL BE INHIBITED.

EXAMPLE
****

```
ERROR 3600 * MULTIPLY * O REG
FPC      =0001 4261
IR       =4411
```

THE ERROR WAS DETECTED IN THE O REGISTER DURRING THE MULTIPLY MAJOR STATE. THE P,C, TELLS US THAT THE ERROR OCCURED IN TIME STATE 2 (THE P,C, POINTS TO THE "FSTEP" INSTRUCTION TO ENTER MULTIPLY STATE 2. THIS IS FOUND IN THE LISTING AT ADDRESS 3600).

THE MULTIPLY INSTRUCTION IS LOCATED IN FIELD 1 AT LOCATION 4257. THE FPC = 1-4261 WHICH IS THE INSTRUCTION LOCATION + 2 OF A DOUBLE WORD INSTRUCTION.

THE INSTRUCTION IS A DOUBLE WORD MULTIPLY USING INDEX REGISTER 1 WITH NO INCREMENT (4411). SEE THE FPP-12 INSTRUCTION CODE CARD FOR DETAILS OF THE ADDRESS MODES AND THE OP CODES.

10',2 ERROR HALT

IF SRØØ=0 THE ERROR ROUTINE GOES TO THE TTY MONITOR AND WAITS FOR THE OPERATOR. TYPING "ALT MODE" WILL CAUSE THE PROGRAM TO CONTINUE.

11', THE PROGRAM LISTING **********

SEVERAL PRE-DEFINED INSTRUCTIONS ARE USED IN THE TRACE SIMULATOR, THESE INSTRUCTIONS ARE DEFINED AT THE BEGINNING OF THE LISTING.

THESE INSTRUCTIONS ARE USED TO SIMPLIFY THE CODING, AND TO MAKE THE LISTING RESEMBLE THE FPP-12 FLOW PRINTS, THE CODING OF THE SIMULATOR WAS WRITTEN USING THE FPP-12 FLOW PRINTS, MANY OF THE COMMENTS ARE COPPIED FROM THE FLOW PRINTS.

ANY TIME THE LISTING IS BEING USED, THE USER SHOULD ALSO HAVE A SET OF FPP-12 FLOWS, USING THE FPP-12 FLOWS WILL CLARIFY THE PROGRAM LISTING AND MAKE IT EASY TO FOLLOW, AS FEW INSTRUCTIONS AS POSSIBLE ARE USED TO ACCOMPLISH THE FUNCTION OF ONE BLOCK IN THE FLOWS.

THE START OF EVERY MAJOR STATE AND TIME STATE IS LABELED WITH A TAG THAT STATES AS CLOSE AS POSSIBLE WHICH MAJOR STATE AND TIME STATE IT IS, MOST OF THE TAGS CONSIST OF THE MAJOR STATE NAME, ABBREVIATED, FOLLOWED BY THE TIME STATE NUMBER, (IE, INIT2 = INITIATE STATE 2), AT THE BEGINNING OF EACH MAJOR STATE IS A TAG WITH THE MAJOR STATE NAME AND NO TIME STATE NUMBER, FROM THIS TAG TO THE FIRST TIME STATE ARE INSTRUCTIONS TO SETUP THE PROGRAM FOR THE MAJOR STATE AND CLEAR TEMPORARY REGISTERS AND FLAGS.

***** NOTE ***** NOTE ***** NOTE ***** NOTE ***** NOTE *****

SINCE DIAL AND PS-8 WILL NOT LOAD INTO THE LAST PAGE OF FIELD 0 THE "STEP" ROUTINE IS LOADED INTO FIELD 1 STARTING AT LOCATION 3200, THE FIRST TIME THE PROGRAM IS STARTED AT LOCATION 20, THE "STEP" ROUTINE IS SWAPPED WITH THE INFORMATION IN THE LAST PAGE OF FIELD 0, THE LISTING SHOWS NO CODE GENERATED FROM 7600-7777 OF FIELD 0 HOWEVER THIS IS WHERE THE "STEP" ROUTINE RESIDES WHILE RUNNING, TO WORK WITH THE "STEP" ROUTINE YOU MUST GO TO THE LISTING IN FIELD 1 AT 3200 AND CONVERT THE CODE TO ADDRESS 7600, SINCE THIS ROUTINE RESIDES IN ONLY ONE PAGE OF MEMORY IT USES ALL RELATIVE ADDRESSING THEREFORE NO CODE NEED BE CHANGED.

