TRACE

IDENTIFICATION

PRODUCT CODE: MAINDEC 12-D0LC-D

PRODUCT NAME: FPP-12 TRACE

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 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.

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 MAINTE- NANCE 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.

2. REQUIREMENTS

2.1 EQUIPMENT

1) AN FPP-12 FLOATING POINT PROCESSOR

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.

THE FPP-12 PROGRAM RUNNING AT THIS TIME IS THE FPP-12 CODE EXTRACTED FROM THE FPP-12 EXERCISER MAINDEC 12-D0QA.

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 12 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. IN THE PROGRAM THERE IS NO DIFFERENCE WHICH TERMINATOR IS USED. 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 FPP-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. WITH THE EXCEPTION OF THE .TY COMMAND THESE ARGUMENTS WILL BE GROUPS OF 1 - 4 OCTAL DIGITS. EACH COMMAND IS SEPERATED FROM ITS ARGUMENT BY A TERMINATOR (SPACE OR RETURN). EACH ARGUMENT IS ALSO TERMINATED BY EITHER OF THE TWO TERMINATORS.
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 (Ø 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)
.OP XXXX	(C)
.TY XX	(1 OR 3)
.SA ADDR	(1)
.RA ADDR	(1)
.EA ADDR	(1)
.CL	(0)
.RT ADDR	(1)
.RF ADDR	(1)
.EX	(0)
.SH XX	(1)

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:
	PO	SIMULATED 0 REGISTER
	FO	FPP-12 0 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)

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 TIME STATE COUNTER
01	BIT 1 OF THE TIME STATE COUTNER
02	BIT 2 OF THE TIME STATE COUNTER
03	BIT 3 OF THE TIME STATE COUNTER
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	AST0 SHFT FAC FRAC H (NOT USED IN TRACE)
11	AST1 NO SHFT (1) H (NOT USED IN TRACE)

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.
,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 0 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,

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,

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,

8,1,4 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 FPP-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.

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,

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 RULES PERTAINING TO THE APT AND AUXILIARY ADDRESSES 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, SWITCH OPTIONS (USED ONLY IN TRACE MODE)

SR@2 (1) INHIBIT ERROR HALT

SR@1 (1) INHIBIT ERROR TYPEOUTS

SR@2 (1) RESTART PROGRAM AFTER AN EXIT, USED IN BOTH TRACE AND FAST MODE.

SR@3 (1) SINGLE STATE MODE, OVERRIDES THE ;SA ,RA ;CL COMMANDS (SEE "SINGLE STATE CONTROLS" 8,2)

SR@4 (1) ENTER STOP, STOPS THE PROGRAM EACH TIME A MAJOR STATE IS ENTERED, TYPE ALT MODE TO CONTINUE.

SR@5 (1) TTY TRACE, TYPES EACH MAJOR STATE AS IT IS ENTERED.

10. 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 * TYPE * FPP YYYY YYYY * PROG ZZZZ ZZZZ"

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.+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 AFTER AN "FSTEP" INSTRUCTION, THE ERROR WAS GENERATED IN THE PREVIOUS TIME STATE.

TYPE TYPE OF ERROR IS THE REGISTER IN WHICH THE ERROR WAS DETECTED (IE, 0 REG).

YYYY THE ERRONEOUS DATA READ FROM THE FPP-12 HARDWARE REGISTER.

ZZZZ WHAT THE DATA SHOULD BE AS DETERMINED BY THE TRACE SIMULATOR.

EXAMPLE

ERROR 2117 * 0 REG * FPP 1234 1234 * PROG 1234 1235

THE P.C. SAYS THAT THE ERROR OCCURED IN DEPOSIT
STATE 11. (THE P.C. POINTS TO THE "FSTEP" TO ENTER
DEPOSIT STATE 12).

THE BAD DATA WAS DETECTED IN THE 0 REGISTER.

THE FPP-12 0 REGISTER CONTAINS 1234 1234

THE FPP-12 0 REGISTER SHOULD CONTAIN 1234 1235.

IT APPEARS THAT THE FPP-12 DID NOT ROUND UP WHEN
IT SHOULD HAVE.

NOTE ******

THE DATA WORDS ARE SPLIT INTO TWO 12 BIT PARTS
EVEN THOUGH THE REGISTER IS A SINGLE 24 BIT REGISTER.
THIS IS SO THAT IT IS EASIER TO DETERMINE WHETHER THE
ERROR WAS IN TTHE MSW OR LSW OF THE REGISTER. THIS IS
DONE BECAUSE MOST SIGNALS IN THE HARDWARE REFERENCE
THE MSW OR LSW OF A REGISTER.

10.1 TYPES OF ERRORS

TIME STATE	AN ERROR IN THE STATE GENERATER, THIS COULD BE A TIME STATE OR MAJOR STATE ERROR.
NOTE ****	SEE PARAGRAPH 7.1 FOR A DESCRIPTION OF THE TIME STATE REGISTER
O REG	A DATA ERROR IN THE O REGISTER.
STATUS	AN ERROR IN THE STATUS REGISTER.
OP ADDR	OP ADDRESS ERROR.
ADDRS	ERROR IN THE ADDRS REGISTER. (APT ADDRESS)
MEMORY	ERRONEOUS DATA STORED INTO MEMORY BY THE FPP-12.
NOTE	THIS ERROR WILL OCCUR IF THE FPP-12 STORES DATA INTO A NON-EXISTENT MEMORY FIELD.

10.2 ERROR HALT

IF SR00=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.

```
/       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

/       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
/       .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 PS8
/       CNTRL C         RETURN TO DTA MONITOR
```

```
/       TYPEOUTS

/       PC      PROGRAM C REGISTER
/       FC      FPP C REGISTER
/       A       A REGISTER
/       B       B REGISTER
/       MQ      MQ REGISTER
/       AC      PROGRAM FAC
/       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
/       AL      ALL OF ABOVE
/       F AAAA N        FIELD, 12 BIT ADDR, COUNT
```

```
                PMODE

                /MACRO DEFINITIONS


                DEFINE  NPAGE<
                        JMP I   ('.+20007600
                        PAGE>



                /BUFFERS FOR FPP CODE

4000            TEST=4000
4100            OPERND=4100


                /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
```

```
                / INSTRUCTION DEFINITIONS

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

```
                / REGISTER DEFINITIONS

7021    OMSW=   OREG
7022    OLSW=   OREG+1
7023    OEXT=   OREG+2

7024    AMSW=   AREG
7025    ALSW=   AREG+1
7026    AEXT=   AREG+2

0027    BMSW=   BREG
0030    BLSW=   BREG+1
0031    BEXT=   BREG+2

0032    MQMSW=  MQREG
0033    MQLSW=  MQREG+1
0034    MQEXT=  MQREG+2

0062    TMSW=   TREG
0063    TLSW=   TREG+1

0035    ACEXP=  PFAC
0036    ACMSW=  PFAC+1
0037    ACLSW=  PFAC+2

0036    FACFR=  PFAC+1

0020    REGS=   OREG-1
0000    ERROR=  0000
7402    HALT=   7402
4000    LDEXT=  4000
```

```
                0000            *0
                                /TEMPORARY LOCATIONS

0000    0000    MT1,    0
0001    0000    MT2,    0
0002    0000    SAVOM,  0
0003    0000    SAVOL,  0
0004    0000    MDFLAG, 0
0005    0000    EMEM,   0
0006    0000    MMEM,   0
0007    0000    LMEM,   0



                0020    *20

0020    5724'           JMP     BEGIN           /FOR USE BY START 20 SWITCH ON PDP-12
```

```
                        /SIMULATED HARDWARE REGISTERS

0021  0000  OREG,   0                   /O REGISTER
0022  0000          0
0023  0000          0

0024  0000  AREG,   0                   /A REGISTER
0025  0000          0
0026  0000          0

0027  0000  BREG,   0                   /B REGISTER
0030  0000          0
0031  0000          0

0032  0000  MQREG,  0                   /MQ REGISTER
0033  0000          0
0034  0000          0

0035  0000  PFAC,   0                   /FPP ACCUMULATOR
0036  0000          0
0037  0000          0
0040  0000          0

0041  0000  PFPC,   0                   /FPP PROGRAM COUNTER
0042  0000          0
0043  0000          0

0044  0000  OPADR,  0                   /OP ADDRESS
0045  0000          0
0046  0000          0
```

```
0047  0000  PAPT,   0                    /ADDRS
0050  0000          0

0051  0000  PIR,    0                    /FPP INSTRUCTION REGISTER

0052  0000  COMREG, 0                    /FPP COMMAND REGISTER

0053  0000  PSTAT,  0                    /FPP STATUS REGISTER

0054  0000  PBASE,  0                    /BASE REGISTER (P0 ADDRESS)
0055  0000          0
0056  0000          0

0057  0200  PXP,    X0                   /PROGRAM INDEX POINTER
0060  0000          0
0061  0000          0

0062  0000  TREG,   0                    /TEMPORARY REGISTER
0063  0000          0
0064  0000          0

0065  0000  X0ADR,  0                    /FPP INDEX POINTER
0066  0000          0
```

```
                        /FLIP FLOPS - TEMPORARY LOCATIONS - CONSTANTS


0067  0000    SHFLAG, 0                       /SPECIAL SHIFT FLAG
0070  0000    SHFCNT, 0                       /SPECIAL SHIFT COUNT
0071  0000    SHREG,  0                       /FPP SHIFT COUNTER
0072  0000    EXITSW, 0                       /EXIT SWITCH


0073  0000    FLAG1,  0                       /GENERAL PURPOSE FLIP FLOPS
0074  0000    FLAG2,  0
0075  0000    FLAG3,  0