****

```
/       SW0     0       ERROR HALT
/       SW1     0       ERROR TYPEOUT
/       SW2     1       RESTART FPP AFTER EXIT
/       SW3     1       STEP MODE
/       SW4     1       ENTER STOP
/       SW5     1       TRACE
/       SW6     1       INHIBIT ERROR LIST
/       SW11    1       OUTPUT TO LINE PRINTER

/       CONTROLS

/       ,SA     LOAD START STEP ADDRESS
/       ,RA     LOAD RESET STEP ADDRESS
/       ,EA     LOAD EXIT ADDRESS
/       ,CL     CLEAR STEP FLAG (EXIT STEP MODE)
/       ,AS     ASSEMBLE
/       ,RT     RUN TRACE MODE
/       ,RF     RUN FAST MODE
/       ,TY     TYPE CONTENTS OF REGISTER
/       ,EL     ENTER ERROR LIST
/       ,CM     LOAD COMMAND REGISTER
/       ,EX     FPP EXIT
/       ,OP     LOAD OPERAND TABLE
/       ,SH     SET SHIFT COUNTER AND FLAG
/               NO INPUT WILL RESET FLAG

/       CNTRL D         RETURN TO DIAL
/       CNTRL P         RETURN TO PSB
/       CNTRL C         RETURN TO DTA MONITOR
```

```
/       TYPEOUTS

/       PO      PROGRAM 0 REGISTER
/       FO      FPP 0 REGISTER
/       A       A REGISTER
/       B       B REGISTER
/       MQ      MQ REGISTER
/       AC      PROGRAM FAC
/       PM      PROGRAM MEMORY REGISTER
/       PC      PROGRAM FPC
/       IR      PROGRAM FIR
/       OP      OP ADDRESS
/       AD      ADDRS
/       ST      STATUS
/       PS      PROGRAM STATE
/       FS      STATE READ FROM FPP
/       AP      APT
/       X       INDEX REGISTERS
/       SH      SHIFT REGISTER
/       M       MODE
/               ALL OF ABOVE
/       F AAAA N        FIELD, 12 BIT ADDR, COUNT


PMODE


/MACRO DEFINITIONS


DEFINE  NPAGE<
        JMP I   (',+200&7600
        PAGE>
```

```
                / FPP-12 IOT DEFINITIONS

6551            FPINT=  6551
6552            FPICL=  6552
6553            FPCOM=  6553
6554            FPHLT=  6554
6555            FPST=   6555
6556            FPRST=  6556
6557            FPIST=  6557

6561            FMAINT= 6561
6562            RSTATE= 6562
6563            ROMSW=  6563
6564            ROLSW=  6564
6565            RAPT=   6565
6567            LSHFT=  6567
6566            RDOP=   6566
6567            ROEPM=  6567

                /  LP08 IOT'S

6663            LSR=    6663
6661            LSF=    6661
6666            LPC=    6666


                /  LP12 IOT'S

6651            LSE=    6651
6664            LPR=    6664
6652            LCF=    6652
6654            LLB=    6654
6661            LSD=    6661


                /  IOT'S FOR BOTH LINE PRINTERS USED TO CHECK WHICH PRINTER TO USE

6662            SFLG=   6662            /SET PRINTER FLAG FOR TEST
6661            CFLG=   6661            /CHECK WHICH PRINTER IS AVAILABLE
```

```
                / INSTRUCTION DEFINITIONS

4577    AMBO=   JMS I [SUBAB    /A MINUS B TO 0
4576    APBO=   JMS I [ADDAB    /A PLUS B TO 0
4575    CAPT=   JMS I [APTC     /COMPARE APT ADDRESS (ADRS)
4574    CLRA=   JMS I [CLRAX    /CLEAR THE A REGISTER
4573    CLRAC=  JMS I [CLRACX   /CLEAR FAC
4572    CLRB=   JMS I [CLRBX    /CLEAR THE B REGISTER
4571    CLRMQ=  JMS I [CLRMQX   /CLEAR MQ
4570    CLRO=   JMS I [CLROX    /CLEAR THE O REGISTER
4567    CMEME=  JMS I [CMEM1    /CHECK MEMORY EXPONENT
4566    CMEMF=  JMS I [CMEM2    /COMPARE MEMORY FRACTION
4565    DECAPT= JMS I [APTDEC   /DECREMENT THE APT ADDRESS (ADRS)
4564    DECOP=  JMS I [OPDEC    /DECREMENT THE OP ADDRESS
4563    ENTER=  JMS I [SETUP    /SETUP TO ENTER A MAJOR STATE
4562    FSTEP=  JMS I [7600     /CHECK REGS AND STEP TO THE NEXT TIME STATE
4561    GETAPT= JMS I [GAPT     /OUTBRK USING ADRS
4560    GETOP=  JMS I [GOP      /OUTBRK USING OP ADDRESS
4557    GETPC=  JMS I [GPC      /OUTBRK USING FPC
4556    GETX=   JMS I [XGETX    /GET PROGRAM INDEX REGISTER
4555    INCAPT= JMS I [INC3     /INCREMENT APT ADDRESS (ADRS)
4554    INCOP=  JMS I [INC1     /INCREMENT THE OP ADDRESS
4553    INCOR=  JMS I [OPLUS1   /INCREMENT THE O REGISTER
4552    INCPC=  JMS I [INC2     /INCREMENT THE FPC
4551    INCST=  JMS I [STINC    /INCREMENT TIME STATE REGISTER
4550    INCX=   JMS I [XPLUS1   /INCREMENT THE PROGRAM INDEX REGISTER
4547    LOADA=  JMS I [TOA      /LOAD THE A REG WITH REG IN NEXT LOCATION
4546    LOADAC= JMS I [TOAC     /LOAD THE FAC WITH REG IN NEXT LOCATION
4545    LOADB=  JMS I [TOB      /LOAD B REG WITH REG IN NEXT LOCATION
4544    LOADMQ= JMS I [TOMQ     /LOAD MQ REG WITH REG IN NEXT LOCATION
4543    LOADO=  JMS I [TOO      /LOAD O REG WITH REG IN NEXT LOCATION
4542    LOADOP= JMS I [TOOP     /LOAD OP ADDRESS WITH REG IN NEXT LOCATION
4541    LOADPC= JMS I [TOPC     /LOAD FPC WITH REG IN NEXT LOCATION
4540    MOVEX=  JMS I [PINDEX   /MOVE USER INDEX REGS TO PROG INDEX REGS
4537    NORM=   JMS I [XNORM    /NORMALIZE THE O REGISTER
4536    PUTX=   JMS I [XPUTX    /STORE DATA IN PROGRAM INDEX REGISTER
4535    REPMEN= JMS I [ENOFF    /RESET EPM ENABLE
4534    SAEZ=   JMS I [AEZ      /SKIP IF FAC = 0
4533    SAGZ=   JMS I [AGZ      /SKIP IF FAC IS GREATER THAN 0
4532    SALZ=   JMS I [ALZ      /SKIP IF FAC IS LESS THAN 0
4531    SAVE=   JMS I [TOTMP    /SAVE REG IN TEMP REG
4530    SEPM=   JMS I [SKPEPM   /SKIP IF EPM MODE
4527    SEPMEN= JMS I [ENON     /SET EPM ENABLE
4526    SHFTB=  JMS I [SHIFTB   /SHIFT B REG RIGHT USING SHIFT COUNT
4525    SHFTO=  JMS I [SHIFTO   /SHIFT O REG LEFT USING SHIFT COUNT
4524    SHIFT=  JMS I [SHIFTR   /SHIFT REG 1 BIT EITHER DIRECTION
4523    SOEZ=   JMS I [OEZ      /SKIP IF O REG = 0
4522    STORA=  JMS I [STRA     /STORE IN ALSW AND EXTEND SIGN TO AMSW
4521    STORB=  JMS I [STRB     /STORE IN BLSW AND EXTENT SIGN TO BMSW
4520    TOMEM=  JMS I [MEMINS   /SKIP IF ANSWER DOES NOT GO TO MEMORY
4517    TRACE=  JMS I [TRSKP    /TYPE TEXT IF TRACING PROGRAM (SR05 = 0)
```

```
                / REGISTER DEFINITIONS

0022    OMSW=   OREG
0023    OLSW=   OREG+1
0024    OLSW1=  OREG+2
0025    OLSW2=  OREG+3
0026    OLSW3=  OREG+4
0027    OEXT=   OREG+5

0030    AMSW=   AREG
0031    ALSW=   AREG+1
0032    ALSW1=  AREG+2
0033    ALSW2=  AREG+3
0034    ALSW3=  AREG+4
0035    AEXT=   AREG+5

0036    BMSW=   BREG
0037    BLSW=   BREG+1
0040    BLSW1=  BREG+2
0041    BLSW2=  BREG+3
0042    BLSW3=  BREG+4
0043    BEXT=   BREG+5

0044    MQMSW=  MQREG
0045    MQLSW=  MQREG+1
0046    MQLSW1= MQREG+2
0047    MQLSW2= MQREG+3
0050    MQLSW3= MQREG+4
0051    MQEXT=  MQREG+5

0060    TMSW=   TREG
0061    TLSW=   TREG+1
0062    TLSW1=  TREG+2
0063    TLSW2=  TREG+3
0064    TLSW3=  TREG+4

0052    ACEXP=  PFAC
0053    ACMSW=  PFAC+1
0054    ACLSW=  PFAC+2
0055    ACLSW1= PFAC+3
0056    ACLSW2= PFAC+4
0057    ACLSW3= PFAC+5

0053    FACFR=  PFAC+1
```

```
                0021    REGS=   OREG+1
                0000    ERROR=  0000
                7402    HALT=   7402
                4000    LDEXT=  4000
                7732    CADST=  CADS&177+7600
                7732    STEPSW= CADST
                7733    EXADDR= CADST+1
                7734    ENTSTP= CADST+2
                7735    EXSTP=  CADST+3
                7736    CKO=    CADST+4
                7737    CKOP=   CADST+5
                4000    LEFT=   4000
                0000    RIGHT=  0000
                4000    TEST=   4000
                4100    OPERND= 4100


                        /**********************BEGINNING OF PROGRAM**************************

                0000            *0
0000            0000    T1,     0
0001            0000    T2,     0
0002            0000    COMREG, 0
0003            0000    EPM,    0
0004            0000    PSTAT,  0
0005            0000    PIR,    0
0006            0200    PXP,    X0
0007            0000    CARYIN, 0




                0020    *20

0020            5716/           JMP     BEGIN           /FOR USE BY START 20 SWITCH ON PDP-12
```

```
                        /SIMULATED HARDWARE REGISTERS

0021  0000  CAROUT, 0
0022  0000  OREG,  0                    /O REGISTER
0023  0000        0
0024  0000        0
0025  0000        0
0026  0000        0
0027  0000        0

0030  0000  AREG,  0                    /A REGISTER
0031  0000        0
0032  0000        0
0033  0000        0
0034  0000        0
0035  0000        0

0036  0000  BREG,  0                    /B REGISTER
0037  0000        0
0040  0000        0
0041  0000        0
0042  0000        0
0043  0000        0

0044  0000  MQREG, 0                    /MQ REGISTER
0045  0000        0
0046  0000        0
0047  0000        0
0050  0000        0
0051  0000        0

0052  0000  FFAC,  0                    /FPP ACCUMULATOR
0053  0000        0
0054  0000        0
0055  0000        0
0056  0000        0
0057  0000        0

0060  0000  TREG,  0                    /TEMP REG
0061  0000        0
0062  0000        0
0063  0000        0
0064  0000        0
0065  0000        0
```

```
0066  0000  PFPC,   0                    /FPP PROGRAM COUNTER
0067  0000          0

0070  0000  OPADR,  0                    /OP ADDRESS
0071  0000          0


0072  0000  PAPT,   0                    /ADDRS
0073  0000          0

0074  0000  PBASE,  0                    /BASE REGISTER (P0 ADDRESS)
0075  0000          0

0076  0000  X0ADR,  0                    /FPP INDEX POINTER
0077  0000          0

0100  0000  SAVOM,  0
0101  0000          0
0102  0000          0
0103  0000          0
0104  0000          0

0105  0000  EMEM,   0
0106  0000  MMEM,   0
0107  0000  LMEM,   0
0110  0000  LMEM1,  0
0111  0000  LMEM2,  0
0112  0000  LMEM3,  0
```

```
                0200            *200

0200  0000     X0,     0                       /PROGRAM INDEX REGISTERS
0201  0000     X1,     0
0202  0000     X2,     0
0203  0000     X3,     0
0204  0000     X4,     0
0205  0000     X5,     0
0206  0000     X6,     0
0207  0000     X7,     0

0210  0211     AAPT,   APT
0211  0000     APT,    0                       /APT USED WHEN NO ADDRESS
0212  0000             0
0213  0000             0
0214  0000             0
0215  0000             0                       /IS GIVEN WITH ;RT OR ;RF COMMANDS
0216  0000             0
0217  0000             0
0220  0000             0
0221  0000             0
0222  0000             0
0223  0000             0

0224  0000     PX0,    0                       /INDEX REGISTERS USED WITH ABOVE APT
0225  0000     PX1,    0
0226  0000     PX2,    0
0227  0000     PX3,    0
0230  0000     PX4,    0
0231  0000     PX5,    0
0232  0000     PX6,    0
0233  0000     PX7,    0
```

```
0234  7300  START, CLA CLL
0235  1777' TAD     APTSAV          /GET LAST APT ADDRESS
0236  3353  DCA     OCTWD           /STORE FOR RERUN
0237  3776' DCA     FPPRUN          /CLEAR FPP RUN FF
0240  7604  LAS                     /GET SWITCH REGISTER
0241  7106  CLL RTL
0242  7710  SPA CLA                 /SR02 = 1 ?
0243  5775' JMP     STFPP+2         /YES = RESTART PROGRAM
0244  4774' JMS     TYP             /TYPE END TRACE
0245  2432  EOP
0246  0000  0
0247  3776' BEGIN, DCA     FPPRUN          /CLEAR PROGRAM RUN FF
0250  4773' JMS     TITLE           /TYPE PROGRAM TITLE
0251  4772' DOTEX, JMS     ASTER           /TYPE AN ASTERISK (*)
0252  4265  JMS     WORD            /ALLOW INPUT FROM TTY
0253  5252  JMP     .-1             /PROGRAM SHOULD NOT RETURN HERE


0254  0260  KEYCK, .+4                  /GET TTY INPUT BETWEEN TIME STATES
0255  7300  CLA CLL                    /AND AFTER ERROR TYPEOUT
0256  1254  TAD     KEYCK           /GET RETURN ADDRESS
0257  3354  DCA     RETURN          /SAVE FOR EXIT TO TRACE
0260  6032  KCC
0261  4772' JMS     ASTER           /TYPE AN ASTERISK (*)
0262  4265  JMS     WORD            /ALLOW TTY INPUT
0263  5654  JMP I   KEYCK           /RETURN TO TRACE

0264  5665  EXITWD, JMP I   WORD          /LINK WITH THIS MEMORY PAGE
```

```
                        /RECEIVE INPUT FROM TTY, ONE WORD AT A TIME,
                        /A CARRIAGE RETURN OR SPACE WILL EXIT BACK TO
                        /THE CALLING ROUTINE,
                        /ALT MODE DOES A SPECIAL EXIT

0265  0000  WORD,   0
0266  7300          CLA CLL
0267  3353          DCA     OCTWD           /CLEAR LINKING WORDS
0270  3352          DCA     ASCWD
0271  6031  READ,   KSF                     /WAIT FOR TTY
0272  5271          JMP     .-1
0273  6036          KRB                     /GET CHARACTER FROM TTY
0274  3351          DCA     ASCCH           /SAVE CHARACTER
0275  1351          TAD     ASCCH           /ECHO CHARACTER
0276  6046          TLS
0277  6041          TSF
0300  5277          JMP     .-1
0301  6042          TCF
0302  1371          TAD     (-215           /CHECK FOR SPECIAL CHARACTERS
0303  7450          SNA                     /RETURN ?
0304  5770'         JMP     RET
0305  1367          TAD     (-23            /SPACE ?
0306  7450          SNA
0307  5665          JMP I   WORD
0310  1366          TAD     (-16            /PERIOD ?
0311  7450          SNA
0312  5765'         JMP     DOT
0313  1364          TAD     (-121           /RUBOUT ?
0314  7450          SNA
0315  5763'         JMP     RUBOUT
0316  1362          TAD     (2              /ALT MODE ?
0317  7650          SNA CLA
0320  5754          JMP I   RETURN          /RETURN TO SIMULATION
0321  1351          TAD     ASCCH           /GET ASCII CHARACTER
0322  1361          TAD     (-240
0323  7710          SPA CLA                 /CNTRL KEY ?
0324  5760'         JMP     TREXIT          /YES = EXIT TO MONITOR
```

```
0325  1351          TAD     ASCCH           /GET CHARACTER
0326  0357          AND     (77             /STRIP TO 6 BITS
0327  3351          DCA     ASCCH           /SAVE 6 BIT ASCII CHARACTER
0330  1352          TAD     ASCWD           /GET ASCII WORD
0331  0357          AND     (77             /SAVE LAST CHARACTER
0332  7106          RTL CLL                 /MOVE TO LEFT HALF OF WORD
0333  7006          RTL
0334  7006          RTL
0335  1351          TAD     ASCCH           /INSERT NEW CHARACTER
0336  3352          DCA     ASCWD           /SAVE PACKED ASCII WORD
0337  1351          TAD     ASCCH           /GET ASCII CHARACTER
0340  0356          AND     (7              /EXTRACT OCTAL DIGIT
0341  3351          DCA     ASCCH           /SAVE OCTAL DIGIT
0342  1353          TAD     OCTWD           /GET OCTAL WORD
0343  0355          AND     (777            /SAVE LAST 3 DIGITS
0344  7104          RAL CLL                 /MOVE 1 DIGIT LEFT
0345  7006          RTL
0346  1351          TAD     ASCCH           /INSERT NEW DIGIT
0347  3353          DCA     OCTWD           /SAVE OCTAL WORD
0350  5271          JMP     READ            /GET NEXT CHARACTER

0351  0000  ASCCH,  0
0352  0000  ASCWD,  0
0353  0000  OCTWD,  0
0354  1002  RETURN, STFPP+2                /SPECIAL RETURN ADDRESS



0355  0777
0356  0007
0357  0077
0360  5462
0361  7540
0362  0002
0363  7125
0364  7657
0365  0413
0366  7762
0367  7755
0370  0404
0371  7563
0372  7362
0373  7367
0374  6703
0375  1002
0376  1043
0377  1042
      0400          PAGE
```

```
                        /SET ENTER SINGLE STATE ADDRESS

0400  4777'  SA,    JMS     WORD            /ALLOW TTY INPUT
0401  1776'          TAD     OCTWD           /GET ADDRESS
0402  3775'          DCA     ENTSTP          /SET ENTER ADDRESS LOCATION
0403  5774'          JMP     DOTEX           /EXIT

0404  1373    RET,    TAD     (212            /INPUT FROM TTY WAS A CARRAIGE
0405  6046            TLS                     /RETURN,  TYPE A LINE FEED
0406  6041            TSF
0407  5206            JMP     .-1
0410  6042            TCF
0411  7200            CLA
0412  5772'           JMP     EXITWD          /EXIT WORD ROUTINE

0413  4777'  DOT,    JMS     WORD            /SET TTY RETURN FOR COMMAND

                        /BRANCH TO COMMAND ROUTINE

0414  1371    DOTCK,  TAD     (BRANCH         /ADDRESS OF CHARACTER TABLE
0415  3010            DCA     10
0416  1370            TAD     (CNTRLS-1       /ADDRESS OF ROUTINE TABLE
0417  3011            DCA     11
0420  1411            TAD I   11              /GET ROUTINE ADDRESS
0421  3000            DCA     T1              /SAVE ADDRESS
0422  1410            TAD I   10              /GET CHARACTER SET
0423  7450            SNA                     /END OF CHARACTER TABLE ?
0424  5232            JMP     DOTERR          /YES = INPUT ERROR
0425  7041            CIA                     /NO
0426  1767'           TAD     ASCWD           /COMPARE WITH TTY ASCII INPUT
0427  7640            SZA CLA                 /COMPARE ?
0430  5220            JMP     .-10            /NO = TRY AGAIN
0431  5400            JMP I   T1              /YES = GO TO ROUTINE

                        /COMMAND INPUT ERROR

0432  4766'  DOTERR, JMS     TYP             /TYPE ??
0433  3004            QMK
0434  0000            0
0435  5765'           JMP     BEGIN+1         /TYPE * AND RESTART
```

```
0436  6556    EX,     FPRST                   /GET FPP STATUS
0437  7010            RAR
0440  7620            SNL CLA                 /IS FPP RUNNING ?
0441  5246            JMP     .+5             /NO
0442  6554            FPHLT                   /CPU FORCE EXIT
0443  6561            FMAINT                  /ADVANCE TIMING UNTILL
0444  6557            FPIST
0445  5243            JMP     .-2             /EXIT IS COMPLETED
0446  6552            FPICL                   /CLEAR FPP-12
0447  7300            CLA CLL
0450  4764'           JMS     SETRET          /SETUP RETURN
0451  5774'           JMP     DOTEX           /EXIT ROUTINE

                      /SET EXIT ADDRESS

0452  4777'   EA,     JMS     WORD            /ALLOW TTY INPUT
0453  1776'           TAD     OCTWD           /GET ADDRESS
0454  3763'           DCA     EXADDR          /SAVE IN COMPARE WORD
0455  5774'           JMP     DOTEX           /EXIT ROUTINE


                      /LOAD OPERAND TABLE

0456  1362    LOOP,   TAD     (OPERND-1       /GET TABLE ADDRESS
0457  3011            DCA     11
0460  4777'           JMS     WORD            /ALLOW TTY INPUT
0461  1776'           TAD     OCTWD           /GET OCTAL WORD
0462  6211            CDF     10              /STORE IN FIELD 1
0463  3411            DCA I   11
0464  6201            CDF     00              /RESTORE DATA FIELD
0465  5260            JMP     .-3             /GET NEXT WORD

                      /LOAD INDEX REGISTER (ENTERED FROM .AS ROUTINE)
0466  1776'   LODX,   TAD     OCTWD           /GET OCTAL WORD
0467  0361            AND     (7              /EXTRACT INDEX REGISTER
0470  1360            TAD     (PX0            /ADD ADDRESS OF INDEX REG 0
0471  3000            DCA     T1              /SAVE INDEX ADDRESS
0472  4777'           JMS     WORD            /ALLOW TTY INPUT
0473  1776'           TAD     OCTWD           /GET OCTAL ARGUMENT
0474  3400            DCA I   T1              /SET INDEX REGISTER
0475  5757'           JMP     AS+4            /GO BACK TO AS ROUTINE
```

```
                        /SET RESET SINGLE STATE ADDRESS

0476  4777'  RA,   JMS     WORD            /ALLOW TTY INPUT
0477  1776'         TAD     OCTWD           /GET OCTAL WORD
0500  3756'         DCA     EXSTP           /SET EXIT COMPARE WORD
0501  5774'         JMP     DOTEX           /EXIT

                        /RUN TRACE MODE

0502  3755'  RT,   DCA     FSTEPM          /RESET EPM START FLAG
0503  3754'         DCA     PSTEPM          /RESET EPM START FLAG
0504  4753'         JMS     STFPP           /SETUP TO START FPP
0505  5752'         JMP     INIT            /START TRACE AND FPP IN INIATE


                        /RUN FAST MODE

0506  3755'  RF,   DCA     FSTEPM          /RESET EPM START FLAG
0507  3754'         DCA     PSTEPM          /RESET EPM START FLAG
0510  4753'         JMS     STFPP           /SETUP TO START FPP
0511  7200          CLA
0512  1755'         TAD     FSTEPM          /GET EPM START FLAG
0513  6567          LSHFT                   /START IN EPM MODE IF FLAG IS SET
0514  1751'         TAD     AAPT            /GET ADDRESS OF APT
0515  6555          FPST                    /FPP-12 START ERROR
0516  7402          HLT
0517  7200  RFA,    CLA
0520  1300          TAD     (-20            /GET TIME CONSTANT
0521  3000          DCA     T1              /SET TIMER
0522  3001          DCA     T2
0523  1342          TAD     ACBIT           /GET CONTENTS OF BIT PATTERN
0524  7040          CMA                     /COMPLIMENT
0525  3342          DCA     ACBIT           /RESTORE
0526  1342          TAD     ACBIT           /LOAD BIT PATTERN
0527  6031  RFB,    KSF                     /CHECK KEYBOARD FLAG
0530  7410          SKP
0531  4747'         JMS     KEYCK           /GET TTY INPUT
0532  6557          FPIST                   /IS FPP-12 FINISHED ?
0533  7410          SKP                     /NO
0534  5746'         JMP     START           /YES - EXIT
0535  2001          ISZ     T2              /TIMEOUT BEFORE COMPLIMENTING AC
0536  5327          JMP     RFB             /AC FLASHES ON AND OFF WHILE WAITING
0537  2000          ISZ     T1
0540  5327          JMP     RFB
0541  5317          JMP     RFA
```

```
0542  0000    ACBIT, 0

0543  3745/   CL,    DCA     STEPSW          /CLEAR STEP SWITCH
0544  5774/          JMP     DOTEX           /EXIT

0545  7732
0546  0234
0547  0254
0550  7760
0551  0210
0552  1046
0553  1000
0554  1044
0555  1045
0556  7735
0557  0604
0560  0224
0561  0007
0562  4077
0563  7733
0564  7354
0565  0250
0566  6703
0567  0352
0570  7336
0571  6341
0572  0264
0573  0212
0574  0251
0575  7734
0576  0353
0577  0265
      0600          PAGE
```

```
                        /ASSEMBLE AN FPP-12 PROGRAM

0600  1377    AS,     TAD     (TEST           /GET BUFFER ADDRESS
0601  3262            DCA     ASPNTR          /SET POINTER
0602  1376            TAD     (CDF 10         /ASSEMBLE IN FIELD 1
0603  3233            DCA     ASFLD
0604  1775'           TAD     ASCCH           /GET ASCII CHARACTER
0605  1374            TAD     (-215           /SUBTRACT RETURN
0606  7640            SZA CLA                 /WAS INPUT TERMINATED BY A RETURN
0607  5216            JMP     ASGET           /NO - GET NEXT WORD
0610  4773'           JMS     ASC             /CONVERT ADDRESS TO ASCII
0611  0662            ASPNTR
0612  2710            ASADDR
0613  4772'           JMS     TYPNCR          /TYPE ADDRESS
0614  2710            ASADDR
0615  0000            0
0616  4771'   ASGET,  JMS     WORD            /ALLOW TTY INPUT
0617  1770'           TAD     ASCWD           /GET ASCII WORD
0620  0367            AND     (7700           /EXTRACT LEFT CHARACTER
0621  1367            TAD     (-0100          /SUBTRACT "A"
0622  7450            SNA                     /INPUT = AC ?
0623  5250            JMP     LDAC            /YES - LOAD FAC IN APT
0624  1366            TAD     (-2700          /SUBTRACT "X"
0625  7450            SNA                     /INPUT = X ?
0626  5765'           JMP     LODX            /YES - SET INDEX REGISTER
0627  1364            TAD     (-2200          /SUBTRACE *
0630  7650            SNA CLA                 /NEW LOCATION ?
0631  5277            JMP     ASLOC           /YES - SET LOCATION COUNTER
0632  1763'           TAD     OCTWD           /NO - INPUT IS DATA
0633  6211    ASFLD,  CDF     10              /CHANGE DATA FIELD
0634  3662            DCA I   ASPNTR          /STORE DATA
0635  6201            CDF     00              /RESTORE DATA FIELD
0636  2262            ISZ     ASPNTR          /INCREMENT POINTER
0637  5204            JMP     AS+4            /POINTER IS OK
0640  1233            TAD     ASFLD           /POINTER OVERFLOWED FIELD
0641  1362            TAD     (10             /INCREMENT FIELD POINTER
0642  0361            AND     (70             /EXTRACT FIELD BITS
0643  7450            SNA                     /OVERFLOW TO FIELD 0 ?
0644  5760'           JMP     DOTERR          /YES - ERROR
0645  1357            TAD     (CDF            /OK - MAKE CDF INSTRUCTION
0646  3233            DCA     ASFLD           /MODIFY PROGRAM
0647  5204            JMP     AS+4            /YES - TYPE POINTER
```

```
                        /LOAD FAC IN APT (ENTERED FROM AS COMMAND)

0650  1356    LDAC,   TAD     (APT+4          /ADDRESS OF FAC
0651  3011            DCA     11
0652  1355            TAD     (-3             /WORD COUNT
0653  3000            DCA     T1
0654  4771'           JMS     WORD            /ALLOW TTY INPUT
0655  1763'           TAD     OCTWD           /GET OCTAL WORD
0656  3411            DCA I   11              /STORE IN AC
0657  2000            ISZ     T1
0660  5254            JMP     .-4             /GET NEXT WORD
0661  5204            JMP     AS+4            /GO BACK TO AS ROUTINE

0662  0000    ASPNTR, 0

                        /SET SPECIAL SHIFT

0663  4771'   SH,     JMS     WORD            /ALLOW INPUT FROM TTY
0664  1770'           TAD     ASCWD           /GET ASCII WORD
0665  7640            SZA CLA                 /WERE ANY CHARACTERS INPUT ?
0666  7240            STA                     /YES = AC = 7777
0667  3322            DCA     SHFLAG          /SET OR RESET SHIFT FLAG
0670  1763'           TAD     OCTWD           /GET OCTAL INPUT
0671  3323            DCA     SHFCNT          /SET SPECIAL SHIFT COUNT
0672  5754'           JMP     DOTEX           /EXIT


                        /SET THE COMMAND REGISTER

0673  4771'   CM,     JMS     WORD            /ALLOW TTY INPUT
0674  1763'           TAD     OCTWD           /GET OCTAL WORD
0675  3002            DCA     COMREG          /SET COMMAND REGISTER
0676  5754'           JMP     DOTEX           /EXIT

                        /CHANGE ASSEMBLY LOCATOR

0677  1763'   ASLOC,  TAD     OCTWD           /GET FIELD DESIGNATOR
0700  0353            AND     (7              /EXTRACT LAST DIGIT
0701  7104            CLL RAL                 /MOVE TO BITS 6-8
0702  7006            RTL
0703  1357            TAD     (CDF            /CREATE CDF INSTRUCTION
0704  3233            DCA     ASFLD           /CHANGE AS ROUTINE
0705  4771'           JMS     WORD            /ALLOW TTY INPUT
0706  1763'           TAD     OCTWD           /GET NEW ADDRESS
0707  3262            DCA     ASPNTR          /SET POINTER
0710  5204            JMP     AS+4            /RETURN TO AS ROUTINE


0711  0000    MFLD,   0                       /THESE 2 CONSTANTS ARE USED BY
0712  0000    SAVMEM, 0                       /ANOTHER ROUTINE ( CMEM1 )
```

```
0713  0000  STEPM,  0                        /START IN EPM MODE
0714  7330          CLA STL RAR              /AC = 4000
0715  3752'         DCA     FSTEPM           /STORE IN START WORD
0716  7201          CLA IAC                  /AC = 0001
0717  3751'         DCA     PSTEPM           /STORE IN PROGRAM START WORD
0720  4771'         JMS     WORD             /ALLOW TTY INPUT FOR NEXT ARGUMENT
0721  5713          JMP I   STEPM            /RETURN

                    /FLIP FLOPS - TEMPORARY LOCATIONS & CONSTANTS

0722  0000  SHFLAG, 0                        /SPECIAL SHIFT FLAG
0723  0000  SHFCNT, 0                        /SPECIAL SHIFT COUNT
0724  0000  SHREG,  0                        /FPP SHIFT COUNTER
0725  0000  EXITSW, 0                        /EXIT SWITCH
0726  0000  FLAG1,  0                        /GENERAL PURPOSE FLIP FLOPS
0727  0000  FLAG2,  0
0730  0000  FLAG3,  0
0731  0000  CSTATE, 0                        /PROGRAM STATE GENERATOR
0732  0000  EXWD,   0
0733  0000  OVFL,   0                        /OVERFLOW
0734  0000  IGNFL,  0                        /IGNORE UNDERFLOW FLIP FLOP
0735  0000  MT1,    0
0736  0000  MT2,    0
0737  0000  MDFLAG, 0


0751  1044
0752  1045
0753  0007
0754  0251
0755  7775
0756  0215
0757  6201
0760  0432
0761  0070
0762  0010
0763  0353
0764  5600
0765  0466
0766  5100
0767  7700
0770  0352
0771  0265
0772  7143
0773  7154
0774  7563
0775  0351
0776  6211
0777  4000
      1000          PAGE
```

```
                        /SETUP TO START THE FPP

1000  0503    STFPP,  RT+1                    /ENTERED WITH A JMS
1001  4777'            JMS     WORD            /ALLOW TTY INPUT
1002  6552            FPICL                   /CLEAR THE FPP-12
1003  3003            DCA     EPM             /CLEAR EPM CONTROL REGISTER
1004  3776'           DCA     CSTATE          /CLEAR PROGRAM STATE GENERATOR
1005  1002            TAD     COMREG          /GET THE COMMAND REGISTER
1006  6553            FPCOM                   /SET THE FPP-12 COMMAND REGISTER
1007  0375            AND     (7              /EXTRACT FIELD BITS
1010  3072            DCA     PAPT            /SET PROGRAM ADRS FIELD BITS
1011  1374            TAD     (TEST           /GET ADDRESS OF FPP INSTRUCTIONS
1012  3773'           DCA     APT+1           /SET FPC IN PROGRAM APT
1013  1372            TAD     (PX0            /GET INDEX POINTER
1014  3771'           DCA     APT+2           /SET INDEX POINTER IN APT
1015  1370            TAD     (OPERND         /GET ADDRESS OF OPERAND TABLE
1016  3767'           DCA     APT+3           /SET BASE ADDRESS IN APT
1017  1366            TAD     (0101           /GET FIELD BITS
1020  3765'           DCA     APT             /SET APT FIELD BITS
1021  3764'           DCA     APT+4           /CLEAR OP ADDRESS IN APT
1022  1763'           TAD     ASCWD           /GET ASCII INPUT
1023  1362            TAD     (-5             /SUBTRACT ASCII "E"
1024  7650            SNA CLA                 /START IN EPM MODE ?
1025  4761'           JMS     STEPM           /YES - SET EPM MODE
1026  1760'           TAD     OCTWD           /GET TTY INPUT
1027  7450            SNA                     /IS IT 0000 ?
1030  1365            TAD     (APT            /YES - GET ADDR OF PROGRAM APT
1031  3757'           DCA     AAPT            /SET APT ADDRESS
1032  3756'           DCA     STEPSW          /CLEAR SINGLE STATE SWITCH
1033  1757'           TAD     AAPT
1034  3242            DCA     APTSAV          /SAVE APT ADDRESS
1035  7240            STA                     /SET PROGRAM RUN FF
1036  3243            DCA     FPPRUN
1037  3004            DCA     PSTAT           /CLEAR STATUS WORD
1040  4755'           JMS     TITLE           /SYNC THE 0 REGISTERS
1041  5600            JMP I   STFPP           /EXIT


1042  0000    APTSAV, 0
1043  0000    FPPRUN, 0
1044  0000    PSTEPM, 0
1045  0000    FSTEPM, 0
```

```
                        / FPP FLOWS

1046  4754'   INIT,   JMS     MSTATE          /CHECK  STATES
1047  7610            SKP CLA                 /TIME STATE ERROR
1050  5254            JMP     .+4             /TIME STATE GENERATER IS OK
1051  1353            TAD     (INIT+1         /GET PC FOR ERROR TYPEOUT
1052  4752'           JMS     ERR
1053  0000            ERROR   0000            / STATE ERROR CODE
1054  1757'           TAD     AAPT            /GET ADDRESS OF APT
1055  3073            DCA     PAPT+1          /SET APT COMPARE ADDRESS
1056  1244            TAD     PSTEPM          /GET PROGRAM START WORD
1057  3003            DCA     EPM             /SET PROGRAM  EPM STATUS
1060  1245            TAD     FSTEPM          /GET FPP START WORD
1061  6567            LSHFT                   /LOAD FPP STATUS WORD
1062  4563            ENTER                   /TYPE ENTER INITIATE
1063  6561    ENINIT, FMAINT                  /SET FPP MAINT MODE
1064  7200            CLA
1065  1757'           TAD     AAPT            /GET ADDRESS OF APT

1066  6555    INIT0,  FPST                    /START FPP
1067  7610            SKP CLA                 /FPP DID NOT START
1070  5274            JMP     .+4
1071  4751'           JMS     TYP             /FPP START ERROR
1072  2021            STRTER                  /TYPE ERROR MESSAGE
1073  4750'           JMS     KEYCK           /WAIT FOR KEYBOARD INPUT
1074  4561            GETAPT                  /OUTBRK USING ADRS
1075  3030            DCA     AMSW            /MB TO AMSW
1076  1030            TAD     AMSW            /AND FIELD BITS
1077  0375            AND     (7
1100  3066            DCA     PFPC            /FIELD BITS OF FPC
1101  1030            TAD     AMSW            /YES-GET FIELD BITS
1102  7012            RTR
1103  7010            RAR
1104  0375            AND     (7
1105  3076            DCA     X0ADR           /FIELD BITS OF INDEX POINTER
1106  1030            TAD     AMSW            /GET FIELD BITS
1107  7012            RTR
1110  7012            RTR
1111  7012            RTR
1112  0375            AND     (7
1113  3074            DCA     PBASE           /FIELD BITS OF P0 ADDR
1114  4555            INCAPT                  /INCREMENT ADRS
1115  4575            CAPT                    /CHECK APT
1116  3747'           DCA     CK0             /CLEAR CHECK 0 FLAG
1117  3746'           DCA     CKOP            /CLEAR CHECK OP ADDR FLAG
```

```
1120  4562  INIT1,  FSTEP                 /STEP TO STATE 1
1121  4561           GETAPT                /OUTBRK USING ADRS
1122  3031           DCA     ALSW          /MB TO ALSW
1123  4543           LOADOI  AREG          /A TO 0
1124  0030
1125  4555           INCAPT                /INC ADRS
1126  4575           CAPT                  /CHECK APT ADDRESS
1127  4542           LOADOPI OREG          /0 TO OP ADDR
1130  0022
1131  1071           TAD     OPADR+1       /OP ADDR TO FPC
1132  3067           DCA     PFPC+1
1133  5334           JMP     INIT2         /GO TO STATE 2


1134  4562  INIT2,   FSTEP                 /STEP TO  STATE 2
1135  4561           GETAPT                /OUTBRK USING ADRS
1136  3077           DCA     X0ADR+1       /MB TO X0 ADDR
1137  4555           INCAPT                /INC ADRS
1140  4575           CAPT                  /CHECK APT ADDRESS
1141  5745           JMP     INIT3         /GO TO STATE 3

1145  1200
1146  7737
1147  7736
1150  0254
1151  6703
1152  6707
1153  1047
1154  5516
1155  7367
1156  7732
1157  0210
1160  0353
1161  0713
1162  7773
1163  0352
1164  0215
1165  0211
1166  0101
1167  0214
1170  4100
1171  0213
1172  0224
1173  0212
1174  4000
1175  0007
1176  0731
1177  0265
      1200           PAGE
```

```
1200  4562    INIT3,  FSTEP                   /STEP TO STATE 3
1201  4561            GETAPT                  /OUTBRK USING ADRS
1202  3075            DCA     PBASE+1         /MB TO P0 ADDR
1203  4555            INCAPT                  /INC ADRS
1204  4555            INCAPT                  /INC ADRS
1205  4575            CAPT                    /CHECK APT ADDRESS

1206  4562    INIT4,  FSTEP                   /STEP TO STATE 4
1207  4561            GETAPT                  /OUTBRK USING ADRS
1210  3052            DCA     ACEXP           /SAVE FAC EXPONENT
1211  4555            INCAPT                  /INC ADRS
1212  4575            CAPT                    /CHECK APT

1213  4562    INIT5,  FSTEP                   /STEP TO STATE 5
1214  4561            GETAPT                  /OUTBRK USING ADRS
1215  3030            DCA     AMSW            /MB TO AMSW
1216  4555            INCAPT                  /INC ADRS
1217  4530            SEPM                    /EPM?
1220  5247            JMP     INIT6           /NO = GO TO STATE 6
1221  4555            INCAPT                  /YES = INC ADRS
1222  4575            CAPT                    /CHECK APT ADDRESS
1223  4777/           JMS     SST             /1 TO SPECIAL STATE
1224  4527            SEPMEN                  /1 TO EPM ENABLE

1225  4562    INIT50, FSTEP                   /STEP TO SPECIAL STATE 0
1226  4561            GETAPT                  /OUTBRK USING ADRS
1227  3032            DCA     ALSW1           /MB TO ALS 1
1230  4555            INCAPT                  /INC ADRS
1231  4575            CAPT                    /CHECK APT ADDRESS

1232  4562    INIT51, FSTEP                   /STEP TO SPECIAL STATE 1
1233  4561            GETAPT                  /OUTBRK USING ADRS
1234  3033            DCA     ALSW2           /MB TO ALS 2
1235  4555            INCAPT                  /INC ADRS
1236  4575            CAPT                    /CHECK APT ADDRESS
```

```
1237  4562  INIT52, FSTEP                 /STEP TO SPECIAL STATE 2
1240  4561          GETAPT                /OUTBRK USING ADRS
1241  3034          DCA     ALSW3         /MB TO ALS3
1242  4565          DECAPT                /DEC ADRS
1243  4565          DECAPT                /DEC ADRS
1244  4565          DECAPT                /DEC ADRS
1245  4575          CAPT                  /CHECK APT ADDRESS
1246  4776          JMS     RST           /Ø TO SPECIAL STATE

1247  4562  INIT6,  FSTEP                 /STEP TO STATE 6
1250  4575          CAPT                  /CHECK APT ADDRESS
1251  4561          GETAPT                /OUTBRK USING ADRS
1252  3031          DCA     ALSW          /MB TO ALSW
1253  4543          LOADOI  AREG          /A TO O
1254  0030
1255  4546          LOADACI OREG          /O TO FAC FRACTION
1256  0022

1257  4540  INENDØ, MOVEX                 /SET PROGRAM INDEX REGISTERS
1260  5261          JMP     FETCH         /RETURN TO CONTROL PROGRAM
```

```
1261  4563  FETCH,  ENTER                 /TYPE ENTER FETCH
1262  4574           CLRA                 /CLEAR A REG
1263  4572           CLRB                 /CLEAR B REG
1264  6567           LSHFT                /ZERO FPP SHIFT REGISTER
1265  3775/          DCA     SHREG        /ZERO PROGRAM SHIFT REGISTER
1266  3007           DCA     CARYIN       /CLEAR CARRY IN FF
1267  3774/          DCA     MDFLAG       /CLEAR MULT OR DIV FLAG
1270  4535           REPMEN               /0 TO EPM ENABLE

1271  4562  FECH0,   FSTEP               /STEP TO FETCH STATE 0
1272  4570           CLRO
1273  4571           CLRMQ                /CLEAR MQ
1274  4557           GETPC                /OUTBRK USING FPC
1275  3005           DCA     PIR          /STORE IN PROGRAM INST REG
1276  4552           INCPC                /INCREMENT PROGRAM FPC
1277  1115           TAD     [23          /SET A REG = 23
1300  4522           STORA
1301  1052           TAD     ACEXP        /SET B REG = FAC EXPONENT
1302  4521           STORB                /STORE LSW AND EXTEND SIGN
1303  4577           AMBO                 /A MINUS B TO O REG
1304  1005           TAD     PIR          /GET FPP INSTRUCTION
1305  0373           AND     (600         /EXTRACT BITS 3 AND 4
1306  7450           SNA                  /SPECIAL INSTRUCTION?
1307  5772/          JMP     FEND         /YES-END OF FETCH
1310  1371           TAD     (-400
1311  7640           SZA CLA              /DOUBLE WORD INS?
1312  5317           JMP     FECH1        /NO-GO TO STATE 1
1313  4770/          JMS     SETST3       /SET STATE 3
1314  7240           STA
1315  3767/          DCA     FLAG1        /SET FLAG1 FOR DOUBLE WORD INST.
1316  5766/          JMP     FECH34

1317  4562  FECH1,   FSTEP
1320  1005           TAD     PIR          /GET INSTRUCTION
1321  7106           CLL RTL              /SAVE BIT PIR3 IN THE LINK
1322  7106           CLL RTL
1323  7200           CLA
1324  1005           TAD     PIR          /GET INSTRUCTION
1325  0365           AND     (177         /GET BITS 9-11
1326  7430           SZL                  /BIT 3 = 1?
1327  0364           AND     (7           /YES-EXTRACT BITS 9-11
1330  4522           STORA                /PIR 5-11 OR 9-11 TO ALSW
1331  4545           LOADBI  AREG         /MOVE A TO B FOR ADD (A+A TO O)
1332  0030
1333  4576           APBO                 /A PLUS B TO O REG
1334  4545           LOADBI  OREG         /O TO B
1335  0022
1336  4576           APBO                 /A PLUS B TO OREG
```

```
1337  4562    FECH2,  FSTEP                   /STEP TO STATE 2
1340  4547            LOADA!  OREG            /0 TO A
1341  0022
1342  4545            LOADB!  PBASE           /P0 ADDR TO B
1343  0074
1344  4576            APB0                    /A PLUS B TO 0
1345  4542            LOADOP! OREG            /0 TO OP ADDR
1346  0022
1347  1005            TAD     PIR             /GET INSTRUCTION
1350  0363            AND     (400            /EXTRACT BIT 3
1351  7640            SZA CLA                 /BIT 3 = 1?
1352  5762/           JMP     FECH24          /YES-INC OP ADDR
1353  6556            FPRST                   /GET FPP STATUS
1354  7710            SPA CLA                 /DOUBLE PRECISION?
1355  5762/           JMP     FECH24          /YES-INC OP ADDR

                      NPAGE

1356  5761            JMP I   (,+20007600
1361  1400
1362  1411
1363  0400
1364  0007
1365  0177
1366  1427
1367  0726
1370  6250
1371  7400
1372  1621
1373  0600
1374  0737
1375  0724
1376  7525
1377  7516
      1400            PAGE
```

```
1400  1005          TAD     PIR             /GET INSTRUCTION
1401  0377          AND     (7000           /EXTRACT OP CODE
1402  1376          TAD     (-3000
1403  7450          SNA                     /FDIV?
1404  5211          JMP     FECH24          /YES-INC OP ADDR
1405  1377          TAD     (-1000
1406  7450          SNA                     /FMUL?
1407  5211          JMP     FECH24          /YES-INC OP ADDR
1410  1376          TAD     (-3000
1411  7640  FECH24, SZA CLA                /FMULM?
1412  5214          JMP     .+2             /NO-DO NOT INC OP ADDR
1413  4554          INCOP                   /INC OP ADDR
1414  1005          TAD     PIR             /GET INSTRUCTION
1415  0375          AND     (400            /EXTRACT BIT 3
1416  7650          SNA CLA                 /BIT 3 = 0?
1417  5774'         JMP     FEND            /END OF FETCH
1420  5221          JMP     FECH3           /NO - GO TO STATE 3

1421  4562  FECH3,  FSTEP                   /STEP TO STATE 3
1422  4560          GETOP                   /OUTBRK USING OP ADDR
1423  0373          AND     (7              /EXTRACT BITS 9-11
1424  3044          DCA     MQMSW           /MB9-11 TO MQMSW
1425  4554          INCOP
1426  3772'         DCA     FLAG1           /CLEAR FLAG 1
1427  1005  FECH34, TAD     PIR             /GET INSTRUCTION
1430  0371          AND     (70             /EXTRACT BITS 6-8
1431  7112          CLL RTR
1432  7010          RAR
1433  4522          STORA                   /FIR 6-8 TO ALSW - 0 TO AMSW
1434  4545          LOADB;  X0ADR           /X0 ADDR TO B
1435  0076

1436  4562  FECH4,  FSTEP                   /STEP TO STATE 4
1437  1772'         TAD     FLAG1           /GET DOUBLE WORD FLAG
1440  7640          SZA CLA                 /DOUBLE WORD INSTRUCTION ?
1441  5244          JMP     .+3             /YES - NO OUTBREAK
1442  4560          GETOP                   /OUTBRK USING OP ADDR
1443  3045          DCA     MQLSW           /MB TO MQLSW
1444  3772'         DCA     FLAG1           /CLEAR DOUBLE WORD FLAG
```

```
1445  4576  FECH42, APBO                    /A PLUS B TO O
1446  4542         LOADOP! OREG              /O TO OP ADDR
1447  0022
1450  4570         CLRO                     /0 TO O REG
1451  1005         TAD     PIR              /GET INSTRUCTION
1452  0370         AND     (170             /EXTRACT BITS 5-8
1453  7106         CLL RTL
1454  7006         RTL
1455  7006         RTL
1456  7470         SZL SNA                  /FIR5 = 0 AND FIR 6-8 NE 0?
1457  5262         JMP     .+3
1460  4767!        JMS     SETST5           /YES-SETUP FOR STATE 6
1461  5275         JMP     FECH6            /GO TO MAJOR STATE 6
1462  7010         RAR
1463  7640         SZA CLA                  /FIR 5-8 = 0?
1464  5267         JMP     .+3
1465  4766!        JMS     SETST6           /YES-SETUP FOR MAJOR STATE 7
1466  5324         JMP     FECH7            /GO TO STATE 7

1467  4562  FECH5,  FSTEP                   /STEP TO STATE 5
1470  4550         INCX
1471  1005         TAD     PIR              /GET INSTRUCTION
1472  0371         AND     (70              /EXTRACT BITS 6-8
1473  7650         SNA CLA                  /FIR6-8 = 0?
1474  5265         JMP     FECH5-2          /YES-SETUP FOR STATE 7

1475  4562  FECH6,  FSTEP                   /STEP TO STATE 6
1476  4556         GETX                     /OUTBRK USING OP ADDR
1477  3031         DCA     ALSW             /MB TO ALSW
1500  3030         DCA     AMSW             /0 TO AMSW
1501  4545         LOADB!  AREG             /A TO B FOR ADD (A+A TO O)
1502  0030
1503  4576         APBO                     /A PLUS B TO O
1504  1002         TAD     COMREG           /GET COMMAND REGISTER
1505  7710         SPA CLA                  /FLOATING POINT MODE?
1506  5312         JMP     .+4              /NO
1507  4545         LOADB!  OREG             /O TO B
1510  0022
1511  4576         APBO                     /A PLUS TO O
1512  4530         SEPM                     /IN EPM MODE?
1513  5324         JMP     FECH7            /NO GO TO STATE 7
1514  4765!        JMS     SST              /1 TO SPECIAL STATE

1515  4562  FECH60, FSTEP                   /STEP TO SPECIAL STATE 0
1516  4545         LOADB!  OREG             /O TO B FOR ADD SINCE I CAN'T
1517  0022
1520  4547         LOADA!  OREG             /O TO A
1521  0022
1522  4576         APBO                     /A+B HERE = A+A IN THE FLOWS
1523  4764!        JMS     RST              /0 TO SPECIAL STATE

1524  4562  FECH7,  FSTEP                   /STEP TO STATE 7
1525  4547         LOADA!  MQREG            /MQ TO A
1526  0044
```

```
1527  4545          LOADB!  OREG        /0 TO B
1530  0022
1531  1005          TAD     PIR         /GET INSTRUCTION
1532  0363          AND     (200        /EXTRACT BIT 4
1533  7640          SZA CLA             /FIR4=1
1534  5762'         JMP     FECH72      /YES-GO TO STATE 7-2
1535  4557          GETPC               /OUTBRK USING FPC
1536  3031          DCA     ALSW        /MB TO ALSW
1537  1005          TAD     PIR         /GET INSTRUCTION
1540  0373          AND     (7          /EXTRACT BITS 9-11
1541  3030          DCA     AMSW        /FIR 9-11 TO AMSW
1542  4545          LOADB!  OREG        /0 TO B
1543  0022
1544  4552          INCPC               /INC FPC
1545  5762'         JMP     FECH72

1562  1600
1563  0200
1564  7525
1565  7516
1566  6257
1567  6241
1570  0170
1571  0070
1572  0726
1573  0007
1574  1621
1575  0400
1576  5000
1577  7000
      1600          PAGE
```

```
1600  4576  FECH72, APBO                    /A PLUS B TO O
1601  4542          LOADOP; OREG            /O TO OP ADDR
1602  0022
1603  1002          TAD     COMREG          /GET COMMAND REG
1604  7710          SPA CLA                 /D, P, MODE ?
1605  5221          JMP     FEND            /YES = END OF FETCH
1606  1005          TAD     PIR             /GET INSTRUCTION
1607  0377          AND     (7000           /EXTRACT OP CODE
1610  1376          TAD     (-3000
1611  7450          SNA                     /FDIV?
1612  5220          JMP     FECH73          /YES=INC OP ADDR
1613  1377          TAD     (-1000
1614  7450          SNA                     /FMUL?
1615  5220          JMP     FECH73          /YES=INC OP ADDR
1616  1376          TAD     (-3000
1617  7650          SNA CLA                 /FMULM?
1620  4554  FECH73, INCOP                   /YES=INC OP ADDR
1621  5222  FEND,   JMP     EXEC            /GO TO EXECUTE


1622  7300  EXEC,   CLA CLL                 /EXECUTE SKIP CHAIN
1623  3775' DCA     FLAG1
1624  3774' DCA     FLAG2                   /CLEAR FLAGS AND TEMP LOCATIONS
1625  3773' DCA     FLAG3
1626  3000          DCA     T1
1627  3001          DCA     T2
1630  1005          TAD     PIR             /GET INSTRUCTION REG
1631  0372          AND     (600            /EXTRACT BITS 3-4
1632  7640          SZA CLA
1633  5771' JMP     ARITH                   /ARITHMETIC INSTRUCTIONS
1634  1005  PROCES, TAD     PIR             /PROCESS SPECIAL INSTRUCTIONS
1635  0377          AND     (7000           /GET OP CODE
1636  1370          TAD     (-2000
1637  7420          SNL                     /WHICH SPECIAL FORMAT?
1640  5260          JMP     SPEC2           /SPECIAL FORMAT 2
1641  7650  SPEC1,  SNA CLA                 /JXN?
1642  5767' JMP     JXN                     /YES
1643  4563  TRAPED, ENTER                   /ENTER TRAPPED INSTRUCTIONS
1644  4562  TRAP1,  FSTEP                   /STEP TO STATE 1
1645  4557          GETPC                   /OUTBRK USING FPC
1646  3031          DCA     ALSW            /MB TO ALSW
1647  1005          TAD     PIR             /GET INSTRUCTION REG
1650  0366          AND     (7              /EXTRACT BITS 9-11
1651  3030          DCA     AMSW            /FIR9-11 TO AMSW
1652  4543          LOADO;  AREG            /A TO O
1653  0030
1654  4542          LOADOP; OREG            /O TO OP ADDR
1655  0022
1656  4552          INCPC                   /INC FPC
1657  5765' JMP     EXIT                    /GO TO EXIT
```

```
1660  7300  SPEC2,  CLA CLL              /SPECIAL FORMAT 2 INSTRUCTIONS
1661  1005          TAD     PIR         /GET INSTRUCTIONS
1662  0364          AND     (7770
1663  7450          SNA                 /SPEC FMAT 2 OR 3
1664  5763'         JMP     SPEC3       /SPEC FMAT 3
1665  7104          CLL RAL
1666  7006          RTL
1667  7420          SNL                 /OP CODE 0 OR 1
1670  5315          JMP     SPEC20      /OP CODE 0

1671  1362  SPEC21, TAD     (-1400
1672  7710          SPA CLA             /NOP?
1673  5277          JMP     .+4         /NO
1674  4517          TRACE               /YES-TRACING PROGRAM?
1675  7000          NOP                 /YES-TYPE NOP
1676  5761'         JMP     FETCH       /GO TO FETCH
1677  1005          TAD     PIR         /GET INSTRUCTION
1700  0360          AND     (170        /EXTRACT EXTENSION
1701  7110          CLL RAR
1702  7012          RTR                 /RIGHT JUSTIFY
1703  7040          CMA                 /NEGATE
1704  3000          DCA     T1
1705  1357          TAD     (INS21-1    /GET ADDRESS OF INSTRUCTION TABLE
1706  3001          DCA     T2
1707  2001          ISZ     T2          /FIND INSTRUCTION
1710  2000          ISZ     T1
1711  5307          JMP     .-2
1712  1401          TAD I   T2          /GET INSTRUCTION ADDRESS
1713  3001          DCA     T2
1714  5401          JMP I   T2
```

```
1715  7112    SPEC20, CLL RTR
1716  7012            RTR
1717  7012            RTR
1720  7041            CIA
1721  7001            IAC
1722  7450            SNA
1723  5756/           JMP     FALN            /ALIGN INSTRUCTION
1724  7001            IAC
1725  7450            SNA
1726  5755/           JMP     FATX            /ATX INSTRUCTION
1727  7001            IAC
1730  7450            SNA
1731  5754/           JMP     FXTA            /XTA INSTRUCTION
1732  7001            IAC
1733  7001            IAC
1734  7450            SNA
1735  5753/           JMP     FSTE            /START E
1736  1352            TAD     (3
1737  7450            SNA
1740  5751/           JMP     FLDX            /LDX INSTRUCTION
1741  7001            IAC
1742  7650            SNA CLA
1743  5750/           JMP     ADDX            /ADDX INSTRUCTION
1744  5274            JMP     SPEC21+3        /NOP

1750  5200
1751  4114
1752  0003
1753  5354
1754  4500
1755  4330
1756  4142
1757  7316
1760  0170
1761  1261
1762  6400
1763  2000
1764  7770
1765  2427
1766  0007
1767  4537
1770  6000
1771  2030
1772  0600
1773  0730
1774  0727
1775  0726
1776  5000
1777  7000
      2000            PAGE
```

```
2000  1005    SPEC3,  TAD     PIR             /SPECIAL FORMAT 3
2001  0377            AND     (7              /EXTRACT BITS 9-11
2002  7041            CIA                     /NEGATE
2003  7450            SNA
2004  5776'           JMP     EXIT            /EXIT
2005  7001            IAC
2006  7450            SNA
2007  5775'           JMP     FPAUSE          /PAUSE
2010  7001            IAC
2011  7450            SNA
2012  5774'           JMP     FCLA            /FCLA
2013  7001            IAC
2014  7450            SNA
2015  5773'           JMP     FNEG            /FNEG
2016  7001            IAC
2017  7450            SNA
2020  5772'           JMP     FNORM           /FNORM
2021  7001            IAC
2022  7450            SNA
2023  5771'           JMP     FSTF            /START F
2024  7001            IAC
2025  7450            SNA
2026  5770'           JMP     FSTD            /START D
2027  5767'           JMP     JAC             /JAC


2030  1005    ARITH,  TAD     PIR             /GET INSTRUCTION
2031  0366            AND     (7000           /EXTRACT OP CODE
2032  7106            CLL RTL                 /RIGHT JUSTIFY
2033  7006            RTL
2034  7040            CMA                     /NEGATE+1
2035  3000            DCA     T1              /SAVE MINUS OP CODE
2036  1002            TAD     COMREG          /GET COMMAND REG
2037  7004            RAL                     /D,P, BIT TO LINK
2040  7206            CLA RTL                 /MOVE D,P, BIT TO AC0
2041  7006            RTL                     /GET ADDR OF FLOATING PT,
2042  1365            TAD     (INS0-1         /OR D,P, INSTRUCTION TABLE
2043  3001            DCA     T2
2044  2001            ISZ     T2              /INC ADDR
2045  2000            ISZ     T1              /INC INSTRUCTION
2046  5244            JMP     .-2             /NOT THIS INS--TRY AGAIN
2047  1401            TAD I   T2              /GET ADDR OF INST
2050  3001            DCA     T2
2051  5401            JMP I   T2              /GO TO INSTRUCTION
```

```
2052  4563    DEP,    ENTER                 /TYPE ENTER DEPOSIT
2053  4527            SEPMEN                /1 TO EPM ENABLE
2054  4562    DEP11,  FSTEP                 /STEP TO STATE11
2055  1002            TAD     COMREG        /GET COMMAND REG
2056  7700            SMA CLA               /FIXED PNT NOS',?
2057  4537            NORM                  /CHECK AND NORMALIZE
2060  4547            LOADAI  OREG          /0 TO A
2061  0022
2062  4572            CLRB                  /ZERO TO B
2063  1027            TAD     OEXT
2064  3000            DCA     T1            /SAVE O EXT FOR ROUNDING
2065  1022            TAD     OMSW          /CHECK FOR OVERFLOW?
2066  7700            SMA CLA               /NO = CLEAR FLAG
2067  7040            CMA                   /YES = SET FLAG
2070  3764'           DCA     FLAG1         /STORE FLAG
2071  4576            APBO
2072  1000            TAD     T1            /GET EXT REG FOR ROUNDING
2073  7710            SPA CLA               /CHECK CARRY IN
2074  4553            INCOR                 /INCREMENT O REG
2075  1764'           TAD     FLAG1         /CHECK FOR OVERFLOW
2076  7650            SNA CLA               /OVERFLOW?
2077  5303            JMP     .+4           /NO = BYPASS O CHECK
2100  1022            TAD     OMSW          /YES = CHECK OVERFLOW
2101  7710            SPA CLA
2102  5305            JMP     .+3           /OVERFLOW
2103  4551            INCST                 /NO OVERFLOW
2104  5332            JMP     CKMEM         /BYPASS MAJOR STATE 12
2105  1002            TAD     COMREG        /GET COMMAND REG
2106  7700            SMA CLA               /FIXED PT, NOS',?
2107  5317            JMP     .+10          /NO
2110  1004            TAD     PSTAT         /GET STATUS WORD
2111  0363            AND     (7577         /SAVE OTHER BITS
2112  1362            TAD     (200          /SET FRAC OVERFLOW
2113  3004            DCA     PSTAT
2114  7040            CMA
2115  3761'           DCA     EXITSW        /SET EXIT SWITCH
2116  5760'           JMP     DEPEND        /END OF DEPOSIT
```

```
2117  4562  DEP12,  FSTEP                 /STEP TO STATE 12
2120  1022           TAD     OMSW         /12=1 AND I2=SHIFT
2121  7110           CLL RAR              /0 TO B0
2122  3036           DCA     BMSW
2123  1023           TAD     OLSW
2124  7010           RAR
2125  3037           DCA     BLSW
2126  4543           LOADOI  BREG         /B TO 0
2127  0036
2130  2757'          ISZ     SHREG        /INC SHFT CNTR
2131  7000           NOP
2132  4520  CKMEM,   TOMEM                /RESULTS TO MEMORY?
2133  5336           JMP     .+3
2134  4551           INCST                /NO - INCREMENT MAJOR STATE
2135  5756'          JMP     DEP14        /BYPASS STATE I3
                     .
2136  4562  DEP13,   FSTEP                /STEP TO STATE 13
2137  1023           TAD     OLSW         /OLSW TO MEMORY
2140  3107           DCA     LMEM
2141  4564           DECOP
2142  4530           SEPM                 /EPM MODE?
2143  5756'          JMP     DEP14        /NO=GO TO STATE I4
2144  5755'          JMP     DEPM3

2155  2200
2156  2223
2157  0724
2160  2415
2161  0725
2162  0200
2163  7577
2164  0726
2165  7276
2166  7000
2167  5027
2170  5057
2171  5042
2172  5100
2173  5066
2174  5144
2175  5122
2176  2427
2177  0007
      2200           PAGE
```

```
2200  4777   DEPM3,  JMS     SST         /1 TO SPECIAL STATE
2201  4554           INCOP
2202  4554           INCOP                /INC OP ADDRESS
2203  4554           INCOP
2204  4554           INCOP

2205  4562   DEP130, FSTEP               /STEP TO SPECIAL STATE 0
2206  1026           TAD     OLSW3        /OLSW3 TO MEMORY
2207  3112           DCA     LMEM3
2210  4564           DECOP                /DEC OP ADDRESS

2211  4562   DEP131, FSTEP               /STEP TO SPECIAL STATE 1
2212  1025           TAD     OLSW2        /0 LSW2 TO MEMORY
2213  3111           DCA     LMEM2
2214  4564           DECOP                /DEC OP ADDRESS

2215  4562   DEP132, FSTEP               /STEP TO SPECIAL STATE 2
2216  1024           TAD     OLSW1        /OLSW1 TO MEMORY
2217  3110           DCA     LMEM1
2220  4564           DECOP
2221  4564           DECOP
2222  4776           JMS     RST          /0 TO SPECIAL STATE
```

```
2223  4562     DEP14,  FSTEP                   /STEP TO STATE 14
2224  4520             TOMEM                   /RESULTS TO MEMORY?
2225  7410             SKP                     /YES
2226  5234             JMP     .+6             /NO
2227  1022             TAD     OMSW            /OMSW TO MEMORY
2230  3106             DCA     MMEM
2231  4566             CMEMF                   /CHECK MEMORY FRACTION
2232  4564             DECOP                   /DEC OP ADDR
2233  5236             JMP     .+3             /BYPASS 0 TO FAC
2234  4546             LOADAC; OREG            /0 TO FAC FRACTION
2235  0022
2236  1045             TAD     MQLSW           /MQLSW TO BLSW
2237  4521             STORB                   /SIGN EXTEND TO BMSW
2240  4523             SOEZ                    /0 = 0 ?
2241  7610             SKP CLA                 /NO
2242  7240             STA                     /YES = SET ZERO TO
2243  3775'            DCA     FLAG3           /FAC FF
2244  4535             REPMEN                  /0 TO EPM ENABLE
2245  1774'            TAD     MDFLAG          /GET MULT OR DIV FLAG
2246  7650             SNA CLA                 /MULT OR DIV?
2247  5264             JMP     DEP144          /NO = GO TO STATE I4=4
2250  1052             TAD     ACEXP           /FAC EXP TO ALSW
2251  4522             STORA                   /SIGN EXTEND TO AMSW
2252  1005             TAD     PIR             /GET INSTRUCTION REG
2253  0373             AND     (7000           /EXTRACT OP CODE
2254  1372             TAD     (-3000
2255  7640             SZA CLA                 /DIVIDE?
2256  5261             JMP     .+3             /NO
2257  4577             AMBO                    /YES = A MINUS B TO 0
2260  7410             SKP
2261  4576             APBO                    /A PLUS B TO 0
2262  4545             LOADB;  OREG            /0 TO B
2263  0022
2264  1771'    DEP144, TAD     SHREG           /SHFT CNTR TO ALSW
2265  4522             STORA                   /SIGN EXTEND TO AMSW
2266  1775'            TAD     FLAG3           /ZERO TO FAC FF SET ?
2267  7650             SNA CLA
2270  5273             JMP     ACNEZ           /NO ADD A AND B
2271  4570             CLRO                    /YES = 0 TO 0
2272  5274             JMP     ACNEZ+1         /DO NOT ADD A PLUS B

2273  4576     ACNEZ,  APBO                    /A PLUS B TO 0
2274  1002             TAD     COMREG          /GET COMMAND REG
2275  7700             SMA CLA                 /FIXED PNT, NOS,?
2276  5301             JMP     DEP15           /NO
2277  3770'            DCA     EXITSW          /CLEAR EXIT SWITCH
2300  5767'            JMP     DEPEND          /END OF DEPOSIT
```

```
2301  4562    DEP15,  FSTEP                   /STEP TO STATE 15
2302  4520            TOMEM                   /RESULTS TO MEMORY?
2303  7410            SKP                     /YES
2304  5311            JMP     .+5             /NO - BYPASS INBRK
2305  1023            TAD     OLSW            /OLSW TO MEM
2306  3105            DCA     EMEM
2307  4567            CMEME                   /CHECK MEMORY EXPONENT
2310  5313            JMP     .+3
2311  1023            TAD     OLSW            /OLSW TO FAC EXP
2312  3052            DCA     ACEXP
2313  4545            LOADBI  OREG            /0 TO B
2314  0022
2315  7330            CLA STL RAR
2316  3031            DCA     ALSW            /4000 TO A
2317  3030            DCA     AMSW
2320  1022            TAD     OMSW
2321  7700            SMA CLA                 /IS OLT 0
2322  5766'           JMP     OPOS            /NO
2323  4576            APBO                    /YES - A PLUS B TO O
2324  1022            TAD     OMSW
2325  7700            SMA CLA                 /IS O LT 0
2326  5341            JMP     END15           /NO - OK
2327  1004            TAD     PSTAT           /GET STATUS WORD
2330  0365            AND     (7737           /SAVE OTHER BITS
2331  1364            TAD     (40             /SET UNDERFLOW
2332  3004            DCA     PSTAT
2333  1002            TAD     COMREG          /GET COMMAND REGISTER
2334  7004            RAL
2335  7700            SMA CLA                 /TRAP ON UNDERFLOW ?
2336  7040            CMA                     /NO - SET IGNORE UNDERFLOW FF
2337  3763'           DCA     IGNFL
2340  7040            CMA
2341  3770'   END15,  DCA     EXITSW          /SET OR CLEAR EXIT SWITCH
2342  5767'           JMP     DEPEND          /END OF DEPOSIT

2363  0734
2364  0040
2365  7737
2366  2400
2367  2415
2370  0725
2371  0724
2372  5000
2373  7000
2374  0737
2375  0730
2376  7525
2377  7516
      2400            PAGE
```

```
2400  4577      OPOS,   AMBO                    /A MINUS B TO 0
2401  4523              SOEZ
2402  7610              SKP CLA
2403  5207              JMP     .+4
2404  1022              TAD     OMSW
2405  7700              SMA  CLA                /IS 0 LEQ 0
2406  5214              JMP     .+6             /NO
2407  1004              TAD     PSTAT           /YES - SET OVERFLOW
2410  0377              AND     (7637           /SAVE ALL BITS EXCEPT UNDERFLOW
2411  1376              TAD     (100
2412  3004              DCA     PSTAT
2413  7040              CMA
2414  3775'             DCA     EXITSW          /SET OR CLEAR EXIT SWITCH
2415  6556      DEPEND, FPRST                   /READ FPP STATUS WORD
2416  0374              AND     (740            /EXTRACT OVERFLOW BITS
2417  7041              CIA
2420  1004              TAD     PSTAT           /COMPARE WITH PROGRAM STATUS
2421  7640              SZA CLA
2422  4773'             JMS     STERR           /STATUS ERROR
2423  1775'             TAD     EXITSW
2424  7640              SZA CLA
2425  5227              JMP     EXIT
2426  5772'             JMP     FETCH
```

```
2427  4563  EXIT,   ENTER
2430  1771/         TAD     IGNFL           /GET IGNORE TRAP FF
2431  7650          SNA CLA                 /UNDERFLOW IGNORED ?
2432  5267          JMP     EXSAV           /NO - SAVE APT
2433  3771/         DCA     IGNFL           /RESET UNDERFLOW FLAG
2434  4562  EXIT0U, FSTEP                   /STEP TO STATE 0
2435  4520          TOMEM                   /ANSWER STORED IN FAC?
2436  5241          JMP     .+3             /NO - 0 TO MEM
2437  4573          CLRAC                   /CLEAR THE FAC
2440  5264          JMP     TOFECH
2441  4554          INCOP                   /INCREMENT OP ADDRESS
2442  4562  EXIT1U, FSTEP                   /STEP TO STATE 1
2443  4554          INCOP                   /INCREMENT OP ADDRESS
2444  4530          SEPM                    /EPM MODE?
2445  5261          JMP     EXIT2U          /NO
2446  4770/         JMS     SST             /1 TO SPECIAL STATE
2447  4554          INCOP                   /INC OP ADDRESS
2450  4562  EXIT10, FSTEP                   /STEP TO SPECIAL STATE 0
2451  4554          INCOP                   /INC OP ADDRESS
2452  4562  EXIT11, FSTEP                   /STEP TO SPECIAL STATE 1
2453  4554          INCOP                   /INC OP ADDRESS
2454  4562  EXIT12, FSTEP                   /STEP TO SPECIAL STATE 2
2455  4564          DECOP                   /DEC OP ADDRESS
2456  4564          DECOP                   /DEC OP ADDRESS
2457  4564          DECOP                   /DEC OP ADDRESS
2460  4767/         JMS     RST             /0 TO SPECIAL STATE

2461  4562  EXIT2U, FSTEP                   /STEP TO STATE 2
2462  4564          DECOP                   /DECREMENT THE OP ADDRESS
2463  4564          DECOP                   /DECREMENT THE OP ADDRESS
2464  4517  TOFECH, TRACE                   /TRACING PROGRAM?
2465  2424          VFER                    /UNDERFLOW ERROR - GO TO FETCH
2466  5772/         JMP     FETCH

2467  3766/ EXSAV,  DCA     FLAG1           /CLEAR ERROR FLAG
2470  1002          TAD     COMREG          /GET THE COMMAND REGISTER
2471  0365          AND     (20             /EXTRACT CR7
2472  7640          SZA CLA                 /SAVE THE FAC?
2473  5324          JMP     EXIT0           /NO
2474  4530          SEPM                    /EXTEND PRECISION MODE?
2475  5324          JMP     EXIT0           /NO
2476  4770/         JMS     SST             /YES - 1 TO SPECIAL STATE

2477  4562  EXIT00, FSTEP                   /STEP TO SPECIAL STATE 0
2500  4555          INCAPT
2501  4555          INCAPT                  /INC ADRS
2502  4555          INCAPT
2503  4575          CAPT                    /CHECK APT ADDRESS
```

```
2504  4562    EXITØ1, FSTEP                 /STEP TO SPECIAL STATE 1
2505  1057            TAD     ACLSW3        /GET FAC LSW3
2506  4764'           JMS     EXCOM         /CHECK FPP DATA
2507  4565            DECAPT                /DEC ADRS
2510  4575            CAPT                  /CHECK APT ADDRESS

2511  4562    EXITØ2, FSTEP                 /STEP TO SPECIAL STATE 2
2512  1056            TAD     ACLSW2        /GET FAC LSW2
2513  4764'           JMS     EXCOM         /CHECK FPP DATA
2514  4565            DECAPT                /DEC ADRS
2515  4575            CAPT                  /CHECK APT ADDRESS

2516  4562    EXITØ3, FSTEP                 /STEP TO SPECIAL STATE 3
2517  1055            TAD     ACLSW1        /GET FAC LSW1
2520  4764'           JMS     EXCOM         /CHECK FPP DATA
2521  4565            DECAPT                /DEC ADRS
2522  4575            CAPT                  /CHECK APT ADDRESS
2523  4767'           JMS     RST           /Ø TO SPECIAL STATE

2524  4562    EXITØ,  FSTEP                 /STEP TO STATE Ø
2525  1002            TAD     COMREG
2526  0365            AND     (20
2527  7640            SZA CLA               /SAVE THE FAC?
2530  5333            JMP     .+3           /NO
2531  1054            TAD     ACLSW         /YES - COMPARE LSW
2532  4764'           JMS     EXCOM
2533  4565            DECAPT                /DEC ADRS
2534  4575            CAPT                  /CHECK APT ADDRESS

2535  4562    EXITI,  FSTEP                 /STEP TO STATE 1
2536  1002            TAD     COMREG
2537  0365            AND     (20
2540  7640            SZA CLA               /SAVE FAC?
2541  5344            JMP     .+3           /NO
2542  1053            TAD     ACMSW         /YES
2543  4764'           JMS     EXCOM         /COMPARE MSW
2544  4565            DECAPT                /DEC, ADRS
2545  4575            CAPT                  /CHECK APT ADDRESS
```

```
2546  4562  EXIT2,  FSTEP                 /STEP TO MAJOR STATE 2
2547  1002           TAD      COMREG
2550  0365           AND      (20
2551  7640           SZA CLA               /SAVE FAC?
2552  5355           JMP      .+3          /NO
2553  1052           TAD      ACEXP
2554  4764'          JMS      EXCOM        /YES - COMPARE EXPONENT
2555  4565           DECAPT                /DEC ADRS
2556  4575           CAPT                  /COMPARE APT ADDRESS

2557  5763'          JMP      EXIT3        /GO TO STATE 3

2563  2600
2564  2663
2565  0020
2566  0726
2567  7525
2570  7516
2571  0734
2572  1261
2573  7114
2574  0740
2575  0725
2576  0100
2577  7637
      2600           PAGE
```

```
2600  4562  EXIT3,  FSTEP                 /STEP TO STATE 3
2601  1002           TAD     COMREG
2602  0377           AND     (200
2603  7640           SZA CLA               /SAVE OP ADDR?
2604  5207           JMP     .+3           /NO
2605  1071           TAD     OPADR+1
2606  4263           JMS     EXCOM         /YES - COMPARE OP ADDR
2607  4565           DECAPT                /DEC, ADRS
2610  4575           CAPT                  /CHECK APT ADDRESS

2611  4562  EXIT4,  FSTEP                 /STEP TO STATE 4
2612  1002           TAD     COMREG
2613  0376           AND     (40
2614  7640           SZA CLA               /SAVE P0 ADDR?
2615  5220           JMP     .+3           /NO
2616  1075           TAD     PBASE+1
2617  4263           JMS     EXCOM         /YES - COMPARE P0 ADDR
2620  4565           DECAPT                /DEC, ADRS
2621  4575           CAPT                  /CHECK APT ADDRESS

2622  4562  EXIT5,  FSTEP                 /STEP TO STATE 5
2623  1002           TAD     COMREG
2624  0375           AND     (100
2625  7640           SZA CLA               /SAVE X0 ADDRESS?
2626  5231           JMP     .+3           /NO
2627  1077           TAD     X0ADR+1       /YES
2630  4263           JMS     EXCOM         /COMPARE X0 ADDRESS
2631  4565           DECAPT                /DEC, ADRS
2632  4575           CAPT                  /CHECK APT ADDRESS
2633  5234           JMP     EXIT6         /GO TO STATE 6
```

```
2634  4562  EXIT6,  FSTEP                    /STEP TO STATE 6
2635  1067           TAD     PFPC+1
2636  4263           JMS     EXCOM           /COMPARE FPC
2637  4565           DECAPT                  /DEC, ADRS
2640  4575           CAPT                    /CHECK APT ADDRESS

2641  4562  EXIT7,  FSTEP                    /STEP TO STATE 7
2642  4774' EXIT7+1  JMS     APTPAC          /PACK APT FIELD BITS
2643  4263           JMS     EXCOM           /COMPARE FIELD BITS
2644  6557           FPIST                   /CHECK FLAG
2645  7402           ERROR   HALT            /FLAG IS NOT SET
2646  0373           AND     (740            /GET OVERFLOW STATUS
2647  7041           CIA
2650  1004           TAD     PSTAT
2651  7640           SZA CLA
2652  4772' EXIT7+6  JMS     STERR           /STATUS ERROR
2653  1771' EXIT7+7  TAD     FLAG1           /CHECK ERROR FLAG
2654  7650           SNA CLA                 /WAS APT DATA STORED CORRECTLY ?
2655  5261           JMP     .+4             /YES
2656  4770' EXIT7+10 JMS     TYP             /NO - TYPE ERROR MESSAGE
2657  2721           APTERR
2660  0000           0
2661  4767' EXIT7+13 JMS     SETRET          /SET REENTER ADDRESS
2662  5766' EXIT7+14 JMP     START           /GO TO CONTROL PROGRAM

2663  0000  EXCOM,   0
2664  3765' EXCOM+1  DCA     EXWD            /SAVE COMPARE WORD
2665  1072           TAD     PAPT            /GET APT FIELD BITS
2666  7104           CLL RAL
2667  7006           RTL
2670  1364           TAD     (CDF            /CREATE FPP CDF
2671  3272           DCA     .+1
2672  6201           CDF                     /CHANGE TO FPP FIELD
2673  1473           TAD I   PAPT+1          /GET APT INFO
2674  6201           CDF     00              /PROGRAM FIELD
2675  7041           CIA
2676  1765' EXCOM+14 TAD     EXWD
2677  7650           SNA CLA
2700  5303           JMP     .+3
2701  7040           CMA
2702  3771' EXCOM+17 DCA     FLAG1           /SET ERROR FLAG
2703  5663           JMP I   EXCOM           /RETURN
```

```
2704  4563  DPADD,  ENTER                 /D,P, ADD AND SUB
2705  4562  DPADD0, FSTEP                 /STEP TO STATE 0
2706  4560           GETOP                 /OUTBRK USING OP ADDR
2707  3036           DCA     BMSW          /MB TO BMSW
2710  4554           INCOP                 /INC OF ADDR
2711  4547           LOADAI  FACFR         /FAC FRAC TO A
2712  0053

2713  4562  DPADD1, FSTEP                 /STEP TO STATE 1
2714  4560           GETOP                 /OUTBRK USING OP ADDR
2715  3037           DCA     BLSW          /MB TO BLSW
2716  1005           TAD     PIR
2717  0363           AND     (2000
2720  7650           SNA CLA               /SUBTRACTION?
2721  5324           JMP     .+3           /NO - GO TO ADD
2722  4577           AMBO                  /A MINUS B TO 0
2723  7410           SKP
2724  4576           APBO                  /A PLUS B TO 0
2725  1762'          TAD     OVFL          /OVERFLOW?
2726  7640           SZA CLA
2727  5331           JMP     DPOVFL        /YES
2730  5761'          JMP     DEP           /TO DEPOSIT

2731  1004  DPOVFL, TAD     PSTAT
2732  0360           AND     (7577
2733  1377           TAD     (200          /SET ARITH FLOW
2734  3004           DCA     PSTAT
2735  5757'          JMP     EXIT          /TO EXIT
```

```
2736  4563    PFADD,  ENTER                   /ADD - SUB OF F.P. NOS.
2737  4562    FADD2,  FSTEP                   /STEP TO STATE 0
2740  4560            GETOP                   /OUTBRK USING OP ADDR
2741  4521            STORB                   /MB TO BLSW SIGN TO BMSW
2742  1037            TAD     BLSW
2743  3045            DCA     MQLSW           /MB TO MQ LSW
2744  4554            INCOP                   /INC OP ADDR
2745  1052            TAD     ACEXP           /GET FAC EXPONENT
2746  4522            STORA                   /FAC EXP TO ALSW SIGN TO AMSW
2747  4577            AMBO                    /A MINUS B TO 0
2750  1022            TAD     OMSW
2751  7700            SMA CLA                 /IS 0 LESS THAN 0
2752  5756'           JMP     SHFOP           /NO - SHIFT OPERAND PATH

                      NPAGE

2753  5755            JMP I   (',+2008&7600
2755  3000
2756  3112
2757  2427
2760  7577
2761  2052
2762  0733
2763  2000
2764  6201
2765  0732
2766  0234
2767  7354
2770  6703
2771  0726
2772  7114
2773  0740
2774  6011
2775  0100
2776  0040
2777  0200
      3000            PAGE
```

```
3000  7240          STA                 /SHIFT FAC FRACTION PATH
3001  3777'         DCA     FLAG1       /SET SHFT FAC FF
3002  1023          TAD     OLSW
3003  3776'         DCA     SHREG       /0 TO SHFT CNTR
3004  4545          LOADB!  OREG        /0 TO B
3005  0022
3006  4530          SEPM                /EPM MODE?
3007  1375          TAD     (-44        /NO 73-44=27
3010  1374          TAD     (73         /73(8)=59(10)
3011  4522          STORA               /+27 OR +59 TO ALSW
3012  4576          APBO                /A PLUS B TO O
```

```
3013  4562    FADD1,  FSTEP                   /STEP TO STATE 1
3014  4527            SEPMEN                  /1 TO EPM ENABLE
3015  4560            GETOP                   /OUTBRK USING OP ADDR
3016  4773/           JMS     CKSUB           /SUBTRACTION?
3017  7040            CMA                     /YES = COMPLIMENT MB
3020  3030            DCA     AMSW            /MB TO AMSW
3021  4554            INCOP                   /INC OP ADDR
3022  1022            TAD     OMSW
3023  7710            SPA CLA
3024  7040            CMA
3025  3772/           DCA     FLAG2           /0 Ø TO OVERSHFT FF
3026  4572            CLRB                    /Ø TO B
3027  4530            SEPM                    /EPM MODE
3030  5260            JMP     FADD2           /NO = GO TO STATE 2
3031  4771/           JMS     SEX             /SET TMSC EXECUTE
3032  4554            INCOP                   /INC OP ADDRESS

3033  4562    FADD10, FSTEP                   /STEP TO TMSC STATE Ø
3034  4560            GETOP                   /OUTBRK USING OP ADDRESS
3035  4773/           JMS     CKSUB           /SUBTRACTION?
3036  7040            CMA                     /YES = COMPLIMENT MB
3037  3032            DCA     ALSW1           /MB TO ALSW1
3040  4554            INCOP                   /INC OP ADDRESS

3041  4562    FADD11, FSTEP                   /STEP TO TMSC STATE 1
3042  4560            GETOP                   /OUTBRK USING OP ADDRESS
3043  4773/           JMS     CKSUB           /SUBTRACTION?
3044  7040            CMA                     /YES = COMPLIMENT MB
3045  3033            DCA     ALSW2           /MB TO ALSW2
3046  4554            INCOP                   /INC OP ADDRESS

3047  4562    FADD12, FSTEP                   /STEP TO TMSC STATE 2
3050  4560            GETOP                   /OUTBRK USING OP ADDRESS
3051  4773/           JMS     CKSUB           /SUBTRACTION?
3052  7040            CMA                     /YES = COMPLIMENT MB
3053  3034            DCA     ALSW3           /MB TO ALSW3
3054  4564            DECOP                   /DEC OP ADDRESS
3055  4564            DECOP                   /DEC OP ADDRESS
3056  4564            DECOP                   /DEC OP ADDRESS
3057  4770/           JMS     REX             /RESET TMSC EXECUTE
```

```
3060  4562  FADD2,  FSTEP                 /STEP TO STATE 2
3061  4560           GETOP                /OUTBRK USING OP ADDR
3062  4773'          JMS      CKSUB       /SUBTRACTION?
3063  5266           JMP      .+3         /YES = NOT MB TO ALSW
3064  3031           DCA      ALSW        /NO = MB TO ALSW
3065  5274           JMP      .+7
3066  7040           CMA                  /COMPLIMENT MB
3067  3031           DCA      ALSW        /NOT MB TO ALSW
3070  1367           TAD      C7400       /COMPLIMENT 0 TO AEXT
3071  3035           DCA      AEXT
3072  1114           TAD      C400        /1 TO ACRY IN
3073  3007           DCA      CARYIN
3074  4576           APBO                 /A PLUS B TO 0 B=0 IF ADDITION
3075  4545           LOADB;   FACFR       /FAC FRAC TO B
3076  0053
3077  4523           SQEZ                 /0 = 0 ?
3100  5306           JMP      .+6         /NO
3101  1052           TAD      ACEXP       /YES =0=0
3102  3045           DCA      MQLSW       /FAC EXP TO MQLSW
3103  4543           LOAD0;   BREG        /B TO 0
3104  0036
3105  5766'          JMP      FADEND      /GO TO DEPOSIT
3106  1772'          TAD      FLAG2       /OVERSHIFT FF = 1 ?
3107  7640           SZA CLA
3110  5766'          JMP      FADEND      /YES = GO TO DEPOSIT
3111  5765'          JMP      FADD3       /NO = GO TO STATE 3
```

```
3112  1023    SHFOP,  TAD     OLSW            /STATE 0 SHIFT OPERAND
3113  7040            CMA
3114  3776/           DCA     SHREG           /0 COMPLIMENT TO SHFT CNTR
3115  4545            LOADBI  OREG            /0 TO B
3116  0022
3117  4530            SEPM                    /EPM MODE?
3120  1375            TAD     (-44            /NO, 73-44=27
3121  1374            TAD     (73             /73(8)=59(10)
3122  4522            STORA                   /+27 OR +59 TO ALSW
3123  4577            AMBO                    /A MINUS B TO 0
3124  2776/           ISZ     SHREG           /INC SHFT COUNTER
3125  7000            NOP

3126  4562    FADD01, FSTEP                   /STEP TO STATE 1 (SHFT OP PATH)
3127  4527            SEPMEN                  /1 TO EPM ENABLE
3130  4560            GETOP                   /OUTBRK USING OP ADDRESS
3131  3036            DCA     BMSW            /MB TO BMSW
3132  4554            INCOP                   /INC OP ADDR
3133  1022            TAD     OMSW
3134  7710            SPA CLA
3135  7040            CMA
3136  3772/           DCA     FLAG2           /0 0 TO OVERSHFT FF
3137  4547            LOADAI  FACFR           /FAC FRAC TO A
3140  0053
3141  4543            LOADOI  AREG            /A TO O
3142  0030
3143  4523            SOEZ                    /O = 0 ?
3144  5764/           JMP     FAD01A          /NO
3145  7040            CMA                     /YES
3146  3763/           DCA     FLAG3           /1 TO ADD ZERO FF
3147  4574            CLRA                    /0 TO A
3150  5762/           JMP     FAD01B

3162  3205
3163  0730
3164  3200
3165  3242
3166  3275
3167  7400
3170  7507
3171  7500
3172  0727
3173  6433
3174  0073
3175  7734
3176  0724
3177  0726
      3200            PAGE
```

```
3230  1052   FAD01A, TAD     ACEXP
3231  3045           DCA     MQLSW          /FAC EXP TO MQLSW
3232  1777'          TAD     FLAG2
3233  7640           SZA CLA                /OVERSHFT FF=1?
3234  5275           JMP     FADEND         /YES - GO TO DEPOSIT
3235  4530   FAD01B, SEPM                   /EPM MODE
3236  5230           JMP     FADD02         /NO - GO TO STATE 2
3237  4776'          JMS     SEX            /SET TMSC EXECUTE
3240  4554           INCOP                  /INC OP ADDRESS

3241  4562   FAD010, FSTEP                  /STEP TO TMSC STATE 0
3242  4560           GETOP                  /OUTBRK USING OP ADDRESS
3243  3040           DCA     BLSW1          /MB TO BLSW1
3244  4554           INCOP                  /INC OP ADDRESS

3245  4562   FAD011, FSTEP                  /STEP TO TMSC STATE 1
3246  4560           GETOP                  /OUTBRK USING OP ADDRESS
3247  3041           DCA     BLSW2          /MB TO BLSW2
3250  4554           INCOP                  /INC OP ADDRESS

3251  4562   FAD012, FSTEP                  /STEP TO TMSC STATE 2
3252  4560           GETOP                  /OUTBRK USING OP ADDRESS
3253  3042           DCA     BLSW3          /MB TO BLSW3
3254  4564           DECOP                  /DEC OP ADDRESS
3255  4564           DECOP                  /DEC OP ADDRESS
3256  4564           DECOP                  /DEC OP ADDRESS
3257  4775'          JMS     REX            /RESET TMSC EXECUTE
```

```
3230  4562   FADD02, FSTEP                    /STEP TO STATE 2
3231  4560           GETOP                    /OUTBRK USING OP ADDR
3232  3037           DCA     BLSW             /MB TO BLSW
3233  1774'          TAD     FLAG3
3234  7650           SNA CLA                  /ADD ZERO FF=1
3235  5242           JMP     FADD3            /NO - GO TO STATE 3
3236  4543           LOADOI  BREG             /B TO O
3237  0036
3240  4577           AMBO                     /YES - A MINUS B TO O
3241  5275           JMP     FADEND           /GO TO DEPOSIT

3242  4562   FADD3,  FSTEP                    /STEP TO STATE 3
3243  4526           SHFTB                    /IF SHFT CNTR NE 0-SHIFT B
3244  4773'          JMS     CKSUB            /SUBTRACTION?
3245  7610           SKP CLA                  /YES - CHECK SHFT FAC FF
3246  5256           JMP     .+10             /NO - ADD A AND B
3247  1772'          TAD     FLAG1
3250  7640           SZA CLA                  /SHFT FF SET?
3251  5254           JMP     .+3              /NO - +1 TO CARRY IN
3252  4577           AMBO                     /YES - A MINUS B TO O
3253  5257           JMP     .+4
3254  1114           TAD     [400
3255  3007           DCA     CARYIN           /1 TO CARRYING
3256  4576           APBO                     /A PLUS B TO O
3257  1771'          TAD     OVFL
3260  7650           SNA CLA                  /FRAC OVERFLOW?
3261  5275           JMP     FADEND           /NO - GO TO DEPOSIT
3262  4545           LOADBI  OREG             /O TO B
3263  0022
3264  4524           SHIFTI  BREG    RIGHT    /SHIFTED RIGHT
3265  0036
3266  1036           TAD     BMSW
3267  1021           TAD     CAROUT           /INSERT ORIGINAL SIGN
3270  3036           DCA     BMSW
3271  4543           LOADOI  BREG             /B TO O
3272  0036
3273  2770'          ISZ     SHREG            /INC SHFT CNTR
3274  7410           SKP
3275  3770'  FADEND, DCA     SHREG
3276  3772'          DCA     FLAG1
3277  3777'          DCA     FLAG2            /CLEAR FLAGS
3300  3774'          DCA     FLAG3
3301  5767'          JMP     DEP              /GO TO DEPOSIT
```

```
3302  4563    FMULT,  ENTER                   /ENTER MULTIPLY
3303  7240            STA
3304  3766/           DCA     MDFLAG          /SET MULT, DIV, FLAG FOR MULTIPLY

3305  4562    MULT0,  FSTEP                   /ENTER STATE 0
3306  4527            SEPMEN                  /1 TO EPM ENABLE
3307  4560            GETOP                   /OUTBRK USING OP ADDR
3310  3036            DCA     BMSW
3311  1036            TAD     BMSW
3312  7710            SPA CLA
3313  7040            CMA
3314  3772/           DCA     FLAG1           /MB0 TO OP SIGN
3315  4554            INCOP                   /INCREMENT OP ADDR
3316  4574            CLRA                    /0 TO A
3317  4530            SEPM                    /EPM MODE ?
3320  1365            TAD     (-44            /NO - MAKE SHIFT COUNT 23
3321  1364            TAD     (73             /23 OR 59 TO THE SHIFT COUNT
3322  3770/           DCA     SHREG           /23 TO SHIFT CNTR
3323  1763/           TAD     SHFLAG
3324  7700            SMA CLA                 /OPERATOR SELECTED SHIFT CNT?
3325  5331            JMP     .+4             /NO-USE NORMAL SHIFT CNT
3326  1762/           TAD     SHFCNT          /YES-GET SPECIAL COUNT
3327  6567            LSHFT                   /LOAD FPP SHIFT REG
3330  3770/           DCA     SHREG
3331  1770/           TAD     SHREG           /NEGATE SHIFT REG
3332  7040            CMA                     /FOR DECREMENT WITH
3333  3770/           DCA     SHREG           /ISZ LOOP
3334  4530            SEPM                    /EPM MODE ?
3335  5761/           JMP     MULT1           /NO - GO TO STATE 1
3336  4776/           JMS     SEX             /YES - 1 TO TMSC EXECUTE
3337  4554            INCOP                   /INC THE OP ADDRESS
3340  5760/           JMP     MULT00          /GO TO EPM STATE 0

3360  3400
3361  3417
3362  0723
3363  0722
3364  0073
3365  7734
3366  0737
3367  2052
3370  0724
3371  0733
3372  0726
3373  6433
3374  0730
3375  7507
3376  7500
3377  0727
      3400            PAGE
```

```
3400  4562  MULT00, FSTEP                /STEP TO EPM STATE 0
3401  4560        GETOP                 /OUTBRK USING OP ADDRESS
3402  3040        DCA     BLSW1         /MB TO B LSW1
3403  4554        INCOP                 /INC OP ADDRESS

3404  4562  MULT01, FSTEP                /STEP TO EPM STATE 1
3405  4560        GETOP                 /OUTBRK USING OP ADDRESS
3406  3041        DCA     BLSW2         /MB TO BLSW 2
3407  4554        INCOP                 /INC OP ADDRESS

3410  4562  MULT02, FSTEP                /STEP TO EPM STATE 2
3411  4560        GETOP                 /OUTBRK USING OP ADDRESS
3412  3042        DCA     BLSW3         /MB TO BLSW 3
3413  4564        DECOP                 /DEC OP ADDRESS
3414  4564        DECOP                 /DEC OP ADDRESS
3415  4564        DECOP                 /DEC OP ADDRESS
3416  4777'       JMS     REX           /RESET TMSC EXECUTE


3417  4562  MULT1,  FSTEP                /STEP TO STATE 1
3420  4560        GETOP                 /OUTBRK USING OP ADDR
3421  3037        DCA     BLSW          /MB TO BLSW
3422  1776'       TAD     FLAG1         /GET OP SIGN
3423  7710        SPA CLA               /OP SIGN=1?
3424  5227        JMP     .+3           /YES
3425  4576        APBO                  /NO=A PLUS B TO 0
3426  7610        SKP CLA
3427  4577        AMBO                  /A MINUS B TO 0
3430  4544        LOADMQ; OREG          /0 TO MQ
3431  0022
3432  1053        TAD     ACMSW
3433  7710        SPA CLA               /GAC FRAC LT 0?
3434  5240        JMP     .+4           /YES
3435  4547        LOADA;  FACFR         /FAC FRAC TO A
3436  0053
3437  5252        JMP     MULT1A
3440  4531        SAVE;   OREG          /SAVE THE 0 REG TO USE THE ADDER
3441  0022
3442  4574        CLRA                  /CLEAR THE A REG
3443  4545        LOADB;  FACFR         /FAC FRAC TO B FOR NEGATION
3444  0053
                                        /THIS REPLACES COMPLIMENT FAC
3445  4577        AMBO                  /TO A AND 1 TO CARRY IN
3446  4547        LOADA;  OREG          /MINUS FAC FRAC TO A
3447  0022
3450  4543        LOADO;  TREG          /RESTORE THE 0 REGISTER
3451  0060
3452  4572  MULT1A, CLRB                /0 TO B
3453  5254        JMP     MULT2         /GO TO STATE 2
```

```
3454  4562  MULT2,  FSTEP                    /STEP TO STATE 2
3455  3007  MADD,   DCA     CARYIN
3456  4576           APBO                    /ENABLE A PLUS B TO 0
3457  2775           ISZ     SHREG           /DEC SHFT CNTR -S',C',=0
3460  7410           SKP                     /NO=CONTINUE
3461  5307           JMP     MULT21          /YES=GO TO STATE 2
3462  4530           SEPM                    /EPM MODE ?
3463  5266           JMP     .+3             /NO
3464  1050           TAD     MQLSW3          /YES = GET MQ 59
3465  7410           SKP
3466  1045           TAD     MQLSW           /NO = GET MQ 23
3467  7010           RAR                     /MOVE LSB TO LINK
3470  7620           SNL CLA                 /MQ 23 OR MQ 59 = 1 ?
3471  5276           JMP     .+5             /NO
3472  4545           LOADB;  OREG            /YES = STROBE B
3473  0022
3474  1027           TAD     OEXT            /0 EXT TO B EXT
3475  3043           DCA     BEXT            /NOT DONE IN LOAD ROUTINE
3476  4524           SHIFT;  BREG    RIGHT   /SHIFT B
3477  0036
3500  4524           SHIFT;  MQREG   RIGHT   /MQ(N) TO MQ(N+1)
3501  0044
3502  1044           TAD     MQMSW           /INSERT ONES INTO MQ00
3503  1374           TAD     (4000
3504  3044           DCA     MQMSW
3505  3051           DCA     MQEXT           /CLEAR MQ EXT
3506  5255           JMP     MADD            /CONTINUE UNTILL S',C',=0
```

```
3507  4543  MULT21, LOADDI  BREG          /B TO 0
3510  0036
3511  1002          TAD      COMREG
3512  7710          SPA CLA                /DOUBLE PRECISION?
3513  5316          JMP      .+3           /YES
3514  4564          DECOP                  /DEC OR ADDR
3515  4564          DECOP
3516  4545          LOADBI   OREG          /O TO B
3517  0022
3520  1027          TAD      OEXT          /O EXT TO B EXT
3521  3043          DCA      BEXT
3522  4574          CLRA                   /0 TO A
3523  1053          TAD      ACMSW         /
3524  7104          CLL RAL                /
3525  7200          CLA                    /PROG NEG?
3526  1776'         TAD      FLAG1
3527  7530          SPA SZL
3530  7410          SKP
3531  5336          JMP      .+5           /NO-PROD IS POS
3532  7060          CMA CML
3533  7520          SMA SNL
3534  5336          JMP      .+2           /NO-PROD IS POS
3535  4577          AMBO                   /PROD IS NEG-A MINUS B TO 0
3536  1002          TAD      COMREG
3537  7710          SPA CLA                /DOUBLE PRECISION?
3540  5773'         JMP      MULEND        /YES-GO TO DEPOSIT

                    NPAGE

3541  5772          JMP I    (.+200&7600
3572  3600
3573  3605
3574  4000
3575  0724
3576  0726
3577  7507
      3600          PAGE
```

```
3600  4562    MULT3,  FSTEP                   /STEP TO STATE 3
3601  4560            GETOP                   /OUTBRK USING OP ADDR
3602  3045            DCA     MQLSW           /MB TO MQLSW
3603  4554            INCOP                   /INC OP ADDR
3604  4554            INCOP
3605  5777/   MULEND,  JMP     DEP             /GO TO DEPOSIT


3606  4563    PFDIV,  ENTER                   /ENTER DIVIDE
3607  7240            STA
3610  3776/           DCA     MDFLAG          /SET MULT, DIV, FLAG

3611  4562    DIV0,   FSTEP                   /STEP TO STATE 0
3612  4527            SEPMEN                  /1 TO EPM ENABLE
3613  4547            LOADAI  FACFR           /FAC FRAC TO A
3614  0053
3615  1053            TAD     ACMSW           /B CAN BE USED FOR NEGATION
3616  7700            SMA CLA                 /FAC < 0?
3617  5227            JMP     .+10            /NO - FAC TO A IS ALREADY DONE
3620  4545            LOADBI  FACFR           /*****
3621  0053
3622  4574            CLRA                    /COMP FAC TO A, A+1 TO 0, 0 TO A
3623  4577            AMBO                    /BELIEVE ME IT WORKS AND DOESN'T
3624  3027            DCA     OEXT            /TAKE MUCH ROOM,
3625  4547            LOADAI  OREG            /
3626  0022
3627  4560            GETOP                   /OUTBRK USING OP ADDR
3630  3036            DCA     BMSW            /MB TO BMSW
3631  1036            TAD     BMSW
3632  7710            SPA CLA
3633  7040            CMA
3634  3775/           DCA     FLAG1           /MB0 TO OP SIGN
3635  4554            INCOP                   /INC OP ADDR
3636  1774/   DIV0A,  TAD     SHFLAG
3637  7640            SZA CLA                 /SPECIAL SHIFT?
3640  5245            JMP     .+5             /YES - GET COUNT
3641  4530            SEPM                    /EPM MODE?
3642  1373            TAD     (-37            /NO, 73-37=34
3643  1372            TAD     (73             /GET SHFT CNT, 34 OR 59
3644  3771/           DCA     SHFCNT          /SET SHIFT COUNT
3645  1771/           TAD     SHFCNT          /YES--SHFCNT=0 IF NOT USED
3646  6567            LSHFT                   /LOAD FPP SHIFT REG
3647  1771/           TAD     SHFCNT
3650  7041            CIA
3651  3770/           DCA     SHREG
3652  4571            CLRMQ                   /CLEAR MQ REG
3653  4530            SEPM                    /EPM MODE?
3654  5276            JMP     DIV1            /NO - GO TO STATE 1
3655  4767/           JMS     SEX             /SET TMSC EXECUTE
3656  4554            INCOP                   /INCREMENT THE OP ADDRESS
```

```
3657  4562  DIV00,  FSTEP                  /STEP TO TMSC STATE 0
3660  4560          GETOP                  /OUTBRK USING OP ADDRESS
3661  3040          DCA     BLSW1          /MB TO BLSW1
3662  4554          INCOP                  /INC OP ADDRESS

3663  4562  DIV01,  FSTEP                  /STEP TO TMSC STATE 1
3664  4560          GETOP                  /OUTBRK USING OP ADDRESS
3665  3041          DCA     BLSW2          /MB TO BLSW2
3666  4554          INCOP                  /INC OP ADDRESS

3667  4562  DIV02,  FSTEP                  /STEP TO TMSC STATE 2
3670  4560          GETOP                  /OUTBRK USING OP ADDRESS
3671  3042          DCA     BLSW3          /MB TO BLSW3
3672  4564          DECOP                  /DEC OP ADDRESS
3673  4564          DECOP                  /DEC OP ADDRESS
3674  4564          DECOP                  /DEC OP ADDRESS
3675  4766'         JMS     REX            /RESET TMSC EXECUTE

3676  4562  DIV1,   FSTEP                  /STEP TO STATE 1
3677  4560          GETOP                  /OUTBRK USING OP ADDR
3700  3037          DCA     BLSW           /MB TO BLSW
3701  3043          DCA     BEXT           /CLEAR B EXT
3702  4543          LOADOI  BREG           /B TO O
3703  0036
3704  4523          SOEZ                   /O = 0 ?
3705  5315          JMP     .+10           /NO
3706  1004          TAD     PSTAT          /YES--SET DIVIDE BY 0 BIT
3707  0365          AND     (7377          /SAVE OTHER BITS
3710  1364          TAD     (400
3711  3004          DCA     PSTAT
3712  4517          TRACE                  /TRACING PROGRAM?
3713  2364          DIVZ                   /YES--DIVIDE BY ZERO
3714  5763'         JMP     EXIT           /GO TO EXIT
3715  4543          LOADOI  AREG           /A TO O
3716  0030
```

```
3717  4562  DIV2,   FSTEP                      /STEP STATE 2
3720  1770'         TAD     SHREG             /CHECK SHIFT REG FOR ZERO
3721  7650          SNA CLA
3722  5762'         JMP     ENDDIV            /NO DIVIDE IF SH REG=0
3723  4531  DIVIDE, SAVE)   OREG              /SAVE THE O REG IN CASE CARRYOUT=0
3724  0022
3725  1775'         TAD     FLAG1             /GET OP SIGN
3726  7700          SMA CLA                   /OP SIGN=0?
3727  5332          JMP     .+3               /YES
3730  4576          APBO                      /NO = A+B TO O
3731  7410          SKP
3732  4577          AMBO                      /OP SIGN=0, ENABLE A-B
3733  4524          SHIFT)  MQREG   LEFT      /MQ(N) TO MQ(N-1)
3734  4044
3735  4530          SEPM                      /EPM MODE?
3736  5343          JMP     .+5               /NO
3737  1021          TAD     CAROUT            /GET CARRY OUT
3740  7640          SZA CLA                   /CARRY OUT=1?
3741  2050          ISZ     MQLSW3            /YES = CARRY OUT TO MQ39
3742  5350          JMP     .+6
3743  1021          TAD     CAROUT            /GET CARRY OUT
3744  7110          CLL RAR
3745  7012          RTR
3746  1051          TAD     MQEXT             /CARRY OUT TO MQ27
3747  3051          DCA     MQEXT

                    NPAGE

3750  5761          JMP I   (.+2000&7600
3761  4000
3762  4015
3763  2427
3764  0400
3765  7377
3766  7507
3767  7500
3770  0724
3771  0723
3772  0073
3773  7741
3774  0722
3775  0726
3776  0737
3777  2052
      4000          PAGE
```

```
4070  1021          TAD     CAROUT
4071  7710          SPA CLA                 /CARRY OUT=1
4072  5205          JMP     .+3             /YES = 0 IS ALREADY LOADED
4073  4543          LOADOI  TREG            /NO = RESTORE THE 0 REG
4074  0060
4075  4524          SHIFTI  OREG    LEFT    /SHIFT 0
4076  4022
4077  2777'         ISZ     SHREG           /DEC SHIFT COUNTER
4010  7410          SKP                     /SHIFT COUNTER NOT=Ø
4011  5215          JMP     ENDDIV          /S,C,=Ø END OF DIVIDE
4012  4547          LOADAI  OREG            /O(N) TO A(N)
4013  0022
4014  5776'         JMP     DIVIDE


4015  4545    ENDDIV, LOADBI  MQREG         /YES = MQ TO B
4016  0044
4017  1051          TAD     MQEXT           /LOAD EXTENSION
4020  3043          DCA     BEXT
4021  4574          CLRA                    /ZERO TO A
4022  3007          DCA     CARYIN          /CLEAR CARRY IN FF
4023  4543          LOADOI  BREG            /B TO 0
4024  0036
4025  1002          TAD     COMREG          /GET COMMAND REGISTER
4026  7700          SMA CLA                 /FIXED POINT MODE ?
4027  5261          JMP     FLDIV           /NO = GO TO FLOATING DIVIDE
4030  1022          TAD     OMSW
4031  7700          SMA CLA                 /O LESS THAN Ø?
4032  5243          JMP     DIVOK           /NO--CONTINUE
4033  4247          JMS     QUONEG          /A = B IF QUOTIENT IS NEGATIVE
4034  1004          TAD     PSTAT           /GET STATUS REG
4035  0375          AND     (7577           /SAVE OTHER BITS
4036  1374          TAD     (200            /SET FRAC OVERFLOW BIT
4037  3004          DCA     PSTAT
4040  4517          TRACE                   /TRACING PROGRAM?
4041  2375          DIVOV                   /YES=DIVIDE F.P. OVERFLOW
4042  5773'         JMP     EXIT            /GO TO EXIT
4043  4543    DIVOK,  LOADOI  BREG          /B TO 0 IF SIGNS ARE LIKE
4044  0036
4045  4247          JMS     QUONEG          /A = B IF QUOTIENT IS NEGATIVE
4046  5313          JMP     DVEND           /GO TO DEPOSIT

4047  0000    QUONEG, Ø
4050  1053          TAD     ACMSW           /GET FAC MSW
4051  7104          CLL RAL                 /SAVE SIGN
4052  7200          CLA
4053  1772'         TAD     FLAG1           /GET OP SIGN
4054  7530          SPA SZL                 /OP SIGN = FAC(Ø)?
4055  7060          CMA CML
4056  7530          SPA SZL
4057  4577          AMBO                    /NO=A MINUS B TO 0
4060  5647          JMP I   QUONEG          /RETURN
```

```
4061  4564  FLDIV,  DECOP                    /DEC OP ADDRESS
4062  4564          DECOP                    /DEC OP ADDR
4063  1022          TAD     OMSW
4064  7700          SMA CLA                  /O LT 0?
4065  5271          JMP     .+4              /NO
4066  4524          SHIFT;  BREG    RIGHT    /SHIFT B
4067  0036
4070  2777'         ISZ     SHREG            /INC SHFT CNTR
4071  4543          LOADO;  BREG             /B TO O IF SIGNS ARE LIKE
4072  0036
4073  1053          TAD     ACMSW            /GET FAC MSW
4074  7104          CLL RAL                  /SAVE SIGN
4075  7200          CLA
4076  1772'         TAD     FLAG1            /GET OP SIGN
4077  7530          SPA SZL
4100  7060          CMA CML                  /OP SIGN=FAC(0)?
4101  7530          SPA SZL
4102  4577          AMBO                     /NO=A MINUS B TO O
4103  5304          JMP     DIV3             /GO TO STATE 3


4104  4562  DIV3,   FSTEP                    /STEP TO STATE 3
4105  4560          GETOP                    /OUTBRK USING OP ADDR
4106  3045          DCA     MQLSW            /MB TO MQLSW
4107  3044          DCA     MQMSW            /CLEAR MQ MSW
4110  3051          DCA     MQEXT            /CLEAR MQ EXT
4111  4554          INCOP
4112  4554          INCOP                    /INC OP ADDR
4113  5771'  DVEND,  JMP     DEP              /GO TO DEPOSIT

4114  4563  FLDX,   ENTER                    /ENTER LDX
4115  4562  LDX1,   FSTEP                    /STEP TO STATE 1
4116  4574          CLRA                     /0 TO AMSW
4117  1005          TAD     PIR              /GET INSTRUCTION
4120  0370          AND     (7
4121  3031          DCA     ALSW
4122  4545          LOADB;  X0ADR            /X0 ADDR TO B
4123  0076
4124  4576          APBO                     /A PLUS B TO O
4125  4542          LOADOP; OREG             /O TO OP ADDR
4126  0022
```

```
4127  4562  LDX2,  FSTEP                 /STEP TO STATE 2
4130  4557         GETPC                 /OUTBRK USING FPC
4131  3031         DCA     ALSW          /MB TO ALSW
4132  3030         DCA     AMSW          /0 TO AMSW
4133  4552         INCPC                 /INC FPC
4134  4543         LOADO)  AREG          /A TO O
4135  0030

4136  4562  LDX3,  FSTEP                 /STEP TO STATE 3
4137  1023         TAD     OLSW
4140  4536         PUTX                  /O LSW TO INDEX REG
4141  5767'        JMP     FETCH         /GO TO FETCH


4142  4563  FALN,  ENTER                 /ENTER ALN
4143  4562  ALN1,  FSTEP                 /STEP TO STATE 1
4144  3030         DCA     AMSW          /0 TO AMSW
4145  1005         TAD     PIR           /GET INSTRUCTION
4146  0370         AND     (7
4147  3031         DCA     ALSW          /FIR 9-11 TO ALSW
4150  4545         LOADB)  X0ADR         /X0 ADDR TO B
4151  0076
4152  4576         APBO                  /A PLUS B TO O
4153  4542         LOADOP) OREG          /O TO OP ADDR
4154  0022
4155  4572         CLRB                  /0 TO B
4156  1002         TAD     COMREG        /GET COMMAND REGISTER
4157  7710         SPA CLA               /D, P, MODE ?
4160  5363         JMP     .+3           /YES - DO NOT STORE EXPONENT
4161  1052         TAD     ACEXP         /FAC EXP TO B LSW
4162  4521         STORB                 /SIGN EXTEND TO BMSW
4163  5766'        JMP     ALN2          /GO TO STATE 2

4166  4200
4167  1261
4170  0007
4171  2052
4172  0726
4173  2427
4174  0200
4175  7577
4176  3723
4177  0724
      4200         PAGE
```

```
4230  4562  ALN2,   FSTEP                  /STEP TO STATE 2
4231  1005          TAD     PIR           /GET INSTRUCTION
4232  0377          AND     (7
4233  7450          SNA                   /X = 0 ?
4234  5211          JMP     .+5           /YES = 27 TO A
4235  1376          TAD     (X0           /GET PROGRAM INDEX REG
4236  3000          DCA     T1            /INSTEAD OF OUTBRK
4237  1400          TAD I   T1            /CONTENTS OF X
4240  7410          SKP
4241  1115          TAD     (27           /GET OCTAL 27
4242  4522          STORA                 /MB OR 27 TO A
4243  1002          TAD     COMREG        /GET COMMAND REGISTER
4244  7710          SPA CLA               /D, P, MODE
4245  5220          JMP     ALN22         /LEAVE EXP ALONE
4246  1031          TAD     ALSW
4247  3052          DCA     ACEXP         /MB TO FAC EXPONENT

4250  4577  ALN22,  AMBO                  /A MINUS B TO 0
4251  1023          TAD     OLSW
4252  3775          DCA     SHREG         /0 TO SHIFT CNTR
4253  4545          LOADBI  OREG          /0 TO B
4254  0022
4255  1022          TAD     OMSW
4256  7710          SPA CLA
4257  7040          CMA
4260  3774          DCA     FLAG1         /00 TO SHIFT 0 FF
4261  1005          TAD     PIR           /GET INSTRUCTION REGISTER
4262  0377          AND     (7            /EXTRACT INDEX BITS
4263  7640          SZA CLA               /X0 ?
4264  4530          SEPM                  /EPM MODE ?
4265  1373          TAD     (-44          /27 = OVERSHIFT
4266  1372          TAD     (73           /23 OR 59
4267  4522          STORA                 /TO THE A REG
4270  1774          TAD     FLAG1
4271  7640          SZA CLA               /SHFT 0 FF=1?
4272  5245          JMP     .+3           /YES
4273  4577          AMBO                  /NO--A MINUS B TO 0
4274  7410          SKP
4275  4576          APBO                  /A PLUS B TO 0
```

```
4246  4562    ALN3,   FSTEP                   /STEP TO STATE 3
4247  4527            SEPMEN                  /1 TO EPM ENABLE
4250  1022            TAD     OMSW
4251  7700            SMA CLA                 /00=1?
4252  5255            JMP     .+3             /NO
4253  4572            CLRB                    /YES = 0 TO B
4254  5257            JMP     .+3
4255  4545            LOADB;  FACFR           /FAC FRAC TO B
4256  0053
4257  4543            LOADO;  BREG            /B TO O
4260  0036
4261  4523            SOEZ                    /O=0?
4262  5264            JMP     ALN4            /NO=GO TO STATE 4
4263  3775'           DCA     SHREG           /YES=0 TO SHFT CNTR


4264  4562    ALN4,   FSTEP                   /STEP TO STATE 4
4265  1774'           TAD     FLAG1
4266  7004            RAL                     /SAVE SHFT O FF
4267  7200            CLA
4270  1775'           TAD     SHREG           /GET SHFT CNTR
4271  7420            SNL
4272  7041            CIA                     /ADJUST SHIFT CNTR FOR ISZ
4273  3775'           DCA     SHREG
4274  1774'           TAD     FLAG1
4275  7650            SNA CLA                 /SHFT O FF=0?
4276  5301            JMP     .+3             /YES
4277  4525            SHFTO                   /NO=SHIFTO UNTIL S'C'=0
4300  5325            JMP     ALNEND
4301  4526            SHFTB                   /SHIFT B UNTIL S'C'=0
4302  1053            TAD     FACFR           /GET FAC FRACTION
4303  7710            SPA CLA                 /O = O AND FAC FR NEG ?
4304  4523            SOEZ                    /O = 0 ?
4305  5323            JMP     ALNEND=2        /NO = B TO O
4306  7240            STA                     /YES = LOGICAL 1 TO O
4307  3022            DCA     OMSW            /LOGICAL 1 TO OMSW
4310  7240            STA
4311  3023            DCA     OLSW            /LOGICAL 1 TO OLSW
4312  4530            SEPM                    /EPM MODE ?
4313  5325            JMP     ALNEND          /NO = FINISHED
4314  7240            STA                     /YES = LOGICAL 1 TO EPM LSW'S
4315  3024            DCA     OLSW1
4316  7240            STA
4317  3025            DCA     OLSW2
4320  7240            STA
4321  3026            DCA     OLSW3
4322  5325            JMP     ALNEND          /DO NOT MOVE B TO O
4323  4543            LOADO;  BREG            /B TO O
4324  0036
4325  4546    ALNEND, LOADAC; OREG            /O TO FAC FRACTION
4326  0022
4327  5771'           JMP     FETCH           /GO TO FETCH
```

```
4330  4563  FATX,   ENTER                 /ENTER ATX
4331  4562  ATX1,   FSTEP                 /STEP TO STATE 1
4332  3030          DCA      AMSW        /0 TO A MSW
4333  1005          TAD      PIR         /GET INSTRUCTION
4334  0377          AND      (7
4335  3031          DCA      ALSW        /PIR9-11 TO ALSW
4336  4545          LOADBI   X0ADR        /X0 ADDR TO B
4337  0076
4340  4576          APBO                 /A PLUS B TO O
4341  4542          LOADOPI  OREG         /O TO OP ADDR
4342  0022
4343  1002          TAD      COMREG      /GET COMMAND REG
4344  7700          SMA CLA              /FLOATING POINT?
4345  5353          JMP      .+6         /YES
4346  4545          LOADBI   FACFR       /NO-FAC FRAC TO B
4347  0053
4350  3775'         DCA      SHREG       /0 TO SHIFT CNTR
4351  4770'         JMS      SETST3      /SET PROG STATE 3
4352  5767'         JMP      ATX4        /GO TO STATE 4
4353  1052          TAD      ACEXP       /FAC EXP TO B LSW
4354  4521          STORB                /SIGN EXTEND TO BMSW
4355  5766'         JMP      ATX2        /GO TO STATE 2

4366  4400
4367  4444
4370  6250
4371  1261
4372  0073
4373  7734
4374  0726
4375  0724
4376  0200
4377  0007
      4400          PAGE
```

```
4400  4562    ATX2,   FSTEP                   /STEP TO STATE 2
4401  1377            TAD     (27
4402  4522            STORA                   /27 TO ALSW = 0 TO AMSW
4403  4577            AMBO                    /A MINUS B TO 0
4404  1023            TAD     OLSW
4405  3776/           DCA     SHREG           /0 TO SHIFT CNTR
4406  4545            LOADBI  OREG           /0 TO B
4407  0022
4410  1022            TAD     OMSW            /GET SIGN OF 0
4411  7710            SPA CLA
4412  7040            CMA
4413  3775/           DCA     FLAG1           /00 TO SHFT 0 FF
4414  1377            TAD     (27
4415  3031            DCA     ALSW            /27 TO ALSW
4416  3030            DCA     AMSW            /0 TO AMSW
4417  1775/           TAD     FLAG1
4420  7650            SNA CLA                 /SHFT 0 FF=1?
4421  5224            JMP     .+3             /NO
4422  4576            APBO                    /YES=A PLUS B TO 0
4423  7410            SKP
4424  4577            AMBO                    /A MINUS B TO 0


4425  4562    ATX3,   FSTEP                   /STEP TO STATE 3
4426  1022            TAD     OMSW
4427  7700            SMA CLA                 /00=1?
4430  5234            JMP     .+4             /NO
4431  3036            DCA     BMSW            /YES=0 TO B
4432  3037            DCA     BLSW
4433  5236            JMP     .+3
4434  4545            LOADBI  FACFR           /FAC FRAC TO B
4435  0053
4436  4543            LOADOI  BREG            /B TO 0
4437  0036
4440  4523            SOEZ                    /0=0?
4441  5244            JMP     ATX4            /NO=GO TO STATE 4
4442  3776/           DCA     SHREG           /YES=0 TO SHFT CNTR
4443  5244            JMP     ATX4            /GO TO STATE 4
```

```
4444  4562     ATX4,   FSTEP                  /STEP TO STATE 4
4445  1775/            TAD     FLAG1         /GET SHIFT O FF
4446  7640             SZA CLA                /SHIFT O FF =1?
4447  5270             JMP     ATX5-1         /YES
4450  1776/            TAD     SHREG
4451  7041             CIA                    /NEGATE SHIFT CNTR FOR ISZ
4452  3776/            DCA     SHREG
4453  4526             SHFTB                  /SHIFT B UNTIL S'C'=0
4454  1053             TAD     FACFR          /GET FAC FRACTION
4455  7710             SPA CLA                /0 = 0 & FAC FRAC NEG ?
4456  4523             SOEZ                   /0 = 0 ?
4457  5265             JMP     ATX4A          /NO - B TO 0
4460  7240             STA
4461  3022             DCA     OMSW           /LOGICAL 1 TO OMSW
4462  7240             STA
4463  3023             DCA     OLSW           /LOGICAL 1 TO OLSW
4464  5271             JMP     ATX5           /GO TO STATE 5
4465  4543     ATX4A,  LOADOI  BREG           /B TO 0 (STATE 4-1)
4466  0036
4467  7410             SKP
4470  4525             SHFTO                  /SHIFT O UNTIL S'C'=0

4471  4562     ATX5,   FSTEP                  /STEP TO STATE 5
4472  1023             TAD     OLSW           /SIMULATE INBRK USING
4473  3105             DCA     EMEM           /PROGRAM MEMORY
4474  1023             TAD     OLSW
4475  4536             PUTX                   /PUT OLSW IN PROG INDEX REG
4476  4567             CMEME                  /COMPARE MEMORY
4477  5774/            JMP     FETCH          /GO TO FETCH
```

```
4500  4563    FXTA,   ENTER                   /ENTER XTA
4501  4562    XTA1,   FSTEP                   /STEP TO STATE 1
4502  3030            DCA     AMSW            /0 TO AMSW
4503  1005            TAD     PIR             /GET INSTRUCTION
4504  0373            AND     (7
4505  3031            DCA     ALSW            /PIR9-11 TO ALSW
4506  4545            LOADB!  X0ADR           /X0 ADDR TO B
4507  0076
4510  4576            APBO                    /A PLUS B TO O
4511  4542            LOADOP! OREG            /0 TO OP ADDR
4512  0022

4513  4562    XTA2,   FSTEP                   /STEP TO STATE 2
4514  1005            TAD     PIR             /GET INSTRUCTION
4515  0373            AND     (7
4516  1372            TAD     (X0             /COMPUTE INDEX REG ADDR
4517  3000            DCA     T1              /GET PROG INDEX REG INSTEAD
4520  1400            TAD I   T1              /OF OUTBRK
4521  4521            STORB                   /MB TO BLSW - SIGN EXTEND TO BMSW
4522  4543            LOADO!  BREG            /B TO O
4523  0036
4524  3776/           DCA     SHREG           /0 TO SHIFT COUNTER
4525  1002            TAD     COMREG          /GET COMMAND REGISTER
4526  7710            SPA CLA                 /D, P, MODE ?
4527  5334            JMP     .+5             /YES - LEAVE EXPONENT ALONE
4530  1115            TAD     (27
4531  3052            DCA     ACEXP           /27 TO FAC EXP
4532  1052            TAD     ACEXP
4533  3045            DCA     MQLSW           /FAC EXP TO MQ LSW
4534  7240            STA                     /1 TO MULTIPLY - DIVIDE FLAG
4535  3771/           DCA     MDFLAG          /FOR USE IN DEPOSIT
4536  5770/           JMP     DEP             /GO TO DEPOSIT
```

```
4537  4563  JXN,    ENTER                 /ENTER JXN
4540  4562  JXN1,   FSTEP                 /STEP TO STATE 1
4541  1005           TAD     PIR          /GET INSTRUCTION
4542  0367           AND     (70          /EXTRACT BITS 6-8
4543  7110           CLL RAR
4544  7012           RTR
4545  3031           DCA     ALSW         /FIR6-8 TO ALSW
4546  3030           DCA     AMSW
4547  4545           LOADBI  X0ADR        /X0 ADDRESS TO B
4550  0076
4551  4576           APB0                 /A PLUS B TO 0
4552  4542           LOADOPI OREG         /O TO OP ADDR
4553  0022
4554  7040           CMA
4555  3022           DCA     OMSW         /LOGICAL 1 TO 0
4556  7040           CMA
4557  3023           DCA     OLSW
4560  5766           JMP     JXN2


4566  4600
4567  0070
4570  2052
4571  0737
4572  0200
4573  0007
4574  1261
4575  0726
4576  0724
4577  0027
      4600           PAGE
```

```
4600  4562  JXN2,   FSTEP                  /STEP TO STATE 2
4601  1005           TAD      PIR         /GET INSTRUCTION
4602  0377           AND      (100        /EXTRACT BIT 5
4603  7640           SZA CLA              /FIR 5=1?
4604  5213           JMP      JXN2A       /YES
4605  4556           GETX                 /NO-OUTBRK USING OP ADDR
4606  3031           DCA      ALSW        /MB TO ALSW
4607  3030           DCA      AMSW        /0 TO AMSW
4610  4543           LOADO)   AREG        /A TO 0
4611  0030
4612  5217           JMP      .+5
4613  4550  JXN2A,   INCX                 /REQUEST INC BRK (X=X+1)
4614  4556           GETX                 /GET X REG
4615  7650           SNA CLA              /OVERFLOW?
4616  4570           CLRO                 /YES - LOGICAL 0 TO 0
4617  4523           SOEZ                 /SKIP IF 0=0
4620  5223           JMP      JXN3        /0 NOT EQUAL 0
4621  4552           INCPC                /INC FPC
4622  5237           JMP      JXNEND      /GO TO FETCH

4623  4562  JXN3,    FSTEP                /STEP TO STATE 3
4624  4557           GETPC                /OUTBRK USING FPC
4625  3031           DCA      ALSW        /MB TO ALSW
4626  1005           TAD      PIR         /GET INSTRUCTION
4627  0376           AND      (7
4630  3030           DCA      AMSW        /FIR9-11 TO AMSW
4631  4543           LOADO)   AREG        /A TO 0
4632  0030
4633  4542           LOADOP)  OREG        /0 TO OP ADDR
4634  0022
4635  4541           LOADPC)  OPADR       /OP ADDR TO FPC
4636  0070

4637  5775'  JXNEND, JMP      FETCH       /GO TO FETCH
```

```
4640  4563  JSR,   ENTER                 /ENTER JSR
4641  4562  JSR1,  FSTEP                 /STEP TO STATE 1
4642  4545         LOADB;  PBASE          /P0 ADDR TO B
4643  0074
4644  4543         LOADO;  BREG           /B TO O
4645  0036
4646  4542         LOADOP; OREG           /O TO OP ADDR
4647  0022
4650  4554         INCOP                  /INC OP ADDR

4651  4562  JSR2,  FSTEP                 /STEP TO STATE 2
4652  4557         GETPC                  /OUTBRK USING FPC
4653  3031         DCA     ALSW           /MB TO ALSW
4654  1005         TAD     PIR            /GET INSTRUCTION
4655  0376         AND     (7
4656  3030         DCA     AMSW           /FIR9-11 TO AMSW
4657  4552         INCPC                  /INC FPC
4660  4543         LOADO;  AREG           /A TO O
4661  0030

4662  4562  JSR3,  FSTEP                 /STEP TO STATE 3
4663  1066         TAD     PFPC           /GET FPC FIELD BITS
4664  0376         AND     (7
4665  1374         TAD     (1030          /SEE JMK STATE 3-1
4666  3106         DCA     MMEM           /SIMULATE INBRK
4667  4554         INCOP                  /INC OP ADDR

4670  4562  JSR4,  FSTEP                 /STEP TO STATE 4
4671  1067         TAD     PFPC+1         /SIMULATE INBRK
4672  3107         DCA     LMEM
4673  4542         LOADOP; OREG           /O TO OP ADDR
4674  0022
4675  4541         LOADPC; OPADR          /OP ADR TO FPC
4676  0070
4677  5775;        JMP     FETCH          /GO TO FETCH
```

```
4700  4563  JSA,   ENTER              /ENTER JSA
4701  4562  JSA2,  FSTEP              /STEP TO STATE 2
4702  4557         GETPC              /OUTBRK USING FPC
4703  3031         DCA    ALSW        /MB TO ALSW
4704  1005         TAD    PIR         /GET INSTRUCTION
4705  0376         AND    (7
4706  3030         DCA    AMSW        /FIR9-11 TO AMSW
4707  4552         INCPC              /INC FPC
4710  4543         LOADO) AREG        /A TO O
4711  0030
4712  4542         LOADOP) OREG       /O TO OP ADDR
4713  0022

4714  4562  JSA3,  FSTEP              /STEP TO STATE 3
4715  1066         TAD    PFPC        /SEE JSB STATE 3-1
4716  0376         AND    (7          /FPC F,B,
4717  1374         TAD    (1030       /JA= 1030 = JUMP ALWAYS
4720  3106         DCA    MMEM        /SIMULATE INBRK TO PROG MEMORY
4721  4554         INCOP              /INC OP ADDR

4722  4562  JSA4,  FSTEP              /STEP TO STATE 4
4723  1067         TAD    PFPC+1      /GET FPC
4724  3107         DCA    LMEM        /SIMULATE INBRK
4725  4554         INCOP              /INC OP ADDR (STATE 4-2)
4726  4541         LOADPC) OPADR      /OP ADDR TO FPC
4727  0070
4730  5775'        JMP    FETCH       /GO TO FETCH
```

```
4731  4534  JEQ,   SAEZ                  /FAC=0?
4732  5773'        JMP    JFALSE         /NO
4733  5772'        JMP    JTRUE          /YES

4734  4532  JGE,   SALZ                  /FAC GT OR=0?
4735  5772'        JMP    JTRUE          /YES
4736  5773'        JMP    JFALSE         /NO

4737  4533  JLE,   SAGZ                  /FAC LT OR=0?
4740  5772'        JMP    JTRUE          /YES
4741  5773'        JMP    JFALSE         /NO

4742  4534  JNE,   SAEZ                  /FAC=0?
4743  5772'        JMP    JTRUE          /NO
4744  5773'        JMP    JFALSE         /YES

4745  4532  JLT,   SALZ                  /FAC LT 0?
4746  5773'        JMP    JFALSE         /NO
4747  5772'        JMP    JTRUE          /YES

4750  4533  JGT,   SAGZ                  /FAC GT 0?
4751  5773'        JMP    JFALSE         /NO
4752  5772'        JMP    JTRUE          /YES

4753  4563  SETB,  ENTER                 /ENTER SET BASE
4754  4562  SETB1, FSTEP                 /STEP TO STATE 1
4755  4557         GETPC                 /OUTBRK USING FPC
4756  3075         DCA    PBASE+1        /MB TO P0 ADDR
4757  1005         TAD    PIR            /GET INSTRUCTION
4760  0376         AND    (7
4761  3074         DCA    PBASE          /PIR9-11 TO P0 F.B.
4762  4552         INCPC                 /INC FPC
4763  5775'        JMP    FETCH          /GO TO FETCH

4772  5005
4773  5024
4774  1030
4775  1261
4776  0007
4777  0100
      5000         PAGE
```

```
5000  7300  JAL,    CLA CLL
5001  1022          TAD     OMSW        /GET O SIGN
5002  7700          SMA CLA             /IS O NEG
5003  5224          JMP     JFALSE      /NO
5004  5205          JMP     JTRUE       /YES

5005  4563  JTRUE,  ENTER               /ENTER JMP CONDITION TRUE
5006  4562  JTRUE1, FSTEP               /STEP TO STATE 1
5007  4527          SEPMEN              /1 TO EPM ENABLE
5010  4557          GETPC               /OUTBRK USING PFC
5011  3031          DCA     ALSW        /MB TO ALSW
5012  1005          TAD     PIR         /GET INSTRUCTION
5013  0377          AND     (7
5014  3030          DCA     AMSW        /FIR9-11 TO AMSW
5015  4543          LOADO;  AREG        /A TO O
5016  0030
5017  4542          LOADOP; OREG        /O TO OP ADDR
5020  0022
5021  4541          LOADPC; OPADR       /OP ADDR TO FPC
5022  0070
5023  5776'         JMP     FETCH       /GO TO FETCH

5024  4563  JFALSE, ENTER               /ENTER JMP CONDITION FALSE
5025  4552          INCPC               /INC FPC
5026  5776'         JMP     FETCH       /GO TO FETCH

5027  4563  JAC,    ENTER               /ENTER JAC
5030  4562  JAC1,   FSTEP              /STEP TO STATE 1
5031  4545          LOADB;  FACFR       /FAC FRAC TO B
5032  0053
5033  4543          LOADO;  BREG        /B TO O
5034  0036
5035  4542          LOADOP; OREG        /O TO OP ADDR
5036  0022
5037  4541          LOADPC; OPADR       /OP ADDR TO FPC
5040  0070
5041  5776'         JMP     FETCH       /GO TO FETCH
```

```
5042  4563   FSTF,   ENTER                  /ENTER STF
5043  4530           SEPM                   /EPM MODE ?
5044  7610           SKP CLA                /NO = 0 TO ROUND FLIP-FLOP
5045  7240           STA                    /YES = 1 TO ROUND FLIP-FLOP
5046  3321           DCA     RNDFF          /0 OR 1 TO ROUND FLIP-FLOP
5047  3003           DCA     EPM            /CLEAR EPM MODE
5050  1002           TAD     COMREG
5051  0375           AND     (3777          /RESET D,P, MODE FF
5052  3002           DCA     COMREG
5053  1321           TAD     RNDFF          /GET ROUND FF
5054  7650           SNA CLA                /IS IT SET ?
5055  5776'          JMP     FETCH          /NO = GO TO FETCH
5056  5300           JMP     FNORM          /YES = GO TO NORMALIZE

5057  4563   FSTD,   ENTER                  /ENTER STD
5060  1002           TAD     COMREG
5061  7104           CLL RAL
5062  7130           STL RAR                /SET D,P, MODE FF
5063  3002           DCA     COMREG
5064  3003           DCA     EPM            /CLEAR EPM MODE
5065  5776'          JMP     FETCH          /GO TO FETCH

5066  4563   FNEG,   ENTER                  /ENTER NEG
5067  4562   NEG1,   FSTEP                 /STEP TO STATE 1
5070  4527           SEPMEN                 /SET EPM ENABLE
5071  4545           LOADBI  FACFR          /FAC TO B
5072  0053
5073  4574           CLRA                   /0 TO A
5074  4577           AMBO                   /A-B TO O
5075  4546           LOADACI OREG           /O TO FAC
5076  0022
5077  5776'          JMP     FETCH          /AS THE FLOWS BUT TAKES LESS CORE

5100  4563   FNORM,  ENTER                  /ENTER FNORM
5101  4562   NORM1,  FSTEP                 /STEP TO STATE 1
5102  4527           SEPMEN                 /1 TO EPM ENABLE
5103  4545           LOADBI  FACFR          /FAC FRAC TO B
5104  0053
5105  1321           TAD     RNDFF          /GET ROUND FLIP-FLOP
5106  7640           SZA CLA                /ROUND F/F = 1 ?
5107  1055           TAD     ACLSW1         /YES = GET AC24
5110  0374           AND     (4000          /DELETE OTHER AC BITS
5111  3043           DCA     BEXT           /AC24 TO B EXT
5112  3321           DCA     RNDFF          /CLEAR ROUND FLIP-FLOP
5113  4543           LOADOI  BREG           /B TO O
5114  0036
5115  1052           TAD     ACEXP          /FAC EXP TO MQLSW
5116  3045           DCA     MQLSW
5117  3773'          DCA     SHREG          /0 TO SHIFT CNTR
5120  5772'          JMP     DEP            /GO TO DEPOSIT

5121  0000   RNDFF,  0                      /ROUND FLIP-FLOP FOR START F
```

```
5122  4563   FPAUSE, ENTER                  /ENTER PAUSE
5123  4562   PAUS0,  FSTEP                  /STEP TO STATE 1
5124  4771'         JMS     TYP             /TYPE FPP PAUSE
5125  2315           TPAUSE
5126  0000           0
5127  1770'          TAD     STEPSW          /SAVE SINGLE STEP SWITCH
5130  3000           DCA     T1
5131  7240           STA
5132  3770'          DCA     STEPSW          /SET SINGLE STEP SW
5133  1767'          TAD     CSTATE
5134  0366           AND     (377            /RESET PROG MAJOR STATE
5135  3767'          DCA     CSTATE

5136  4562   PAUS1,  FSTEP                  /WAIT FOR OPERATOR
5137  1000           TAD     T1              /RESTORE SINGLE STEP SWITCH
5140  3770'          DCA     STEPSW
5141  6555           FPST                    /RESTART FPP
5142  7000           NOP
5143  5776'          JMP     FETCH           /GO TO FETCH

5144  4563   FCLA,   ENTER                  /ENTER CLA
5145  4574           CLRA                    /0 TO A
5146  4546           LOADAC' AREG            /A TO FAC
5147  0030
5150  1002           TAD     COMREG          /GET THE COMMAND REGISTER
5151  7700           SMA CLA                 /D.P. MODE ?
5152  3052           DCA     ACEXP           /NO - CLEAR FAC EXPONENT
5153  5776'          JMP     FETCH           /GO TO FETCH



5154  4563   SETX,   ENTER                  /ENTER SET X
5155  4562   SETX1,  FSTEP                  /STEP TO STATE 1
5156  4557           GETPC                   /OUTBRK USING FPC
5157  3077           DCA     X0ADR+1         /MB TO X0 ADDR
5160  1005           TAD     PIR             /GET INSTRUCTION
5161  0377           AND     (7
5162  3076           DCA     X0ADR           /FIR9-11 TO X0 F.B.
5163  4552           INCPC                   /INC FPC
5164  4540           MOVEX                   /SET PROGRAM INDEX REGS
5165  5776'          JMP     FETCH           /GO TO FETCH


5166  0377
5167  0731
5170  7732
5171  6703
5172  2052
5173  0724
5174  4000
5175  3777
5176  1261
5177  0007
```