0076  0000    CARYIN, 0                       /ADDER CARRIES
0077  0000    CAROUT, 0

0100  0000    D27,    0                       /DOUBLE PRECISION 27
0101  0027            27
0102  0000            0

0103  0000    T1,     0                       /TEMPORARY STORAGE
0104  0000    T2,     0

0105  0000    CSTATE, 0                       /PROGRAM STATE GENERATOR
0106  0000    STEPSW, 0                       /PROGRAM SINGLE STATE SWITCH
0107  0210    AAPT,   APT                     /ADDRESS OF THE APT
0110  0000    EXWD,   0
0111  0000    OVFL,   0                       /OVERFLOW
0112  0000    IGNFL,  0                       /IGNORE UNDERFLOW FLIP FLOP
0113  0000    EXADDR, 0                       /EXIT ADDRESS (,EX COMMAND)
0114  0000    ENTSTP, 0                       /ENTER STEP ADDRESS (,SA COMMAND)
0115  0000    EXSTP,  0                       /EXIT STEP ADDRESS (,RA COMMAND)
0116  0000    MCNT,   0
0117  0000    CKO,    0                       /CHECK O REGISTER SWITCH
0120  0000    CKOP,   0                       /CHECK OP ADDRESS SWITCH
0121  0000    SAVOP,  0
```

```
                0200            PAGE


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  0000    APT,    0                       /APT USED WHEN NO ADDRESS
0211  0000            0                       /IS GIVEN WITH ,RT OR ,RF COMMANDS
0212  0000            0
0213  0000            0
0214  0000            0
0215  0000            0
0216  0000            0
0217  0000            0

0220  0000    PX0,    0                       /INDEX REGISTERS USED WITH ABOVE APT
0221  0000    PX1,    0
0222  0000    PX2,    0
0223  0000    PX3,    0
0224  0000    PX4,    0
0225  0000    PX5,    0
0226  0000    PX6,    0
0227  0000    PX7,    0
```

```
0232  7300    START,  CLA CLL
0231  1777'            TAD     APTSAV          /GET LAST APT ADDRESS
0232  3347            DCA     OCTWD           /STORE FOR RERUN
0233  3776'           DCA     FPPRUN          /CLEAR FPP RUN FF
0234  7604            LAS                     /GET SWITCH REGISTER
0235  7106            CLL RTL
0236  7710            SPA CLA                 /SR02 = 1 ?
0237  5775'           JMP     STFPP+2         /YES = RESTART PROGRAM
0240  4774'           JMS     TYP             /TYPE END TRACE
0241  2354            EOP
0242  0000            0
0243  3776'   BEGIN,  DCA     FPPRUN          /CLEAR PROGRAM RUN FF
0244  4773'           JMS     TITLE           /TYPE PROGRAM TITLE
0245  4772'   DOTEX,  JMS     ASTER           /TYPE AN ASTERISK (*)
0246  4261            JMS     WORD            /ALLOW INPUT FROM TTY
0247  5246            JMP     .-1             /PROGRAM SHOULD NOT RETURN HERE


0250  0254    KEYCK,  .+4                     /GET TTY INPUT BETWEEN TIME STATES
0251  7300            CLA CLL                 /AND AFTER ERROR TYPEOUT
0252  1250            TAD     KEYCK           /GET RETURN ADDRESS
0253  3350            DCA     RETURN          /SAVE FOR EXIT TO TRACE
0254  6032            KCC
0255  4772'           JMS     ASTER           /TYPE AN ASTERISK (*)
0256  4261            JMS     WORD            /ALLOW TTY INPUT
0257  5650            JMP I   KEYCK           /RETURN TO TRACE

0260  5661    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

0261  0000    WORD,   0
0262  7300            CLA CLL
0263  3347            DCA     OCTWD           /CLEAR LINKING WORDS
0264  3346            DCA     ASCWD
0265  6031    READ,   KSF                     /WAIT FOR TTY
0266  5265            JMP     .-1
0267  6036            KRB                     /GET CHARACTER FROM TTY
0270  3345            DCA     ASCCH           /SAVE CHARACTER
0271  1345            TAD     ASCCH           /ECHO CHARACTER
0272  6046            TLS
0273  6041            TSF
0274  5273            JMP     .-1
0275  6042            TCF
0276  1371            TAD     (-215           /CHECK FOR SPECIAL CHARACTERS
0277  7450            SNA                     /RETURN ?
0300  5770'           JMP     RET
0301  1367            TAD     (-23            /SPACE ?
0302  7450            SNA
0303  5661            JMP I   WORD
0304  1366            TAD     (-16            /PERIOD ?
0305  7450            SNA
0306  5765'           JMP     DOT
0307  1364            TAD     (-121           /RUBOUT ?
0310  7450            SNA
0311  5763'           JMP     RUBOUT
0312  1362            TAD     (2              /ALT MODE ?
0313  7650            SNA CLA
0314  5750            JMP I   RETURN          /RETURN TO SIMULATION
0315  1345            TAD     ASCCH           /GET ASCII CHARACTER
0316  1361            TAD     (-240
0317  7710            SPA CLA                 /CNTRL KEY ?
0320  5760'           JMP     TREXIT          /YES - EXIT TO MONITOR
```

```
0321  1345          TAD     ASCCH           /GET CHARACTER
0322  0357          AND     (77             /STRIP TO 6 BITS
0323  3345          DCA     ASCCH           /SAVE 6 BIT ASCII CHARACTER
0324  1346          TAD     ASCWD           /GET ASCII WORD
0325  0357          AND     (77             /SAVE LAST CHARACTER
0326  7106          RTL CLL                 /MOVE TO LEFT HALF OF WORD
0327  7006          RTL
0330  7006          RTL
0331  1345          TAD     ASCCH           /INSERT NEW CHARACTER
0332  3346          DCA     ASCWD           /SAVE PACKED ASCII WORD
0333  1345          TAD     ASCCH           /GET ASCII CHARACTER
0334  0356          AND     (7              /EXTRACT OCTAL DIGIT
0335  3345          DCA     ASCCH           /SAVE OCTAL DIGIT
0336  1347          TAD     OCTWD           /GET OCTAL WORD
0337  0355          AND     (777            /SAVE LAST 3 DIGITS
0340  7104          RAL CLL                 /MOVE 1 DIGIT LEFT
0341  7006          RTL
0342  1345          TAD     ASCCH           /INSERT NEW DIGIT
0343  3347          DCA     OCTWD           /SAVE OCTAL WORD
0344  5265          JMP     READ            /GET NEXT CHARACTER

0345  0000  ASCCH,  0
0346  0000  ASCWD,  0
0347  0000  OCTWD,  0
0350  1020  RETURN, STFPP+2                /SPECIAL RETURN ADDRESS
0351  0000  STSAVE, 0




0355  0777
0356  0007
0357  0077
0360  5705
0361  7540
0362  0002
0363  1000
0364  7657
0365  0413
0366  7762
0367  7755
0370  0404
0371  7563
0372  7261
0373  7266
0374  7313
0375  1020
0376  1045
0377  1044
      0400          PAGE
```

```
                        /SET ENTER SINGLE STATE ADDRESS

0400  4777'  SA,    JMS     WORD            /ALLOW TTY INPUT
0401  1776'        TAD     OCTWD           /GET ADDRESS
0402  3114        DCA     ENTSTP          /SET ENTER ADDRESS LOCATION
0403  5775'        JMP     DOTEX           /EXIT

0404  1374  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  5773'        JMP     EXITWD          /EXIT WORD ROUTINE

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

                        /BRANCH TO COMMAND ROUTINE

0414  1372  DOTCK, TAD     (BRANCH         /ADDRESS OF CHARACTER TABLE
0415  3010        DCA     10
0416  1371        TAD     (CNTRLS-1       /ADDRESS OF ROUTINE TABLE
0417  3011        DCA     11
0420  1411        TAD I   11              /GET ROUTINE ADDRESS
0421  3103        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  1770'        TAD     ASCWD           /COMPARE WITH TTY ASCII INPUT
0427  7640        SZA CLA                 /COMPARE ?
0430  5220        JMP     .-10            /NO - TRY AGAIN
0431  5503        JMP I   T1              /YES = GO TO ROUTINE

                        /COMMAND INPUT ERROR

0432  4767'  DOTERR, JMS     TYP             /TYPE ??
0433  2736        QMK
0434  0000        0
0435  5766'        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  4765'           JMS     SETRET          /SETUP RETURN
0451  5775'           JMP     DOTEX           /EXIT ROUTINE

                      /SET EXIT ADDRESS

0452  4777'   EA,     JMS     WORD            /ALLOW TTY INPUT
0453  1776'           TAD     OCTWD           /GET ADDRESS
0454  3113            DCA     EXADDR          /SAVE IN COMPARE WORD
0455  5775'           JMP     DOTEX           /EXIT ROUTINE


                      /LOAD OPERAND TABLE

0456  1364    LDOP,   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     .-5             /GET NEXT WORD

                      /LOAD INDEX REGISTER (ENTERED FROM ,AS ROUTINE)
0466  1776'   LDDX,   TAD     OCTWD           /GET OCTAL WORD
0467  0363            AND     (7              /EXTRACT INDEX REGISTER
0470  1362            TAD     (PX0            /ADD ADDRESS OF INDEX REG 0
0471  3103            DCA     T1              /SAVE INDEX ADDRESS
0472  4777'           JMS     WORD            /ALLOW TTY INPUT
0473  1776'           TAD     OCTWD           /GET OCTAL ARGUMENT
0474  3503            DCA I   T1              /SET INDEX REGISTER
0475  5761'           JMP     AS+4            /GO BACK TO AS ROUTINE
```

```
                        /SET THE COMMAND REGISTER

0476  4777/  CM,   JMS     WORD            /ALLOW TTY INPUT
0477  1776/        TAD     OCTWD           /GET OCTAL WORD
0500  3052          DCA     COMREG          /SET COMMAND REGISTER
0501  5775/        JMP     DOTEX           /EXIT

                        /SET RESET SINGLE STATE ADDRESS

0502  4777/  RA,   JMS     WORD            /ALLOW TTY INPUT
0503  1776/        TAD     OCTWD           /GET OCTAL WORD
0504  3115          DCA     EXSTP           /SET EXIT COMPARE WORD
0505  5775/        JMP     DOTEX           /EXIT

                        /RUN TRACE MODE

0506  4760/  RT,   JMS     STFPP           /SETUP TO START FPP
0507  5757/        JMP     INIT            /START TRACE AND FPP IN INIATE


                        /RUN FAST MODE

0510  4760/  RF,   JMS     STFPP           /SETUP TO START FPP
0511  1052          TAD     COMREG          /GET COMMAND REGISTER
0512  6553          FPCOM                   /SET FPP-12 COMMAND REGISTER
0513  7200          CLA
0514  1107          TAD     AAPT            /GET ADDRESS OF APT
0515  6555          FPST                    /FPP-12 START ERROR
0516  7402          HLT
0517  1356  RFA,    TAD     (-20            /GET TIME CONSTANT
0520  3103          DCA     T1              /SET TIMER
0521  3104          DCA     T2
0522  1341          TAD     ACBIT           /GET CONTENTS OF BIT PATTERN
0523  7040          CMA                     /COMPLIMENT
0524  3341          DCA     ACBIT           /RESTORE
0525  1341          TAD     ACBIT           /LOAD BIT PATTERN
0526  6031  RFB,    KSF                     /CHECK KEYBOARD FLAG
0527  7410          SKP
0530  4755/        JMS     KEYCK           /GET TTY INPUT
0531  6557          FPIST                   /IS FPP-12 FINISHED ?
0532  7410          SKP                     /NO
0533  5754/        JMP     START           /YES - EXIT
0534  2104          ISZ     T2              /TIMEOUT BEFORE COMPLIMENTING AC
0535  5326          JMP     RFB             /AC FLASHES ON AND OFF WHILE WAITING
0536  2103          ISZ     T1
0537  5326          JMP     RFB
0540  5317          JMP     RFA
```

```
0541  0000  ACBIT,  0

0542  3106  CL,     DCA     STEPSW          /CLEAR STEP SWITCH
0543  5775  /       JMP     DOTEX           /EXIT

0554  0230
0555  0250
0556  7760
0557  1064
0560  1016
0561  0604
0562  0220
0563  0007
0564  4077
0565  7254
0566  0244
0567  7313
0570  0346
0571  7237
0572  1046
0573  0260
0574  0212
0575  0245
0576  0347
0577  0261
      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  2646            ASADDR
 0613  4772'           JMS     TYPNCR          /TYPE ADDRESS
 0614  2646            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  5273            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)

 2650  1356   LDAC,    TAD     (APT+4          /ADDRESS OF FAC
 2651  3011            DCA     11
 2652  1355            TAD     (-3             /WORD COUNT
 2653  3103            DCA     T1
 2654  4771'           JMS     WORD            /ALLOW TTY INPUT
 2655  1763'           TAD     OCTWD           /GET OCTAL WORD
 2656  3411            DCA I   11              /STORE IN AC
 2657  2103            ISZ     T1
 2660  5254            JMP     .-4             /GET NEXT WORD
 2661  5204            JMP     AS+4            /GO BACK TO AS ROUTINE

 2662  0000   ASPNTR,  0

                        /SET SPECIAL SHIFT

 2663  4771'   SH,     JMS     WORD            /ALLOW INPUT FROM TTY
 2664  1770'           TAD     ASCWD           /GET ASCII WORD
 2665  7640            SZA CLA                 /WERE ANY CHARACTERS INPUT ?
 2666  7240            STA                     /YES - AC = 7777
 2667  3067            DCA     SHFLAG          /SET OR RESET SHIFT FLAG
 2670  1763'           TAD     OCTWD           /GET OCTAL INPUT
 2671  3070            DCA     SHFCNT          /SET SPECIAL SHIFT COUNT
 2672  5754'           JMP     DOTEX           /EXIT
```

```
                        /CHANGE ASSEMBLY LOCATOR

0673  1763'  ASLOC,  TAD      OCTWD           /GET FIELD DESIGNATOR
0674  0353          AND      (7              /EXTRACT LAST DIGIT
0675  7104          CLL RAL                  /MOVE TO BITS 6-8
0676  7006          RTL
0677  1357          TAD      (CDF            /CREATE CDF INSTRUCTION
0700  3233          DCA      ASFLD           /CHANGE AS ROUTINE
0701  4771'         JMS      WORD            /ALLOW TTY INPUT
0702  1763'         TAD      OCTWD           /GET NEW ADDRESS
0703  3262          DCA      ASPNTR          /SET POINTER
0704  5204          JMP      AS+4            /RETURN TO AS ROUTINE


0705  0000  MFLD,   0                        /THESE 2 CONSTANTS ARE USED BY
0706  0000  SAVMEM, 0                        /ANOTHER ROUTINE ( CMEM1 )



0753  0007
0754  0245
0755  7775
0756  0214
0757  6201
0760  0432
0761  0070
0762  0010
0763  0347
0764  5600
0765  0466
0766  5100
0767  7700
0770  0346
0771  0261
0772  7335
0773  7346
0774  7563
0775  0345
0776  6211
0777  4000
      1000          PAGE
```

```
1000  1377    RUBOUT, TAD     (RUTXT-1        /IF RUBOUT WAS INPUT
1001  3010            DCA     10              /ECHO /  AND DELETE
1002  1410            TAD I   10              /DATA WORDS IN WORD ROUTINE
1003  7450            SNA
1004  5776            JMP     WORD+1
1005  6046            TLS
1006  6041            TSF
1007  5206            JMP     .-1
1010  7200            CLA
1011  5202            JMP     .-7
1012  0334    RUTXT,  334
1013  0240            240
1014  0240            240
1015  0000            0
```

```
                        /SETUP TO START THE FPP

1016  0527    STFPP,  PT+1                    /ENTERED WITH A JMS
1017  4775'            JMS     WORD            /ALLOW TTY INPUT
1020  1374            TAD     (TEST           /GET ADDRESS OF FPP INSTRUCTIONS
1021  3773'           DCA     APT+1           /SET FPC IN PROGRAM APT
1022  1372            TAD     (PX0            /GET INDEX POINTER
1023  3771'           DCA     APT+2           /SET INDEX POINTER IN APT
1024  1370            TAD     (OPERND         /GET ADDRESS OF OPERAND TABLE
1025  3767'           DCA     APT+3           /SET BASE ADDRESS IN APT
1026  1366            TAD     (0101           /GET FIELD BITS
1027  3765'           DCA     APT             /SET APT FIELD BITS
1030  3764'           DCA     APT+4           /CLEAR OP ADDRESS IN APT
1031  1763'           TAD     OCTWD           /GET TTY INPUT
1032  7450            SNA                     /IS IT 0000 ?
1033  1365            TAD     (APT            /YES - GET ADDR OF PROGRAM APT
1034  3107            DCA     AAPT            /SET APT ADDRESS
1035  3106            DCA     STEPSW          /CLEAR SINGLE STATE SWITCH
1036  1107            TAD     AAPT
1037  3244            DCA     APTSAV          /SAVE APT ADDRESS
1040  7240            STA                     /SET PROGRAM RUN FF
1041  3245            DCA     FPPRUN
1042  3053            DCA     PSTAT           /CLEAR STATUS WORD
1043  5616            JMP I   STFPP           /EXIT

1044  0000    APTSAV, 0
1045  0000    FPPRUN, 0


1046  0000    BRANCH, 0
1047  2301            2301    /SA     LOAD START STEP ADDRESS
1050  2201            2201    /RA     LOAD RESET STEP ADDRESS
1051  0123            0123    /AS     ASSEMBLE
1052  2224            2224    /RT     RUN TRACE MODE
1053  2206            2206    /RF     RUN FAST MODE
1054  2431            2431    /TY     TYPE CONTENTS OF REGISTER
1055  0315            0315    /CM     LOAD COMMAND REGISTER
1056  0530            0530    /EX     FPP EXIT
1057  1720            1720    /OP     LOAD    OPERAND TABLE
1060  2310            2310    /SH     SET SHIFT COUNT AND FLAG
1061  0314            0314    /CL     CLEAR STEP SWITCH
1062  0501            0501    /EA     SET EXIT ADDRESS
1063  0000            0000
```

```
                        / FPP FLOWS

1064  4574  INIT,   CLEAR                  /CLEAR FPP AND PROGRAM REGISTERS
1065  4762'         JMS     MSTATE         /CHECK  STATES
1066  7610          SKP CLA                /TIME STATE ERROR
1067  5273          JMP     .+4            /TIME STATE GENERATER IS OK
1070  1361          TAD     (INIT+1        /GET PC FOR ERROR TYPEOUT
1071  4760'         JMS     ERR
1072  0000          ERROR   0000           / STATE ERROR CODE
1073  1107          TAD     AAPT           /GET ADDRESS OF APT
1074  3050          DCA     PAPT+1         /SET APT COMPARE ADDRESS
1075  4564          ENTER                  /TYPE ENTER INITIATE
1076  6561  ENINIT, FMAINT                 /SET FPP MAINT MODE
1077  7200          CLA
1100  1052          TAD     COMREG         /GET FPP COMMAND WORD
1101  6553          FPCOM                  /SET FPP COMMAND REG
1102  0357          AND     (7
1103  3047          DCA     PAPT
1104  1107          TAD     AAPT           /GET ADDRESS OF APT