5200 PAGE

```
5200  4563    ADDX,   ENTER                   /ENTER ADDX
5201  4562    ADDX1,  FSTEP                   /STEP TO STATE 1
5202  3030            DCA     AMSW            /0 TO AMSW
5203  1005            TAD     PIR             /GET INSTRUCTION
5204  0377            AND     (7
5205  3031            DCA     ALSW            /FIR9-11 TO ALSW
5206  4545            LOADB!  X0ADR           /X0 ADDR TO B
5207  0076
5210  4576            APB0                    /A PLUS B TO 0
5211  4542            LOADOP! OREG            /0 TO OP ADDR
5212  0022

5213  4562    ADDX2,  FSTEP                   /STEP TO STATE 2
5214  1005            TAD     PIR
5215  0377            AND     (7              /GET PROGRAM INDEX REG INSTEAD
5216  1006            TAD     PXP             /OF OUTBRK USING OP ADDR
5217  3000            DCA     T1
5220  1400            TAD I   T1
5221  3037            DCA     BLSW            /MB TO BLSW
5222  3036            DCA     BMSW            /0 TO BMSW

5223  4562    ADDX3,  FSTEP                   /STEP TO STATE 3
5224  4557            GETPC                   /OUTBRK USING FPC
5225  3031            DCA     ALSW            /BM TO ALSW
5226  3030            DCA     AMSW            /0 TO AMSW
5227  4552            INCPC                   /INC PFC
5230  4576            APB0                    /A PLUS B TO 0

5231  4562    ADDX4,  FSTEP                   /STEP TO STATE 4
5232  1023            TAD     OLSW
5233  4536            PUTX                    /O LSW TO X REG
5234  5776!           JMP     FETCH           /GO TO FETCH
```

```
5235  4563     FSTA,   ENTER                   /TYPE ENTER FSTA
5236  4562     STA0,   FSTEP                   /STEP TO MAJOR STATE 0
5237  1002             TAD     COMREG         /GET COMMAND REGISTER
5240  7710             SPA CLA                /D, P, MODE ?
5241  5246             JMP     STA1           /YES - GO TO STATE 1
5242  1052             TAD     ACEXP          /INBRK USING OP ADDR
5243  3105             DCA     EMEM           /FAC EXP TO MB
5244  4567             CMEME                  /COMPARE MEMORY EXPONENT
5245  4554             INCOP                  /INC OP ADDR

5246  4562     STA1,   FSTEP                   /STEP TO MAJOR STATE 1
5247  1053             TAD     ACMSW          /INBRK USING OP ADDR
5250  3106             DCA     MMEM           /FAC MSW TO MB
5251  4554             INCOP                  /INC OF ADDR

5252  4562     STA2,   FSTEP                   /STEP TO MAJOR STATE 2
5253  1054             TAD     ACLSW          /INBRK USING OP ADDR
5254  3107             DCA     LMEM           /FAC LSW TO MB
5255  4530             SEPM                   /EXTENDED PRECISION MODE?
5256  5300             JMP     STAEND         /NO - END OF FSTA
5257  4775'            JMS     SEX            /YES - SET TMSC EXECUTE
5260  4554             INCOP                  /INC OP ADDRESS

5261  4562     STA20,  FSTEP                   /STEP TO TMSC EX STATE 0
5262  1055             TAD     ACLSW1         /GET FAC LSW1
5263  3110             DCA     LMEM1          /FAC LSW1 TO MB
5264  4554             INCOP                  /INC OP ADDRESS

5265  4562     STA21,  FSTEP                   /STEP TO TMSC EX STATE 1
5266  1056             TAD     ACLSW2         /GET FAC LSW2
5267  3111             DCA     LMEM2          /FAC LSW2 TO MB
5270  4554             INCOP                  /INC OP ADDRESS

5271  4562     STA22,  FSTEP                   /STEP TO TMSC EX STATE 2
5272  1057             TAD     ACLSW3         /GET FAC LSW3
5273  3112             DCA     LMEM3          /FAC LSW3 TO MB
5274  4564             DECOP
5275  4564             DECOP                  /DEC OP ADDRESS
5276  4564             DECOP
5277  4774'            JMS     REX            /0 TO TMSC EXECUTE

5300  4564     STAEND, DECOP                   /DEC OP ADDR FOR COMPARE
5301  4566             CMEMF                  /COMPARE MEMORY
5302  4554             INCOP                  /RESTORE OP ADDR
5303  5776'            JMP     FETCH          /GO TO FETCH
```

```
5304  4563    FLDA,   ENTER                   /TYPE ENTER LDA
5305  4562    LDA0,   FSTEP                   /STEP TO  STATE 0
5306  1002            TAD     COMREG          /GET COMMAND REGISTER
5307  7710            SPA CLA                 /D, P, MODE ?
5310  5314            JMP     LDA1            /YES = GO TO STATE 1
5311  4560            GETOP                   /OUTBRK USING OP ADDR
5312  3052            DCA     ACEXP           /MB TO FAC EXP
5313  4554            INCOP

5314  4562    LDA1,   FSTEP                   /STEP TO MAJOR STATE 1
5315  4560            GETOP                   /OUTBRK USING OP ADDR
5316  3030            DCA     AMSW
5317  4554            INCOP
5320  4530            SEPM                    /EXTENDED PRECISION MODE?
5321  5344            JMP     LDA2            /NO GO TO STATE 2
5322  4775'           JMS     SEX             /1 TO TMSC EXECUTE
5323  4554            INCOP                   /INC OP ADDRESS
5324  4527            SEPMEN                  /1 TO EPM ENABLE

5325  4562    LDA10,  FSTEP                   /STEP TO TMSC EX STATE 0
5326  4560            GETOP                   /OUTBRK USING OP ADDRESS
5327  3032            DCA     ALSW1           /MB TO ALSW1
5330  4554            INCOP                   /INC OP ADDRESS

5331  4562    LDA11,  FSTEP                   /STEP TO TMSC EX STATE 1
5332  4560            GETOP                   /OUTBRK USING OP ADDRESS
5333  3033            DCA     ALSW2           /MB TO ALSW2
5334  4554            INCOP                   /INC OP ADDRESS

5335  4562    LDA12,  FSTEP                   /STEP TO TMSC EX STATE 2
5336  4560            GETOP                   /OUTBRK USING OP ADDRESS
5337  3034            DCA     ALSW3           /MB TO ALSW3
5340  4564            DECOP                   /DEC OP ADDRESS
5341  4564            DECOP                   /DEC OP ADDRESS
5342  4564            DECOP                   /DEC OP ADDRESS
5343  4774'           JMS     REX             /0 TO TMSC EXECUTE


5344  4562    LDA2,   FSTEP                   /STEP TO MAJOR STATE 2
5345  4560            GETOP                   /OUTBRK USING OP ADDR
5346  3031            DCA     ALSW
5347  4543            LOADOI  AREG            /A TO 0
5350  0030
5351  4546            LOADACI OREG            /0 TO FAC FRAC
5352  0022
5353  5776'           JMP     FETCH
```

```
5354  4563    FSTE,   ENTER                   /SET EPM MODE
5355  7324            CLA STL RAL             /AC=0001
5356  3003            DCA     EPM             /1 TO EPM
5357  1002            TAD     COMREG          /GET THE COMMAND REGISTER
5360  0373            AND     (3777           /DELETE D.P. BIT
5361  3002            DCA     COMREG          /RESTORE THE COMMAND REGISTER
5362  4772'           JMS     TITLE           /SYNC THE 0 REGISTERS
5363  5776'           JMP     FETCH           /GO TO FETCH

5372  7367
5373  3777
5374  7507
5375  7500
5376  1261
5377  0007
      5400            PAGE
```

```
                / MINI ROUTINES (TR3)

5400  0000  INC2,  0                    /INCREMENT FPC
5401  7300         CLA CLL
5402  2067         ISZ     PFPC+1       /FPC ADDR
5403  7410         SKP
5404  2066         ISZ     PFPC         /FPC FIELD
5405  7000         NOP
5406  5600         JMP I   INC2         /RETURN

5407  0000  INC3,  0                    /INCREMENT ADRS
5410  7300         CLA CLL
5411  2073         ISZ     PAPT+1       /APT ADDRESS
5412  7410         SKP
5413  2072         ISZ     PAPT         /APT FIELD
5414  7000         NOP
5415  5607         JMP I   INC3         /RETURN


5416  0000  INC1,  0                    /INCREMENT OP ADDR
5417  7300         CLA CLL
5420  2071         ISZ     OPADR+1      /OP ADDR
5421  7410         SKP
5422  2070         ISZ     OPADR        /OP FLD
5423  7000         NOP
5424  5616         JMP I   INC1         /RETURN

5425  0000  APTC,  0                    /COMPARE APT ADDRESS
5426  7200         CLA
5427  6565         RAPT                 /GET APT ADDR FROM FPP
5430  3261         DCA     SAVAPT       /SAVE FOR ERROR REPORT
5431  1261         TAD     SAVAPT       /RESTORE AC
5432  7041         CIA
5433  1073         TAD     PAPT+1       /COMPARE WITH PROGRAM ADDR
5434  7650         SNA CLA              /APT ADDR OK?
5435  5625         JMP I   APTC         /RETURN
5436  1225         TAD     APTC         /GET RETURN ADDRESS
5437  4777'        JMS     ERR          /TYPE ERROR DATA
5440  0005         ERROR   0005         /ADDRS ERROR CODE
5441  5625         JMP I   APTC         /RETURN TO MAIN PROGRAM
```

```
5442  0000  SPECFL, 0                       /SPECIAL OVERFLOW CONDITION
5443  1022         TAD     OMSW            /GET O MSW
5444  3261         DCA     SAVAPT          /SAVE IT
5445  1022         TAD     OMSW            /GET O MSW AGAIN
5446  7004         RAL
5447  7110         CLL RAR                 /DELETE THE SIGN BIT
5450  3022         DCA     OMSW            /PUT OTHER BITS BACK
5451  4523         SOEZ                    /IS O=0?
5452  5255         JMP     .+3             /NO - OVERFLOW IS IMPOSSIBLE
5453  1261         TAD     SAVAPT          /GET ORIGINAL OMSW
5454  7700         SMA CLA                 /DOES 0=40000000,,,0000?
5455  2242         ISZ     SPECFL          /NO - RETURN+1
5456  1261         TAD     SAVAPT          /GET ORIGINAL OMSW
5457  3022         DCA     OMSW            /RESTORE O REG
5460  5642         JMP I   SPECFL          /RETURN

5461  0000  SAVAPT, 0

5462  1776' TREXIT, TAD     ASCCH           /GET ASCII CHARACTER
5463  0375         AND     (77             /STRIP TO 6 BITS
5464  1374         TAD     (-3
5465  7450         SNA                     /CNTRL C ?
5466  5310         JMP     DTAM            /YES - RETURN TO DTA MONITOR
5467  1373         TAD     (-1
5470  7450         SNA                     /CNTRL D ?
5471  5276         JMP     .+5             /YES - SETUP TO READ IN DIAL
5472  1372         TAD     (-14
5473  7650         SNA CLA                 /CNTRL P ?
5474  5311         JMP     PS8M            /YES - RETURN TO PS-8 MONITOR
5475  5771' ,      JMP     DOTERR          /ILLEGAL INPUT
5476  1305         TAD     ER1             /MOVE DIAL LOADER TO
5477  3770' ,      DCA     4015            /4015 FOR TAPE READ
5500  1306         TAD     ER2             /INTO PROPER LOCATION
5501  3767' ,      DCA     4016            /AND AUTO START
5502  1307         TAD     ER3
5503  3766' ,      DCA     4017
5504  5770' ,      JMP     4015            /READ DIAL TAPE AND EXECUTE

5505  6141  ER1,    6141                    /LINC
5506  0701  ER2,    0701                    /RCG
5507  7300  ER3,    7300                    /BLOCK NUMBER

5510  1365  DTAM,   TAD     (73
5511  1364  PS8M,   TAD     (7605
5512  6212         CIF     10
5513  6211         CDF     10
5514  5715         JMP I   .+1
5515  1502         RESPS8
```

```
                        /CHECK MAJOR STATE REGISTER

5516  3000    MSTATE, 0                       /GET PROGRAM STATE GENERATOR
5517  7300            CLA CLL
5520  1763'           TAD     CSTATE
5521  0362            AND     (7774           /DELETE BITS 10 & 11
5522  3761'           DCA     MT1             /SAVE FOR ERROR
5523  6562            RSTATE                  /READ STATE FROM FPP
5524  0362            AND     (7774           /DELETE BITS 10 & 11
5525  3760'           DCA     MT2             /SAVE FOR ERROR
5526  1760'           TAD     MT2
5527  7041            CIA
5530  1761'           TAD     MT1             /COMPARE WITH PROGRAM STATE
5531  7650            SNA CLA                 /IS MAJOR STATE OK?
5532  2316            ISZ     MSTATE          /YES-RETURN+1
5533  5716            JMP I   MSTATE

5534  0000    TOPC,   0                       /LOAD THE FPC
5535  7200            CLA
5536  1334            TAD     TOPC            /GET RETURN -1
5537  4757'           JMS     LOAD
5540  0066            PFPC                    /ADDRESS OF FPC
5541  0000    TOAC,   0                       /LOAD THE FAC
5542  7200            CLA
5543  1341            TAD     TOAC            /GET RETURN-1
5544  4757'           JMS     LOAD
5545  0053            FACFR                   /ADDR OF FAC

5557  5636
5560  0736
5561  0735
5562  7774
5563  0731
5564  7605
5565  0073
5566  4017
5567  4016
5570  4015
5571  0432
5572  7764
5573  7777
5574  7775
5575  0077
5576  0351
5577  6707
      5600            PAGE
```

```
5600 0000 TOO,   0                    /LOAD THE OREG
5601 7200        CLA
5602 1200        TAD    TOO           /GET RETURN-1
5603 4236        JMS    LOAD
5604 0022        OMSW                 /ADDR OF O
5605 0000 TOA,   0                    /LOAD THE A REG
5606 7200        CLA
5607 1205        TAD    TOA           /GET RETURN-1
5610 4236        JMS    LOAD
5611 0030        AMSW                 /ADDR OF A

5612 0000 TOB,   0                    /LOAD THE A REG
5613 7200        CLA
5614 1212        TAD    TOB           /GET RETURN-1
5615 4236        JMS    LOAD
5616 0036        BMSW                 /ADDR OF B

5617 0000 TOTMP, 0
5620 7200        CLA
5621 1217        TAD    TOTMP
5622 4236        JMS    LOAD
5623 0060        TREG

5624 0000 TOMQ,  0
5625 7200        CLA
5626 1224        TAD    TOMQ
5627 4236        JMS    LOAD
5630 0044        MQREG                /ADDR OF MQ

5631 0000 TOOP,  0                    /LOAD OP ADDR
5632 7200        CLA
5633 1231        TAD    TOOP          /GET RETURN-1
5634 4236        JMS    LOAD
5635 0070        OPADR                /ADDR OF OP ADDR
```

```
5636  0000  LOAD,  0
5637  3000        DCA     T1              /SAVE RETURN ADDRESS
5640  3321        DCA     LREG            /CLEAR LONG REG FLAG
5641  1400        TAD I   T1              /GET "FROM" REG
5642  7041        CIA
5643  1377        TAD     (PFPC           /COMPARE WITH SHORT REG ADDR
5644  7710        SPA CLA                 /ARITHMETIC REG?
5645  5257        JMP     SHORT           /NO
5646  1636        TAD I   LOAD            /GET "TO" REG
5647  7041        CIA
5650  1377        TAD     (PFPC           /COMPARE WITH SHORT REG ADDR
5651  7710        SPA CLA                 /ARITHMETIC REG?
5652  5257        JMP     SHORT           /NO
5653  7240        STA
5654  3321        DCA     LREG            /SET LONG REG FLAG
5655  1003        TAD     EPM             /GET EPM CONTROL REGISTER
5656  7700        SMA CLA                 /EPM ENABLE = 1 ?
5657  1376  SHORT, TAD     (3              /NO = MOVE 2 WORDS
5660  1375        TAD     (-5             /YES = MOVE 5 WORDS
5661  3001        DCA     T2              /SET WORD COUNT
5662  7240        STA
5663  1400        TAD I   T1              /GET "FROM" REG
5664  3010        DCA     10
5665  2000        ISZ     T1              /INC RETURN
5666  7240        STA
5667  1636        TAD I   LOAD            /GET "TO" REG
5670  3011        DCA     11
5671  1410        TAD I   10              /GET "FROM" DATA
5672  3411        DCA I   11              /STORE IN "TO" REG
5673  2001        ISZ     T2              /INC WORD COUNT
5674  5271        JMP     .-3             /MORE TO GO
5675  1321        TAD     LREG            /GET LONG REG FLAG
5676  7650        SNA CLA                 /ARITHMETIC REG?
5677  5774        JMP     LTRUNK          /NO = FINISHED
5700  1003        TAD     EPM             /GET EPM CONTROL REGISTER
5701  7710        SPA CLA                 /EPM ENABLE = 1 ?
5702  5317        JMP     LEXT            /YES = CLEAR EXT
5703  2010        ISZ     10
5704  2011        ISZ     11
5705  2010        ISZ     10
5706  2011        ISZ     11
5707  2010        ISZ     10
5710  2011        ISZ     11
5711  1011        TAD     11              /GET "TO" REG
5712  7041        CIA
5713  1373        TAD     (OLSW3          /IS IT THE 0 REG?
5714  7650        SNA CLA
5715  1410        TAD I   10              /YES = GET "FROM" EXT
5716  0372        AND     (7400
5717  3411  LEXT, DCA I   11              /LOAD OR CLEAR EXT
5720  5400        JMP I   T1              /RETURN

5721  0000  LREG,  0
```

```
5722  0000  COMPO,  0                    /COMPARE FPP 0 REG
5723  7200          CLA
5724  1371          TAD     (SAVOM-1             /GET ADDR OF SAVE BUFFER
5725  3010          DCA     10           /STORE IN AUTO INDEX REG
5726  6563          RQMSW                /GET MSW
5727  3410          DCA I   10           /STORE IN BUFFER
5730  6564          RQLSW                /GET LSW
5731  3410          DCA I   10
5732  1370          TAD     (400
5733  6567          RQEPM                /READ OLSW1
5734  3410          DCA I   10
5735  1367          TAD     (200
5736  6567          RQEPM                /READ OLSW2
5737  3410          DCA I   10
5740  1366          TAD     (100
5741  6567          RQEPM                /READ OLSW3
5742  3410          DCA I   10
5743  1365          TAD     (OREG-1      /GET ADDRESS OF PROGRAM 0 REG
5744  3010          DCA     10
5745  1371          TAD     (SAVOM-1             /GET ADDRESS OF FPP 0 REG
5746  3011          DCA     11
5747  4530          SEPM                 /EPM MODE?
5750  1376          TAD     (3           /NO - 2 WORD REGISTER
5751  1375          TAD     (-5          /YES - 5 WORD REGISTER
5752  3000          DCA     T1           /SAVE WORD COUNT
5753  1410          TAD I   10           /GET PROGRAM DATA
5754  7041          CIA
5755  1411          TAD I   11           /COMPARE WITH FPP DATA
5756  7640          SZA CLA              /IS FPP DATA CORRECT?
5757  5764          JMP     ORERR        /NO - 0 REG ERROR
5760  2000          ISZ     T1           /FINISHED?
5761  5353          JMP     .-6          /NO
5762  5722  COMPOR, JMP I   COMPO

5764  6000
5765  0021
5766  0100
5767  0200
5770  0400
5771  0077
5772  7400
5773  0026
5774  6232
5775  7773
5776  0003
5777  0066
      6000          PAGE
```

```
6000  1777'  ORERR,  TAD     7600          /GET P,C,
6001  4776'          JMS     ERR           /GO TO ERROR ROUTINE
6002  0021           ERROR   0021          /ERROR CODE
6003  5775'          JMP     COMPOR        /RETURN

6004  0000   SAVO,   0                     /SAVE FPP 0 REG HERE
6005  0000           0
6006  0000           0
6007  0000           0
6010  0000           0

6011  0000   APTPAC, 0
6012  1070           TAD     OPADR         /FIELD BITS OF OP ADDR
6013  7104           RAL CLL
6014  7006           RTL
6015  1074           TAD     PBASE         /FIELD BITS OF P0 ADDR
6016  7104           RAL CLL
6017  7006           RTL
6020  1076           TAD     X0ADR         /FIELD BITS OF X0 ADDR
6021  7104           RAL CLL
6022  7006           RTL
6023  1066           TAD     PFPC          /FIELD BITS OF FPC
6024  5611           JMP I   APTPAC        /RETURN

6025  0000   STRA,   0
6026  7100           CLL
6027  7510           SPA                   /CHECK SIGN
6030  7120           STL                   /SAVE SIGN IN LINK
6031  3031           DCA     ALSW          /STORE LSW
6032  7430           SZL                   /WAS LSW MINUS ?
6033  7240           STA                   /YES = MSW = 7777
6034  3030           DCA     AMSW
6035  3035           DCA     AEXT          /CLEAR EXT
6036  5625           JMP I   STRA          /RETURN

6037  0000   STRB,   0
6040  7100           CLL
6041  7510           SPA                   /CHECK SIGN
6042  7120           STL                   /SAVE SIGN IN LINK
6043  3037           DCA     BLSW          /STORE LSW
6044  7430           SZL                   /WAS SIGN MINUS ?
6045  7240           STA                   /YES = MSW = 7777
6046  3036           DCA     BMSW
6047  3043           DCA     BEXT          /CLEAR EXT
6050  5637           JMP I   STRB          /RETURN
```

```
6051  0000  SETUP,  0
6052  7200          CLA
6053  1374          TAD     (SETTAB-1       /GET ADDRESS OF TABLE
6054  3010          DCA     10
6055  6211          CDF     10
6056  1410          TAD I   10              /GET MAJOR STATE WORD
6057  6201          CDF     00              /RESTORE PROGRAM DATA FIELD
6060  3773'         DCA     CSTATE
6061  6211          CDF     10              /GET DATA FROM FILED 1
6062  1410          TAD I   10              /GET TEXT ADDR
6063  3306          DCA     ENTTXT
6064  1410          TAD I   10
6065  7440          SZA
6066  5270          JMP     .+2
6067  5315          JMP     BADNWS          /SOMETHING IS WRONG
6070  7041          CIA
6071  1251          TAD     SETUP           /CHECK FOR PROPER ADDRESS
6072  7640          SZA CLA
6073  5255          JMP     SETUP+4
6074  6201          CDF     00              /RESET PROGRAM FIELD
6075  7604          LAS
6076  0372          AND     (100            /GET SW 5
6077  7640          SZA CLA                 /TRACING PROGRAM ?
6100  5305          JMP     ENTYP           /YES = TYPE ENTER
6101  7604          LAS
6102  0371          AND     (200            /GET SW 4
6103  7650          SNA CLA                 /STOP ON ENTER ?
6104  5651          JMP I   SETUP           /NO = RETURN
6105  4770'  ENTYP, JMS     TYP             /TYPE ENTER
6106  3006   ENTTXT, CRLF
6107  0000          0
6110  7604          LAS
6111  0371          AND     (200            /GET SW 4
6112  7640          SZA CLA                 /STOP ON ENTER ?
6113  4767'         JMS     KEYCK           /YES = WAIT FOR KEYBOARD INPUT
6114  5651          JMP I   SETUP

6115  6201  BADNWS, CDF     00
6116  7402          HLT                     /PROGRAM MUST BE RE LOADED
6117  5316          JMP     .-1

6120  0000  ESTOP,  0
```

```
6121  0000  GAPT,  0                    /OUTBRK USING ADDRS
6122  4335        JMS     GET           /GET DATA
6123  0072        PAPT                  /DATA ADDRESS
6124  5721        JMP I   GAPT

6125  0000  GPC,   0                    /OUTBRK USING PPC
6126  4335        JMS     GET           /GET DATA
6127  0066        PFPC                  /ADDRESS OF DATA
6130  5725        JMP I   GPC

6131  0000  GOP,   0                    /OUTBRK USING OP ADDRESS
6132  4335        JMS     GET           /GET DATA
6133  0070        OPADR                 /DATA ADDRESS
6134  5731        JMP I   GOP

6135  0000  GET,   0                    /GET DATA
6136  7200        CLA
6137  1735        TAD I   GET           /GET ADDRESS REGISTER
6140  3000        DCA     T1            /SAVE
6141  2335        ISZ     GET           /INCREMENT RETURN
6142  1400        TAD I   T1            /GET FIELD BITS
6143  2000        ISZ     T1            /INC TO RIGHT HALF OF REGISTER
6144  0366        AND     (7            /3 BITS ONLY
6145  7106        CLL RTL               /MOVE 3 BITS LEFT
6146  7004        RAL
6147  1365        TAD     (6201         /CREATE CDF INST
6150  3353        DCA     .+3
6151  1400        TAD I   T1            /GET ABSOLUTE ADDRESS
6152  3000        DCA     T1            /SAVE
6153  6201        CDF     00            /CHANGE TO CORRECT FIELD
6154  1400        TAD I   T1            /GET DATA
6155  6201        CDF     00            /RESTORE DATA FIELD
6156  5735        JMP I   GET           /RETURN

6165  6201
6166  0007
6167  0254
6170  6703
6171  0200
6172  0100
6173  0731
6174  1577
6175  5762
6176  6707
6177  7600
      6200        PAGE
```

```
                        /CHECK SR05

6200  0000    TRSKP,    0
6201  7604              LAS
6202  0377              AND     (100          /GET SWITCH 5
6203  7650              SNA CLA               /IS SWITCH 5 SET ?
6204  5212              JMP     .+6           /NO - BYPASS TYPEOUT
6205  1600              TAD I   TRSKP         /GET TEXT ADDRESS
6206  3210              DCA     .+2
6207  4776'             JMS     TYP
6210  0000              0
6211  0000              0
6212  2200              ISZ     TRSKP
6213  5600              JMP I   TRSKP

                        /CHECK OP ADDRESS

6214  0000    CKOPAD,   0
6215  7200              CLA
6216  6566              RDOP                  /GET OP ADDR FROM FPP
6217  3231              DCA     SAVOP         /SAVE FOR TYPEOUT
6220  1071              TAD     OPADR+1       /GET PROGRAM OP ADDR
6221  7041              CIA
6222  1231              TAD     SAVOP         /COMPARE THE ADDRESSES
6223  7650              SNA CLA               /FPP OP ADDR CORRECT ?
6224  5614              JMP I   CKOPAD        /YES
6225  1775'             TAD     7600          /NO - GET PC
6226  4774'             JMS     ERR           /GO TO ERROR ROUTINE
6227  0003              ERROR   0003          /OP ADDR ERROR CODE
6230  5614              JMP I   CKOPAD        /RETURN
6231  0000    SAVOP,    0

6232  7240    LTRUNK,   STA                   /DEL BITS 0-8 IN 2 WORD REG
6233  1011              TAD     11            /GET REGISTER ADDRESS
6234  3001              DCA     T2            /STORE IN POINTER
6235  1401              TAD I   T2            /GET REG MSW
6236  0373              AND     (7            /DEL BITS 0-8
6237  3401              DCA I   T2            /STORE REG DATA IN MSW
6240  5400              JMP I   T1            /EXIT
```

```
6241  0000  SETST5, 0                        /SET PROG STATE 5
6242  7200         CLA
6243  1772'        TAD     CSTATE
6244  0371         AND     (377
6245  1370         TAD     (2400
6246  3772'        DCA     CSTATE
6247  5641         JMP I   SETST5

6250  0000  SETST3, 0                        /SET PROG STATE 3
6251  7200         CLA
6252  1772'        TAD     CSTATE
6253  0371         AND     (377
6254  1367         TAD     (1400
6255  3772'        DCA     CSTATE
6256  5650         JMP I   SETST3

6257  0000  SETST6, 0                        /SET PROG STATE 6
6260  7200         CLA
6261  1772'        TAD     CSTATE
6262  0371         AND     (377
6263  1366         TAD     (3000
6264  3772'        DCA     CSTATE
6265  5657         JMP I   SETST6


                   /NORMALIZE THE O REGISTER

6266  0000  XNORM, 0
6267  1022         TAD     OMSW            /GET MSW
6270  3000         DCA     T1              /SAVE IT
6271  1022         TAD     OMSW            /GET MSW AGAIN
6272  0365         AND     (1777           /DELETE BITS 0 AND 1
6273  3022         DCA     OMSW            /STORE BITS 2-11
6274  4523         SOEZ                    /IS O=0?
6275  7120         STL
6276  1000         TAD     T1              /RESTORE MSW
6277  3022         DCA     OMSW
6300  7420         SNL                     /WAS O = 0 ?
6301  5666         JMP I   XNORM           /NO NORMALIZE REQUIRED
6302  1022  XNCK,  TAD     OMSW            /GET MSW
6303  7104         RAL CLL
6304  7520         SMA SNL                 /IS BIT 0 OR 1=1?
6305  5311         JMP     NXSHFT          /NO - NO, IS NOT NORMALIZED
6306  7060         CMA CML                 /COMPLIMENT SIGN AND LINK
6307  7530         SPA SZL                 /ARE BOTH BITS=1
6310  5666         JMP I   XNORM           /NO - NO, IS NORMALIZED
6311  4524  NXSHFT, SHIFT                  /SHIFT O LEFT 1 BIT
6312  4022         OREG    LEFT
6313  1764'        TAD     SHREG           /GET SHIFT REGISTER
6314  1363         TAD     (-1             /DECREMENT
6315  3764'        DCA     SHREG           /RESTORE IT
6316  5267         JMP     XNORM+1         /REPEAT
```

```
6317  0000  INDEX,  0                       /FIND INDEX REG SPECIFIED
6320  7300          CLA CLL                 /BY BITS 6-8 OF FPP INSTRUCTION
6321  1005          TAD     PIR             /GET INSTRUCTION
6322  0362          AND     (70             /EXTRACT X BITS
6323  7010          RAR                     /RIGHT JUSTIFY
6324  7012          RTR
6325  1361          TAD     (X0             /ADD TO ADDRESS OF X0
6326  3000          DCA     T1              /SAVE X ADDR
6327  5717          JMP I   INDEX           /RETURN

6330  0000  XPLUS1, 0                       /INCREMENT X
6331  4317          JMS     INDEX           /GET ADDR OF X IN T1
6332  2400          ISZ I   T1              /INC X
6333  7000          NOP
6334  5730          JMP I   XPLUS1          /RETURN

6335  0000  XGETX,  0                       /GET INDEX REG
6336  4317          JMS     INDEX           /FIND ADDR OF X
6337  1400          TAD I   T1              /GET DATA IN X
6340  5735          JMP I   XGETX           /RETURN

6341  0000  BRANCH, 0
6342  2301          2301    /SA     LOAD START STEP ADDRESS
6343  2201          2201    /RA     LOAD RESET STEP ADDRESS
6344  0123          0123    /AS     ASSEMBLE
6345  2224          2224    /RT     RUN TRACE MODE
6346  2206          2206    /RF     RUN FAST MODE
6347  2431          2431    /TY     TYPE CONTENTS OF REGISTER
6350  0514          0514    /EL     ERROR LIST
6351  0315          0315    /CM     LOAD COMMAND REGISTER
6352  0530          0530    /EX     FPP EXIT
6353  1720          1720    /OP     LOAD    OPERAND TABLE
6354  2310          2310    /SH     SET SHIFT COUNT AND FLAG
6355  0314          0314    /CL     CLEAR STEP SWITCH
6356  0501          0501    /EA     SET EXIT ADDRESS
6357  0000          0000

6361  0200
6362  0070
6363  7777
6364  0724
6365  1777
6366  3000
6367  1400
6370  2400
6371  0377
6372  0731
6373  0007
6374  6707
6375  7600
6376  6703
6377  0100
      6400          PAGE
```

```
6400  0000  SHIFTB, 0                    /SHIFT B UNTIL SHFT CNT=0
6401  7300        CLA CLL
6402  1777'       TAD     SHREG          /GET SHIFT COUNT
6403  7650        SNA CLA                /SHFT CNT=0?
6404  5600        JMP I   SHIFTB         /YES - NO SHIFT
6405  1036  SAVSN, TAD     BMSW           /GET B MSW
6406  0376        AND     (4000          /SAVE SIGN BIT
6407  3220        DCA     BSIGN
6410  4524        SHIFT                  /SHIFT B RIGHT 1 BIT
6411  0036        BREG    RIGHT
6412  1036        TAD     BMSW
6413  1220        TAD     BSIGN          /EXTEND THE SIGN
6414  3036        DCA     BMSW
6415  2777'       ISZ     SHREG          /SHIFT COUNT=0
6416  5205        JMP     SAVSN          /NO - SHIFT AGAIN
6417  5600        JMP I   SHIFTB         /YES - RETURN

6420  0000  BSIGN, 0
```

```
6421  0000  APTDEC, 0                        /DECREMENT PROGRAM
6422  7340           STA CLL                 /APT POINTER
6423  1073           TAD     PAPT+1          /12 BIT ABSOLUTE ADDR-1
6424  3073           DCA     PAPT+1
6425  7420           SNL                     /OVERFLOW?
6426  7040           CMA                     /YES
6427  1072           TAD     PAPT            /FIELD BITS
6430  3072           DCA     PAPT
6431  7300           CLA CLL
6432  5621           JMP I   APTDEC          /RETURN

6433  0000  CKSUB,  0                        /SKIP IF INSTRUCTION IS
6434  3244           DCA     CKST            /NOT A SUBTRACT
6435  1005           TAD     PIR             /GET INSTRUCTION
6436  7006           RTL
6437  7420           SNL                     /SUBTRACT?
6440  2233           ISZ     CKSUB           /NO-INCREMENT RETURN
6441  7300           CLA CLL
6442  1244           TAD     CKST            /RESTORE AC
6443  5633           JMP I   CKSUB           /RETURN

6444  0000  CKST,   0

                     /INCREMENT THE O REGISTER

6445  0000  OPLUS1, 0                        /INCOR-INC THE O REG
6446  7300           CLA CLL
6447  3775           DCA     OVFL            /CLEAR OVERFLOW
6450  2023           ISZ     OLSW            /INC LSW
6451  5645           JMP I   OPLUS1          /RETURN
6452  2022           ISZ     OMSW            /INC OMSW
6453  7000           NOP
6454  5645           JMP I   OPLUS1          /RETURN

                     /DOES INSTRUCTION STORE THE ANSWER IN MEMORY ?

6455  0000  MEMINS, 0                        /SKIP IF ANSWER IS NOT TO GO TO MEMORY
6456  7300           CLA CLL
6457  1005           TAD     PIR             /GET INSTRUCTION
6460  0374           AND     (7000           /EXTRACT OP CODE
6461  1373           TAD     (-5000
6462  7450           SNA                     /FADDM?
6463  5267           JMP     .+4             /YES-RETURN
6464  1372           TAD     (-2000
6465  7640           SZA CLA                 /FMULM?
6466  2255           ISZ     MEMINS          /NO-INCREMENT RETURN
6467  7300           CLA CLL
6470  5655           JMP I   MEMINS          /RETURN
```

```
                        /CHECK DATA THE FPP STORED IN MEMORY

6471  0000    CMEM1,    0                       /COMPARE MEMORY EXPONENT
6472  7300              CLA CLL
6473  1371              TAD     (-1             /GET COUNT
6474  3000              DCA     T1              /SAVE COUNT
6475  1000              TAD     T1
6476  7001              IAC
6477  0370              AND     (20             /1 OR 2 WORD ERROR CODE
6500  1367              TAD     (2              /MEMORY ERROR CODE
6501  3334              DCA     MCODE           /STORE FOR ERROR
6502  1000              TAD     T1              /GET WORD COUNT
6503  7001              IAC
6504  7640              SZA CLA
6505  7001              IAC
6506  1366              TAD     (EMEM-1
6507  3765'             DCA     SAVMEM          /SAVE FOR ERROR REPORT
6510  1765'             TAD     SAVMEM
6511  3010              DCA     10
6512  7040              CMA
6513  1071              TAD     OPADR+1         /GET OP ADDRESS
6514  3011              DCA     11
6515  1070              TAD     OPADR           /GET FIELD BITS
6516  0364              AND     (7              /DELETE EXTRA BITS
6517  7104              CLL RAL
6520  7006              RTL                     /MOVE 3 BITS LEFT
6521  1363              TAD     (CDF            /CREATE CDF INSTRUCTION
6522  3323              DCA     .+1             /MODIFY PROGRAM
6523  6201    CMEM,     CDF     00              /USER FIELD
6524  1411              TAD I   11              /GET FPP DATA
6525  6201              CDF     00              /PROGRAM FIELD
6526  7041              CIA
6527  1410              TAD I   10              /PROGRAM DATA
6530  7650              SNA CLA                 /IS DATA OK ?
6531  5336              JMP     .+3             /YES
6532  1271              TAD     CMEM1           /NO
6533  4762'             JMS     ERR             /GO TO ERROR ROUTINE
6534  0002    MCODE,    ERROR   0002            /MEMORY ERROR CODE
6535  5340              JMP     .+3             /RETURN
6536  2000              ISZ     T1              /YES--END OF COMPARE?
6537  5323              JMP     CMEM            /NO--COMPARE NEXT WORD
6540  5671              JMP I   CMEM1           /RETURN
```

```
6541  0000  CMEM2,  0                    /COMPARE MEMORY FRACTION
6542  7300          CLA CLL
6543  1341          TAD     CMEM2        /GET RETURN
6544  3271          DCA     CMEM1        /SET RETURN IN EXP COMPARE
6545  4530          SEPM                 /EPM MODE?
6546  1361          TAD     (3           /NO - WORD COUNT=-1
6547  1360          TAD     (-4          /WORD COUNT=-1 OR -4
6550  5273          JMP     CMEM1+2      /GO TO EXPONENT COMPARE



6560  7774
6561  0003
6562  6707
6563  6201
6564  0007
6565  0712
6566  0104
6567  0002
6570  0020
6571  7777
6572  6000
6573  3000
6574  7000
6575  0733
6576  4000
6577  0724
      6600          PAGE
```

```
                        /MOVE USER INDEX REGS TO PROGRAM INDEX REGS

6600  0000    PINDEX, 0                       /SET PROGRAM INDEX REGS
6601  7300            CLA CLL
6602  1076            TAD     X0ADR           /USER INDEX POINTER FIELD BITS
6603  7104            CLL RAL
6604  7006            RTL
6605  1377            TAD     (CDF            /CREATE CDF INST
6606  3216            DCA     PINCF           /MODIFY PROGRAM
6607  7240            STA
6610  1077            TAD     X0ADR+1         /USER INDEX POINTER
6611  3010            DCA     10
6612  1376            TAD     (X0-1           /ADDR OF PROG INDEX REGS
6613  3011            DCA     11
6614  1375            TAD     (-10            /COUNT
6615  3000            DCA     T1
6616  6201    PINCF,  CDF     00              /USER FIELD
6617  1410            TAD I   10
6620  6201            CDF     00              /PROGRAM FIELD
6621  3411            DCA I   11
6622  2000            ISZ     T1              /FINISHED?
6623  5216            JMP     PINCF           /NO
6624  5600            JMP I   PINDEX          /RETURN

6625  0000    CLREG,  0
6626  1625            TAD I   CLREG
6627  3010            DCA     10
6630  1374            TAD     (-6
6631  3000            DCA     T1
6632  2225            ISZ     CLREG
6633  3410            DCA I   10
6634  2000            ISZ     T1
6635  5233            JMP     .-2
6636  5625            JMP I   CLREG
```

```
6637  0000  CLRAX,  0
6640  4225        JMS     CLREG
6641  0027        AREG-1
6642  5637        JMP I   CLRAX

6643  0000  CLRBX,  0
6644  4225        JMS     CLREG
6645  0035        BREG-1
6646  5643        JMP I   CLRBX

6647  0000  CLROX,  0
6650  1003        TAD     EPM             /GET THE EPM CONTROL REGISTER
6651  7710        SPA CLA                 /EPM ENABLE = 1 ?
6652  5257        JMP     .+5             /YES - CLEAR ENTIRE REGISTER
6653  3022        DCA     OMSW            /NO - CLEAR MSW,LSW,EXT ONLY
6654  3023        DCA     OLSW
6655  3027        DCA     OEXT
6656  5647        JMP I   CLROX           /RETURN
6657  4225        JMS     CLREG
6660  0021        OREG-1
6661  5647        JMP I   CLROX

6662  0000  CLRMQX, 0
6663  4225        JMS     CLREG
6664  0043        MQREG-1
6665  5662        JMP I   CLRMQX

6666  0000  CLRACX, 0
6667  4225        JMS     CLREG
6670  0051        PFAC-1
6671  5666        JMP I   CLRACX

6672  0000  WORDL,  0
6673  4773'       JMS     WORD
6674  6212        CIF     10
6675  5672        JMP I   WORDL

6676  0000  KEYL1,  0                     /LINK TO KEYCK FROM FIELD 1
6677  4772'       JMS     KEYCK
6700  6211        CDF     10
6701  6212        CIF     10
6702  5676        JMP I   KEYL1
```

```
6703  0000  TYP,   0                    /LINK TO TYP10 IN FIELD 1
6704  6212        CIF     10
6705  5706        JMP I   .+1
6706  0036        TYPL

6707  0000  ERR,   0                    /LINK TO ERR10 IN FIELD 1
6710  6211        CDF     10
6711  6212        CIF     10
6712  4714        JMS I   .+2
6713  5707        JMP I   ERR
6714  1200        ERR10

                  /INCREMENT THE PROGRAM STATE GENERATOR

6715  0000  STINC, 0                    /INCST=INCREMENT THE
6716  7300        CLA CLL               /PROGRAM MAJOR STATE GEN.
6717  1771'       TAD     CSTATE
6720  1370        TAD     (400
6721  3771'       DCA     CSTATE
6722  5715        JMP I   STINC         /RETURN

                  /DECREMENT THE OP ADDRESS

6723  0000  OPDEC, 0                    /DECREMENT OP ADDRESS
6724  7340        CLL STA               /AC=-1
6725  1071        TAD     OPADR+1       /12 BIT ABSOLUTE ADDR
6726  3071        DCA     OPADR+1
6727  7420        SNL
6730  7040        CMA
6731  1070        TAD     OPADR         /FIELD BITS
6732  3070        DCA     OPADR
6733  7300        CLA CLL
6734  5723        JMP I   OPDEC         /RETURN
```

```
                        /ADD A AND B REGISTERS AND STORE THE
                        /ANSWER IN THE O REGISTER,
                        /THIS ROUTINE AND "SUBAB" SIMULATE THE
                        /ARITHMETIC FUNCTIONS OF THE ADDER

6735  0000    ADDAB,  0
6736  7300            CLA CLL
6737  1003            TAD     EPM             /GET EPM CONTROLS
6740  7700            SMA CLA                 /EPM ENABLE = 1?
6741  5767'           JMP     ADEXT           /NO - ADD EXTENSIONS
6742  1007            TAD     CARYIN          /GET CARRY INSERT
6743  7640            SZA CLA                 /CARRY IN = 1?
6744  7001            IAC                     /YES - ADD IT IN
6745  1034            TAD     ALSW3
6746  1042            TAD     BLSW3           /ADD EPM LSW3
6747  3026            DCA     OLSW3
6750  7004            RAL                     /INSERT CARRY OUT OF LSW3
6751  1033            TAD     ALSW2
6752  1041            TAD     BLSW2           /ADD EPM LSW2
6753  3025            DCA     OLSW2
6754  7004            RAL                     /INSERT CARRY OUT OF LSW2
6755  1032            TAD     ALSW1
6756  1040            TAD     BLSW1           /ADD EPM LSW1
6757  3024            DCA     OLSW1
6760  5766'           JMP     ADEXT+3         /CLEAR O EXTENSION
6761  5767'           JMP     ADEXT
6762  5735    ADRET,  JMP I   ADDAB

6766  7003
6767  7000
6770  0400
6771  0731
6772  0254
6773  0265
6774  7772
6775  7770
6776  0177
6777  6201
      7000            PAGE
```

```
7000  1035    ADEXT,  TAD     AEXT            /ADD EXTENSIONS WHEN NOT
7001  1043            TAD     BEXT            /IN EPM

7002  1007            TAD     CARYIN          /ADD CARRY INSERT
7003  3027            DCA     OEXT
7004  7004            RAL                     /INSERT CARRY OUT OF EXT
7005  1031            TAD     ALSW            /ADD LSW
7006  1037            TAD     BLSW
7007  3023            DCA     OLSW
7010  3007            DCA     CARYIN          /RESET CARRIN IN
7011  7004            RAL                     /INSERT CARRY OUT OF SLW
7012  1030            TAD     AMSW            /ADD MSW
7013  1036            TAD     BMSW
7014  3022            DCA     OMSW
7015  7010            RAR                     /CARRY OUT TO SIGN BIT
7016  3021            DCA     CAROUT          /SAVE CARRY OUT
7017  4777/           JMS     SPECFL          /CHECK FOR 4000 0000 IN O
7020  5240            JMP     SETFL           /O = 4000 0000 SPECIAL OVERFLOW
7021  1030            TAD     AMSW            /CHECK OVERFLOW
7022  7104            RAL CLL                 /SIGN OF A IN LINK
7023  7200            CLA
7024  1036            TAD     BMSW            /GET B SIGN
7025  7530            SPA SZL                 /ARE BOTH SIGNS POS
7026  7060            CMA CML                 /NO=COMPLIMENT
7027  7730            SPA SZL CLA             /ARE BOTH SIGNS NEG
7030  5241            JMP     SETFL+1         /NO=NO OVERFLOW
7031  1030            TAD     AMSW            /GET SIGN OF OPERAND
7032  7104            RAL CLL                 /SAVE IN THE LINK
7033  7200            CLA
7034  1022            TAD     OMSW            /GET SIGN OF ANSWER
7035  7530            SPA SZL                 /ARE BOTH SIGNS POS?
7036  7060            CMA CML                 /COMPLIMENT
7037  7730            SPA SZL CLA             /ARE BOTH SIGNS NEG?
7040  7340    SETFL,  STA CLL                 /NO=SET OVERFLOW
7041  3776/           DCA     OVFL            /STORE OVERFLOW
7042  5775/           JMP     ADRET           /RETURN
```

```
                        /SUBTRACT B FROM A AND STORE THE
                        /ANSWER IN THE 0 REGISTER
7043  0000    SUBAB,    0
7044  4250              JMS     NEGB            /MAKE B NEGATIVE
7045  4576              APBO                    /ADD A AND -B
7046  4250              JMS     NEGB            /RESTORE B
7047  5643              JMP I   SUBAB           /RETURN

7050  0000    NEGB,     0
7051  1003              TAD     EPM             /GET EPM CONTROL REG
7052  7710              SPA CLA                 /EPM ENABLE = 1?
7053  7346              STA CLL RTL             /AC = -3
7054  1374              TAD     (-2             /WORD COUNT IS -2 OR -5
7055  3000              DCA     T1              /STORE WORD COUNT
7056  1000              TAD     T1              /GET WORD COUNT
7057  7160              CMA STL                 /MAKE IT POSITIVE AND SUBTRACT 1
7060  1373              TAD     (BREG           /ADD BASE ADDRESS
7061  3001              DCA     T2              /STORE STARTING ADDRESS
7062  1003              TAD     EPM             /GET EPM CONTROL REG
7063  7710              SPA CLA                 /IS EPM ENABLE = 1?
7064  5270              JMP     BCOM            /YES = BYPASS EXT,
7065  1043              TAD     BEXT            /NO = EXT IS USED INSTEAD OF LSW/S
7066  7141              CIA CLL                 /NEGATE EXT,
7067  3043              DCA     BEXT            /STORE = EXT,

7070  1401    BCOM,     TAD I   T2              /GET REGISTER DATA
7071  7040              CMA                     /COMPLIMENT IT
7072  7430              SZL                     /CARRY OUT FROM PREVIOUS WORD?
7073  7101              IAC CLL                 /YES = ADD CARRY OUT
7074  3401              DCA I   T2              /STORE = DATA
7075  7060              CML CMA                 /AC = -1 AND COMPLIMENT THE LINK
7076  1001              TAD     T2              /SUBTRACT 1 FROM ADDR AND RESTORE LINK
7077  3001              DCA     T2              /STORE ADDRESS -1
7100  2000              ISZ     T1              /END OF WORD COUNT?
7101  5270              JMP     BCOM            /NO = NEGATE NEXT WORD
7102  5650              JMP I   NEGB            /YES = RETURN

7103  0000    XPUTX,    0                       /PUT DATA IN INDEX REG
7104  3001              DCA     T2              /SAVE DATA
7105  1005              TAD     PIR             /GET INSTRUCTION
7106  0372              AND     (7              /EXTRACT BITS 9-11
7107  1006              TAD     PXP             /ADD X0 ADDR
7110  3000              DCA     T1              /SAVE ADDR
7111  1001              TAD     T2              /GET DATA
7112  3400              DCA I   T1              /STORE DATA
7113  5703              JMP I   XPUTX           /RETURN
```

```
7114  0000  STERR,  0                        /STATUS ERROR
7115  7300          CLA CLL
7116  6556          FPRST                    /GET STATUS FROM FPP-12
7117  3324          DCA     STSAVE           /SAVE FOR ERROR TYPEOUT
7120  1314          TAD     STERR            /GET RETURN ADDRESS
7121  4771'         JMS     ERR              /GO TO ERROR ROUTINE
7122  0004          ERROR   0004             /STATUS ERROR CODE
7123  5714          JMP I   STERR            /RETURN
7124  0000  STSAVE, 0

7125  1370  RUBOUT, TAD     (RUTXT-1         /IF RUBOUT WAS INPUT
7126  3010          DCA     10               /ECHO /  AND DELETE
7127  1410          TAD I   10               /DATA WORDS IN WORD ROUTINE
7130  7450          SNA
7131  5767'         JMP     WORD+1
7132  6046          TLS
7133  6041          TSF
7134  5333          JMP     .-1
7135  7200          CLA
7136  5327          JMP     .-7
7137  0334  RUTXT,  334
7140  0240          240
7141  0240          240
7142  0000          0
7143  0000  TYPNCR, 0                        /TYPE WITH NO CARRIAGE RETURN
7144  7300          CLA CLL
7145  1343          TAD     TYPNCR           /GET RETURN ADDRESS
7146  3766'         DCA     TYP              /SET TYP ENTRY FOR RETURN
7147  7240          STA                      /SET THE AC = 7777
7150  6211          CDF     10               /FIELD 1
7151  3765          DCA I   (CRSW            /SET NO RETURN SWITCH
7152  6201          CDF     00               /RESTORE FIELD
7153  5764'         JMP     TYP+1            /GO TO TYPE ROUTINE

7154  0000  ASC,    0                        /LINK TO ASCII IN FIELD 1
7155  6212          CIF     10
7156  5757          JMP I   .+1
7157  0103          ASCL

7164  6704
7165  0027
7166  6703
7167  0266
7170  7136
7171  6707
7172  0007
7173  0036
7174  7776
7175  6762
7176  0733
7177  5442
      7200          PAGE
```

```
7200  0000  OEZ,   0                       /SKIP IF O=0
7201  7300         CLA CLL
7202  1003         TAD     EPM             /GET EPM CONTROL REGISTER
7203  7700         SMA CLA                 /EPM ENABLE = 1 ?
7204  5213         JMP     .+7             /NO - BYPASS EPM LSW'S
7205  1026         TAD     OLSW3           /GET EPM LSW3
7206  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7207  1025         TAD     OLSW2           /GET EPM LSW2
7210  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7211  1024         TAD     OLSW1           /GET EPM LSW1
7212  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7213  1023         TAD     OLSW            /GET LSW
7214  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7215  1022         TAD     OMSW            /GET MSW
7216  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7217  7650         SNA CLA                 /AC = 0?
7220  2200         ISZ     OEZ             /YES-INC RETURN
7221  5600         JMP I   OEZ

7222  0000  AEZ,   0                       /SKIP IF FAC=0
7223  7300         CLA CLL
7224  4530         SEPM                    /EPM MODE?
7225  5234         JMP     .+7             /NO - BYPASS EPM LSW'S
7226  1057         TAD     ACLSW3          /GET EPM LSW3
7227  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7230  1056         TAD     ACLSW2          /GET EPM LSW2
7231  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7232  1055         TAD     ACLSW1          /GET EPM LSW1
7233  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7234  1054         TAD     ACLSW           /GET LSW
7235  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7236  1053         TAD     ACMSW
7237  4243         JMS     CKLINK          /INHIBIT OVERFLOW
7240  7650         SNA CLA                 /AC = 0?
7241  2222         ISZ     AEZ             /YES-INC RETURN
7242  5622         JMP I   AEZ             /RETURN

7243  0000  CKLINK, 0
7244  7420         SNL                     /LINK = 1 ?
7245  5643         JMP I   CKLINK          /NO - RETURN
7246  7101         CLL IAC                 /YES - INSERT LINK INTO THE AC
7247  5244         JMP     .-3             /CHECK LINK AGAIN
```

```
7250  0000  ALZ,   0                    /SKIP IF FAC IS NEG
7251  7300         CLA CLL
7252  1053         TAD     ACMSW        /GET FAC MSW
7253  7710         SPA CLA              /IS FAC MINUS
7254  2250         ISZ     ALZ          /YES-INC RETURN
7255  5650         JMP I   ALZ          /RETURN

7256  0000  AGZ,   0                    /SKIP IF FAC IS GREATER THAN 0
7257  4250         JMS     ALZ          /IS FAC MINUS
7260  7410         SKP                  /NO-CHECK IF 0
7261  5656         JMP I   AGZ          /YES-DO NOT SKIP
7262  4222         JMS     AEZ          /IS FAC 0?
7263  2256         ISZ     AGZ          /NO-INC RETURN
7264  5656         JMP I   AGZ          /RETURN

7265  0000  SHIFTO, 0                    /SHIFT O LEFT UNTIL
7266  7300         CLA CLL              /SHIFT CNTR=0
7267  1777'        TAD     SHREG
7270  7650         SNA CLA              /SHIFT CNT=0
7271  5665         JMP I   SHIFTO       /YES-NO SHIFT
7272  4524         SHIFT
7273  4022         OREG    LEFT
7274  2777'        ISZ     SHREG        /SHIFT CNTR=0?
7275  5272         JMP     SHIFTO+5     /NO-SHIFT AGAIN
7276  5665         JMP I   SHIFTO       /YES RETURN
```

```
                        /FPP-12 ARITHMETIC INSTRUCTIONS

7277  5304    INS0,   FLDA                    /FLDA
7300  2736            PFADD                   /F,P, ADD AND SUB
7301  2736            PFADD                   /F,P, ADD AND SUB
7302  3606            PFDIV                   /FDIV
7303  3302            FMULT                   /FMUL
7304  2736            PFADD                   /F,P, ADD AND SUB
7305  5235            FSTA                    /FSTA
7306  3302            FMULT                   /FMULM
7307  5304            FLDA                    /FLDA
7310  2704            DPADD                   /D,P, ADD AND SUB
7311  2704            DPADD                   /D,P, ADD AND SUB
7312  3606            PFDIV                   /FDIV
7313  3302            FMULT                   /FMULM
7314  2704            DPADD                   /D,P, ADD AND SUB
7315  5235            FSTA                    /FSTA
7316  3302            FMULT                   /FMULM
7317  4731    INS2I,  JEQ                     /TABLE OF INSTRUCTION
7320  4734            JGE                     /ADDRESSES FOR SPECIAL
7321  4737            JLE                     /FORMAT 2 OP CODE I
7322  5005            JTRUE                   /JUMP ALWAYS (JA)
7323  4742            JNE
7324  4745            JLT
7325  4750            JGT
7326  5000            JAL
7327  5154            SETX
7330  4753            SETB
7331  4700            JSA
7332  4640            JSR
7333  1674            SPEC21+3
7334  1674            SPEC21+3
7335  1674            SPEC21+3
7336  1674            SPEC21+3

7337  0400    CNTRLS, SA
7340  0476            RA
7341  0600            AS
7342  0502            RT
7343  0506            RF
7344  7400            TY
7345  7404            EL
7346  0673            CM
7347  0436            EX
7350  0456            LDQP
7351  0663            SH
7352  0543            CL
7353  0452            EA
```

```
                        /SET RETURN FOR AUTO RESTART

7354  0000    SETRET, 0
7355  7300            CLA CLL
7356  1376            TAD     (STFPP+2        /GET REENTER ADDR OF NORMAL MODE
7357  3775'           DCA     RETURN          /SET REENTER ADDRESS
7360  3003            DCA     EPM             /RESET EPM MODE FLIP-FLOP
7361  5754            JMP I   SETRET          /RETURN

7362  0000    ASTER,  0                       /LINK TO WORD FROM FIELD 1
7363  4774'           JMS     TYP
7364  2457            AST
7365  0000            0
7366  5762            JMP I   ASTER


7367  0000    TITLE,  0
7370  6212            CIF     10
7371  4773            JMS I   .+2
7372  5767            JMP I   TITLE
7373  1422            TITL


7374  6703
7375  0354
7376  1002
7377  0724
      7400            PAGE
```

```
7400  6212  TY,     CIF      10              /GO TO TYPE ROUTINE IN FIELD 1
7401  6211           CDF      10
7402  5603           JMP I    .+1
7403  0600           TY10

7404  6212  EL,     CIF      10
7405  6211           CDF      10
7406  5607           JMP I    .+1
7407  1407           EL10

7410  0000  SHIFTR, 0                        /SHIFT REG LEFT OR RIGHT
7411  7200           CLA
7412  1610           TAD I    SHIFTR         /GET THE ARGUMENT
7413  7700           SMA CLA                 /LEFT OR RIGHT
7414  1377           TAD      (7704          /RIGHT = MAKE RAR
7415  1376           TAD      (7104          /MAKE RAR OR RAL
7416  3234           DCA      ROT            /SET ROTATE INSTRUCTION
7417  1610           TAD I    SHIFTR         /GET REG
7420  7004           RAL
7421  7110           CLL RAR                 /DELETE SIGN
7422  3000           DCA      T1             /START REG ADDRESS
7423  1000           TAD      T1
7424  3001           DCA      T2             /I NEED THE ADDRESS TWICE
7425  1003           TAD      EPM            /GET EPM CONTROL REG
7426  7710           SPA CLA                 /EPM ENABLE=1?
7427  7346           STA CLL RTL             /YES = AC=-3
7430  1375           TAD      (-1            /WORD COUNT=-1 OR -4
7431  3277           DCA      SWCNT          /SET SHIFT WORD COUNT
7432  1400           TAD I    T1             /GET MSW
7433  7100           CLL
7434  7104  ROT,    RAL CLL                 /ROTATE LEFT OR RIGHT
7435  3400           DCA I    T1             /STORE SHIFTED MSW
7436  1234           TAD      ROT            /GET ROTATE INSTRUCTION
7437  3242           DCA      .+3            /STORE IN NEXT ROTATE INST
7440  2000  LOOPR,  ISZ      T1
7441  1400           TAD I    T1             /GET NEXT WORD
7442  7104           RAL CLL                 /ROTATE LEFT OR RIGHT
7443  3400           DCA I    T1
7444  1610           TAD I    SHIFTR         /GET ARGUMENT
7445  7700           SMA CLA                 /LEFT OR RIGHT
7446  5252           JMP      .+4            /RIGHT = CHECK WORD COUNT
7447  7430           SZL                     /LEFT = CARRY OUT?
7450  2401           ISZ I    T2             /YES = INSERT IN PREVIOUS WORD
7451  2001           ISZ      T2             /INC PREVIOUS ADDRESS
7452  2277           ISZ      SWCNT          /END OF REGISTER ?
7453  5240           JMP      LOOPR          /GET NEXT WORD
```

```
                    /EPM CONTROL REGISTER OPERATIONS
                    /BIT 11          EPM MODE
                    /BIT 00          EPM ENABLE
                    /BIT 01          EPM EXECUTE STATE
                    /BIT 02          EPM SPECIAL STATE

7454  2000  REXT,   ISZ      T1
7455  1003          TAD      EPM             /GET EPM CONTROLS
7456  7710          SPA CLA                  /EPM ENABLE=1?
7457  5274          JMP      REND            /YES = END OF ROTATE
7460  1000          TAD      T1              /INC T1 OVER EPM LSW'S
7461  1374          TAD      (3
7462  3000          DCA      T1
7463  1234          TAD      ROT             /GET ROTATE INSTRUCTION
7464  3266          DCA      .+2
7465  1400          TAD I    T1              /GET EXT
7466  7110          RAR CLL                  /ROTATE LEFT OR RIGHT
7467  0373          AND      (7400           /MAKE IT 4 BITS
7470  3400          DCA I    T1              /STORE ROTATED EXT
7471  7430          SZL                      /CARRY OUT LEFT?
7472  2401          ISZ I    T2              /YES = INSERT IN PREVIOUS WORD
7473  7410          SKP
7474  3400  REND,   DCA I    T1              /IF EPM=1 CLEAR EXT
7475  2210          ISZ      SHIFTR          /INC RETURN
7476  5610          JMP I    SHIFTR          /FINISHED = RETURN

7477  0000  SWCNT,  0

7500  0000  SEX,    0                        /SET TMSC EXECUTE
7501  7300          CLA CLL
7502  1003          TAD      EPM             /GET EPM CONTROL REG
7503  7006          RTL                      /MOVE BIT 01 TO THE LINK
7504  7132          STL RTR                  /SET BIT 01 AND RESTORE THE AC
7505  3003          DCA      EPM             /STORE STATE REG
7506  5700          JMP I    SEX             /RETURN

7507  0000  REX,    0                        /RESET TMSC EXECUTE
7510  7300          CLA CLL
7511  1003          TAD      EPM             /GET STATE REG
7512  7006          RTL                      /MOVE BIT 01 TO THE LINK
7513  7112          CLL RTR                  /CLEAR BIT 01 AND RESTORE AC
7514  3003          DCA      EPM             /STORE STATE REG
7515  5707          JMP I    REX             /RETURN
```

```
7516  0000  SST,   0                    /SET TMSC SPECIAL STATE
7517  7300         CLA CLL
7520  1003         TAD     EPM          /GET STATE REG
7521  0372         AND     (6777        /SAVE ALL BITS EXCEPT BIT 02
7522  1371         TAD     (1000        /SET BIT 02
7523  3003         DCA     EPM          /STORE STATE REG
7524  5716         JMP I   SST          /RETURN

7525  0000  RST,   0                    /RESET TMSC SPECIAL STATE
7526  7300         CLA CLL
7527  1003         TAD     EPM          /GET STATE REG
7530  0372         AND     (6777        /CLEAR BIT 02
7531  3003         DCA     EPM          /STORE STATE REG
7532  5725         JMP I   RST          /RETURN
7533  0000  SKPEPM, 0                   /SKIP IF IN EPM MODE
7534  7300         CLA CLL
7535  1003         TAD     EPM          /GET EPM CONTROLS
7536  7012         RTR                  /MOVE MODE BIT TO SIGN
7537  7710         SPA CLA              /EPM MODE?
7540  2333         ISZ     SKPEPM       /YES - INC RETURN
7541  5733         JMP I   SKPEPM

7542  0000  ENOFF, 0                    /RESET EPM ENABLE
7543  7300         CLA CLL
7544  1003         TAD     EPM          /GET EPM CONTROL REG
7545  7004         RAL                  /MOVE ENABLE TO LINK
7546  7110         CLL RAR              /CLEAR ENABLE AND RESTORE REG
7547  3003         DCA     EPM          /RESTORE CONTROL REG
7550  5742         JMP I   ENOFF        /RETURN

7551  0000  ENON,  0                    /SET EPM ENABLE
7552  7300         CLA CLL
7553  4530         SEPM                 /EPM MODE ?
7554  5751         JMP I   ENON         /NO - DO NOT SET EPM ENABLE
7555  1003         TAD     EPM          /GET CONTROL REG
7556  7004         RAL                  /MOVE ENABLE TO LINK
7557  7130         STL RAR              /SET ENABLE AND RESTORE REG
7560  3003         DCA     EPM          /RESTORE CONTROL REG
7561  5751         JMP I   ENON         /RETURN
```

```
7571  1000
7572  6777
7573  7400
7574  0003
7575  7777
7576  7104
7577  7704
0114  0400
0115  0027
0116  0247
0117  6200
0120  6455
0121  6037
0122  6025
0123  7200
0124  7410
0125  7265
0126  6400
0127  7551
0130  7533
0131  5617
0132  7250
0133  7256
0134  7222
0135  7542
0136  7103
0137  6266
0140  6600
0141  5534
0142  5631
0143  5600
0144  5624
0145  5612
0146  5541
0147  5605
0150  6330
0151  6715
0152  5400
0153  6445
0154  5416
0155  5407
0156  6335
0157  6125
0160  6131
0161  6121
0162  7600
0163  6051
0164  6723
0165  6421
0166  6541
0167  6471
0170  6647
```

```
 2171  6662
 2172  6643
 2173  6666
 2174  6637
 2175  5425
 2176  6735
 2177  7043
       0001          FIELD 1
```

```
0000  11111111  00000000  1111I111  11I1111I  11111111  11111111  I1111I11  11I11111
0100  11111111  11I01111  11111111  11111111  11111111  11111111  I1111111  11111111

0200  11111111  11I11111  1111I111  11I1111I  11111111  11111111  I111I1I1  11I11111
0300  11111111  11111111  1111I111  11I1111I  11111111  11111111  I111I111  11I11111

0400  11111111  11111111  11111111  11I1111I  11111111  11111111  I1111111  11I11111
0500  11111111  11111111  1111I111  11I1111I  11111111  11111111  I1111111  11I11111

0600  11111111  11I1111I  11I11111  11111111  11111111  11111111  I111I111  11I11111
0700  11111111  11I11111  1111I111  11I1111I  00000000  01111111  I1111111  11111111


1000  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I111I111  11I11111
1100  11111111  11I11111  1111I111  11I1111I  11000111  11111111  I1111111  11111111

1200  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I111I111  11I11111
1300  11111111  11I11111  1111I111  11I1111I  11111111  11111110  01111111  11I11111

1400  11111111  11I1111I  11111111  11I1111I  11111111  11111111  I1111111  11I11111
1500  11111111  11I11111  1111I111  11I1111I  11111100  00000000  00111111  11I11111

1600  11111111  11I1111I  11I11111  11111111  11111111  11111111  I111I111  11I11111
1700  11111111  11I11111  1111I111  11I1111I  11111000  11111111  I1111111  11I11111


2000  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I111I111  11I11111
2100  11111111  11I11111  1111I111  11I1111I  11111000  00000111  I1111111  11I11111

2200  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I111I111  11I11111
2300  11111111  11I11111  1111I111  11I1111I  11100000  00000000  0001I111  11I11111

2400  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I1111111  11I11111
2500  11111111  11I11111  1111I111  11I1111I  11111111  11111111  0001I111  11I11111

2600  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I111I111  11I11111
2700  11111111  11I11111  1111I111  11I1111I  11111111  11110111  I1111111  11I11111


3000  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I111I111  11I11111
3100  11111111  11I11111  1111I111  11I1111I  11111111  10000000  00111111  11I11111

3200  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I1111111  11I11111
3300  11111111  11I11111  1111I111  11I1111I  10000000  00000000  I1111111  11I11111

3400  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I111I111  11I11111
3500  11111111  11I11111  1111I111  11I1111I  11000000  00000000  00000000  00I11111

3600  11111111  11I1111I  11I11111  11I1111I  11111111  11111111  I1111111  11I11111
3700  11111111  11I11111  1111I111  11I1111I  11111111  10000000  01111111  11I11111
```

```
4000  11111111  111111I1  111I1111  11I111I1  I1111111  11111111  I1111I11  I1I11111
4100  11111111  11111111  11111111  11111111  11111111  11111111  I1110011  11I11111

4200  11111111  11111111  11111111  11I111I1  I1111111  11111111  I111I1I1  I1I11111
4300  11111111  11111111  11111111  11I11111  11111111  11111100  00000011  11111111

4400  11111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1I1I1I1  I1I11111
4500  11111111  11I111I1  11I1I111  11I111I1  11111111  11111111  I0000011  11I11111

4600  11111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1I1I1I1  I1I11111
4700  11111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1I10000  00I11111

5000  I1111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1111I1I  I1I11111
5100  I1111111  11111111  11111111  11I111I1  11111111  11111111  I1I1I1I1  I1I11111

5200  I1111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1I1I1I1  I1I11111
5300  I1111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1I10000  00I11111

5400  I1111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1111I1I  I1I11111
5500  11111111  11I111I1  11I1I111  11I111I1  11111100  00000001  I1111I1I  I1I11111

5600  I1111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1I1I1I1  I1I11111
5700  11111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1I0I1I1  I1I11111

6000  11111I11  11I111I1  I1I1I111  11I111I1  I1111111  11111111  I1I1I1I1  I1I111I1
6100  11111111  11I11111  11I11111  11I11111  11111111  11111110  00000I11  11I11111

6200  11111I11  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1I1I1I1  I1I11111
6300  11111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  01I1I1I1  11I11111

6400  I1111I11  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1I1I1I1  I1I11111
6500  11111111  11111111  11111111  11I11111  11111111  10000000  I1I1I1I1  11I11111

6600  I1111111  11I111I1  I1I1I111  11I111I1  I1111111  11111111  I1I1I1I1  I1I11111
6700  11111111  11I111I1  11I1I111  11I11111  11111111  11111111  I1I000I1  11I11111

7000  I1111111  11I111I1  11I1I111  11I111I1  I1111111  11111111  I1111I1I  11I11111
7100  11111111  11111111  11111111  11111111  11111111  11111111  0000I1I1  I1I11111

7200  11111111  11I111I1  11111111  11I111I1  11111111  11111111  I1111I1I  11I11111
7300  11111111  11I111I1  11I1I111  11I11111  11111111  11111111  I1111I1I  11I11111

7400  11111111  11111111  11I1I111  11I111I1  11111111  11111111  I1111I1I  11I11111
7500  11111111  11111111  11I1I111  11111111  11111111  11111111  I1000000  01I11111

7600
7700
```

```
        0000    *0
0000    0247    BEGIN



        0020    *20
0020    0000    T10,    0
0021    0000    T20,    0
0022    0000    TPNTR,  0
0023    0000    TCNTR,  0
0024    0000    ASCCH0, 0
0025    0000    ASCWD0, 0
0026    0000    OCTWD0, 0
0027    0000    CRSW,   0
0030    0000    REGAD,  0
0031    0000    RWCNT,  0
0032    0000    RTAD,   0
0033    0000    REGFLD, 0
0034    0000    DADD,   0
0035    0000    DWRD,   0
```

```
                /LINK TYP IN FIELD 0 TO TYPE IN FIELD 1

0036  7240  TYPL,  STA                     /AC = -1
0037  1577         TAD I   (TYP            /SUBTRACT 1 FROM RETURN ADDRESS
0040  6211         CDF     10              /DATA FIELD 1
0041  3010         DCA     10              /SAVE RETURN ADDRESS
0042  1176         TAD     (-6             /GET TEXT COUNT
0043  3062         DCA     TYPLX           /SET LOOP COUNTER
0044  1175         TAD     (TYPLD-1        /TEXT TABLE ADDRESS
0045  3011         DCA     11
0046  3411         DCA I   11              /CLEAR TEXT TABLE
0047  2062         ISZ     TYPLX
0050  5046         JMP     .-2
0051  1175         TAD     (TYPLD-1        /TEXT TABLE ADDRESS
0052  3011         DCA     11
0053  6201         CDF     00              /MOVE TEXT TABLE
0054  1410         TAD I   10              /FROM FIELD 0 TO FIELD 1
0055  6211         CDF     10
0056  7450         SNA                     /END OF TEXT TABLE ?
0057  5063         JMP     .+4             /YES
0060  3411         DCA I   11              /NO - SAVE TEXT ADDRESS
0061  5053         JMP     .-6             /GET NEXT ADDRESS
0062  0000  TYPLX,  0

0063  1174         TAD     (TYPLD          /GET DATA ADDRESS
0064  3773'        DCA     TYPE            /SET TYP RETURN
0065  1027         TAD     CRSW            /GET CARRAIGE RETURN SWITCH
0066  7650         SNA CLA                 /RETURN WANTED ?
0067  5772'        JMP     TYPE+1          /YES
0070  3027         DCA     CRSW            /RESET CARRAIGE RETURN SWITCH
0071  5771'        JMP     TYPE+5          /BYBASS CARRAIGE RETURN

0072  0000  TYPLD,  0
0073  0000         0
0074  0000         0
0075  0000         0
0076  0000         0
0077  0000         0
0100  6201         CDF     00
0101  6202         CIF     00
0102  5410         JMP I   10              /RETURN TO FIELD 0
```

```
                /LINK ASC IN FIELD 0 TO ASC10 IN FIELD 1

0103  7240  ASCL,  STA                  /AC = -1
0104  1570         TAD I   (ASC        /RETURN ADDRESS -1
0105  3010         DCA     10          /DATA ADDRESS
0106  1410         TAD I   10          /GET FIRST WORD FRO FIELD 0
0107  3123         DCA     ASCDAT      /SAVE
0110  1410         TAD I   10          /GET SECOND WORD FROM FIELD 0
0111  3117         DCA     ASCL2       /SAVE
0112  1523         TAD I   ASCDAT      /SET DATA FIELD 1
0113  3123         DCA     ASCDAT      /CONVERT TO ASCII
0114  6211         CDF     10
0115  4767         JMS     ASC10
0116  0123  ASCL1, ASCDAT
0117  0000  ASCL2, 0
0120  6201         CDF     00
0121  6202         CIF     00
0122  5410         JMP I   10
0123  0000  ASCDAT, 0


                /LINK TO WORD IN FIELD 1

0124  0000  WORD0, 0
0125  6201         CDF     00          /DATA FIELD 0
0126  6202         CIF     00          /INSTRUCTION FIELD 0
0127  4566         JMS I   (WORDL      /JMS TO WORD LINK IN FIELD 0
0130  1565         TAD I   (ASCWD      /GET THE ASCII WORD FROM FIELD 0
0131  3025         DCA     ASCWD0      /SAVE IN FIELD 1
0132  1564         TAD I   (OCTWD      /GET THE OCTAL WORD FROM FIELD 0
0133  3026         DCA     OCTWD0      /STORE IN FIELD 1
0134  1563         TAD I   (ASCCH      /GET TERMINATOR CHAR FROM FIELD 0
0135  6211         CDF     10          /RETURN TO FIELD 1
0136  3024         DCA     ASCCH0      /STORE IN FIELD 1
0137  5524         JMP I   WORD0

                /LINK TO KEYCK IN FIELD 0

0140  0000  KEYCKL, 0
0141  6201         CDF     00          /CHANGE TO FIELD 1
0142  6202         CIF     00
0143  4562         JMS I   (KEYL1      /JMS TO KEYCK LINK IN FIELD 0
0144  5540         JMP I   KEYCKL      /RETURN

0145  6201  KEYCK0, CDF     00          /GO TO KEYCK+4 IN FIELD 0
0146  6202         CIF     00
0147  5561         JMP I   (KEYCK+4

      0200         *200
```

```
0200  0000    TYREG,  0                       /TYPE CONTENTS OF A REGISTER
0201  4211            JMS     GETREG          /GET DATA FROM REGISTER
0202  1032            TAD     RTAD            /GET REG TEXT ADDRESS
0203  3205            DCA     .+2
0204  4777'           JMS     TYPE            /TYPE TEXT AND DATA
0205  3006            CRLF                    /TEXT ADDRESS
0206  3007            ROUT                    /DATA ADDRESS
0207  0000            0                       /TERMINATOR
0210  5600            JMP I   TYREG           /EXIT

0211  0000    GETREG, 0                       /CONVERT REGISTER TO ASCII
0212  7240            STA                     /AND STORE IN TEXT BUFFER
0213  1376            TAD     (ROUT           /GET TEXT BUFFER ADDRESS
0214  3010            DCA     10
0215  1375            TAD     (-36            /60 CHARACTER WORD COUNT
0216  3020            DCA     T10             /SET COUNTER
0217  1374            TAD     (4040           /STORE ASCII SPACES
0220  3410            DCA I   10              /IN TEXT BUFFER
0221  2020            ISZ     T10             /END OF BUFFER ?
0222  5217            JMP     .-3             /NO - CONTINUE
0223  1033            TAD     REGFLD          /GET REG DATA FIELD
0224  7104            RAL CLL                 /MOVE TO BITS 6-8
0225  7006            RTL
0226  1373            TAD     (CDF            /MAKE "CDF" INSTRUCTION
0227  3232            DCA     OUTFLD          /STORE IN OUTPUT FIELD
0230  1376            TAD     (ROUT           /GET DATA TEXT ADDRESS
0231  3240            DCA     GETXT           /SET TEXT POINTER
0232  6201    OUTFLD, CDF                     /SET REG DATA FIELD
0233  1430            TAD I   REGAD           /GET REGISTER DATA
0234  6211            CDF     10              /SET PROGRAM DATA FIELD
0235  3020            DCA     T10             /SAVE REG DATA
0236  4772'           JMS     ASC10           /CONVERT REG DATA TO ASCII
0237  0020            T10
0240  3007    GETXT,  ROUT
0241  2030            ISZ     REGAD           /INC REG POINTER
0242  2240            ISZ     GETXT           /INC TEXT POINTER 3 TIMES
0243  2240            ISZ     GETXT
0244  2240            ISZ     GETXT
0245  2031            ISZ     RWCNT           /END OF REGISTER ?
0246  5232            JMP     OUTFLD          /CONVERT NEXT WORD
0247  3640            DCA I   GETXT           /YES - STORE TERMINATING ZERO
0250  5611            JMP I   GETREG          /EXIT
```

```
0251  0000  TCHECK, 0
0252  1771'       TAD     TYBUF           /GET FIRST ARGUMENT
0253  1370        TAD     (-60
0254  7510        SPA                     /IS ARG LESS THAN 60
0255  5261        JMP     .+4             /YES - CANNOT BE A DATA REQUEST
0256  1367        TAD     (-10
0257  7710        SPA CLA                 /IS ARG GREATER THAN 67 ?
0260  5263        JMP     .+3             /NO - THIS IS A DATA REQUEST
0261  4766'       JMS     SCAN            /ARGUMENT IS A REG REQUEST
0262  5651        JMP I   TCHECK          /RETURN TO TTY MONITOR
0263  1365        TAD     (CRLF           /GET TEXT ADDRESS
0264  3032        DCA     RTAD            /STORE IN REGISTER TEXT ADDRESS
0265  1764'       TAD     TDAD            /GET FIELD DESIGNATOR
0266  7104        RAL CLL                 /MOVE TO BITS 6-8
0267  7006        RTL
0270  1373        TAD     (CDF            /MAKE CORRECT "CDF" INSTRUCTION
0271  3310        DCA     TDFLD           /CORRECT PROGRAM
0272  1763'       TAD     TDAD+1          /GET DATA ADDRESS
0273  3034        DCA     DADD            /STORE IN ADDRESS POINTER
0274  1762'       TAD     TDAD+2          /GET WORD COUNT
0275  7040        CMA
0276  3023        DCA     TCNTR           /STORE IN TEMP COUNTER
0277  1361  TDATA, TAD     (DADD           /GET DATA ADDRESS
0300  3030        DCA     REGAD           /STORE IN REGISTER ADDRESS
0301  2023        ISZ     TCNTR           /END OF WORD COUNT ?
0302  7610        SKP CLA                 /NO - TYPE NEXT WORD
0303  5651        JMP I   TCHECK          /YES - FINISHED
0304  7344        STA CLL RAL             /AC = -2
0305  3031        DCA     RWCNT           /SET REGISTER WORD COUNT
0306  7001        IAC
0307  3033        DCA     REGFLD          /SET REGISTER FIELD TO 1
```

```
0310  6201  TDFLD,  CDF
0311  1434         TAD I   DADD      /SET DATA FIELD
0312  6211         CDF     10        /GET DATA
0313  3035         DCA     DWRD      /RESTORE DATA FIELD
0314  4200         JMS     TYREG     /STORE IN DATA WORD
0315  2034         ISZ     DADD      /TYPE DATA
0316  5277         JMP     TDATA     /INC DATA ADDRESS
0317  1310         TAD     TDFLD     /GET NEXT DATA WORD
0320  1360         TAD     (10       /FIELD OVERFLOW
0321  0357         AND     (70       /INC CDF INSTRUCTION
0322  1373         TAD     (CDF      /EXTRACT FIELD BITS
0323  3310         DCA     TDFLD     /MAKE NEW CDF INSTRUCTION
0324  5277         JMP     TDATA     /UPDATE PROGRAM
                                     /GET NEXT DATA WORD

0357  0070
0360  0010
0361  0034
0362  0463
0363  0462
0364  0461
0365  3006
0366  0400
0367  7770
0370  7720
0371  0464
0372  0725
0373  6201
0374  4040
0375  7742
0376  3007
0377  1000
      0400         PAGE
```

```
0400  0000  SCAN,   0
0401  7300          CLA CLL
0402  1264          TAD     TYBUF           /GET FIRST ARGUMENT
0403  7650          SNA CLA                 /IS IT 0 ?
0404  4315          JMS     LDBUF           /YES = LOAD BUFFER WITH ALL ARGUMENTS
0405  1377          TAD     (TYBUF          /GET BUFFER ADDRESS
0406  3260          DCA     TPNTR1          /STORE IN POINTER
0407  1376  SCANL,  TAD     (TWD-1          /GET REGISTER TABLE ADDRESS
0410  3017          DCA     17
0411  1417          TAD I   17              /GET ARGUMENT FROM TABLE
0412  7450          SNA                     /END OF TABLE ?
0413  5253          JMP     SCEND           /YES = NO COMPARE
0414  7041          CIA
0415  1660          TAD I   TPNTR1          /COMPARE WITH INPUT ARGUMENT
0416  7650          SNA CLA                 /MATCH ?
0417  5222          JMP     .+3             /YES
0420  2017          ISZ     17              /INC TABLE ADDRESS
0421  5211          JMP     SCANL+2         /CHECK NEXT ARGUMENT
0422  1417          TAD I   17              /GET REGISTER ADDRESS
0423  3021          DCA     T20             /SAVE IT
0424  1021          TAD     T20             /RESTORE AC
0425  7450          SNA                     /ADDRESS = 0000 ?
0426  5330          JMP     MODE            /YES = MODE IS A SPECIAL CASE
0427  0375          AND     (77             /EXTRACT ADDRESS POINTER BITS
0430  1374          TAD     (RPNTR          /ADD POINTER TABLE ADDRESS
0431  3030          DCA     REGAD
0432  1430          TAD I   REGAD           /GET REGISTER ADDRESS
0433  3030          DCA     REGAD           /STORE IN REGISTER ADDRESS
0434  3033          DCA     REGFLD          /SET REGISTER FIELD = 0
0435  1021          TAD     T20             /GET REG ADDRESS
0436  0373          AND     (7700           /EXTRACT WORD COUNT
0437  7112          RTR CLL                 /MOVE TO BITS 6-11
0440  7012          RTR
0441  7012          RTR
0442  7041          CIA
0443  3031          DCA     RWCNT           /SET REG WORD COUNT
0444  1021          TAD     T20             /GET ARGUMENT
0445  0375          AND     (77             /EXTRACT ADDRESS POINTER
0446  1372          TAD     (REGEQ          /ADDRESS OF TEXT TABLE
0447  3032          DCA     RTAD
0450  1432          TAD I   RTAD            /GET TEXT ADDRESS
0451  3032          DCA     RTAD            /STORE IN REG TEXT POINTER
0452  4771          JMS     TYREG           /TYPE REGISTER CONTENTS
0453  2260  SCEND,  ISZ     TPNTR1          /INC ARGUMENT POINTER
0454  1660          TAD I   TPNTR1
0455  7640          SZA CLA                 /END OF ARGUMENT LIST ?
0456  5207          JMP     SCANL           /NO - PROCESS NEXT ARGUMENT
0457  5600          JMP I   SCAN            /YES = EXIT
```

```
0460  0000    TPNTR1, 0
0461  0000    TDAD,   0
0462  0000            0
0463  0000            0
0464  0000    TYBUF,  2
0465  0000    ZBLOCK  30
```

```
0515  0000  LDBUF,  0                    /LOAD ALL ARGUMENTS INTO BUFFER
0516  1370          TAD     (TYBUF-1     /GET BUFFER ADDRESS
0517  3017          DCA     17
0520  1376          TAD     (TWD-1       /ADDR OF ARGUMENT TABLE
0521  3016          DCA     16
0522  1416          TAD I   16           /GET ARGUMENT FROM TABLE
0523  7450          SNA                  /END OF TABLE ?
0524  5715          JMP I   LDBUF        /YES - EXIT
0525  3417          DCA I   17           /NO - STORE IN BUFFER
0526  2016          ISZ     16           /INC POINTER OVER ADDRESS
0527  5322          JMP     .-5          /GET NEXT ENTRY

0530  6201  MODE,   CDF     00           /DATA IS IN FIELD 0
0531  1767          TAD I   (EPM         /GET EPM CONTROL REGISTER
0532  7010          RAR
0533  7630          SZL CLA              /EPM MODE ?
0534  5341          JMP     EMODE        /YES
0535  1766          TAD I   (COMREG      /GET COMMAND REGISTER
0536  7710          SPA CLA              /FLOATING POINT MODE ?
0537  5342          JMP     DMODE        /NO - MUST BE DOUBLE PRECISION
0540  7001  FMODE,  IAC
0541  7001  EMODE,  IAC
0542  7001  DMODE,  IAC
0543  6211          CDF     10           /RESTORE PROGRAM DATA FIELD
0544  7106          RTL CLL              /MOVE CHARACTER TO LEFT
0545  7006          RTL                  /HALF OF THE AC
0546  7006          RTL
0547  1365          TAD     (0340        /MAKE ASCII CHARACTER
0550  3764'         DCA     MTEXT+7      /STORE IN TEXT STRING
0551  4763'         JMS     TYPE         /TYPE MODE - X
0552  2054          MTEXT
0553  0000          0
0554  5253          JMP     SCEND        /RETURN TO SCAN

0563  1000
0564  2063
0565  0340
0566  0002
0567  0003
0570  0463
0571  0200
0572  1746
0573  7700
0574  0701
0575  0077
0576  0633
0577  0464
      0600          PAGE
```

```
0600  1377     TY10,   TAD      (-24            /GET BUFFER COUNT
0601  3020             DCA      T10             /SET COUNTER
0602  1376             TAD      (TDAD-1         /GET ADDR OF ARGUMENT BUFFER
0603  3010             DCA      10
0604  3410             DCA I    10              /CLEAR ARGUMENT BUFFER
0605  2020             ISZ      T10             /FINISHED ?
0606  5204             JMP      .-2             /NO - CONTINUE
0607  1375             TAD      (TYBUF-1        /ADDR OF ARGUMENT BUFFER
0610  3017             DCA      17
0611  1374     TY10L,  TAD      (-3             /GET DATA ARGUMENT COUNT
0612  3020             DCA      T10             /SET 3 WORD COUNTER
0613  1376             TAD      (TDAD-1         /ADDRESS OF DATA ARGUMENT BUFFER
0614  3016             DCA      16
0615  4124             JMS      WORD0           /ALLOW TTY INPUT
0616  1025             TAD      ASCWD0          /GET ASCII INPUT
0617  3417             DCA I    17              /STORE IN ARGUMENT BUFFER
0620  1026             TAD      OCTWD0          /GET OCTAL INPUT
0621  3416             DCA I    16              /STORE IN DATA ARGUMENT BUFFER
0622  1024             TAD      ASCCH0          /GET TERMINATOR CHARACTER
0623  1373             TAD      (-215           /SUBTRACT CARRIAGE RETURN
0624  7640             SZA CLA                  /INPUT TERMINATED BY RETURN ?
0625  5231             JMP      .+4             /NO - GET MORE ARGUMENTS
0626  3417             DCA I    17              /STORE TERMINATOR ZERO
0627  4772             JMS      TCHECK          /YES - CHECK TYPE OF REQUEST
0630  5145             JMP      KEYCK0          /RETURN TO MONITOR
0631  2020             ISZ      T10             /THIRD INPUT ?
0632  5215             JMP      TY10L+4         /NO - GET NEXT ARGUMENT
0633  5211             JMP      TY10L           /YES - RESET COUNTER
```

```
                    /,TY COMMAND ARGUMENT TABLE

0634  2017  TWD,    2017          /PO - PROGRAM O REG
0635  0600          0600
0636  0617          0617          /FO - FPP O REGISTER
0637  0501          0501
0640  0001          0001          /A
0641  0602          0602
0642  0002          0002          /B
0643  0603          0603
0644  1521          1521          /MQ
0645  0604          0604
0646  0103          0103          /AC
0647  0605          0605
0650  2015          2015          /PM - PROGRAM MEMORY REGISTER
0651  0620          0620
0652  2003          2003          /PC - PROGRAM FPC
0653  0206          0206
0654  1720          1720          /OP - OP ADDRESS
0655  0210          0210
0656  0104          0104          /AD - ADRS
0657  0211          0211
0660  1122          1122          /IR
0661  0107          0107
0662  2324          2324          /ST - STATUS
0663  0112          0112
0664  2023          2023          /PS - PROGRAM STATES
0665  0113          0113
0666  0623          0623          /FS - READ FPP STATES
0667  0114          0114
0670  2310          2310          /SH - SHIFT REGISTER
0671  0117          0117
0672  0015          0015          /M - MODE
0673  0000          0000
0674  0030          0030          /X - INDEX REGISTERS
0675  1016          1016
0676  0120          0120          /AP - APT
0677  1315          1315
0700  0000          0000
```

```
0701  0022  RPNTR,  OREG          /REGISTER POINTER TABLE FOR TY ROUTINE
0702  0100          SAVOM
0703  0030          AREG
0704  0036          BREG
0705  0044          MQREG
0706  0052          PFAC
0707  0066          PFPC
0710  0005          PIR
0711  0070          OPADR
0712  0072          PAPT
0713  0004          PSTAT
0714  0731          CSTATE
0715  0736          MT2
0716  0211          APT
0717  0200          X0
0720  0724          SHREG
0721  0105          EMEM
0722  0000          0
0723  0000          0
0724  0000          0
```

```
                /CONVERT OCTAL WORD TO 6 BIT ASCII

0725  0000  ASC10,  0
0726  1725          TAD I   ASC10           /GET ADDRESS OF DATA WORD
0727  3367          DCA     ASC4            /SAVE
0730  2325          ISZ     ASC10           /INCREMENT RETURN
0731  1725          TAD I   ASC10           /GET TEXT ADDRESS
0732  6211          CDF     10              /DATA FIELD 1
0733  3366          DCA     ASC3            /SAVE TEXT ADDRESS
0734  2325          ISZ     ASC10           /INCREMENT RETURN
0735  1371          TAD     ASC77           /GET MASK
0736  7040          CMA                     /LEFT HALF
0737  0767          AND I   ASC4            /EXTRACT LEFT HALF OF DATA
0740  7112          CLL RTR                 /MOVE TO RIGHT HALF
0741  7012          RTR
0742  7012          RTR
0743  4351          JMS     ASCB            /CONVERT LEFT HALF
0744  2366          ISZ     ASC3            /INCREMENT TEXT ADDRESS
0745  1371          TAD     ASC77           /GET MASK
0746  0767          AND I   ASC4            /EXTRACT RIGHT HALF OF DATA WORD
0747  4351          JMS     ASCB            /CONVERT RIGHT HALF
0750  5725          JMP I   ASC10           /EXIT

0751  0000  ASCB,   0                       /CONVERT 2 OCTAL DIGITS
0752  3370          DCA     ASC5            /SAVE DATA
0753  1370          TAD     ASC5            /RESTORE DATA
0754  7006          RTL                     /MOVE DATA 1 DIGIT LEFT
0755  7004          RAL
0756  0364          AND     ASC1            /DELETE RIGHT DIGIT
0757  1370          TAD     ASC5            /GET CORRECT RIGHT DIGIT
0760  0364          AND     ASC1            /SAVE ONLY 2 CORRECT DIGITS
0761  1365          TAD     ASC2            /INSERT ASCII MODIFIER
0762  3766          DCA I   ASC3            /STORE CONVERTED DATA
0763  5751          JMP I   ASCB            /RETURN

0764  0707  ASC1,   0707
0765  6060  ASC2,   6060
0766  0000  ASC3,   0
0767  0000  ASC4,   0
0770  0000  ASC5,   0
0771  0077  ASC77,  77

0772  0251
0773  7563
0774  7775
0775  0463
0776  0460
0777  7754
      1000          PAGE
```

```
                /THIS ROUTINE UNPACKES 6 BIT PACKED ASCII CHARACTERS AND
                /OUTPUTS THEM TO THE TELETYPE OR LINE PRINTER IF SR11 = 1.
1070  7000      TYPE,   0                       /TYPE 6 BIT PACKED ASCII
1071  7200              CLA
1072  1350              TAD     T215            /ASCII RETURN
1073  4251              JMS     TOUT            /TYPE IT
1074  1351              TAD     T212            /LINE FEED
1075  4251              JMS     TOUT            /TYPE IT
1076  1600              TAD I   TYPE            /GET TEXT ADDRESS
1077  2200              ISZ     TYPE            /INCREMENT RETURN
1100  7450              SNA                     /END OF ARGUMENT STRING ?
1101  5267              JMP     PRT             /YES - PRINT LINE AND EXIT
1102  3022              DCA     TPNTR           /NO - STORE TEXT ADDRESS
1103  1422  TRET,       TAD I   TPNTR           /GET TEXT WORD
1104  7112              CLL RTR                 /MOVE LEFT CHARACTER
1105  7012              RTR                     /TO RIGHT HALF OF AC
1106  7012              RTR
1107  4224              JMS     TYPA            /CONVERT AND TYPE LEFT CHARACTER
1110  1422              TAD I   TPNTR           /GET TEXT WORD AGAIN
1111  4224              JMS     TYPA            /CONVERT AND TYPE RIGHT CHARACTER
1112  2022              ISZ     TPNTR           /INC TEXT POINTER
1113  5213              JMP     TRET            /GET NEXT WORD

1114  0000  TYPA,       0                       /CONVERT AC 6-11 TO TRUE ASCII
1115  0352              AND     T77             /DELETE AC 0-5
1116  7450              SNA                     /END OF TEXT STRING?
1117  5205              JMP     TYPE+5          /YES - GET NEXT ARGUMENT
1120  1353              TAD     TM40            /SUBTRACT 40
1121  7510              SPA                     /LESS THAN 40?
1122  1354              TAD     T100            /YES-300 SERIES CHAR-ADD 100
1123  1355              TAD     T240            /ADD ORIGINAL 40+200
1124  4251              JMS     TOUT            /TYPE CHARACTER
1125  5624              JMP I   TYPA            /GET NEXT CHARACTER

1126  0000  TTY,        0
1127  1266              TAD     TCHR            /GET CHARACTER
1130  6046              TLS                     /OUTPUT CHARACTER
1131  6041              TSF                     /WAIT FOR FLAG
1132  5241              JMP     .-1
1133  6042              TCF                     /CLEAR THE TTY FLAG
1134  7200              CLA
1135  6031              KSF                     /IS KEYBOARD FLAG SET?
1136  5636              JMP I   TTY             /NO-RETURN
1137  6032              KCC                     /YES-CLEAR FLAG
1140  5145              JMP     KEYCK0          /BYPASS REMAINING TEXT
```

```
1051  0000  TOUT,  0                    /CHECK OUTPUT DEVICE
1052  3266        DCA     TCHR          /SAVE CHARACTER
1053  7604        LAS                   /GET SWITCH REGISTER
1054  7010        RAR
1055  7630        SZL CLA               /IS OUTPUT TO A PRINTER ?
1056  5261        JMP     .+3           /YES
1057  4236        JMS     TTY           /NO - OUTPUT TO THE TTY
1060  5651        JMP I   TOUT          /RETURN
1061  6662        SFLG                  /SET OR CLEAR FLAG IN PRINTER
1062  6661        CFLG                  /WHICH PRINTER IS AVAILABLE ?
1063  4312        JMS     LP08          /IT IS THE LP08
1064  4325        JMS     LP12          /IT IS THE LP12
1065  5651        JMP I   TOUT          /RETURN

1066  0000  TCHR,  0                    /SAVE OUTPUT CHARACTER HERE

                  /IF A PRINTER WAS USED, PRINT THE LINE AND EXIT

1067  7604  PRT,   LAS                   /GET THE SWITCH REGISTER
1070  7010        RAR                   /MOVE SR11 TO LINK
1071  7620        SNL CLA               /WAS THE TTY USED ?
1072  5600        JMP I   TYPE          /YES - EXIT
1073  6662        SFLG                  /SET OR CLEAR PRINTER FLAG
1074  6661        CFLG                  /WHICH LINE PRINTER ?
1075  5304        JMP     PRT8          /LP08
1076  1356        TAD     PT10          /LP12 END LINE CONTROL
1077  6652        LCF                   /CLEAR THE FLAGS
1100  6664        LPR                   /PRINT THE LINE
1101  6661        LSD                   /WAIT FOR PRINTER TO FINISH
1102  5301        JMP     .-1
1103  5600        JMP I   TYPE          /EXIT
1104  1350  PRT8,  TAD     T215          /CARRIAGE RETURN CHARACTER
1105  6666        LPC                   /PRINT THE LINE
1106  6661        LSF                   /WAIT FOR PRINTER TO FINISH
1107  5306        JMP     .-1
1110  7300        CLA CLL
1111  5600        JMP I   TYPE          /EXIT
```

```
                /OUTPUT TO THE LP08 LINE PRINTER

1112  0000      LP08,   0
1113  6663              LSR                     /CHECK FOR PRINTER ERROR
1114  7610              SKP CLA                 /PRINTER IS OK
1115  5712              JMP I   LP08            /PRINTER IS NOT READY - EXIT
1116  1266              TAD     TCHR            /GET OUTPUT CHARACTER
1117  6666              LPC                     /OUTPUT IT TO THE PRINTER
1120  6661              LSF                     /WAIT FOR PRINTER TO FINISH
1121  5320              JMP     .-1
1122  7300              CLA CLL
1123  2312              ISZ     LP08            /INC RETURN OVER LP12 CALL
1124  5712              JMP I   LP08            /EXIT


                /OUTPUT TO THE LP12 PRINTER

1125  0000      LP12,   0
1126  6651              LSE                     /CHECK FOR PRINTER ERROR
1127  7610              SKP CLA                 /PRINTER IS OK
1130  5725              JMP I   LP12            /PRINTER IS NOT READY - EXIT
1131  1266              TAD     TCHR            /GET OUTPUT CHARACTER
1132  1357              TAD     TM212           /SUBTRACT LINE FEED
1133  7450              SNA                     /LINE FEED ?
1134  5346              JMP     LP12E           /YES - IGNORE IT
1135  1360              TAD     TM3             /SUBTRACT CARRIAGE RETURN
1136  7650              SNA CLA                 /CARRIAGE RETURN ?
1137  5346              JMP     LP12E           /YES - IGNORE IT
1140  1266              TAD     TCHR            /GET OUTPUT CHARACTER
1141  0352              AND     T77             /STRIP IT TO 6 BITS
1142  6652              LCF                     /CLEAR FLAGS
1143  6654              LLB                     /LOAD CHAR INTO PRINT BUFFER
1144  6661              LSD                     /WAIT FOR PRINTER TO FINISH
1145  5344              JMP     .-1
1146  7300      LP12E,  CLA CLL
1147  5725              JMP I   LP12            /EXIT

1150  0215      T215,   215
1151  0212      T212,   212
1152  0077      T77,    77
1153  7740      TM40,   -40
1154  0100      T100,   100
1155  0240      T240,   240
1156  0010      PT10,   10
1157  7566      TM212,  -212
1160  7775      TM3,    -3

      1200              PAGE
```

```
1200  0000    ERR10,  0
1201  3255            DCA     ERRET           /SAVE RETURN ADDR
1202  7604            LAS
1203  7004            RAL
1204  7710            SPA CLA                 /IS SW 1 SET ?
1205  5247            JMP     NOREP           /YES - NO REPORT
1206  4777'           JMS     ASC10           /CONV RETURN ADDR TO ASCII
1207  1255            ERRET
1210  2417            EROUT+3
1211  4776'   DATERR,  JMS     CODE            /GET ERROR CODE
1212  0375            AND     (7              /TYPE OF ERROR
1213  1374            TAD     (DATTXT         /ADDRESS OF TEXT TABLE
1214  3256            DCA     ERRT1
1215  1656            TAD I   ERRT1
1216  3226            DCA     ETXT2           /STORE IN TYPE ADDR STRING
1217  6201            CDF     00              /INSTRUCTION IS IN FIELD 0
1220  1773            TAD I   (ENTTXT         /GET INSTRICTION TEXT
1221  6211            CDF     10              /RESTORE PROGRAM DATA FIELD
1222  3225            DCA     ETXT1
1223  4772'   ERRTP,  JMS     TYPE            /TYPE DATA
1224  2414            EROUT
1225  3006    ETXT1,  CRLF
1226  3006    ETXT2,  CRLF
1227  0000            0
1230  7604            LAS                     /GET SWITCH REG
1231  0371            AND     (40             /EXTRACT SR06
1232  7640            SZA CLA                 /TYPE ERROR LIST ?
1233  5247            JMP     NOREP           /NO - END OF TYPEOUT
1234  1370            TAD     (ERREG-1        /GET ADDRESS OF ERROR LIST
1235  3017            DCA     17
1236  1367            TAD     (TYBUF-1        /ADDRESS OF TYPE OUT BUFFER
1237  3016            DCA     16
1240  1417            TAD I   17
1241  7450            SNA                     /END OF LIST ?
1242  5245            JMP     .+3             /YES
1243  3416            DCA I   16              /NO - STORE ARGUMENT
1244  5240            JMP     .-4             /MOVE NEXT ENTRY
1245  3416            DCA I   16              /ZERO LAST WORD OF BUFFER
1246  4766'           JMS     TCHECK
1247  7604    NOREP,  LAS
1250  7700            SMA CLA                 /SW0 SET ?
1251  4140            JMS     KEYCKL          /NO - WAIT FOR KEYBOARD INPUT
1252  6201            CDF     00
1253  6202            CIF     00              /GO BACK TO FIELD 0
1254  5600            JMP I   ERR10           /EXIT

1255  0000    ERRET,  0
1256  0000    ERRT1,  0
```

```
1257  6201  DOTER0, CDF      00              /LINK TO DOTERR IN FIELD 0
1260  6202           CIF      00
1261  5765           JMP I    (DOTERR


1262  2003  ERREG,  2003
1263  1122           1122
1264  0000  ZBLOCK  20

1365  0432
1366  0251
1367  0463
1370  1261
1371  0040
1372  1000
1373  6106
1374  1735
1375  0007
1376  1400
1377  0725
      1400           PAGE
```

```
1400  0000  CODE,  0
1401  6201         CDF     00          /GET CODE FROM FIELD 0
1402  1777         TAD I   (ERR        /GET ADDRESS OF ERR CODE
1403  3020         DCA     T10         /SAVE IT
1404  1420         TAD I   T10         /GET ERROR CODE
1405  6211         CDF     10          /RESTORE DATA FIELD
1406  5600         JMP I   CODE        /RETURN

1407  1376  EL10,  TAD     (ERREG-1
1410  3017         DCA     17
1411  4124         JMS     WORD0       /ALLOW TTY INPUT
1412  1025         TAD     ASCWD0      /GET ASCII INPUT
1413  3417         DCA I   17          /STORE IN ERROR LIST
1414  1024         TAD     ASCCH0      /GET TERMINATOR CHARACTER
1415  1375         TAD     (-215       /SUBTRACT CARRIAGE RETURN
1416  7640         SZA CLA             /INPUT TERMINATED BY A RETURN ?
1417  5211         JMP     EL10+2      /NO - GET MORE INPUT
1420  3417         DCA I   17          /YES - ZERO LAST WORD OF LIST
1421  5145         JMP     KEYCK0      /RETURN TO TTY MONITOR


1422  0000  TITL,  0
1423  6211         CDF     10          /SET CURREND DATA FIELD
1424  7610  TNOP,  SKP CLA             /WILL BE A CLA AFTER FIRST ENTRY
1425  5257         JMP     TEND        /THIS IS USED AFTER FIRST ENTRY
1426  1160         TAD     [7600       /GET ADDRESS OF LAST PAGE
1427  3224         DCA     TNOP        /SAVE
1430  1374         TAD     (STEP       /GET STEP ROUTINE
1431  3020         DCA     T10
1432  1420  SWAP,  TAD I   T10         /SWAP STEP WITH LAST PAGE
1433  3021         DCA     T20         /IN FIELD 0
1434  6201         CDF     00          /CHANGE TO FIELD 0
1435  1624         TAD I   TNOP        /GET WORD FROM FIELD 0
1436  6211         CDF     10          /CHANGE BACK TO FIELD 1
1437  3420         DCA I   T10         /STORE WHERE STEP WAS
1440  1021         TAD     T20         /GET WORD FROM STEP
1441  6201         CDF     00          /CHANGE TO FIELD 0
1442  3624         DCA I   TNOP        /STORE WHERE PS8 WAS
1443  6211         CDF     10          /BACK TO FIELD 1
1444  2020         ISZ     T10         /INC STEP ADDRESS
1445  2224         ISZ     TNOP        /INC PS8 ADDRESS
1446  5232         JMP     SWAP        /CONTINUE
1447  1160         TAD     [7600       /GET A "CLA"
1450  3224         DCA     TNOP        /DELETE ENTRY SKP
1451  4773'        JMS     TYPE        /TYPE MAINDEC NUMBER
1452  2736         MDEC
1453  0000         0
1454  4773'        JMS     TYPE        /TYPE ECO NUMBER
1455  2760         ECONO
1456  0000         0
```

```
1457  1372  TEND,   TAD     (OREG-1         /SYNC THE 0 REGISTERS
1460  3010          DCA     10
1461  6201          CDF     00              /THE PROGRAM 0 REG IS IN FIELD 0
1462  6563          ROMSW                   /READ FPP-12 0 REG MSW
1463  3410          DCA I   10              /STORE IN THE PROGRAM 0 REG
1464  6564          ROLSW                   /READ FPP-12 0 REG LSW
1465  3410          DCA I   10              /STORE IN PROGRAM 0 REG
1466  1371          TAD     (400
1467  6567          ROEPM                   /READ FPP-12 0 REG LSW1
1470  3410          DCA I   10              /STORE IN PROGRAM 0 REG
1471  1370          TAD     (200
1472  6567          ROEPM                   /READ FPP-12 0 REG LSW2
1473  3410          DCA I   10              /STORE IN PROGRAM 0 REG
1474  1367          TAD     (100
1475  6567          ROEPM                   /READ FPP-12 0 REG LSW3
1476  3410          DCA I   10              /STORE INPROGRAM 0 REG
1477  3410          DCA I   10              /CLEAR PROGRAM 0 REG EXT,
1500  6202          CIF     00              /RETURN TO FIELD 0
1501  5622          JMP I   TITL            /RETURN
1502  3000  RESPS8, DCA     0               /RESTORE LAST PAGE IN FIELD 0
1503  1366          TAD     (STEP-1         /ADDRESS OF PS8 MONITOR
1504  3010          DCA     10
1505  1160          TAD     [7600           /ADDRESS OF LAST PAGE
1506  3020          DCA     T10             /SAVE IN POINTER
1507  6211          CDF     10              /SET CURRENT FIELD
1510  1410          TAD I   10              /GET PS8 MONITOR
1511  6201          CDF     00              /DATA FIELD 0
1512  3420          DCA I   T10             /STORE IN FIELD 0
1513  2020          ISZ     T10             /INC ADDRESS SKP WHEN FINISHED
1514  5307          JMP     .-5             /NOT DONE
1515  6202          CIF     00
1516  5400          JMP I   0               /GO TO MONITOR IN FIELD 0


1566  3177
1567  0100
1570  0200
1571  0400
1572  0021
1573  1000
1574  3200
1575  7563
1576  1261
1577  6707
      1600          PAGE
```

```
1600  0010    SETTAB, 0010                    /INIATE
1601  2103            TINIT
1602  1063            ENINIT

1603  7500            7500                    /FETCH
1604  2110            TFETCH
1605  1262            FETCH+1

1606  0004            0004                    /TRAPPED INSTRUCTIONS
1607  2120            TTRAP
1610  1644            TRAP1

1611  5200            5200                    /DEPOSIT
1612  2133            TDEP
1613  2053            DEP+1

1614  7420            7420                    /FEXIT
1615  2114            TEXIT
1616  2430            EXIT+1

1617  7440            7440                    /D, P, ADD - SUBTRACT
1620  2140            TDPAS
1621  2705            DPADD0

1622  7440            7440                    /FLOATING PT, ADD & SUBTRACT
1623  2154            TFADD
1624  2737            FADD0

1625  7440            7440                    /MULTIPLY
1626  2171            TMULT
1627  3303            FMULT+1

1630  7440            7440                    /DIVIDE
1631  2177            TDIV
1632  3607            FFDIV+1

1633  0004            0004                    /LDX
1634  2204            TLDX
1635  4115            LDX1

1636  0004            0004                    /ALN
1637  2207            TALN
1640  4143            ALN1

1641  0004            0004                    /ATX
1642  2212            TATX
1643  4331            ATX1
```

```
1644  2004          0004                /XTA
1645  2215          TXTA
1646  4501          XTA1

1647  0004          0004                /JXN
1650  2220          TJXN
1651  4540          JXN1

1652  0004          0004                /JSR
1653  2223          TJSR
1654  4641          JSR1

1655  0404          0404                /JSA
1656  2226          TJSA
1657  4701          JSA2

1660  0004          0004                /JAC
1661  2231          TJAC
1662  5030          JAC1

1663  0004          0004                /JMP TRUE
1664  2234          TTRUE
1665  5006          JTRUE1

1666  0004          0004                /JMP FALSE
1667  2247          TFALSE
1670  5025          JFALSE+1

1671  0004          0004                /CLA
1672  2262          TCLA
1673  5145          FCLA+1

1674  0004          0004                /STF
1675  2266          TSTF
1676  5043          FSTF+1

1677  0004          0004                /STD
1700  2300          TSTD
1701  5060          FSTD+1

1702  0004          0004                /SET EPM
1703  2273          TSTE
1704  5355          FSTE+1
```

```
1705  0004          0004                /FNEG
1706  2305          TNEG
1707  5067          NEG1

1710  0004          0004                /NORM
1711  2311          TNORM
1712  5101          NORM1

1713  0004          0004                /PAUSE
1714  3006          CRLF
1715  5123          FPAUSE+1


1716  0004          0004                /SETB
1717  2340          TSETB
1720  4754          SETB1

1721  0004          0004                /SETX
1722  2344          TSETX
1723  5155          SETX1

1724  0004          0004                /ADDX
1725  2350          TADDX
1726  5201          ADDX1

1727  7440          7440                /STA
1730  2354          TSTA
1731  5236          FSTA+1

1732  7440          7440                /LDA
1733  2360          TLDA
1734  5305          FLDA+1
```

```
1735  2000    DATTXT, FSIS
1736  2034            FOIS
1737  2044            FMIS
1740  2065            FOPIS
1741  1770            FSTIS
1742  2012            FAPIS
1743  3006            CRLF
1744  3006            CRLF
1745  3006            CRLF


                      /REGISTER NAME TABLE

1746  2461    REGEQ,  POEQ
1747  2471            FOEQ
1750  2501            AEQ
1751  2511            BEQ
1752  2521            MQEQ
1753  2531            ACEQ
1754  2616            PCEQ
1755  2636            PIREQ
1756  2541            OPEQ
1757  2626            ADEQ
1760  2551            STATEQ
1761  2561            PSEQ
1762  2606            RSEQ
1763  2571            APEQ
1764  2576            XEQ
1765  2656            SHEQ
1766  2646            MEMEQ
1767  0000            0
```

```
1770  4052    FSTIS,  TEXT    " * STATUS *  "
1771  4023
1772  2401
1773  2425
1774  2340
1775  5240
1776  4000
1777  0000            0
2000  4052    FSIS,   TEXT    " * TIME STATE *   "
2001  4024
2002  1115
2003  0540
2004  2324
2005  0124
2006  0540
2007  5240
2010  4040

2011  0000            0
2012  4052    FAPIS,  TEXT    " * ADDRS *  "
2013  4001
2014  0404
2015  2223
2016  4052
2017  4040

2020  0000            0
2021  0620    STRTER, TEXT    "FPP-12 DID NOT START"
2022  2055
2023  6162
2024  4004
2025  1104
2026  4016
2027  1724
2030  4023
2031  2401
2032  2224

2033  0000            0
2034  4052    FOIS,   TEXT    " * O REG *   "
2035  4017
2036  4022
2037  0507
2040  4052
2041  4040
2042  4000
2043  0000            0
```

```
2044  4052  FMIS,   TEXT   " * MEMORY *   "
2045  4015
2046  0515
2047  1722
2050  3140
2051  5240
2052  4040

2053  0000          0
2054  1517  MTEXT,  TEXT   "MODE       *    "
2055  0405
2056  4040
2057  4040
2060  4040
2061  4075
2062  4040
2063  4040

2064  0000          0
2065  4052  FOPIS,  TEXT   " * OP ADDR *   "
2066  4017
2067  2040
2070  0104
2071  0422
2072  4052
2073  4040
2074  4000
2075  0000          0
2076  0516  ENT,    TEXT   "ENTER  "
2077  2405
2100  2240
2101  4000
2102  0000          0
2103  1116  TINIT,  TEXT   "INIATE "
2104  1101
2105  2405
2106  4000
2107  0000          0
2110  0605  TFETCH, TEXT   "FETCH"
2111  2403
2112  1000
2113  0000          0
2114  0605  TEXIT,  TEXT   "FEXIT"
2115  3011
2116  2400
2117  0000          0
```

```
2120  2422    TTRAP,  TEXT    "TRAPPED INSTRUCTION"
2121  0120
2122  2005
2123  0440
2124  1116
2125  2324
2126  2225
2127  0324
2130  1117
2131  1600
2132  0000            0
2133  0405    TDEP,   TEXT    "DEPOSIT"
2134  2017
2135  2311
2136  2400
2137  0000            0
2140  0456    TDPAS,  TEXT    "D', P', ADD - SUBTRACT "
2141  4020
2142  5640
2143  0104
2144  0440
2145  5540
2146  2325
2147  0224
2150  2201
2151  0324
2152  4000
2153  0000            0
2154  0614    TFADD,  TEXT    "FLOATING PT, ADD - SUB "
2155  1701
2156  2411
2157  1607
2160  4020
2161  2456
2162  4001
2163  0404
2164  4055
2165  4023
2166  2502
2167  4000
2170  0000            0
2171  1525    TMULT,  TEXT    "MULTIPLY "
2172  1424
2173  1120
2174  1431
2175  4000
2176  0000            0
2177  0411    TDIV,   TEXT    "DIVIDE "
2200  2611
2201  0405
2202  4000
2203  0000            0
```

```
2204  1404    TLDX,   TEXT    "LDX"
2205  3000
2206  0000            Ø
2207  0114    TALN,   TEXT    "ALN"
2210  1600
2211  0000            Ø
2212  0124    TATX,   TEXT    "ATX"
2213  3000
2214  0000            Ø
2215  3024    TXTA,   TEXT    "XTA"
2216  0100
2217  0000            Ø
2220  1230    TJXN,   TEXT    "JXN"
2221  1600
2222  0000            Ø
2223  1223    TJSR,   TEXT    "JSR"
2224  2200
2225  0000            Ø
2226  1223    TJSA,   TEXT    "JSA"
2227  0100
2230  0000            Ø
2231  1201    TJAC,   TEXT    "JAC"
2232  0300
2233  0000            Ø
2234  1215    TTRUE,  TEXT    "JMP CONDITION TRUE "
2235  2040
2236  0317
2237  1604
2240  1124
2241  1117
2242  1640
2243  2422
2244  2505
2245  4000
2246  0000            Ø
2247  1215    TFALSE, TEXT    "JMP CONDITION FALSE"
2250  2040
2251  0317
2252  1604
2253  1124
2254  1117
2255  1640
2256  0601
2257  1423
2260  0500
2261  0000            Ø
2262  0603    TCLA,   TEXT    "FCLA "
2263  1401
2264  4000
2265  0000            Ø
```

```
2266  2324    TSTF,   TEXT    "START F"
2267  0122
2270  2440
2271  0600
2272  0000            Ø
2273  2324    TSTE,   TEXT    "START E"
2274  0122
2275  2440
2276  0500
2277  0000            Ø
2300  2324    TSTD,   TEXT    "START D"
2301  0122
2302  2440
2303  0400
2304  0000            Ø
2305  0616    TNEG,   TEXT    "FNEG "
2306  0507
2307  4000
2310  0000            Ø
2311  0616    TNORM,  TEXT    "FNORM"
2312  1722
2313  1500
2314  0000            Ø
2315  0620    TPAUSE, TEXT    "FPP PAUSE...;ALT MODE TO CONTINUE "
2316  2040
2317  2001
2320  2523
2321  0556
2322  5656
2323  5656
2324  0114
2325  2440
2326  1517
2327  0405
2330  4024
2331  1740
2332  0317
2333  1624
2334  1116
2335  2505
2336  4000
2337  0000            Ø
2340  2305    TSETB,  TEXT    "SETB "
2341  2402
2342  4000
2343  0000            Ø
```

```
2344  2305    TSETX,  TEXT    "SETX "
2345  2430
2346  4000
2347  0000            0
2350  0104    TADDX,  TEXT    "ADDX "
2351  0430
2352  4000
2353  0000            0
2354  0623    TSTA,   TEXT    "FSTA "
2355  2401
2356  4000
2357  0000            0
2360  0614    TLDA,   TEXT    "FLDA "
2361  0401
2362  4000
2363  0000            0



2364  0411    DIVZ,   TEXT    "DIVIDE BY ZERO "
2365  2611
2366  0405
2367  4002
2370  3140
2371  3205
2372  2217
2373  4000
2374  0000            0
2375  0411    DIVOV,  TEXT    "DIVIDE FIXED POINT OVERFLOW"
2376  2611
2377  0405
2400  4006
2401  1130
2402  0504
2403  4020
2404  1711
2405  1624
2406  4017
2407  2605
2410  2206
2411  1417
2412  2700
2413  0000            0
2414  0522    EROUT,  TEXT    "ERROR ???? * "
2415  2217
2416  2240
2417  7777
2420  7777
2421  4052
2422  4000
2423  0000            0
```

```
2424  2516    VFER,   TEXT    "UNDERFLOW"
2425  0405
2426  2206
2427  1417
2430  2700
2431  0000            0
2432  0516    EOP,    TEXT    "END TRACE"
2433  0440
2434  2422
2435  0103
2436  0500
2437  0000            0
2440  0456    DPFLOW, TEXT    "D',P', ADD-SUBTRACT OVERFLOW "
2441  2056
2442  4001
2443  0404
2444  5523
2445  2502
2446  2422
2447  0103
2450  2440
2451  1726
2452  0522
2453  0614
2454  1727
2455  4000
2456  0000            0
2457  5200    AST,    TEXT    "*"
2460  0000            0
2461  2022    POEQ,   TEXT    "PROG 0 REG =  "
2462  1707
2463  4017
2464  4022
2465  0507
2466  4075
2467  4040

2470  0000            0
2471  0620    FOEQ,   TEXT    "FPP  0 REG =  "
2472  2040
2473  4017
2474  4022
2475  0507
2476  4075
2477  4040

2500  0000            0
```

```
2501  0140    AEQ,    TEXT    "A REG      =  "
2502  2205
2503  0740
2504  4040
2505  4040
2506  4075
2507  4040

2510  0000            0
2511  0240    BEQ,    TEXT    "B REG      =  "
2512  2205
2513  0740
2514  4040
2515  4040
2516  4075
2517  4040

2520  0000            0
2521  1521    MQEQ,   TEXT    "MQ REG     =  "
2522  4022
2523  0507
2524  4040
2525  4040
2526  4075
2527  4040

2530  0000            0
2531  0601    ACEQ,   TEXT    "FAC        =  "
2532  0340
2533  4040
2534  4040
2535  4040
2536  4075
2537  4040

2540  0000            0
2541  1720    OPEQ,   TEXT    "OP ADDR    =  "
2542  4001
2543  0404
2544  2240
2545  4040
2546  4075
2547  4040

2550  0000            0
```

```
2551  2324    STATEQ, TEXT    "STATUS     =  "
2552  0124
2553  2523
2554  4040
2555  4040
2556  4075
2557  4040

2560  0000            Ø
2561  2022    PSEQ,   TEXT    "PROG STATE =  "
2562  1707
2563  4023
2564  2401
2565  2405
2566  4075
2567  4040

2570  0000            Ø
2571  0120    APEQ,   TEXT    "APT  =  "
2572  2440
2573  4075
2574  4040

2575  0000            Ø
2576  1116    XEQ,    TEXT    "INDEX REGS =  "
2577  0405
2600  3040
2601  2205
2602  0723
2603  4075
2604  4040

2605  0000            Ø
2606  0620    RSEQ,   TEXT    "FPP STATE  =  "
2607  2040
2610  2324
2611  0124
2612  0540
2613  4075
2614  4040

2615  0000            Ø
2616  0620    PCEQ,   TEXT    "FPC        =  "
2617  0340
2620  4040
2621  4040
2622  4040
2623  4075
2624  4040

2625  0000            Ø
2626  0104    ADEQ,   TEXT    "ADRS       =  "
2627  2223
2630  4040
```

```
2631  4040
2632  4040
2633  4075
2634  4040

2635  0000          0
2636  1122  PIREQ,  TEXT   "IR          =  "
2637  4040
2640  4040
2641  4040
2642  4040
2643  4075
2644  4040

2645  0000          0
2646  1505  MEMEQ,  TEXT   "MEMORY REG =  "
2647  1517
2650  2231
2651  4022
2652  0507
2653  4075
2654  4040

2655  0000          0
2656  2310  SHEQ,   TEXT   "SHFT CNTR  =  "
2657  0624
2660  4003
2661  1624
2662  2240
2663  4075
2664  4040

2665  0000          0
2666  0516  INSTEP, TEXT   "ENTER STEP MODE"
2667  2405
2670  2240
2671  2324
2672  0520
2673  4015
2674  1704
2675  0500
2676  0000          0
2677  0530  OUTSTP, TEXT   "EXIT STEP MODE "
2700  1124
2701  4023
2702  2405
2703  2040
2704  1517
2705  0405
2706  4030
2707  0000          0
```

```
2710  7777    ASADDR, TEXT    "????   "
2711  7777
2712  4040
2713  4000
2714  0000            0
2715  7777    ADDR,   TEXT    "???? "
2716  7777
2717  4000
2720  0000            0
2721  0522    APTERR, TEXT    "ERROR SAVING APT IN EXIT"
2722  2217
2723  2240
2724  2301
2725  2611
2726  1607
2727  4001
2730  2024
2731  4011
2732  1640
2733  0530
2734  1124

2735  0000            0
2736  5252    MDEC,   TEXT    "*** TRACE-EPM ***  MAINDEC 12-D0TA"
2737  5240
2740  2422
2741  0103
2742  0555
2743  0520
2744  1540
2745  5252
2746  5240
2747  4015
2750  0111
2751  1604
2752  0503
2753  4061
2754  6255
2755  0460
2756  2401

2757  0000            0
```

```
2760  7114    ECONO,  TEXT    "ALL ECO'S THROUGH #0 MUST BE INSTALLED"
2761  1440
2762  0503
2763  1747
2764  2340
2765  2410
2766  2217
2767  2507
2770  1040
2771  4360
2772  4015
2773  2523
2774  2440
2775  0205
2776  4011
2777  1623
3000  2401
3001  1414
3002  0504

3003  0000            0
3004  7777    QMK,    TEXT    "?? "
3005  4000

3006  0000    CRLF,   0

3007  0000    ROUT,   0
3010  0000    ZBLOCK  60

      3200            PAGE
```

```
3200  0000  STEP,  0
3201  6031       KSF
3202  7610       SKP CLA
3203  4777'      JMS     KEYCK
3204  7240       STA
3205  1200       TAD     STEP            /GET EXIT ADDRESS
3206  3200       DCA     STEP            /STORE RETURN +1
3207  1200       TAD     STEP
3210  7041       CIA
3211  1776'      TAD     EXADDR          /COMPARE WITH RETURN ADDRESS
3212  7640       SZA CLA                 /EXIT AT THIS ADDRESS ?
3213  5221       JMP     .+6             /NO
3214  6552       FPICL                   /YES - CLEAR THE FPP-12
3215  3775'      DCA     CSTATE          /CLEAR THE PROGRAM STATE GENERATOR
3216  3003       DCA     EPM             /CLEAR THE EPM CONTROL REGISTER
3217  4774'      JMS     TITLE           /SYNC THE 0 REGISTERS
3220  5773'      JMP     INIT            /GO TO INIATE
3221  1772'      TAD     CK0             /CHECK 0 FLAG
3222  7640       SZA CLA                 /CHECK 0 REG ?
3223  4771'      JMS     COMP0           /YES
3224  7240       STA
3225  3772'      DCA     CK0             /SET FLAG FOR NEXT STATE
3226  1770'      TAD     CKOP            /CHECK OP ADDR FLAG
3227  7640       SZA CLA                 /CHECK OP ADDR ?
3230  4767'      JMS     CKOPAD          /YES
3231  7240       STA
3232  3770'      DCA     CKOP            /SET FLAG FOR NEXT STATE
3233  7604       LAS
3234  0366       AND     (400            /GET SWITCH 3
3235  7640       SZA CLA                 /IS SWITCH 3 SET ?
3236  5254       JMP     SSTEP           /YES - GO TO SINGLE STEP
3237  1765'      TAD     STEPSW
3240  7640       SZA CLA                 /SINGLE STEP MODE
3241  5254       JMP     SSTEP           /YES
3242  1200       TAD     STEP            /GET RETURN ADDRESS
3243  7041       CIA
3244  1764'      TAD     ENTSTP          /COMPARE WITH ENTER STEP ADDR
3245  7640       SZA CLA                 /ENTER STEP MODE ?
3246  5274       JMP     STEPGO          /NO
3247  7040       CMA
3250  3765'      DCA     STEPSW          /YES - SET STEP SWITCH
3251  4763'      JMS     TYP
3252  2666       INSTEP
3253  0000       0
```

```
3254  4762  SSTEP,  JMS      ASC          /CONVERT RETURN ADDR TO ASCII
3255  7600          7600
3256  2715          ADDR
3257  4763          JMS      TYP          /TYPE RETURN ADDRESS
3260  2715          ADDR
3261  0000          0
3262  4777          JMS      KEYCK        /WAIT FOR TTY INPUT

3263  1200          TAD      STEP         /GET RETURN ADDRESS
3264  7041          CIA
3265  1761          TAD      EXSTP        /EXIT STEP ADDRESS
3266  7640          SZA CLA               /EXIT STEP MODE ?
3267  5274          JMP      STEPGO       /NO