1105  6555  INIT0,  FPST                   /START FPP
1106  7610          SKP CLA                /FPP DID NOT START
1107  5313          JMP     .+4
1110  4756'         JMS     TYP            /FPP START ERROR
1111  1644          STRTER                 /TYPE ERROR MESSAGE
1112  4755'         JMS     KEYCK          /WAIT FOR KEYBOARD INPUT
1113  4562          GETAPT                 /OUTBRK USING ADRS
1114  3024          DCA     AMSW           /MB TO AMSW
1115  1024          TAD     AMSW           /AND FIELD BITS
1116  0357          AND     (7
1117  3041          DCA     PFPC           /FIELD BITS OF FPC
1120  1024          TAD     AMSW           /YES-GET FIELD BITS
1121  7012          RTR
1122  7010          RAR
1123  0357          AND     (7
1124  3065          DCA     X0ADR          /FIELD BITS OF INDEX POINTER
1125  1024          TAD     AMSW           /GET FIELD BITS
1126  7012          RTR
1127  7012          RTR
1130  7012          RTR
1131  0357          AND     (7
1132  3054          DCA     PBASE          /FIELD BITS OF P0 ADDR
1133  4556          INCAPT                 /INCREMENT ADRS
1134  4575          CAPT                   /CHECK APT
1135  3117          DCA     CK0            /CLEAR CHECK 0 FLAG
1136  3120          DCA     CKOP           /CLEAR CHECK OP ADDR FLAG
```

```
1137  4563    INIT1,  FSTEP                   /STEP TO STATE 1
1140  4562            GETAPT                  /OUTBRK USING ADRS
1141  3025            DCA     ALSW            /MB TO ALSW
1142  4544            LOADO                   /A TO 0
1143  0024            AREG
1144  4556            INCAPT                  /INC ADRS
1145  4575            CAPT                    /CHECK APT ADDRESS
1146  4543            LOADOP                  /0 TO OP ADDR
1147  0021            OREG
1150  1045            TAD     OPADR+1         /OP ADDR TO FPC
1151  3042            DCA     PFPC+1
1152  5754/           JMP     INIT2           /GO TO STATE 2

1154  1200
1155  0250
1156  7313
1157  0007
1160  7317
1161  1065
1162  6000
1163  0347
1164  0214
1165  0210
1166  0101
1167  0213
1170  4100
1171  0212
1172  0220
1173  0211
1174  4000
1175  0261
1176  0262
1177  1011
      1200            PAGE
```

```
1200  4563    INIT2,  FSTEP                   /STEP TO  STATE 2
1201  4562            GETAPT                  /OUTBRK USING ADRS
1202  3066            DCA     X0ADR+1         /MB TO X0 ADDR
1203  4556            INCAPT                  /INC ADRS
1204  4575            CAPT                    /CHECK APT ADDRESS

1205  4563    INIT3,  FSTEP                   /STEP TO STATE 3
1206  4562            GETAPT                  /OUTBRK USING ADRS
1207  3055            DCA     PBASE+1         /MB TO P0 ADDR
1210  4556            INCAPT                  /INC ADRS
1211  4556            INCAPT                  /INC ADRS
1212  4575            CAPT                    /CHECK APT ADDRESS


1213  4563    INIT4,  FSTEP                   /STEP TO STATE 4
1214  4562            GETAPT                  /OUTBRK USING ADRS
1215  3035            DCA     ACEXP           /SAVE FAC EXPONENT
1216  4556            INCAPT                  /INC ADRS
1217  4575            CAPT                    /CHECK APT

1220  4563    INIT5,  FSTEP                   /STEP TO STATE 5
1221  4562            GETAPT                  /OUTBRK USING ADRS
1222  3024            DCA     AMSW            /MB TO AMSW
1223  4556            INCAPT                  /INC ADRS
1224  4575            CAPT                    /CHECK APT ADDRESS


1225  4563    INIT6,  FSTEP                   /STEP TO STATE 6
1226  4562            GETAPT                  /OUTBRK USING ADRS
1227  3025            DCA     ALSW            /MB TO ALSW
1230  4544            LOADO                   /A TO O
1231  0024            AREG
1232  4547            LOADAC                  /O TO FAC FRACTION
1233  0021            OREG

1234  4541    INEND,  MQVEX                   /SET PROGRAM INDEX REGISTERS
1235  5236            JMP     FETCH           /RETURN TO CONTROL PROGRAM
```

```
1236  4564    FETCH,  ENTER                   /TYPE ENTER FETCH
1237  4573            CLRA                    /CLEAR A REG
1240  4572            CLRB                    /CLEAR B REG
1241  1041            TAD     PFPC            /CLEAR BITS 0-8 OF FPC
1242  0377            AND     (7              /BECAUSE THE HARDWARE HAS
1243  3041            DCA     PFPC            /ONLY 15 BITS
1244  6567            LSHFT                   /ZERO FPP SHIFT REGISTER
1245  3071            DCA     SHREG           /ZERO PROGRAM SHIFT REGISTER
1246  3076            DCA     CARYIN          /CLEAR CARRY IN FF
1247  3004            DCA     MDFLAG          /CLEAR MULT OR DIV FLAG

1250  4563    FECH0,  FSTEP                   /STEP TO FETCH STATE 0
1251  4571            CLRO
1252  3032            DCA     MQMSW
1253  3033            DCA     MQLSW           /CLEAR MQ
1254  3034            DCA     MQEXT
1255  4560            GETPC                   /OUTBRK USING FPC
1256  3051            DCA     PIR             /STORE IN PROGRAM INST REG
1257  4553            INCPC                   /INCREMENT PROGRAM FPC
1260  4550            LOADA                   /SET A REG = 23
1261  0100            D27                     /DOUBLE WORD 27
1262  1035            TAD     ACEXP           /SET B REG = FAC EXPONENT
1263  4527            STORB                   /STORE LSW AND EXTEND SIGN
1264  4577            AMBO                    /A MINUS B TO O REG
1265  1051            TAD     PIR             /GET FPP INSTRUCTION
1266  0376            AND     (600            /EXTRACT BITS 3 AND 4
1267  7450            SNA                     /SPECIAL INSTRUCTION?
1270  5775'           JMP     FEND            /YES-END OF FETCH
1271  1374            TAD     (-400
1272  7640            SZA CLA                 /DOUBLE WORD INS?
1273  5300            JMP     FECH1           /NO-GO TO STATE 1
1274  4773'           JMS     SETST3          /SET STATE 3
1275  7240            STA
1276  3073            DCA     FLAG1           /SET FLAG1 FOR DOUBLE WORD INST.
1277  5772'           JMP     FECH34

1300  4563    FECH1,  FSTEP
1301  1051            TAD     PIR             /GET INSTRUCTION
1302  7106            CLL RTL                 /SAVE BIT FIR3 IN THE LINK
1303  7106            CLL RTL
1304  7200            CLA
1305  1051            TAD     PIR             /GET INSTRUCTION
1306  0371            AND     (177            /GET BITS 5-11
1307  7430            SZL                     /BIT 3 = 1?
1310  0377            AND     (7              /YES-EXTRACT BITS 9-11
1311  4530            STORA                   /FIR 5-11 OR 9-11 TO ALSW
1312  4546            LOADB                   /MOVE A TO B FOR ADD (A+A TO O)
1313  0024            AREG
1314  4576            APBO                    /A PLUS B TO O REG
1315  4546            LOADB                   /O TO B
1316  7021            OREG
1317  4576            APBO                    /A PLUS B TO OREG
```

```
1320  4563   FECH2,  FSTEP                  /STEP TO STATE 2
1321  4552           LOADA                  /0 TO A
1322  7021           OREG
1323  4546           LOADB                  /PC ADDR TO B
1324  0054           PBASE
1325  4576           APBO                   /A PLUS B TO 0
1326  4543           LOADOP                 /0 TO OP ADDR
1327  0021           OREG
1330  1051           TAD     PIR            /GET INSTRUCTION
1331  0370           AND     (400           /EXTRACT BIT 3
1332  7640           SZA CLA                /BIT 3 = 1?
1333  5350           JMP     FECH24         /YES-INC OP ADDR
1334  6556           FPRST                  /GET FPP STATUS
1335  7710           SPA CLA                /DOUBLE PRECISION?
1336  5350           JMP     FECH24         /YES-INC OP ADDR

1337  1051           TAD     PIR            /GET INSTRUCTION
1340  0367           AND     (7000          /EXTRACT OP CODE
1341  1366           TAD     (-3000
1342  7450           SNA                    /FDIV?
1343  5350           JMP     FECH24         /YES-INC OP ADDR
1344  1367           TAD     (-1000
1345  7450           SNA                    /FMUL?
1346  5350           JMP     FECH24         /YES-INC OP ADDR
1347  1366           TAD     (-3000
1350  7640   FECH24, SZA CLA                /FMULM?
1351  5353           JMP     .+2            /NO-DO NOT INC OP ADDR
1352  4555           INCOP                  /INC OP ADDR
1353  1051           TAD     PIR            /GET INSTRUCTION
1354  0370           AND     (400           /EXTRACT BIT 3
1355  7650           SNA CLA                /BIT 3 = 0?
1356  5775'          JMP     FEND           /END OF FETCH
1357  5765'          JMP     FECH3          /NO - GO TO STATE 3

1365  1400
1366  5000
1367  7000
1370  0400
1371  0177
1372  1406
1373  6344
1374  7400
1375  1533
1376  0600
1377  0007
      1400           PAGE
```

```
1400  4563  FECH3,  FSTEP                 /STEP TO STATE 3
1401  4561           GETOP                /OUTBRK USING OP ADDR
1402  0377           AND     (7           /EXTRACT BITS 9-11
1403  3032           DCA     MQMSW        /MB9-11 TO MQMSW
1404  4555           INCOP
1405  3073           DCA     FLAG1        /CLEAR FLAG 1
1406  1051  FECH34,  TAD     PIR          /GET INSTRUCTION
1407  0376           AND     (70          /EXTRACT BITS 6-8
1410  7112           CLL RTR
1411  7010           RAR
1412  4530           STORA                /FIR 6-8 TO ALSW = 0 TO AMSW
1413  4546           LOADB                /X0 ADDR TO B
1414  0065           X0ADR

1415  4563  FECH4,   FSTEP                 /STEP TO STATE 4
1416  1073           TAD     FLAG1        /GET DOUBLE WORD FLAG
1417  7640           SZA CLA              /DOUBLE WORD INSTRUCTION ?
1420  5223           JMP     .+3          /YES = NO OUTBREAK
1421  4561           GETOP                /OUTBRK USING OP ADDR
1422  3033           DCA     MQLSW        /MB TO MQLSW
1423  3073           DCA     FLAG1        /CLEAR DOUBLE WORD FLAG



1424  4576  FECH42,  APBO                 /A PLUS B TO O
1425  4543           LOADOP               /O TO OP ADDR
1426  0021           OREG
1427  4571           CLRO                 /0 TO O REG
1430  1051           TAD     PIR          /GET INSTRUCTION
1431  0375           AND     (170         /EXTRACT BITS 5-8
1432  7106           CLL RTL
1433  7006           RTL
1434  7006           RTL
1435  7470           SZL SNA              /FIR5 = 0 AND FIR 6-8 NE 0?
1436  5241           JMP     .+3
1437  4774'          JMS     SETST5       /YES-SETUP FOR STATE 6
1440  5254           JMP     FECH6        /GO TO MAJOR STATE 6
1441  7010           RAR
1442  7640           SZA CLA              /FIR 5-8 = 0?
1443  5246           JMP     .+3
1444  4773'          JMS     SETST6       /YES-SETUP FOR MAJOR STATE 7
1445  5271           JMP     FECH7        /GO TO STATE 7
```

```
1446  4563  FECH5,  FSTEP                   /STEP TO STATE 5
1447  4551           INCX
1450  1051           TAD     PIR            /GET INSTRUCTION
1451  0376           AND     (70            /EXTRACT BITS 6-8
1452  7650           SNA CLA                /FIR6-8 = 0?
1453  5244           JMP     FECH5-2        /YES-SETUP FOR STATE 7

1454  4563  FECH6,  FSTEP                   /STEP TO STATE 6
1455  4557           GETX                   /OUTBRK USING OP ADDR
1456  3025           DCA     ALSW           /MB TO ALSW
1457  3024           DCA     AMSW           /0 TO AMSW
1460  4546           LOADB                  /A TO B FOR ADD (A+A TO 0)
1461  0024           AREG
1462  4576           APBO                   /A PLUS B TO 0
1463  1052           TAD     COMREG         /GET COMMAND REGISTER
1464  7710           SPA CLA                /FLOATING POINT MODE?
1465  5271           JMP     .+4            /NO
1466  4546           LOADB                  /0 TO B
1467  0021           OREG
1470  4576           APBO                   /A PLUS TO 0
```

```
1471  4563  FECH7,  FSTEP                  /STEP TO STATE 7
1472  4550           LOADA                 /MQ TO A
1473  7032           MQREG
1474  4546           LOADB                 /0 TO B
1475  0021           OREG
1476  1051           TAD     PIR           /GET INSTRUCTION
1477  0372           AND     (200          /EXTRACT BIT 4
1500  7640           SZA CLA               /FIR4=1
1501  5312           JMP     FECH72        /YES-GO TO STATE 7-2
1502  4560           GETPC                 /OUTBRK USING FPC
1503  3025           DCA     ALSW          /MB TO ALSW
1504  1051           TAD     PIR           /GET INSTRUCTION
1505  0377           AND     (7            /EXTRACT BITS 9-11
1506  3024           DCA     AMSW          /FIR 9-11 TO AMSW
1507  4546           LOADB                 /0 TO B
1510  0021           OREG
1511  4553           INCPC                 /INC FPC
1512  4576  FECH72,  APB0                  /A PLUS B TO 0
1513  4543           LOADOP                /0 TO OP ADDR
1514  0021           OREG
1515  1052           TAD     COMREG        /GET COMMAND REG
1516  7710           SPA CLA               /D, P, MODE ?
1517  5333           JMP     FEND          /YES = END OF FETCH
1520  1051           TAD     PIR           /GET INSTRUCTION
1521  0371           AND     (7000         /EXTRACT OP CODE
1522  1370           TAD     (-3000
1523  7450           SNA                   /FDIV?
1524  5332           JMP     FECH73        /YES-INC OP ADDR
1525  1371           TAD     (-1000
1526  7450           SNA                   /FMUL?
1527  5332           JMP     FECH73        /YES-INC OP ADDR
1530  1370           TAD     (-3000
1531  7650           SNA CLA               /FMULM?
1532  4555  FECH73,  INCOP                 /YES-INC OP ADDR
1533  5767  FEND,    JMP     EXEC          /GO TO EXECUTE

1567  1600
1570  5000
1571  7000
1572  0200
1573  6353
1574  6335
1575  0170
1576  0070
1577  0007
      1600           PAGE
```

```
1600  7300  EXEC,   CLA CLL                 /EXECUTE SKIP CHAIN
1601  3073         DCA     FLAG1
1602  3074         DCA     FLAG2           /CLEAR FLAGS AND TEMP LOCATIONS
1603  3075         DCA     FLAG3
1604  3103         DCA     T1
1605  3104         DCA     T2
1606  3777'        DCA     SUBSW
1607  1051         TAD     PIR             /GET INSTRUCTION REG
1610  0376         AND     (600            /EXTRACT BITS 3-4
1611  7640         SZA CLA
1612  5775'        JMP     ARITH           /ARITHMETIC INSTRUCTIONS
1613  1051  PROCES, TAD     PIR             /PROCESS SPECIAL INSTRUCTIONS
1614  0374         AND     (7000           /GET OP CODE
1615  1373         TAD     (-2000
1616  7420         SNL                     /WHICH SPECIAL FORMAT?
1617  5237         JMP     SPEC2           /SPECIAL FORMAT 2
1620  7650  SPECI,  SNA CLA                 /JXN?
1621  5772'        JMP     JXN             /YES
1622  4564  TRAPED, ENTER                   /ENTER TRAPPED INSTRUCTIONS
1623  4563  TRAPI,  FSTEP                   /STEP TO STATE 1
1624  4560         GETPC                   /OUTBRK USING FPC
1625  3025         DCA     ALSW            /MB TO ALSW
1626  1051         TAD     PIR             /GET INSTRUCTION REG
1627  0371         AND     (7              /EXTRACT BITS 9-11
1630  3024         DCA     AMSW            /FIR9-11 TO AMSW
1631  4544         LOADD                   /A TO D
1632  0024         AREG
1633  4543         LOADDP                  /D TO DP ADDR
1634  0021         DREG
1635  4553         INCPC                   /INC FPC
1636  5770'        JMP     EXIT            /GO TO EXIT
1637  7300  SPEC2,  CLA CLL                 /SPECIAL FORMAT 2 INSTRUCTIONS
1640  1051         TAD     PIR             /GET INSTRUCTIONS
1641  0367         AND     (7770
1642  7450         SNA                     /SPEC FMAT 2 OR 3
1643  5766'        JMP     SPEC3           /SPEC FMAT 3
1644  7104         CLL RAL
1645  7006         RTL
1646  7420         SNL                     /OP CODE 0 OR 1
1647  5274         JMP     SPEC20          /OP CODE 0
```

```
1650  1365    SPEC21, TAD     (-1400
1651  7710            SPA CLA                 /NOP?
1652  5256            JMP     .+4             /NO
1653  4525            TRACE                   /YES-TRACING PROGRAM?
1654  7000            NOP                     /YES-TYPE NOP
1655  5764            JMP     FETCH           /GO TO FETCH
1656  1051            TAD     PIR             /GET INSTRUCTION
1657  7363            AND     (170            /EXTRACT EXTENSION
1660  7110            CLL RAR
1661  7012            RTR                     /RIGHT JUSTIFY
1662  7040            CMA                     /NEGATE
1663  3103            DCA     T1
1664  1362            TAD     (INS21-1        /GET ADDRESS OF INSTRUCTION TABLE
1665  3104            DCA     T2
1666  2104            ISZ     T2              /FIND INSTRUCTION
1667  2103            ISZ     T1
1670  5266            JMP     .-2
1671  1504            TAD I   T2              /GET INSTRUCTION ADDRESS
1672  3104            DCA     T2
1673  5504            JMP I   T2
```

```
1674  7112    SPEC20, CLL RTR
1675  7012            RTR
1676  7012            RTR
1677  7041            CIA
1700  7001            IAC
1701  7450            SNA
1702  5761/           JMP     FALN            /ALIGN INSTRUCTION
1703  7001            IAC
1704  7450            SNA
1705  5760/           JMP     FATX            /ATX INSTRUCTION
1706  7001            IAC
1707  7450            SNA
1710  5757/           JMP     FXTA            /XTA INSTRUCTION
1711  1356            TAD     (5
1712  7450            SNA
1713  5755/           JMP     FLDX            /LDX INSTRUCTION
1714  7001            IAC
1715  7650            SNA CLA
1716  5754/           JMP     ADDX            /ADDX INSTRUCTION
1717  5253            JMP     SPEC21+3        /NOP

1754  5212
1755  4010
1756  0005
1757  4434
1760  4265
1761  4036
1762  7217
1763  0170
1764  1236
1765  6400
1766  2000
1767  7770
1770  2427
1771  0007
1772  4473
1773  6000
1774  7000
1775  2030
1776  0600
1777  7112
      2000            PAGE
```

```
2000  1051    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  1051    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  3103            DCA     T1              /SAVE MINUS OP CODE
2036  1052            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  3104            DCA     T2
2044  2104            ISZ     T2              /INC ADDR
2045  2103            ISZ     T1              /INC INSTRUCTION
2046  5244            JMP     .-2             /NOT THIS INS--TRY AGAIN
2047  1504            TAD I   T2              /GET ADDR OF INST
2050  3104            DCA     T2
2051  5504            JMP I   T2              /GO TO INSTRUCTION
```