3270  4763          JMS      TYP
3271  2677          OUTSTP                /EXIT STEP MODE
3272  0000          0
3273  3765          DCA      STEPSW       /YES - RESET STEP SWITCH

3274  1003  STEPGO, TAD      EPM          /GET EPM CONTROLS
3275  7006          RTL
3276  7720          SMA SNL CLA           /IS THIS A TMSC STATE
3277  5315          JMP      NTMSC        /NO - CHECK NORMAL STATE
3300  7001          IAC                   /SET EXECUTE
3301  7420          SNL                   /IS THIS TMSC EXECUTE?
3302  7001          IAC                   /NO - SET SPECIAL STATE
3303  3760          DCA      MT1          /SAVE IT
3304  6562          RSTATE                /GET FPP STATE REG
3305  3757          DCA      MT2          /SAVE
3306  1757          TAD      MT2          /CHECK FPP STATE
3307  0356          AND      (3           /SAVE TMSC BITS ONLY
3310  7041          CIA
3311  1760          TAD      MT1
3312  7640          SZA CLA               /IS FPP IN THE CORRECT STATE?
3313  5323          JMP      STEPER       /NO
3314  5326          JMP      STEPER+3     /YES - ADVANCE AND RETURN
```

```
3315  1775'  NTMSC,  TAD     CSTATE
3316  1366           TAD     (400          /INC, CURRENT STATE REG
3317  3775'          DCA     CSTATE
3320  4755'          JMS     MSTATE        /CHECK STATES
3321  7610           SKP CLA               /STATE ERROR
3322  5326           JMP     .+4           /STATES OK
3323  1200   STEPER, TAD     STEP          /GET RETURN ADDRESS
3324  4754'          JMS     ERR           /GO TO ERROR ROUTINE
3325  0000           ERROR   0000          /TIME STATE ERROR CODE
3326  6561           FMAINT                /STEP FPP TO NEXT STATE
3327  7300           CLA CLL
3330  2200           ISZ     STEP          /RESTORE RETURN ADDRESS
3331  5600           JMP I   STEP

3332  0000   CADS,   0                     /COMPARE ADDRESS STORAGE LOCATIONS
3333  0000           0
3334  0000           0
3335  0000           0
3336  0000           0
3337  0000           0
3340  0000           0
3341  0000           0

3354  6707
3355  5516
3356  0003
3357  0736
3360  0735
3361  7735
3362  7154
3363  6703
3364  7734
3365  7732
3366  0400
3367  6214
3370  7737
3371  5722
3372  7736
3373  1046
3374  7367
3375  0731
3376  7733
3377  0254
      4000           *,&7000+1000
```

```
                /FLOATING POINT EXERCISER
                /
                /FLOATING POINT SYMBOL TABLE
                /
0000                    EFEXIT=0000
0002                    EFCLA=0002
0000                    EFLDA=0000
0003                    EFNEG=0003
0004                    EFNORM=0004
6000                    EFSTA=6000
1000                    EFADD=1000
5000                    EFADDM=5000
2000                    EFSUB=2000
0100                    ELDX=0100
1101                    ESETX=1101
1111                    ESETB=1111
2001                    EJXN=2001
1131                    EJSR=1131
1121                    EJSA=1121
1001                    EJEQ=1001
1031                    EJA= 1031
1051                    EJLT=1051
1061                    EJGT=1061
1041                    EJNE=1041
0007                    EJAC=0007
0010                    EALN=0010
0020                    EATX=0020
0030                    EXTA=0030
0040                    EFNOP=0040
4000                    EFMUL=4000
7000                    EFMULM=7000
3000                    EFDIV=3000
0110                    EADDX=0110
0005                    ESTRTF=0005
0006                    ESTRTD=0006
0050                    ESTRTE=0050
6000                    TABLE1=6000
6400                    TABLE2=6400
7000                    TABLE3=7000
7400                    TABLE4=7400
```

```
////////
/FLOATING POINT CODE AREA
////////
/
/THIS FIRST SECTION OF THE FPP ARITHEMATIC
/TEST OPERATES ON RANDOM NUMBER ARGUMENTS
/IN TABLE1 IF FLOATING POINT MODE
/OR TABLE3 IF DOUBLE PRECISION MODE
/THE OPERATIONS ARE OF THE FORM A*A-A
/OR A*A/A WHERE THE RESULT EXPECTED IS
/THE RANDOM NUMBER ARGUMENT ITSELF
/THE RESULTS ARE STORED IN
/TABLE2 IF FLOATING POINT MODE
/OR TABLE4 IF DOUBLE PRECISION MODE
/THESE TABLES ARE 400 OCTAL LOCATIONS
/IN LENGTH AND ARE LOCATED IN NUMERICAL SEQUENCE
/STARTING AT LOCATION 6000 OF LOWER 4K
/
/
/EXECUTE IN FLOATING POINT MODE
/MOVE RANDOM ARG TABLE1 TO TABLE2
/
```

```
                4000            *4000
4000    1111    ESTART, ESETB;          BASE
4001    5500
4002    1101            ESETX;          XREG1
4003    5375
4004    0006            ESTRTD
4005    0031            EXTA;1                          /LOAD X1
4006    0023            EATX;3                          /RESET X3 FOR REENTRY
4007    1031            EJA;            LINK+2
4010    5420
4011    0040            EFNOP
4012    0040            EFNOP
4013    1131    FPP1,   EJSR;           FSET            /SETUP FOR TST
4014    4745
4015    1131            EJSR;           RANGEN
4016    5313
4017    1121            EJSA;           LINK
4020    5416
4021    1031            EJA;            FPP2
4022    4040
4023    0005            ESTRTF

4024    0511    FPP1R,  EFLDA;511;      TABLE1-3        /LOAD FROM TABLE1
4025    5775
4026    6531            EFSTA;531;      TABLE2-3        /STORE IN TABLE 2
4027    6375
4030    1131            EJSR;           FCMPR           /COMPARE TABLE1 WITH TABLE2
4031    5045
4032    2101            EJXN;100;       FPP1R           /COMPLETED BLOCK
4033    4024
4034    0002            EFCLA                           /YES
4035    1210            EFADD;210                       /LOCK ONTO TST
4036    1041            EJNE;           FPP1            /IF OFFSET 10 IS NON-ZERO
4037    4013
```

```
                        /
                        /EXECUTE IN DOUBLE PRECISION MODE
                        /MOVE RANDOM ARG TABLE3 TO TABLE4
                        /
4040  1131    FPP2,     EJSR;           DPSET           /SETUP FOR TST
4041  5206
4042  1121              EJSA;           LINK
4043  5416
4044  1031              EJA;            FPP2E
4045  4063
4046  0006              ESTRTD
4047  0511    FPP2R,    EFLDA;511;      TABLE3-2        /LOAD RANDOM ARG
4050  6776
4051  6531              EFSTA;531;      TABLE4-2        /STORE SAME
4052  7376
4053  1131              EJSR;           DPCMPR          /JMP-SAVE RETURN
4054  5275
4055  2101              EJXN;100;       FPP2R           /COMPLETED BLOCK
4056  4047
4057  0002              EFCLA                           /YES
4060  1210              EFADD;210                       /LOCK ONTO TST
4061  1041              EJNE;           FPP2            /IF OFFSET 10 IS NON-ZERO
4062  4040

                        /
                        /EXECUTE IN EPM MODE
                        /MOVE RANDOM ARG TABLE1 TO TABLE2
                        /

4063  1131    FPP2E,    EJSR;           ESET
4064  5033
4065  1121              EJSA;           LINK
4066  5416
4067  1031              EJA;            FPP3
4070  4107
4071  0050    FPP2ER,   ESTRTE
4072  0511              EFLDA;511;      TABLE1          /LOAD FROM TABLE1
4073  6000
4074  6531              EFSTA;531;      TABLE2          /STORE IN TABLE2
4075  6400
4076  1131              EJSR;           ECMPR           /COMPARE RESULTS
4077  5067
4100  2101              EJXN;100;       FPP2ER
4101  4071
4102  0006              ESTRTD
4103  0002              EFCLA
4104  1210              EFADD;210                       /LOCK ON TO TEST
4105  1041              EJNE;           FPP2E           /IF OFFSET 10 IS NON-ZERO
4106  4063
```

```
                    /
                    /EXECUTE IN FLOATING POINT MODE
                    /ADD SUBTRACT TEST
                    /
4107  1131  FPP3,   EJSR!          FSET          /JMP-SAVE RETURN
4110  4745
4111  1121          EJSA!          LINK
4112  5416
4113  1031          EJA!           FPP3E
4114  4136
4115  0005          ESTRTF
4116  0511  FPP3R,  EFLDA!511!     TABLE1-3      /LOAD RANDOM ARG
4117  5775
4120  1411          EFADD!411!     TABLE1-3      /A&A
4121  5775
4122  2411          EFSUB!411!     TABLE1-3      /A&A-A
4123  5775
4124  6531          EFSTA!531!     TABLE2-3      /STORE IN TABLE2
4125  6375
4126  1131          EJSR!          FCMPAS        /JMP-SAVE RETURN
4127  5107
4130  2101          EJXN!100!      FPP3R         /COMPLETED BLOCK
4131  4116
4132  0002          EFCLA                        /YES
4133  1210          EFADD!210                    /LOCK ONTO TST
4134  1041          EJNE!          FPP3          /IF OFFSET 10 IS NON-ZERO
4135  4107
```

```
                    /
                    /EXECUTE IN EXTENDED PRECISION MODE
                    /ADD - SUBTRACT TEST
                    /
4136  1131    FPP3E,  EJSR;           ESET            /SETUP INDEX REGISTERS
4137  5033
4140  1121            EJSA;           LINK            /SETUP RE-ENTRY ADDRESS
4141  5416
4142  1031            EJA;            FPP4
4143  4165
4144  0050    FPP3ER, ESTRTE                          /START EPM MODE
4145  0511            EFLDA;511;      TABLE1          /LOAD RANDOM ARGUMENT
4146  6000
4147  1411            EFADD;411;      TABLE1          /A + A
4150  6000
4151  2411            EFSUB;411;      TABLE1          /A + A - A
4152  6000
4153  6531            EFSTA;531;      TABLE2          /STORE IN TABLE 2
4154  6400
4155  1131            EJSR;           ECMPAS          /COMPARE RESULTS
4156  5121
4157  2101            EJXN;100;       FPP3ER          /END OF TEST ?
4160  4144
4161  0006            ESTRTD                          /YES
4162  1210            EFADD;210                       /LOCK ON TO TEST IF
4163  1041            EJNE;           FPP3E           /OFFSET 10 IS NON-ZERO
4164  4136
```

```
                    /
                    /EXECUTE IN DOUBLE PRECISION MODE
                    /ADD-SUBTRACT TEST
                    /
4165  1131  FPP4,   EJSR!          DPSET          /JMP-SAVE RETURN
4166  5206
4167  1121          EJSA!          LINK
4170  5416
4171  1031          EJA!           FPP5
4172  4214
4173  0006          ESTRTD
4174  0511  FPP4R,  EFLDA!511!     TABLE3-2       /LOAD RANDOM ARG
4175  6776
4176  1411          EFADD!411!     TABLE3-2       /A+A
4177  6776
4200  2411          EFSUB!411!     TABLE3-2       /A+A-A
4201  6776
4202  6531          EFSTA!531!     TABLE4-2       /STORE RESULT
4203  7376
4204  1131          EJSR!          DPCMPR         /JMP-SAVE RETURN
4205  5275
4206  2101          EJXN!100!      FPP4R          /COMPLETED BLOCK
4207  4174
4210  0002          EFCLA                         /YES
4211  1210          EFADD!210                     /LOCK ONTO TEST
4212  1041          EJNE!          FPP4           /IF OFFSET 10 IS NON-ZERO
4213  4165
```

```
                        /
                        /EXECUTE IN FLOATING POINT MODE
                        /MULTIPLY-DIVIDE TEST
                        /
4214  1131    FPP5,     EJSR!           FSET          /JMP-SAVE RETERN
4215  4745
4216  1121              EJSA!           LINK
4217  5416
4220  1031              EJA!            FPP5E
4221  4245
4222  0005              ESTRTF
4223  0511    FPP5R,    EFLDA!511!      TABLE1-3      /LOAD RANDOM ARG
4224  5775
4225  0004              EFNORM                        /NORMALIZE
4226  6211              EFSTA!211                     /STORE IN OFFSET 11
4227  4411              EFMUL!411!      TABLE1-3      /A*A
4230  5775
4231  6212              EFSTA!212                     /STORE IN OFFSET 12
4232  3211              EFDIV!211                     /A*A/A
4233  6531              EFSTA!531!      TABLE2-3      /STORE RESULT
4234  6375
4235  1131              EJSR!           FCMPMD        /JMP-SAVE RETURN
4236  5135
4237  2101              EJXN!100!       FPP5R         /COMPLETED BLOCK
4240  4223
4241  0002              EFCLA                         /YES
4242  1210              EFADD!210                     /LOCK ONTO TST
4243  1041              EJNE!           FPP5          /IF OFFSET 10 IS NON-ZERO
4244  4214
```

```
                    /
                    /EXECUTE IN EXTENDED PRECISION MODE
                    /MULTIPLY - DIVIDE TEST
                    /
4245  1131  FPP5E,  EJSR!          ESET          /SETUP INDEX REGISTERS
4246  5033
4247  1121          EJSA!          LINK          /SETUP RE-ENTRY ADDRESS
4250  5416
4251  1031          EJA!           FPP6
4252  4277
4253  0050  FPP5ER, ESTRTE
4254  0511          EFLDA!511!     TABLE1        /LOAD RANDOM ARGUMENT
4255  6000
4256  0004          EFNORM                       /NORMALIZE
4257  6242          EFSTA!242                    /STORE IN OFFSET 42
4260  4411          EFMUL!411!     TABLE1        /A * A
4261  6000
4262  6244          EFSTA!244                    /STORE IN OFFSET 44
4263  3242          EFDIV!242                    /A * A / A
4264  6531          EFSTA!531!     TABLE2        /STORE RESULT IN TABLE 2
4265  6400
4266  1131          EJSR!          ECMPMD        /COMPARE RESULT
4267  5234
4270  2101          EJXN!100!      FPP5ER        /END OF TES ?
4271  4253
4272  0002          EFCLA                        /YES
4273  0006          ESTRTD
4274  1210          EFADD!210                    /LOOK ON TO TEST IF
4275  1041          EJNE!          FPP5E         /OFFSET 10 IS NON-ZERO
4276  4245
```

```
                        /
                        /EXECUTE IN DOUBLE PRECISION MODE
                        /MULTIPLY DIVIDE TEST
                        /
4277  1131    FPP6,     EJSR;          DPSET          /JMP-SAVE RETURN
4300  5206
4301  1121              EJSA;          LINK
4302  5416
4303  1031              EJA;           FPP7
4304  4327
4305  0006              ESTRTD
4306  0511    FPP6R,    EFLDA!511;     TABLE3-2       /LOAD RANDOM ARG
4307  6776
4310  4411              EFMUL!411;     TABLE3-2       /A*A
4311  6776
4312  6212              EFSTA!212                     /STORE IN OFFSET 12
4313  3411              EFDIV!411;     TABLE3-2       /A*A/A
4314  6776
4315  6531              EFSTA!531;     TABLE4-2       /STORE RESULT
4316  7376
4317  1131              EJSR;          DPCMPR         /JMP-SAVE RETURN
4320  5275
4321  2101              EJXN!100;      FPP6R          /COMPLETED BLOCK
4322  4306
4323  0002              EFCLA                         /YES
4324  1210              EFADD!210                     /LOCK ONTO TST
4325  1041              EJNE;          FPP6           /IF OFFSET 10 IS NON-ZERO
4326  4277
```

```
                        /
                        /EXECUTED IN FLOATING POINT MODE
                        /NORMALIZE-ALIGN TEST
                        /
4327  1131  FPP7,   EJSR!          FSET          /JMP-SAVE RETURN
4330  4745
4331  1121          EJSA!          LINK
4332  5416
4333  1031          EJA!           FPP7E
4334  4371
4335  0005          ESTRTF
4336  0411  FPP7R,  EFLDA!411!     TABLE1-2      /LOAD RANDOM ARG
4337  5776
4340  0006          ESTRTD
4341  0022          EATX!2                       /STORE IN FR 2
4342  0005          ESTRTF
4343  0511          EFLDA!511!     TABLE1-3      /LOAD RANDOM ARG
4344  5775
4345  0004          EFNORM                       /NORMALIZE
4346  0012          EALN!2                       /ALIGN ON FR 2
4347  0004          EFNORM                       /ETC
4350  0012          EALN!2
4351  0004          EFNORM
4352  0012          EALN!2
4353  0004          EFNORM
4354  0012          EALN!2
4355  0004          EFNORM
4356  0012          EALN!2
4357  6531          EFSTA!531!     TABLE2-3      /STORE RESULT
4360  6375
4361  1131          EJSR!          FCMPR         /JMP-SAVE RETURN
4362  5045
4363  2101          EJXN!100!      FPP7R         /COMPLED BLOCK
4364  4336
4365  0002          EFCLA                        /YES
4366  1210          EFADD!210                    /LOCK ONTO TST
4367  1041          EJNE!          FPP7          /IF OFFSET 10 IS NONZERO
4370  4327
```

```
                        /
                        /EXECUTE IN EXTENDED PRECISION MODE
                        /NORMALIZE - ALIGN TEST
                        /

4371  1131    FPP7E,  EJSR;           ESET            /SETUP INDEX REGISTERS
4372  5033
4373  1121            EJSA;           LINK            /SETUP RE-ENTRY ADDRESS
4374  5416
4375  1031            EJA;            FPP10
4376  4434
4377  0050    FPP7ER, ESTRTE                          /START EPM MODE
4400  0411            EFLDA;411;      TABLE1+4        /LOAD RANDOM ARG FROM TABLE 1
4401  6004
4402  0006            ESTRTD
4403  0022            EATX;2                          /STORE IN INDEX REG 2
4404  0050            ESTRTE
4405  0511            EFLDA;511;      TABLE1          /LOAD RANDOM ARGUMENT
4406  6000
4407  0004            EFNORM
4410  0012            EALN;2
4411  0004            EFNORM
4412  0012            EALN;2
4413  0004            EFNORM
4414  0012            EALN;2
4415  0004            EFNORM
4416  0012            EALN;2
4417  0004            EFNORM
4420  0012            EALN;2
4421  6531            EFSTA;531;      TABLE2          /STORE RESULT IN TABLE 2
4422  6400
4423  1131            EJSR;           ECMPR
4424  5067
4425  2101            EJXN;100;       FPP7ER          /END OF TEST ?
4426  4377
4427  0002            EFCLA                           /YES
4430  0006            ESTRTD
4431  1210            EFADD;210                       /LOCK ON TO TEST IF
4432  1041            EJNE;           FPP7E           /IF OFFSET 10 IS NON-ZERO
4433  4371
```

```
                        /
                        /EXECUTE IN DOUBLE PRECISION MODE
                        /TESTS SHIFTING OF THE FAC VIA ALN INST
                        /INDEX REG 4 CONTAINS NUM RIGHT SHIFTS
                        /INDEX REG 5 CONTAINS NUM LEFT SHIFTS
                        /
4434  1131    FPP10,    EJSR!           DPSET1          /JMP-SAVE RETURN
4435  5217
4436  1121              EJSA!           LINK
4437  5416
4440  1031              EJA!            FPP11           /GO TO NEXT TEST
4441  4467
4442  0006              ESTRTD
4443  0712    FPP10R,   EFLDA!712                       /LOAD RAN ARG FROM TABLE 3
4444  0014              EALN!4                          /SHIFT VIA IR 4
4445  0015              EALN!5                          /SHIFT VIA IR 5
4446  0014              EALN!4                          /ETC
4447  0015              EALN!5
4450  0014              EALN!4
4451  0015              EALN!5
4452  0014              EALN!4
4453  0015              EALN!5
4454  0014              EALN!4
4455  0015              EALN!5
4456  6731              EFSTA!731                       /STORE RESULT IN TABLE4
4457  1131              EJSR!           DPCPR1          /JMP-SAVE RETURN
4460  5305
4461  2101              EJXN!100!       FPP10R          /COMPLETED BLOCK
4462  4443
4463  0002              EFCLA                           /YES
4464  1210              EFADD!210                       /LOOK ONTO TST
4465  1041              EJNE!           FPP10           /IF OFFSET 1# IS NON-ZERO
4466  4434
```

```
                        /
                        /EXECUTED IN FLOATING POINT MODE
                        /ADD TO MEMORY-SUBTRACT TEST
                        /PRELIMINARY COMPARE OF FADDM AND FADD RESULTS MADE
                        /THEY SHOULD BE EQUAL
                        /
4467  1131  FPP11,  EJSR!          FSET1          /JMP-SAVE RETURN
4470  4757
4471  1121           EJSA!          LINK
4472  5416
4473  1031           EJA!           FPP11E
4474  4522
4475  0005           ESTRTF
4476  0713  FPP11R,  EFLDA!713                    /LOAD RAN ARG FROM TABLE1
4477  1734           EFADD!734                    /A+A
4500  6222           EFSTA!222                    /STORE IN OFFSET 22
4501  0613           EFLDA!613                    /LOAD RAN ARG FROM TABLE1
4502  5634           EFADDM!634                   /A+A TO MEMORY
4503  0634           EFLDA!634                    /LOAD A+A
4504  2222           EFSUB!222                    /SUBTRACT OFFSET 22
4505  1001           EJEQ!          G011          /SHOULD EQUAL ZERO
4506  4510
4507  0000           EFEXIT                       /FADDM AND FADD RESULTS DIFFER
4510  0613  G011,   EFLDA!613                    /LOAD ARG IN TABLE1
4511  6634           EFSTA!634                    /STORE BACK IN TABLE2
4512  1131           EJSR!          FCMPR         /JMP-SAVE RETURN
4513  5045
4514  2101           EJXN!100!      FPP11R        /COMPLETED BLOCK
4515  4476
4516  0002           EFCLA                        /YES
4517  1210           EFADD!210                    /LOCK ONTO TST
4520  1041           EJNE!          FPP11         /IF OFFSET 10 IS NON-ZERO
4521  4467
```

```
                    /
                    /EXECUTE IN EXTENDED PRECISION MODE
                    /ADD TO MEMORY - SUBTRACT TEST
                    /PRELIMINARY COMPARE OF FADDM AND FADD
                    /RESULTS SHOULD BE EQUAL
                    /
4522  1131  FPP11E, EJSR;          ESET1          /SETUP INDEX REGISTERS
4523  5005
4524  1121          EJSA;          LINK           /SETUP RE-ENTRY ADDRESS
4525  5416
4526  1031          EJA;           FPP12
4527  4555
4530  0050  FP11ER, ESTRTE                        /START EPM MODE
4531  0713          EFLDA!713                     /LOAD RANDOM ARG FROM TABLE 1
4532  1734          EFADD!734                     /A + A
4533  6230          EFSTA!230                     /STORE IN OFFSET 30
4534  0613          EFLDA!613                     /LOAD RAN ARG FROM TABLE 1
4535  5634          EFADDM!634                    /A + A TO MEMORY
4536  2230          EFSUB!230                     /SUBTRACT OFFSET 30
4537  1001          EJEQ;          G011E          /SHOULD EQUAL ZERO
4540  4542
4541  0000          EFEXIT
4542  0613  G011E,  EFLDA!613                     /LOAD ARG IN TABLE 1
4543  6634          EFSTA!634                     /STORE BACK IN TABLE 2
4544  1131          EJSR;          ECMPR1         /COMPARE RESULTS
4545  5103
4546  2101          EJXN!100;      FP11ER         /END OF TEST ?
4547  4530
4550  0002          EFCLA                         /YES
4551  0006          ESTRTD
4552  1210          EFADD!210                     /LOCK ON TO TEST IF
4553  1041          EJNE;          FPP11E         /IF OFFSET 10 IS NON-ZERO
4554  4522
```

```
                    /
                    /EXECUTED IN DOUBLE PRECISION MODE
                    /ADD TO MEMORY-SUBTRACT TEST
                    /PRELIMINARY COMPARE OF FADDM AND FADD RESULTS ARE MADE
                    /THEY SHOULD BE EQUAL
                    /
4555  1131  FPP12,  EJSR!          DPSET2          /JMP-SAVE RETURN
4556  5250
4557  1121          EJSA!          LINK
4560  5416
4561  1031          EJA!           FPP13
4562  4610
4563  0006          ESTRTD
4564  0712  FPP12R, EFLDA!712                     /LOAD RAN ARG FROM TABLE3
4565  1731          EFADD!731                     /A+A
4566  6222          EFSTA!222                     /STORE IN OFFSET 22
4567  0612          EFLDA!612                     /LOAD RAN ARG FROM TABLE3
4570  5631          EFADDM!631                    /A+A TO MEMORY
4571  0631          EFLDA!631                     /LOAD RESULT A+A
4572  2222          EFSUB!222                     /SUBTRACT OFFSET 22
4573  1001          EJEQ!          G012           /SHOULD EQUAL ZERO
4574  4576
4575  0000          EFEXIT                        /FADDM AND FADD RESULTS DIFFER
4576  0612  G012,   EFLDA!612                     /LOAD ARG IN TABLE3
4577  6631          EFSTA!631                     /STORE BACK IN TABLE4
4600  1131          EJSR!          DPCMPR         /JMP-SAVE RETURN
4601  5275
4602  2101          EJXN!100!      FPP12R         /COMPLETED BLOCK
4603  4564
4604  0002          EFCLA                         /YES
4605  1210          EFADD!210                     /LOCK ONTO TST
4606  1041          EJNE!          FPP12          /IF OFFSET 10 IS NON-ZERO
4607  4555
```

```
                        /
                        /EXECUTED IN FLOATING POINT MODE
                        /MULTIPLY TO MEMORY DIVIDE TEST
                        /PRELIMINARY COMPARE OF FMULM AND FMUL RESULTS ARE MADE
                        /THEY SHOULD BE EQUAL
                        /
4610  1131      FPP13,  EJSR!           FSET1           /JMP-SAVE RETURN
4611  4757
4612  1121              EJSA!           LINK
4613  5416
4614  1031              EJA!            FPP13E
4615  4645
4616  0005              ESTRTF
4617  0713      FPP13R, EFLDA!713                       /LOAD RAN ARG FROM TABLE1
4620  0004              EFNORM                          /NORMALIZE
4621  6211              EFSTA!211                       /STORE IN OFFSET 11
4622  4734              EFMUL!734                       /A*A
4623  6222              EFSTA!222                       /STORE IN OFFSET 22
4624  0211              EFLDA!211                       /LOAD OFFSET 11
4625  7634              EFMULM!634                      /A*A TO MEMORY
4626  0634              EFLDA!634                       /LOAD RESULT A*A
4627  2222              EFSUB!222                       /SUBTRACT OFFSET 22
4630  1001              EJEQ!           G013            /SHOULD EQUAL ZERO
4631  4633
4632  0000              EFEXIT                          /FMULM AND FMUL RESULTS DIFFER
4633  0613      G013,   EFLDA!613                       /GET ARG IN TABLE1
4634  6634              EFSTA!634                       /STORE BACK IN TABLE2
4635  1131              EJSR!           FCMPR           /JMP-SAVE RETURN
4636  5045
4637  2101              EJXN!100!       FPP13R          /COMPLETED BLOCK
4640  4617
4641  0002              EFCLA                           /YES
4642  1210              EFADD!210                       /LOCK ONTO TST
4643  1041              EJNE!           FPP13           /IF OFFSET 10 IS NON-ZERO
4644  4610
```

```
                    /
                    /EXECUTE IN EXTENDED PRECISION MODE
                    /MULTIPLY TO MEMORY - DIVIDE TEST
                    /PRELIMINARY COMPARE OF FMULM AND FMUL
                    /RESULTS SHOULD BE EQUAL
                    /
4645  1131  FPP13E, EJSR!          ESET1          /SETUP INDEX REGISTERS
4646  5005
4647  1121          EJSA!          LINK           /SETUP RE-ENTRY ADDRESS
4650  5416
4651  1031          EJA!           FPP14
4652  4703
4653  0050  FP13ER, ESTRTE                        /START EPM MODE
4654  0713          EFLDA!713                     /LOAD RANDOM ARG FROM TABLE 1
4655  0004          EFNORM                        /NORMALIZE
4656  6230          EFSTA!230                     /STORE IN OFFSET 30
4657  4734          EFMUL!734                     /A * A
4660  6232          EFSTA!232                     /STORE IN OFFSET 32
4661  0230          EFLDA!230                     /LOAD OFFSET 30
4662  7634          EFMULM!634                    /A * A TO MEMORY
4663  0634          EFLDA!634                     /LOAD FMULM RESULTS
4664  2232          EFSUB!232                     /SUBTRACT FMUL RESULTS
4665  1001          EJEQ!          G013E          /SHOULD EQUAL ZERO
4666  4670
4667  0000          EFEXIT
4670  0613  G013E,  EFLDA!613                     /GET RANDOM ARG FROM TABLE 1
4671  6634          EFSTA!634                     /STORE IN TABLE 2
4672  1131          EJSR!          ECMPR1         /COMPARE RESULTS
4673  5103
4674  2101          EJXN!100!      FP13ER         /END OF TEST ?
4675  4653
4676  0002          EFCLA                         /YES
4677  0006          ESTRTD
4700  1210          EFADD!210                     /LOCK ON TO TEST IF
4701  1041          EJNE!          FPP13E         /OFFSET 10 IS NON-ZERO
4702  4645
```

```
                    /
                    /EXECUTED IN DOUBLE PRECISION MODE
                    /MULTIPLY TO MEMORY-DIVIDE TEST
                    /PRELIMINARY COMPARE OF FMULM AND FMUL RESULTS ARE MADE
                    /THEY SHOULD BE EQUAL
                    /
4703  1131  FPP14,  EJSR!          DPSET2          /JMP-SAVE RETURN
4704  5250
4705  1121          EJSA!          LINK
4706  5416
4707  1031          EJA!           FPP1
4710  4013
4711  0006          ESTRTD
4712  0712  FPP14R, EFLDA!712                      /LOAD RAN ARG FROM TABLE3
4713  4731          EFMUL!731                      /A*A
4714  6222          EFSTA!222                      /STORE IN OFFSET 22
4715  0612          EFLDA!612                      /LOAD RAN ARG FROM TABLE3
4716  7631          EFMULM!631                     /A*A TO MEMORY
4717  0631          EFLDA!631                      /LOAD RESULT A*A
4720  2222          EFSUB!222                      /SUBTRACT OFFSET 22
4721  1001          EJEQ!          G014            /SHOULD EQUAL ZERO
4722  4724
4723  0000          EFEXIT                         /FMULM AND FMUL RESULT DIFFERS
4724  0612  G014,   EFLDA!612                      /LOAD ARG IN TABLE3
4725  6631          EFSTA!631                      /PUT BACK IN TABLE4
4726  1131          EJSR!          DPCMPR          /JMP-SAVE RETURN
4727  5275
4730  2101          EJXN!100!      FPP14R          /COMPLETED BLOCK
4731  4712
4732  0002          EFCLA                          /YES
4733  1210          EFADD!210                      /LOCK ONTO TST
4734  0100          ELDX!0!        -1
4735  7777
4736  1121          EJSA!          LINK
4737  5416
4740  0040          EFNOP
4741  0000          EFEXIT
4742  1031          EJA!           FPP1
4743  4013
```

```
                        /
                        /DUMMY POINTER
                        /END OF FPP TESTS WAS FPP14
                        /
4744  0000    FPP15,    0
                        /
                        /ENTERED ONLY IN FLOATING POINT MODE
                        /SET UP OF FPP INDEX REGISTERS
                        /
4745  0005    FSET,     ESTRTF
4746  0002              EFCLA                              /CLEAR THE FAC
4747  0100              ELDX!0!         7653               /LOAD IR 0
4750  7653
4751  0101              ELDX!1!         0                  /LOAD IR 1
4752  0000
4753  0103              ELDX!3!         0                  /LOAD IR 3
4754  0000
4755  1031              EJA!            BASE+1             /JMP ALWAYS
4756  5501


                        /
                        /ENTERED ONLY IN FLOATING POINT MODE
                        /SETS UP FPP INDEX REGISTERS
                        /DUPLICATES TABLE1 INTO TABLE2
                        /
4757  0005    FSET1,    ESTRTF
4760  0002              EFCLA                              /CLEAR THE FAC
4761  0100              ELDX!0!         7653               /LOAD IR 0
4762  7653
4763  0101              ELDX!1!         0                  /LOAD IR 1
4764  0000
4765  0102              ELDX!2!         7653               /LOAD IR 2
4766  7653
4767  0103              ELDX!3!         0                  /LOAD IR 3
4770  0000
4771  0104              ELDX!4!         0                  /LOAD IR 4
4772  0000
4773  0105              ELDX!5!         0                  /LOAD IR 5
4774  0000
4775  0541    MORE,     EFLDA!541!      TABLE1-3           /LOAD RAN ARG
4776  5775
4777  6551              EFSTA!551!      TABLE2-3           /STORE RAN ARG
5000  6375
5001  2121              EJXN!120!       MORE               /DONE 125 TIMES
5002  4775
5003  1031              EJA!            BASE+1             /JMP ALWAYS
5004  5501
```

```
5005  0050    ESET1,  ESTRTE                          /START EPM MODE
5006  0002            EFCLA
5007  0101            ELDX!1;         0
5010  0000
5011  0103            ELDX!3;         0
5012  0000
5013  0104            ELDX!4;         0
5014  0000
5015  0105            ELDX!5;         0
5016  0000
5017  0100            ELDX!0;         -50
5020  7730
5021  0102            ELDX!2;         -52
5022  7726
5023  0541    EMORE,  EFLDA!541;      TABLE1-6        /LOAD ARG FROM TABLE 1
5024  5772
5025  6551            EFSTA!551;      TABLE2-6        /STORE IN TABLE 2
5026  6372
5027  2121            EJXN!120;               EMORE
5030  5023
5031  1031            EJA;            BASE+1
5032  5501

5033  0050    ESET,   ESTRTE                          /START EPM MODE
5034  0002            EFCLA
5035  0100            ELDX!0;         -52
5036  7726
5037  0101            ELDX!1;         -1
5040  7777
5041  0103            ELDX!3;         -1
5042  7777
5043  1031            EJA;            BASE+1
5044  5501
```

```
                        /
                        /ENTER ONLY IN FLOATING POINT MODE
                        /COMPARE TABLE1 WITH TABLE2
                        /THEY SHOULD BE EQUAL
                        /TEST OFFSET 16 IN BASE REG TABLE
                        /IF NOT ZERO
                        /DECREMENT INDEX REGISTERS AND
                        /EXECUTE SAME ARGUMENT IN TABLE1 AGAIN
                        /THE RET PORTION OF THIS ROUTINE
                        /IS USED BY ALL OTHER COMPARE ROUTINES
                        /IN BOTH FLOATING POINT AND DOUBLE PRECISION MODE
                        /
5045  0411  FCMPR,  EFLDA!411!       TABLE1-3        /LOAD RAN ARG
5046  5775
5047  2431          EFSUB!431!       TABLE2-3        /SUBTRACT RESULT
5050  6375
5051  1001          EJEQ!            ERET            /JMP IF FAC=0
5052  5054
5053  0040          EFNOP
5054  0216  ERET,   EFLDA!216                        /LOAD OFFSET 16 IN BASE REG TABLE
5055  1001          EJEQ!            RETINC          /JMP IF FAC=0--IE, DONT LOCK ONTO TEST SEQ
5056  5065
5057  0110          EADDX!0!         7777            /ADD TO IR 0
5060  7777
5061  0111          EADDX!1!         7777            /ADD TO IR 1
5062  7777
5063  0113          EADDX!3!         7777            /ADD TO IR 3
5064  7777
5065  1031  RETINC, EJA!             BASE+1          /JMP ALWAYS
5066  5501


5067  0411  ECMPR,  EFLDA!411!       TABLE1
5070  6000
5071  2431          EFSUB!431!       TABLE2
5072  6400
5073  1001  ECMPRC, EJEQ!            .+4
5074  5100
5075  0006          ESTRTD
5076  1031          EJA!             ERET-1
5077  5053
5100  0006          ESTRTD
5101  1031          EJA!             ERET
5102  5054

5103  0613  ECMPR1, EFLDA!613
5104  2634          EFSUB!634
5105  1031          EJA!             ECMPRC
5106  5073
```

```
                        /
                        /ENTER ONLY IN FLOATING POINT MODE
                        /FROM AN ADDITION-SUBTRACTION TEST
                        /COMPARE TABLE1 WITH TABLE2
                        /THEY SHOULD BE EQUAL
                        /
5107  0411    FCMPAS, EFLDA!411!      TABLE1-3        /LOAD RAN ARG
5110  5775
5111  2431            EFSUB!431!      TABLE2-3        /SUBTRACT RESULT
5112  6375
5113  1041            EJNE!           FASCK           /JMP IF FAC NOT 0
5114  5117
5115  1031            EJA!            ERET            /JMP ALWAYS
5116  5054
                        /
                        /FLOATING POINT ADD-SUB FAILED
                        /
5117  1031    FASCK,  EJA!            ERET            /DATA ERROR EXIT
5120  5054
                        /
5121  0411    ECMPAS, EFLDA!411!      TABLE1
5122  6000
5123  2431            EFSUB!431!      TABLE2
5124  6400
5125  1041            EJNE!           .+4
5126  5132
5127  0006            ESTRTD
5130  1031            EJA!            ERET
5131  5054
5132  0006            ESTRTD
5133  1031            EJA!            FASCK
5134  5117

                        /ENTERED ONLY IN FLOATING POINT MODE
                        /FROM ROUTINE DOING A MULTIPLY-DIVIDE TEST
                        /THIS ROUTINE COMPARES TABLE1 WITH TABLE2
                        /THEY SHOULD BE EQUAL
                        /
5135  0411    FCMPMD, EFLDA!411!      TABLE1-3        /LOAD RAN ARG
5136  5775
5137  2431            EFSUB!431!      TABLE2-3        /SUBTRACT RESULT
5140  6375
5141  1041            EJNE!           FMDCK           /JMP IF FAC NOT 0 TO FMD
5142  5145
5143  1031            EJA!            ERET            /JMP ALWAYS
5144  5054
```

```
                    /
                    /ENTERED ONLY IN FLOATING POINT MODE
                    /FROM FCMPMD ROUTINE
                    /WHEN RANDOM ARG DIFFERS FROM RESULT
                    /BY PLUS OR MINUS 1 DUE TO ROUNDING
                    /ROUTINE TESTS FOR THIS OCCURANCE
                    /
5145  0431  FMDCK,  EFLDA!431!      TABLE2-3        /LOAD RESULT
5146  6375
5147  1001          EJEQ!           FZMD            /ROUTINE TO TEST FOR LEGAL UNDERFLOW
5150  5175
5151  0006          ESTRTD
5152  1214          EFADD!214                       /SUBTRACT 1 CONTAINED IN OFFSET 14 OF BASE REG
5153  0005          ESTRTF
5154  6215          EFSTA!215                       /STORE IN OFFSET 15 OF BASE REG
5155  0215          EFLDA!215                       /LOAD OFFSET 15 IN BASE REG
5156  2411          EFSUB!411!      TABLE1-3        /SUBTRACT RAN ARG
5157  5775
5160  1001          EJEQ!           ERET            /JMP IF FAC=0
5161  5054
5162  0431          EFLDA!431!      TABLE2-3        /LOAD WITH RESULT
5163  6375
5164  0006          ESTRTD
5165  1217          EFADD!217                       /ADD 1 CONTAINED IN OFFSET 17 OF BASE REG
5166  0005          ESTRTF
5167  6221          EFSTA!221                       /STORE IN OFFSET 21 BASE REG
5170  0221          EFLDA!221                       /LOAD OFFSET 21 IN BASE REG
5171  2411          EFSUB!411!      TABLE1-3        /SUBTRACT RAN ARG
5172  5775
5173  1031          EJA!            ERET
5174  5054
                    /
                    /ENTERED ONLY IN FLOATING POINT MODE
                    /WHEN RESULT HAS ZERO MANTISSA
                    /ROUTINE TESTS FOR A LEGAL UNDERFLOW
                    /
5175  0411  FZMD,   EFLDA!411!      TABLE1-3        /LOAD RANDOM ARG
5176  5775
5177  4213          EFMUL!213                       /MULTIPLY BY CONSTANT IN OFFSET 13 OF BASE REG
5200  1001          EJEQ!           ERET            /JMP IF FAC=0
5201  5054
5202  0002          EFCLA                           /CLEAR FAC
5203  6223          EFSTA!223                       /FLAG 8 UNDERFLOW NOT LEGAL
5204  1031          EJA!            ERET
5205  5054
```

```
                        /
                        /SET UP OF FPP INDEX REGISTERS
                        /FOR OPERATION IN DOUBLE PRECISION MODE
                        /
5206  0006      DPSET,  ESTRTD
5207  0100              ELDX!0!         7600            /LOAD !R 0
5210  7600
5211  0101              ELDX!1!         0               /LOAD !R 1
5212  0000
5213  0103              ELDX!3!         0               /LOAD !R 3
5214  0000
5215  1031              EJA!            BASE+1          /JMP ALWAYS
5216  5501
                        /
                        /SET UP OF FPP INDEX REGISTERS ON PAGE 0
                        /FOR OPERATION IN DOUBLE PRECISION MODE
                        /ENTERED FROM FPP10 TEST
                        /
5217  0006      DPSET1, ESTRTD
5220  0100              ELDX!0!         7600            /LOAD !R 0
5221  7600
5222  0101              ELDX!1!         0               /LOAD !R 1
5223  0000
5224  0103              ELDX!3!         0               /LOAD !R 3
5225  0000
5226  0104              ELDX!4!         14              /LOAD !R 4
5227  0014
5230  0105              ELDX!5!         7764            /LOAD !R 5
5231  7764
5232  1031              EJA!            BASE+1          /JMP ALWAYS
5233  5501

5234  0411      ECMPMD, EFLDA!411!      TABLE1
5235  6000
5236  2431              EFSUB!431!      TABLE2
5237  6400
5240  1041              EJNE!           EMDCK
5241  5245
5242  0006              ESTRTD
5243  1031              EJA!            ERET
5244  5054

5245  0006      EMDCK,  ESTRTD
5246  1031              EJA!            ERET
5247  5054
```

```
                    /
                    /SET UP OF FPP INDEX REGISTERS ON PAGE 0
                    /DUPLICATION OF TABLE3 INTO TABLE4
                    /
5250  0006    DPSET2, ESTRTD
5251  0100            ELDX!0;         7600            /LOAD IR 0
5252  7600
5253  0101            ELDX!1;         0               /LOAD IR 1
5254  0000
5255  0102            ELDX!2;         7600            /LOAD IR 2
5256  7600
5257  0103            ELDX!3;         0               /LOAD IR 3
5260  0000
5261  0104            ELDX!4;         0               /LOAD IR 4
5262  0000
5263  0105            ELDX!5;         0               /LOAD IR 5
5264  0000
5265  0541    MOR,    EFLDA!541;      TABLE3-2        /LOAD RAN ARG
5266  6776
5267  6551            EFSTA!551;      TABLE4-2        /STORE IT
5270  7376
5271  2121            EJXN!120;       MOR             /DONE 200 TIMES
5272  5265
5273  1031            EJA;            BASE+1          /JMP ALWAYS
5274  5501
```

```
                    /
                    /COMPARES DOUBLE PRECISION NUMBERS
                    /TABLE3 WITH TABLE4
                    /USING DOUBLE WORD DIRECT REFERENCE INSTRUCTIONS
                    /
5275  0411  .     DPCMPR, EFLDA!411!      TABLE3-2        /LOAD RANDOM ARG
5276  6776
5277  2431              EFSUB!431!        TABLE4-2        /SUBTRACT RESULT
5300  7376
5301  1001              EJEQ!             ERET            /IF THEY ARE EQUAL
5302  5054
5303  1031              EJA!              ERET
5304  5054
                    /
                    /COMPARES DOUBLE PRECISION NUMBERS
                    /TABLE3 WITH TABLE4
                    /USING SINGLE WORD INDIRECT REFERENCE INSTRUCTIONS
                    /
5305  0631        DPCPR1, EFLDA!631                       /LOAD RESULT FROM TABLE4
5306  2612              EFSUB!612                         /SUBTRACT RAN ARG FROM TABLE3
5307  1001              EJEQ!             ERET            /IF THEY ARE EQUAL
5310  5054
5311  1031              EJA!              ERET
5312  5054
                    /
```

```
5313  1101    RANGEN, ESETX!          XREG2
5314  5405
5315  1111            ESETB!          RBASE
5316  5361
5317  0101            ELDX!1!         0000
5320  0000
5321  0100            ELDX!0!         -200
5322  7600
5323  0006            ESTRTD
5324  0202            EFLDA!202
5325  1051            EJLT!1!         RCONST
5326  5335
5327  2203            EFSUB!203
5330  1051            EJLT!           RCONST
5331  5335
5332  0202            EFLDA!202
5333  2200            EFSUB!200
5334  6202            EFSTA!202
5335  0202    RCONST, EFLDA!202
5336  6411            EFSTA!411!      TABLE1
5337  6000
5340  0411    RGENA,  EFLDA!411!      TABLE1
5341  6000
5342  1201            EFADD!201
5343  6411            EFSTA!411!      TABLE3
5344  7000
5345  1201            EFADD!201
5346  6511            EFSTA!511!      TABLE1
5347  6000
5350  2101            EJXN!100!       RGENA
5351  5340
5352  6202            EFSTA!202
5353  1101            ESETX!          XREG1
5354  5375
5355  1111            ESETB!          BASE
5356  5500
5357  1031            EJA!1!          BASE+1
5360  5501
```

```
5361  0000    RBASE,  0000
5362  2657            2657
5363  1234            1234
5364  0000            0000
5365  0005            0005
5366  0011            0011
5367  0000            0000
5370  2200            2200
5371  0000            0000
5372  0000            0000
5373  1373            1373
5374  0000            0000

5375  0000    XREG1,  0
5376  0000            0
5377  0000            0
5400  0000            0
5401  0000            0
5402  0000            0
5403  0000            0
5404  0000            0
5405  0000    XREG2,  0
5406  0000            0
5407  0000            0
5410  0000            0
5411  0000            0
5412  0000            0
5413  0000            0
5414  0000            0
```

```
5415  0000    EEND,   0
                /
                /
5416  1031    LINK,   EJA!            FPP1-2
5417  4011
5420  2101            EJXN!100!       .+3
5421  5424
5422  1031            EJA!            LINK
5423  5416
5424  0006            ESTRTD
5425  0401            EFLDA!401!      LINK
5426  5416
5427  1401            EFADD!401!      TWO
5430  5432
5431  0007            EJAC

5432  0000    TWO,    0000
5433  0002            0002


5434  0000    XREG,   0

                /
                /BASE REGISTER TABLE
                /CONTAINS CONSTANTS-FLAGS-TEMPORARY STORAGE
                /BASE+1 AND BASE+2 CONTAIN RETURN JMP FROM SUBROUTINE
                /
```

```
        5500          *5500
5500    0000   BASE,  0                    /OFFSET OF 0
5501    0000          0
5502    0000          0
5503    0000          0                    /OFFSET OF 1
5504    0001          0001
5505    7376          7376
5506    0000          0                    /OFFSET OF 2
5507    0001          0001
5510    6776          6776
5511    0000          0                    /OFFSET OF 3
5512    0001          0001
5513    5775          5775
5514    0000          0                    /OFFSET OF 4
5515    0001          0001
5516    6375          6375
5517    0000          0                    /OFFSET OF 5
5520    0000          0
5521    0000          0
5522    0000          0                    /OFFSET OF 6
5523    0000          0
5524    0000          0
5525    0000          0                    /OFFSET OF 7
5526    0000          0
5527    0000          0
5530    0000          0                    /OFFSET OF 10
5531    0000          0
```

```
5532  0000    LOKTST, 0
5533  0000            0                       /OFFSET OF 11
5534  0000            0
5535  0000            0
5536  0000            0                       /OFFSET OF 12
5537  0000            0
5540  0000            0
5541  6000            6000                    /OFFSET OF 13
5542  3777            3777
5543  7777            7777
5544  0000            0000                    /OFFSET OF 14
5545  7777            7777
5546  7777            7777
5547  0000            0                       /OFFSET OF 15
5550  0000            0
5551  0000            0
5552  0000            0                       /OFFSET OF 16
5553  0000            0
5554  0000    LOKSEG, 0
5555  0000            0                       /OFFSET OF 17
5556  0000            0
5557  0001            0001
5560  0000            0                       /OFFSET OF 20
5561  0000            0
5562  0000            0
5563  0000            0                       /OFFSET OF 21
5564  0000            0
5565  0000            0
5566  0000    RESLT,  0                       /OFFSET OF 22
5567  0000            0
5570  0000            0
5571  0000            0000                    /OFFSET OF 23
5572  0000            0000
5573  7777    UFLO,   7777
5574  0000    ZBLOCK  6000-.                  /REST OF BASE TABLE
```

```
                        /
                        /
                        /
                        /FLOATING POINT MODE
                        /RANDOM DATA TABLE 1
                        /400 OCTAL LOCATIONS LONG
                        /
            6000                *6000
    6000    0000        TABLE1, 0
                        /
                        /
                        /FLOATING POINT MODE
                        /RESULTING ARG TABLE 2
                        /USED IN ASSOCIATION WITH TABLE 1
                        /400 OCTAL LOCATIONS LONG
                        /
            6400                *6400
    6400    0000        TABLE2, 0
                        /
                        /
                        /DOUBLE PRECISION MODE
                        /RANDOM DATA TABLE 3
                        /400 OCTAL LOCATIONS LONG
                        /
            7000                *7000
    7000    0000        TABLE3, 0
                        /
                        /
                        /DOUBLE PRECISION MODE
                        /RESULTING ARGUMENT TABLE 4
                        /USED IN ASSOCIATION WITH TABLE 3
                        /400 OCTAL LOCATIONS LONG
                        /
            7400                *7400
    7400    0000        TABLE4, 0
                        /
                        /
                        ////////////////////////////////////
                        ////////////END OF PROGRAM////////////
                        ////////////////////////////////////
```

```
0160  7600
0161  0260
0162  6676
0163  0351
0164  0353
0165  0352
0166  6672
0167  0725
0170  7154
0171  1005
0172  1001
0173  1000
0174  0072
0175  0071
0176  7772
0177  6703
```

```
0000  10000000  00000000  11111111  11111111  11111111  11111111  11111111  11111111
0100  11111111  11111111  11111111  11111111  11111111  00000000  11111111  11111111

0200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0300  11111111  11111111  11111000  00000000  00000000  00000001  11111111  11111111

0400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0500  11111111  11111111  11111111  11111111  11111111  11111000  00011111  11111111

0600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111


1000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1100  11111111  11111111  11111111  11111111  11111111  11111111  10000000  00000000

1200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1300  11110000  00000000  00000000  00000000  00000000  00000000  00000111  11111111

1400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1500  11111111  11111110  00000000  00000000  00000000  00000000  00000011  11111111

1600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111


2000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

2200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

2400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

2600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111


3000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  00000000
3100  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

3200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
3300  11111111  11111111  11111111  11111111  11000000  00001111  11111111  11111111

3400
3500

3600
3700
```

```
4000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

4200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

4400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

4600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

5000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

5200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

5400  11111111  11111111  11111111  11111000  00000000  00000000  00000000  00000000
5500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

5600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

6000  10000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
6100  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

6200
6300

6400  10000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
6500  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

6600
6700

7000  10000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
7100  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

7200
7300

7400  10000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
7500  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

7600
7700
```

```
AAPT    0210    ASC2    0765    CKSUB   6433    DIVIDE  3723
ACBIT   0542    ASC3    0766    CL      0543    DIVOK   4043
ACEQ    2531    ASC4    0767    CLRA    4574    DIVOV   2375
ACEXP   0052    ASC5    0770    CLRAC   4573    DIVZ    2364
ACLSW   0054    ASC77   0771    CLRACX  6666    DMODE   0542
ACLSW1  0055    ASCB    0751    CLRAX   6637    DOT     0413
ACLSW2  0056    ASCCW   0351    CLRB    4572    DOTCK   0414
ACLSW3  0057    ASCCW0  0024    CLRBX   6643    DOTER0  1257
ACMSW   0053    ASCDAT  0123    CLREG   6625    DOTERR  0432
ACNEZ   2273    ASCL    0103    CLRMQ   4571    DOTEX   0251
ADDAB   6735    ASCL1   0116    CLRMQX  6662    DPADD   2704
ADDR    2715    ASCL2   0117    CLRO    4570    DPADD0  2705
ADDX    5200    ASCWD   0352    CLROX   6647    DPADD1  2713
ADDX1   5201    ASCWD0  0025    CM      0673    DPCMPR  5275
ADDX2   5213    ASFLD   0633    CMEM    6523    DPCPR1  5305
ADDX3   5223    ASGET   0616    CMEM1   6471    DPFLOW  2440
ADDX4   5231    ASLOC   0677    CMEM2   6541    DPOVFL  2731
ADEQ    2626    ASPNTR  0662    CMEME   4567    DPSET   5206
ADEXT   7000    AST     2457    CMEMF   4566    DPSET1  5217
ADRET   6762    ASTER   7362    CNTRLS  7337    DPSET2  5250
AEQ     2501    ATX1    4331    CODE    1400    DTAM    5510
AEXT    0035    ATX2    4400    COMPO   5722    DVEND   4113
AEZ     7222    ATX3    4425    COMPOR  5762    DWRD    0035
AGZ     7256    ATX4    4444    COMREG  0002    EA      0452
ALN1    4143    ATX4A   4465    CRLF    3006    EADDX   0110
ALN2    4200    ATX5    4471    CRSW    0027    EALN    0010
ALN22   4220    BADNWS  6115    CSTATE  0731    EATX    0020
ALN3    4246    BASE    5500    DADD    0034    ECMPAS  5121
ALN4    4264    BCOM    7070    DATERR  1211    ECMPMD  5234
ALNEND  4325    BEGIN   0247    DATTXT  1735    ECMPR   5067
ALSW    0031    BEQ     2511    DECAPT  4565    ECMPR1  5103
ALSW1   0032    BEXT    0043    DECOP   4564    ECMPRC  5073
ALSW2   0033    BLSW    0037    DEP     2052    ECONO   2760
ALSW3   0034    BLSW1   0040    DEP11   2054    EEND    5415
ALZ     7250    BLSW2   0041    DEP12   2117    EFADD   1000
AMBO    4577    BLSW3   0042    DEP13   2136    EFADDM  5000
AMSW    0030    BMSW    0036    DEP130  2205    EFCLA   0002
APBO    4576    BRANCH  6341    DEP131  2211    EFDIV   3000
APEQ    2571    BREG    0036    DEP132  2215    EFEXIT  0000
APT     0211    BSIGN   6420    DEP14   2223    EFLDA   0000
APTC    5425    CADS    3332    DEP144  2264    EFMUL   4000
APTDEC  6421    CADST   7732    DEP15   2301    EFMULM  7000
APTERR  2721    CAPT    4575    DEPEND  2415    EFNEG   0003
APTPAC  6011    CAROUT  0021    DEPM3   2200    EFNOP   0040
APTSAV  1042    CARYIN  0007    DIV0    3611    EFNORM  0004
AREG    0030    CFLG    6661    DIV00   3657    EFSTA   6000
ARITH   2030    CKLINK  7243    DIV01   3663    EFSUB   2000
AS      0600    CKMEM   2132    DIV02   3667    EJA     1031
ASADDR  2710    CKO     7736    DIV0A   3636    EJAC    0007
ASC     7154    CKOP    7737    DIV1    3676    EJEQ    1001
ASC1    0764    CKOPAD  6214    DIV2    3717    EJGT    1061
ASC10   0725    CKST    6444    DIV3    4104    EJLT    1051
```

```
EJNE    1041        EXIT00  2477        FECH34  1427        FPP2ER  4071
EJSA    1121        EXIT01  2504        FECH4   1436        FPP2R   4047
EJSR    1131        EXIT02  2511        FECH42  1445        FPP3    4107
EJXN    2001        EXIT03  2516        FECH5   1467        FPP3E   4136
EL      7404        EXIT0U  2434        FECH6   1475        FPP3ER  4144
EL10    1407        EXIT1   2535        FECH60  1515        FPP3R   4116
ELDX    0100        EXIT10  2450        FECH7   1524        FPP4    4165
EMDCK   5245        EXIT11  2452        FECH72  1600        FPP4R   4174
EMEM    0105        EXIT12  2454        FECH73  1620        FPP5    4214
EMODE   0541        EXIT1U  2442        FEND    1621        FPP5E   4245
EMORE   5023        EXIT2   2546        FETCH   1261        FPP5ER  4253
END15   2341        EXIT2U  2461        FLAG1   0726        FPP5R   4223
ENDDIV  4015        EXIT3   2600        FLAG2   0727        FPP6    4277
ENINIT  1063        EXIT4   2611        FLAG3   0730        FPP6R   4306
ENOFF   7542        EXIT5   2622        FLDA    5304        FPP7    4327
ENON    7551        EXIT6   2634        FLDIV   4061        FPP7E   4371
ENT     2076        EXIT7   2641        FLDX    4114        FPP7ER  4377
ENTER   4563        EXITSW  0725        FMAINT  6561        FPP7R   4336
ENTSTP  7734        EXITWD  0264        FMDCK   5145        FPPRUN  1043
ENTTXT  6106        EXSAV   2467        FMIS    2044        FPRST   6556
ENTYP   6105        EXSTP   7735        FMODE   0540        FPST    6555
EOP     2432        EXTA    0030        FMULT   3302        FSET    4745
EPM     0003        EXWD    0732        FNEG    5066        FSET1   4757
ER1     5505        FACFR   0053        FNORM   5100        FSIS    2000
ER2     5506        FAD010  3211        FOEQ    2471        FSTA    5235
ER3     5507        FAD011  3215        FOIS    2034        FSTD    5057
ERET    5054        FAD012  3221        FOPIS   2065        FSTE    5354
EROUT   2414        FAD01A  3200        FP11ER  4530        FSTEP   4562
ERR     6707        FAD01B  3205        FP13ER  4653        FSTEPM  1045
ERR10   1200        FADD0   2737        FPAUSE  5122        FSTF    5042
ERREG   1262        FADD01  3126        FPCOM   6553        FSTIS   1770
ERRET   1255        FADD02  3230        FPHLT   6554        FXTA    4500
ERROR   0000        FADD1   3013        FPICL   6552        FXMD    5175
ERRTI   1256        FADD10  3033        FPINT   6551        GAPT    6121
ERRTP   1223        FADD11  3041        FPIST   6557        GET     6135
ESET    5033        FADD12  3047        FPP1    4013        GETAPT  4561
ESET1   5005        FADD2   3060        FPP10   4434        GETOP   4560
ESETB   1111        FADD3   3242        FPP10R  4483        GETPC   4557
ESETX   1101        FADEND  3275        FPP11   4467        GETREG  0211
ESTART  4000        FALN    4142        FPP11E  4522        GETX    4556
ESTOP   6120        FAPIS   2012        FPP11R  4476        GETXT   0240
ESTRTD  0006        FASCK   5117        FPP12   4555        GO11    4510
ESTRTE  0050        FATX    4330        FPP12R  4564        GO11E   4542
ESTRTF  0005        FCLA    5144        FPP13   4610        GO12    4576
ETXT1   1225        FCMPAS  5107        FPP13E  4645        GO13    4633
ETXT2   1226        FCMPMD  5135        FPP13R  4617        GO13E   4670
EX      0436        FCMPR   5045        FPP14   4703        GO14    4724
EXADDR  7733        FECH0   1271        FPP14R  4712        GOP     6131
EXCOM   2663        FECH1   1317        FPP15   4744        GPC     6125
EXEC    1622        FECH2   1337        FPP1R   4024        HALT    7402
EXIT    2427        FECH24  1411        FPP2    4040        IGNFL   0734
EXIT0   2524        FECH3   1421        FPP2E   4063        INC1    5416
```