```
2052  4564  DEP,    ENTER                 /TYPE ENTER DEPOSIT
2053  4563  DEP11,  FSTEP                 /STEP TO STATE11
2054  1052          TAD     COMREG        /GET COMMAND REG
2055  7700          SMA CLA               /FIXED PNT NOS,?
2056  4540          NORM                  /CHECK AND NORMALIZE
2057  4550          LOADA                 /0 TO A
2060  0021          OREG
2061  4572          CLRB                  /ZERO TO B
2062  1023          TAD     OEXT
2063  3103          DCA     T1            /SAVE O EXT FOR ROUNDING
2064  1021          TAD     OMSW          /CHECK FOR OVERFLOW?
2065  7700          SMA CLA               /NO - CLEAR FLAG
2066  7040          CMA                   /YES - SET FLAG
2067  3073          DCA     FLAG1         /STORE FLAG
2070  4576          APBO
2071  1103          TAD     T1            /GET EXT REG FOR ROUNDING
2072  7710          SPA CLA               /CHECK CARRY IN
2073  4554          INCOR                 /INCREMENT O REG
2074  1073          TAD     FLAG1         /CHECK FOR OVERFLOW
2075  7650          SNA CLA               /OVERFLOW?
2076  5302          JMP     .+4           /NO - BYPASS O CHECK
2077  1021          TAD     OMSW          /YES - CHECK OVERFLOW
2100  7710          SPA CLA
2101  5304          JMP     .+3           /OVERFLOW
2102  4552          INCST                 /NO OVERFLOW
2103  5331          JMP     CKMEM         /BYPASS MAJOR STATE 12
2104  1052          TAD     COMREG        /GET COMMAND REG
2105  7700          SMA CLA               /FIXED PT, NOS,?
2106  5316          JMP     .+10          /NO
2107  1053          TAD     PSTAT         /GET STATUS WORD
2110  0364          AND     (7577         /SAVE OTHER BITS
2111  1363          TAD     (200          /SET FRAC OVERFLOW
2112  3053          DCA     PSTAT
2113  7040          CMA
2114  3072          DCA     EXITSW        /SET EXIT SWITCH
2115  5762          JMP     DEPEND        /END OF DEPOSIT
```

```
2116  4563    DEP12,  FSTEP                   /STEP TO STATE 12
2117  1021            TAD     OMSW            /12-1 AND 12-2=SHIFT
2120  7110            CLL RAR                 /0 TO B0
2121  3027            DCA     BMSW
2122  1022            TAD     OLSW
2123  7010            RAR
2124  3030            DCA     BLSW
2125  4544            LOADO                   /B TO O
2126  0027            BREG
2127  2071            ISZ     SHREG           /INC SHFT CNTR
2130  7000            NOP
2131  4526    CKMEM,  TOMEM                   /RESULTS TO MEMORY?
2132  5335            JMP     .+3
2133  4552            INCST                   /NO - INCREMENT MAJOR STATE
2134  5761'           JMP     DEP14           /BYPASS STATE 13

2135  4563    DEP13,  FSTEP                   /STEP TO STATE 13
2136  1022            TAD     OLSW            /OLSW TO MEMORY
2137  3007            DCA     LMEM
2140  4565            DECOP
2141  5761'           JMP     DEP14           /GO TO STATE 14

2161  2200
2162  2415
2163  0200
2164  7577
2165  7177
2166  7000
2167  5050
2170  5070
2171  5063
2172  5114
2173  5076
2174  5150
2175  5126
2176  2427
2177  0007
      2200            PAGE
```

```
2270  4563  DEP14,  FSTEP            /STEP TO STATE 14
2271  4526          TOMEM            /RESULTS TO MEMORY?
2272  7410          SKP              /YES
2273  5211          JMP     .+6      /NO
2274  1021          TAD     OMSW     /OMSW TO MEMORY
2275  3006          DCA     MMEM
2276  4567          CMEMF            /CHECK MEMORY FRACTION
2277  4565          DECOP            /DEC OP ADDR
2210  5213          JMP     .+3      /BYPASS 0 TO FAC
2211  4547          LOADAC           /0 TO FAC FRACTION
2212  0021          OREG
2213  1033          TAD     MQLSW    /MQLSW TO BLSW
2214  4527          STORB            /SIGN EXTEND TO BMSW
2215  4531          SQEZ             /Q = 0 ?
2216  7410          SKP              /NO
2217  7240          STA              /YES = SET ZERO TO
2220  3075          DCA     FLAG3    /FAC FF
2221  1004          TAD     MDFLAG   /GET MULT OR DIV FLAG
2222  7650          SNA CLA          /MULT OR DIV?
2223  5240          JMP     DEP144   /NO = GO TO STATE 14-4
2224  1035          TAD     ACEXP    /FAC EXP TO ALSW
2225  4530          STORA            /SIGN EXTEND TO AMSW
2226  1051          TAD     PIR      /GET INSTRUCTION REG
2227  0377          AND     (7000    /EXTRACT OP CODE
2230  1376          TAD     (-3000
2231  7640          SZA CLA          /DIVIDE?
2232  5235          JMP     .+3      /NO
2233  4577          AMBO             /YES = A MINUS B TO 0
2234  7410          SKP
2235  4576          APBO             /A PLUS B TO 0
2236  4546          LOADB            /0 TO B
2237  0021          OREG
2240  1071  DEP144, TAD     SHREG    /SHFT CNTR TO ALSW
2241  4530          STORA            /SIGN EXTEND TO AMSW
2242  1075          TAD     FLAG3    /ZERO TO FAC FF SET ?
2243  7650          SNA CLA
2244  5247          JMP     ACNEZ    /NO ADD A AND B
2245  4571          CLRO             /YES = 0 TO 0
2246  5250          JMP     ACNEZ+1  /DO NOT ADD A PLUS B
```

```
2247  4576    ACNE2,  APBO                    /A PLUS B TO 0
2250  1052            TAD     COMREG          /GET COMMAND REG
2251  7700            SMA CLA                 /FIXED PNT, NOS.?
2252  5255            JMP     DEP15           /NO
2253  3072            DCA     EXITSW          /CLEAR EXIT SWITCH
2254  5775i           JMP     DEPEND          /END OF DEPOSIT

2255  4563    DEP15,  FSTEP                   /STEP TO STATE 15
2256  4526            TOMEM                   /RESULTS TO MEMORY?
2257  7410            SKP                     /YES
2260  5265            JMP     .+5             /NO - BYPASS INBRK
2261  1022            TAD     OLSW            /OLSW TO MEM
2262  3005            DCA     EMEM
2263  4570            CMEME                   /CHECK MEMORY EXPONENT
2264  5267            JMP     .+3
2265  1022            TAD     OLSW            /OLSW TO FAC EXP
2266  3035            DCA     ACEXP
2267  4546            LOADB                   /0 TO B
2270  0021            OREG
2271  7330            CLA STL RAR
2272  3025            DCA     ALSW            /4000 TO A
2273  3024            DCA     AMSW
2274  1021            TAD     OMSW
2275  7700            SMA CLA                 /IS OLT 0
2276  5774i           JMP     OPOS            /NO
2277  4576            APBO                    /YES - A PLUS B TO 0
2300  1021            TAD     OMSW
2301  7700            SMA CLA                 /IS 0 LT 0
2302  5315            JMP     END15           /NO - OK
2303  1053            TAD     PSTAT           /GET STATUS WORD
2304  0373            AND     (7737           /SAVE OTHER BITS
2305  1372            TAD     (40             /SET UNDERFLOW
2306  3053            DCA     PSTAT
2307  1052            TAD     COMREG          /GET COMMAND REGISTER
2310  7004            RAL
2311  7700            SMA CLA                 /TRAP ON UNDERFLOW ?
2312  7040            CMA                     /NO - SET IGNORE UNDERFLOW FF
2313  3112            DCA     IGNFL
2314  7040            CMA
2315  3072    END15,  DCA     EXITSW          /SET OR CLEAR EXIT SWITCH
2316  5775i           JMP     DEPEND          /END OF DEPOSIT

2372  0040
2373  7737
2374  2400
2375  2415
2376  5000
2377  7000
      2400            PAGE
```

```
2400  4577  OPOS,   AMBO                 /A MINUS B TO 0
2401  4531           SOEZ
2402  7610           SKP CLA
2403  5207           JMP     .+4
2404  1021           TAD     OMSW
2405  7700           SMA  CLA             /IS 0 LEQ 0
2406  5214           JMP     .+6          /NO
2407  1053           TAD     PSTAT        /YES - SET OVERFLOW
2410  0377           AND     (7637        /SAVE ALL BITS EXCEPT UNDERFLOW
2411  1376           TAD     (100
2412  3053           DCA     PSTAT
2413  7040           CMA
2414  3072           DCA     EXITSW       /SET OR CLEAR EXIT SWITCH
2415  6556  DEPEND,  FPRST                /READ FPP STATUS WORD
2416  0375           AND     (740         /EXTRACT OVERFLOW BITS
2417  7041           CIA
2420  1053           TAD     PSTAT        /COMPARE WITH PROGRAM STATUS
2421  7640           SZA CLA
2422  4774'          JMS     STERR        /STATUS ERROR
2423  1072           TAD     EXITSW
2424  7640           SZA CLA
2425  5227           JMP     EXIT
2426  5773'          JMP     FETCH
```

```
2427  4564  EXIT,  ENTER
2430  4563  EXIT2,  FSTEP                    /STEP TO STATE 2
2431  1112         TAD     IGNFL            /GET IGNORE TRAP FF
2432  7650         SNA CLA                  /UNDERFLOW IGNORED ?
2433  5257         JMP     EXSAV            /NO = SAVE APT
2434  3112         DCA     IGNFL            /RESET UNDERFLOW FLAG
2435  4526         TOMEM                    /ANSWER STORED IN FAC?
2436  5243         JMP     .+5              /NO = 0 TO MEM
2437  3035         DCA     ACEXP            /YES = 0 TO FAC
2440  3036         DCA     ACMSW
2441  3037         DCA     ACLSW
2442  5254         JMP     TOFECH
2443  3005         DCA     EMEM
2444  4555         INCOP                    /INCREMENT OP ADDRESS
2445  4563         FSTEP                    /STEP TO STATE 1
2446  3006         DCA     MMEM
2447  3007         DCA     LMEM
2450  4555         INCOP                    /INCREMENT OP ADDRESS

2451  4563         FSTEP                    /STEP TO STATE 2
2452  4565         DECOP                    /DECREMENT THE OP ADDRESS
2453  4565         DECOP                    /DECREMENT THE OP ADDRESS
2454  4525  TOFECH, TRACE                   /TRACING PROGRAM?
2455  2347         VFER                     /UNDERFLOW ERROR = GO TO FETCH
2456  5773'        JMP     FETCH

2457  3073  EXSAV,  DCA     FLAG1            /CLEAR ERROR FLAG
2460  1052         TAD     COMREG
2461  0372         AND     (20
2462  7640         SZA CLA                  /SAVE FAC?
2463  5266         JMP     .+3              /NO
2464  1037         TAD     ACLSW
2465  4771'        JMS     EXCOM            /YES = COMPARE LSW
2466  4566         DECAPT                   /DEC, ADRS
2467  4575         CAPT                     /COMPARE APT ADDRESS


2470  4563  EXIT1,  FSTEP                   /STEP TO STATE 1
2471  1052         TAD     COMREG
2472  0372         AND     (20
2473  7640         SZA CLA                  /SAVE FAC?
2474  5277         JMP     .+3              /NO
2475  1036         TAD     ACMSW            /YES
2476  4771'        JMS     EXCOM            /COMPARE MSW
2477  4566         DECAPT                   /DEC, ADRS
2500  4575         CAPT                     /CHECK APT ADDRESS
```

```
2501  4563    EXIT2,  FSTEP                   /STEP TO MAJOR STATE 2
2502  1052            TAD     COMREG
2503  0372            AND     (20
2504  7640            SZA CLA                 /SAVE FAC?
2505  5310            JMP     .+3             /NO
2506  1035            TAD     ACEXP
2507  4771'           JMS     EXCOM           /YES - COMPARE EXPONENT
2510  4566            DECAPT                  /DEC ADRS
2511  4575            CAPT                    /COMPARE APT ADDRESS

2512  4563    EXIT3,  FSTEP                   /STEP TO STATE 3
2513  1052            TAD     COMREG
2514  0370            AND     (200
2515  7640            SZA CLA                 /SAVE OP ADDR?
2516  5321            JMP     .+3             /NO
2517  1045            TAD     OPADR+1
2520  4771'           JMS     EXCOM           /YES - COMPARE OP ADDR
2521  4566            DECAPT                  /DEC, ADRS
2522  4575            CAPT                    /CHECK APT ADDRESS

2523  4563    EXIT4,  FSTEP                   /STEP TO STATE 4
2524  1052            TAD     COMREG
2525  0367            AND     (40
2526  7640            SZA CLA                 /SAVE P0 ADDR?
2527  5332            JMP     .+3             /NO
2530  1055            TAD     PBASE+1
2531  4771'           JMS     EXCOM           /YES - COMPARE P0 ADDR
2532  4566            DECAPT                  /DEC, ADRS
2533  4575            CAPT                    /CHECK APT ADDRESS

2534  4563    EXIT5,  FSTEP                   /STEP TO STATE 5
2535  1052            TAD     COMREG
2536  0376            AND     (100
2537  7640            SZA CLA                 /SAVE X0 ADDRESS?
2540  5343            JMP     .+3             /NO
2541  1066            TAD     X0ADR+1         /YES
2542  4771'           JMS     EXCOM           /COMPARE X0 ADDRESS
2543  4566            DECAPT                  /DEC, ADRS
2544  4575            CAPT                    /CHECK APT ADDRESS
2545  5766'           JMP     EXIT6           /GO TO STATE 6

2566  2600
2567  0040
2570  0200
2571  2627
2572  0020
2573  1236
2574  7325
2575  0740
2576  0100
2577  7637
      2600            PAGE
```

```
2620  4563    EXIT6,  FSTEP                   /STEP TO STATE 6
2621  1042            TAD     PFPC+1
2622  4227            JMS     EXCOM           /COMPARE FPC
2623  4566            DECAPT                  /DEC, ADRS
2624  4575            CAPT                    /CHECK APT ADDRESS

2625  4563    EXIT7,  FSTEP                   /STEP TO STATE 7
2626  4777'           JMS     APTPAC          /PACK APT FIELD BITS
2627  4227            JMS     EXCOM           /COMPARE FIELD BITS
2630  6557            FPIST                   /CHECK FLAG
2631  7402            ERROR   HALT            /FLAG IS NOT SET
2632  0376            AND     (740            /GET OVERFLOW STATUS
2633  7041            CIA
2634  1053            TAD     PSTAT
2635  7640            SZA CLA
2636  4775'           JMS     STERR           /STATUS ERROR
2637  1073            TAD     FLAG1           /CHECK ERROR FLAG
2640  7650            SNA CLA                 /WAS APT DATA STORED CORRECTLY ?
2641  5225            JMP     .+4             /YES
2642  4774'           JMS     TYP             /NO - TYPE ERROR MESSAGE
2643  2655            APTERR
2644  0000            0
2645  4773'           JMS     SETRET          /SET REENTER ADDRESS
2646  5772'           JMP     START           /GO TO CONTROL PROGRAM

2647  0000    EXCOM,  0
2650  3110            DCA     EXWD            /SAVE COMPARE WORD
2651  1047            TAD     PAPT            /GET APT FIELD BITS
2652  7104            CLL RAL
2653  7006            RTL
2654  1371            TAD     (CDF            /CREATE FPP CDF
2655  3236            DCA     .+1
2656  6201            CDF                     /CHANGE TO FPP FIELD
2657  1450            TAD I   PAPT+1          /GET APT INFO
2660  6201            CDF     00              /PROGRAM FIELD
2661  7041            CIA
2662  1110            TAD     EXWD
2663  7650            SNA CLA
2664  5247            JMP     .+3
2665  7040            CMA
2666  3073            DCA     FLAG1           /SET ERROR FLAG
2667  5627            JMP I   EXCOM           /RETURN
```

```
2650  4564   DPADD,  ENTER                  /D,P, ADD AND SUB
2651  4563   DPADD0, FSTEP                  /STEP TO STATE 0
2652  4561           GETOP                  /OUTBRK USING OP ADDR
2653  3027           DCA     BMSW           /MB TO BMSW
2654  4555           INCOP                  /INC OF ADDR
2655  4550           LOADA                  /FAC FRAC TO A
2656  2036           ACMSW

2657  4563   DPADD1, FSTEP                  /STEP TO STATE 1
2660  4561           GETOP                  /OUTBRK USING OP ADDR
2661  3030           DCA     BLSW           /MB TO BLSW
2662  1051           TAD     PIR
2663  0370           AND     (2000
2664  7650           SNA CLA                /SUBTRACTION?
2665  5270           JMP     .+3            /NO - GO TO ADD
2666  4577           AMBO                   /A MINUS B TO 0
2667  7410           SKP
2670  4576           APBO                   /A PLUS B TO 0
2671  1111           TAD     OVFL           /OVERFLOW?
2672  7640           SZA CLA
2673  5275           JMP     DPOVFL         /YES
2674  5767/          JMP     DEP            /TO DEPOSIT



2675  1053   DPOVFL, TAD     PSTAT
2676  0366           AND     (7577
2677  1365           TAD     (200           /SET ARITH FLOW
2700  3053           DCA     PSTAT
2701  5764/          JMP     EXIT           /TO EXIT

2764  2427
2765  0200
2766  7577
2767  2052
2770  2000
2771  6201
2772  0230
2773  7254
2774  7313
2775  7325
2776  0740
2777  6104
      3000           PAGE
```

```
3000  4564  FPADD,  ENTER              /ADD = SUB OF F.P. NOS.
3001  4563  FADD0,  FSTEP              /STEP TO STATE 0
3002  4561          GETOP              /OUTBRK USING OP ADDR
3003  4527          STORB              /MB TO BLSW SIGN TO BMSW
3004  1030          TAD     BLSW
3005  3033          DCA     MQLSW      /MB TO MQ LSW
3006  4555          INCOP              /INC OP ADDR
3007  1035          TAD     ACEXP      /GET FAC EXPONENT
3010  4530          STORA              /FAC EXP TO ALSW SIGN TO AMSW
3011  4577          AMBO               /A MINUS B TO 0
3012  1021          TAD     OMSW
3013  7700          SMA CLA            /IS 0 LESS THAN 0
3014  5273          JMP     SHFOP      /NO = SHIFT OPERAND PATH
3015  7240          STA                /SHIFT FAC FRACTION PATH
3016  3073          DCA     FLAG1      /SET SHFT FAC FF
3017  1022          TAD     OLSW
3020  3071          DCA     SHREG      /0 TO SHFT CNTR
3021  4546          LOADB              /0 TO B
3022  0021          OREG
3023  1101          TAD     D27+1
3024  4530          STORA              /+27 TO ALSW
3025  4576          APBO               /A PLUS B TO 0
```

```
3026  4563    FADD1,  FSTEP                   /STEP TO STATE 1
3027  4561            GETOP                   /OUTBRK USING OP ADDR
3030  4777'           JMS     CKSUB           /SUBTRACTION?
3031  7040            CMA                     /YES - COMPLIMENT MB
3032  3024            DCA     AMSW            /MB TO AMSW
3033  4555            INCOP                   /INC OP ADDR
3034  1021            TAD     OMSW
3035  7710            SPA CLA
3036  7040            CMA
3037  3074            DCA     FLAG2           /0 0 TO OVERSHFT FF
3040  4572            CLRB                    /0 TO B