INC2	5400	KEYCK0	0145	MDFLAG	0737	OPOS	2400
INC3	5407	KEYCKL	0140	MEMEQ	2646	OREG	0022
INCAPT	4555	KEYL1	6676	MEMINS	6455	ORERR	6000
INCOP	4554	LCF	6652	MFLD	0711	OUTFLD	0232
INCOR	4553	LDA0	5305	MMEM	0106	OUTSTP	2677
INCPC	4552	LDA1	5314	MODE	0530	OVFL	0733
INCST	4551	LDA10	5325	MOR	5265	PAPT	0072
INCX	4550	LDA11	5331	MORE	4775	PAUS0	5123
INDEX	6317	LDA12	5335	MOVEX	4540	PAUS1	5136
INEND	1257	LDA2	5344	MQEQ	2521	PBASE	0074
INIT	1046	LDAC	0650	MQEXT	0051	PCEQ	2616
INIT0	1066	LDBUF	0515	MQLSW	0045	PFAC	0052
INIT1	1120	LDEXT	4000	MQLSW1	0046	PFADD	2736
INIT2	1134	LDOP	0456	MQLSW2	0047	PFDIV	3606
INIT3	1200	LDX1	4115	MQLSW3	0050	PFPC	0066
INIT4	1206	LDX2	4127	MQMSW	0044	PINCF	6616
INIT5	1213	LDX3	4136	MQREG	0044	PINDEX	6600
INIT50	1225	LEFT	4000	MSTATE	5516	PIR	0005
INIT51	1232	LEXT	5717	MT1	0735	PIREQ	2636
INIT52	1237	LINK	5416	MT2	0736	PQEQ	2461
INIT6	1247	LLB	6654	MTEXT	2054	PROCES	1634
INS0	7277	LMEM	0107	MULEND	3605	PRT	1067
INS21	7317	LMEM1	0110	MULT0	3305	PRT8	1104
INSTEP	2666	LMEM2	0111	MULT00	3400	PSBM	5511
JAC	5027	LMEM3	0112	MULT01	3404	PSEQ	2561
JAC1	5030	LOAD	5636	MULT02	3410	PSTAT	0004
JAL	5000	LOADA	4547	MULT1	3417	PSTEPM	1044
JEQ	4731	LOADAC	4546	MULT1A	3452	PT10	1156
JFALSE	5024	LOADB	4545	MULT2	3454	PUTX	4536
JGE	4734	LOADMQ	4544	MULT21	3507	PX0	0224
JGT	4750	LOADO	4543	MULT3	3600	PX1	0225
JLE	4737	LOADOP	4542	NEG1	5067	PX2	0226
JLT	4745	LOADPC	4541	NEGB	7090	PX3	0227
JNE	4742	LODX	0466	NOREP	1247	PX4	0230
JSA	4700	LOKSEG	5554	NORM	4537	PX5	0231
JSA2	4701	LOKTST	5532	NORM1	5101	PX6	0232
JSA3	4714	LOOPR	7440	NTMSC	3315	PX7	0233
JSA4	4722	LP08	1112	NXSHFT	6311	PXP	0006
JSR	4640	LP12	1125	OCTWD	0393	QMK	3004
JSR1	4641	LP12E	1146	OCTWD0	0026	QUONEG	4047
JSR2	4651	LPC	6666	OEXT	0027	RA	0476
JSR3	4662	LPR	6664	OEZ	7200	RANGEN	5313
JSR4	4670	LREG	5721	OLSW	0023	RAPT	6565
JTRUE	5005	LSD	6661	OLSW1	0024	RBASE	5361
JTRUE1	5006	LSE	6651	OLSW2	0025	RCONST	5335
JXN	4537	LSF	6661	OLSW3	0026	RDOP	6566
JXN1	4540	LSHFT	6567	OMSW	0022	READ	0271
JXN2	4600	LSR	6663	OPADR	0070	REGAD	0030
JXN2A	4613	LTRUNK	6232	OPDEC	6723	REGEQ	1746
JXN3	4623	MADD	3455	OPEQ	2541	REGFLD	0033
JXNEND	4637	MCODE	6534	OPERND	4100	REGS	0021
KEYCK	0254	MDEC	2736	OPLUS1	6445	REND	7474

REPMEN	4535	SETUP	6051	SWAP	1432	TNEG	2305
RESLT	5566	SETX	5154	SWCNT	7477	TNOP	1424
RESPS8	1502	SETX1	5155	T1	0000	TNORM	2311
RET	0404	SEX	7500	T10	0020	TOA	5605
RETINC	5065	SFLG	6662	T100	1154	TOAC	5541
RETURN	0354	SH	0663	T2	0001	TOB	5612
REX	7507	SHEQ	2656	T20	0021	TOFECH	2464
REXT	7454	SHFCNT	0723	T212	1151	TOMEM	4520
RF	0506	SHFLAG	0722	T215	1150	TOMQ	5624
RFA	0517	SHFOP	3112	T240	1155	TOO	5600
RFB	0527	SHFTB	4526	T77	1152	TOOP	5631
RGENA	5340	SHFTO	4525	TABLE1	6000	TOPC	5534
RIGHT	0000	SHIFT	4524	TABLE2	6400	TOTMP	5617
RNDFF	5121	SHIFTB	6400	TABLE3	7000	TOUT	1051
ROEPM	6567	SHIFTO	7265	TABLE4	7400	TPAUSE	2315
ROLSW	6564	SHIFTR	7410	TADDX	2350	TPNTR	0022
ROMSW	6563	SHORT	5657	TALN	2207	TPNTR1	0460
ROT	7434	SHREG	0724	TATX	2212	TRACE	4517
ROUT	3007	SKPEPM	7533	TCHECK	0251	TRAP1	1644
RPNTR	0701	SOEZ	4523	TCHR	1066	TRAPED	1643
RSEQ	2606	SPEC1	1641	TCLA	2262	TREG	0060
RST	7525	SPEC2	1660	TCNTR	0023	TRET	1013
RSTATE	6562	SPEC20	1715	TDAD	0461	TREXIT	5462
RT	0502	SPEC21	1671	TDATA	0277	TRSKP	6200
RTAD	0032	SPEC3	2000	TDEP	2133	TSETB	2340
RUBOUT	7125	SPECFL	5442	TDFLD	0310	TSETX	2344
RUTXT	7137	SST	7516	TDIV	2177	TSTA	2354
RWCNT	0031	SSTEP	3254	TDPAS	2140	TSTD	2300
SA	0400	STA0	5236	TEND	1457	TSTE	2273
SAEZ	4534	STA1	5246	TEST	4000	TSTF	2266
SAGZ	4533	STA2	5252	TEXIT	2114	TTRAP	2120
SALZ	4532	STA20	5261	TFADD	2154	TTRUE	2234
SAVAPT	5461	STA21	5265	TFALSE	2247	TTY	1036
SAVE	4531	STA22	5271	TFETCH	2110	TWD	0634
SAVMEM	0712	STAEND	5300	TINIT	2103	TWO	5432
SAVO	6004	START	0234	TITL	1422	TXTA	2215
SAVOM	0100	STATEQ	2551	TITLE	7367	TY	7400
SAVOP	6231	STEP	3200	TJAC	2231	TY10	0600
SAVSN	6405	STEPER	3323	TJSA	2226	TY10L	0611
SCAN	0400	STEPGO	3274	TJSR	2223	TYBUF	0464
SCANL	0407	STEPM	0713	TJXN	2220	TYP	6703
SCEND	0453	STEPSW	7732	TLDA	2360	TYPA	1024
SEPM	4530	STERR	7114	TLDX	2204	TYPE	1000
SEPMEN	4527	STFPP	1000	TLSW	0061	TYPL	0036
SETB	4753	STINC	6715	TLSW1	0062	TYPLD	0072
SETB1	4754	STORA	4522	TLSW2	0063	TYPLX	0062
SETFL	7040	STORB	4521	TLSW3	0064	TYPNCR	7143
SETRET	7354	STRA	6025	TM212	1157	TYREG	0200
SETST3	6250	STRB	6037	TM3	1160	UFLO	5573
SETST5	6241	STRTER	2021	TM40	1153	VFER	2424
SETST6	6257	STSAVE	7124	TMSW	0060	WORD	0265
SETTAB	1600	SUBAB	7043	TMULT	2171	WORD0	0124

```
        WORDL   6672
        X0      0200
        X0ADR   0076
        X1      0201
        X2      0202
        X3      0203
        X4      0204
        X5      0205
        X6      0206
        X7      0207
        XEQ     2576
        XGETX   6335
        XNCK    6302
        XNORM   6266
        XPLUS1  6330
        XPUTX   7103
        XREG    5434
        XREG1   5375
        XREG2   5405
        XTA1    4501
        XTA2    4513


ERRORS DETECTED: 0

LINKS GENERATED: 437

RUN-TIME:  96 SECONDS

4K CORE USED
```