3041  4563    FADD2,  FSTEP                   /STEP TO STATE 2
3042  4561            GETOP                   /OUTBRK USING OP ADDR
3043  4777'           JMS     CKSUB           /SUBTRACTION?
3044  5247            JMP     .+3             /YES - NOT MB TO ALSW
3045  3025            DCA     ALSW            /NO - MB TO ALSW
3046  5255            JMP     .+7
3047  7040            CMA                     /COMPLIMENT MB
3050  3025            DCA     ALSW            /NOT MB TO ALSW
3051  1376            TAD     [7400           /COMPLIMENT 0 TO AEXT
3052  3026            DCA     AEXT
3053  1123            TAD     [400            /1 TO ACRY IN
3054  3076            DCA     CARYIN
3055  4576            APBO                    /A PLUS B TO 0 B=0 IF ADDITION
3056  4546            LOADB                   /FAC FRAC TO B
3057  0036            FACFR
3060  4531            SQEZ                    /0 = 0 ?
3061  5267            JMP     .+6             /NO
3062  1035            TAD     ACEXP           /YES =0=0
3063  3033            DCA     MQLSW           /FAC EXP TO MQLSW
3064  4544            LOADO                   /B TO 0
3065  0027            BREG
3066  5775'           JMP     FADEND          /GO TO DEPOSIT
3067  1074            TAD     FLAG2           /OVERSHIFT FF = 1 ?
3070  7640            SZA CLA
3071  5775'           JMP     FADEND          /YES - GO TO DEPOSIT
3072  5774'           JMP     FADD3           /NO - GO TO STATE 3
```

```
3073  1022  SHFOP,  TAD     OLSW          /STATE 0 SHIFT OPERAND
3074  7040          CMA
3075  3071          DCA     SHREG         /0 COMPLIMENT TO SHFT CNTR
3076  4546          LOADB                 /0 TO B
3077  2021          OREG
3100  1101          TAD     D27+1
3101  4530          STORA                 /+27 TO ALSW
3102  4577          AMBO                  /A MINUS B TO 0
3103  2071          ISZ     SHREG         /INC SHFT COUNTER
3104  7000          NOP

3105  4563  FADD01, FSTEP                 /STEP TO STATE 1 (SHFT OP PATH)
3106  4561          GETOP                 /OUTBRK USING OP ADDRESS
3107  3027          DCA     BMSW          /MB TO BMSW
3110  4555          INCOP                 /INC OP ADDR
3111  1021          TAD     OMSW
3112  7710          SPA CLA
3113  7040          CMA
3114  3074          DCA     FLAG2         /0 0 TO OVERSHFT FF
3115  4550          LOADA                 /FAC FRAC TO A
3116  0036          FACFR
3117  4544          LOADO                 /A TO O
3120  0024          AREG
3121  4531          SOEZ                  /O = 0 ?
3122  5327          JMP     FADD01A       /NO



3123  7040          CMA                   /YES
3124  3075          DCA     FLAG3         /1 TO ADD ZERO FF
3125  4573          CLRA                  /0 TO A
3126  5334          JMP     .+6
3127  1035  FADD1A, TAD     ACEXP
3130  3033          DCA     MQLSW         /FAC EXP TO MQLSW
3131  1074          TAD     FLAG2
3132  7640          SZA CLA               /OVERSHFT FF=1?
3133  5775          JMP     FADEND        /YES - GO TO DEPOSIT
3134  5773          JMP     FADD02        /NO - GO TO STATE 2

3173  3200
3174  3213
3175  3252
3176  7400
3177  4541
      3200          PAGE
```

```
3200  4563  FADD02, FSTEP                 /STEP TO STATE 2
3201  4561         GETOP                  /OUTBRK USING OP ADDR
3202  3030         DCA      BLSW          /MB TO BLSW
3203  1075         TAD      FLAG3
3204  7650         SNA CLA                /ADD ZERO FF=1
3205  5213         JMP      FADD3         /NO = GO TO STATE 3
3206  4544         LOADO
3207  0027         BREG                   /B TO 0
3210  4777'        JMS      CKSUB         /SUBTRACTION?
3211  4577         AMBO                   /YES = A MINUS B TO 0
3212  5252         JMP      FADEND        /GO TO DEPOSIT

3213  4563  FADD3,  FSTEP                 /STEP TO STATE 3
3214  4533         SHFTB                  /IF SHFT CNTR NE 0=SHIFT B
3215  4777'        JMS      CKSUB         /SUBTRACTION?
3216  7610         SKP CLA                /YES = CHECK SHFT FAC FF
3217  5227         JMP      .+10          /NO = ADD A AND B
3220  1073         TAD      FLAG1
3221  7640         SZA CLA                /SHFT FF SET?
3222  5225         JMP      .+3           /NO = +1 TO CARRY IN
3223  4577         AMBO                   /YES = A MINUS B TO 0
3224  5230         JMP      .+4
3225  1123         TAD      C400
3226  3076         DCA      CARYIN        /1 TO CARRYING
3227  4576         APBO                   /A PLUS B TO 0
3230  1111         TAD      OVFL
3231  7650         SNA CLA                /FRAC OVERFLOW?
3232  5252         JMP      FADEND        /NO = GO TO DEPOSIT
3233  1021         TAD      OMSW          /
3234  7110         CLL RAR
3235  1077         TAD      CAROUT        /SET SIGN BIT
3236  3027         DCA      BMSW          /0 TO B SHIFTED RIGHT
3237  1022         TAD      OLSW
3240  7010         RAR
3241  3030         DCA      BLSW
3242  1023         TAD      OEXT          /GET EXTENSION
3243  7010         RAR
3244  0376         AND      C7400         /KEEP IT 4 BITS LONG
3245  3031         DCA      BEXT
3246  4544         LOADO                  /B TO 0
3247  0027         BREG
3250  2071         ISZ      SHREG         /INC SHFT CNTR
3251  7410         SKP
3252  3071  FADEND, DCA      SHREG
3253  3073         DCA      FLAG1
3254  3074         DCA      FLAG2         /CLEAR FLAGS
3255  3075         DCA      FLAG3
3256  5775'        JMP      DEP           /GO TO DEPOSIT
```

```
3257  4564  FMULT,  ENTER                 /ENTER MULTIPLY
3260  7240           STA
3261  3004           DCA     MDFLAG       /SET MULT, DIV, FLAG FOR MULTIPLY

3262  4563  MULT0,  FSTEP                 /ENTER STATE 0
3263  4561           GETOP                /OUTBRK USING OP ADDR
3264  3027           DCA     BMSW
3265  1027           TAD     BMSW
3266  7710           SPA CLA
3267  7040           CMA
3270  3073           DCA     FLAG1        /MB0 TO OP SIGN
3271  4555           INCOP                /INCREMENT OP ADDR
3272  4573           CLRA                 /0 TO A
3273  1101           TAD     D23+1
3274  3071           DCA     SHREG        /23 TO SHIFT CNTR
3275  1067           TAD     SHFLAG
3276  7700           SMA CLA              /OPERATOR SELECTED SHIFT CNT?
3277  5303           JMP     .+4          /NO-USE NORMAL SHIFT CNT
3300  1070           TAD     SHFCNT       /YES-GET SPECIAL COUNT
3301  6567           LSHFT                /LOAD FPP SHIFT REG
3302  3071           DCA     SHREG
3303  1071           TAD     SHREG        /NEGATE SHIFT REG
3304  7040           CMA                  /FOR DECREMENT WITH
3305  3071           DCA     SHREG        /ISZ LOOP
3306  4563  MULT1,  FSTEP                 /STEP TO STATE 1
3307  4561           GETOP                /OUTBRK USING OP ADDR
3310  3030           DCA     BLSW         /MB TO BLSW
3311  1073           TAD     FLAG1        /GET OP SIGN
3312  7710           SPA CLA              /OP SIGN=1?
3313  5316           JMP     .+3          /YES
3314  4576           APBO                 /NO-A PLUS B TO 0
3315  7610           SKP CLA
3316  4577           AMBO                 /A MINUS B TO 0
3317  4545           LOADMQ               /0 TO MQ
3320  0021           OREG
3321  1036           TAD     ACMSW
3322  7710           SPA CLA              /GAC FRAC LT 0?
3323  5327           JMP     .+4          /YES
3324  4550           LOADA                /FAC FRAC TO A
3325  0036           FACFR
3326  5337           JMP     MULT1A
3327  1036           TAD     ACMSW        /COMPLIMENT FAC FRAC TO A
3330  7040           CMA
3331  3024           DCA     AMSW
3332  1037           TAD     ACLSW
3333  7040           CMA
3334  3025           DCA     ALSW
3335  1376           TAD     (7400        /COMPLIMENT 0 TO EXT
3336  3026           DCA     AEXT
3337  4572  MULT1A, CLRB                 /0 TO B
3340  5774           JMP     MULT2        /GO TO STATE 2
3374  3400
3375  2052
3376  7400
```

```
3377  6541
      3430          PAGE
```

```
3400  4563  MULT2,  FSTEP                 /STEP TO STATE 2
3401  1036  MADD,   TAD     FACFR
3402  7710           SPA CLA              /FAC FRAC LT 0?
3403  1123           TAD     C400         /YES -1 TO CARRY INSERT
3404  3076           DCA     CARYIN
3405  4576           APBO                 /ENABLE A PLUS B TO 0
3406  2071           ISZ     SHREG        /DEC SHFT CNTR -S,C,=0
3407  7410           SKP                  /NO-CONTINUE
3410  5227           JMP     MULT21       /YES-GO TO STATE 2-
3411  1032           TAD     MQMSW        /SHIFT MQ
3412  7130           STL RAR
3413  3032           DCA     MQMSW
3414  1033           TAD     MQLSW
3415  7010           RAR
3416  3033           DCA     MQLSW
3417  7420           SNL                  /MQ(23)=1?
3420  5225           JMP     .+5          /NO
3421  4546           LOADB                /YES-STROBE B
3422  0021           OREG
3423  1023           TAD     OEXT         /O EXT TO B EXT
3424  3031           DCA     BEXT
3425  4777'          JMS     RARB         /SHIFT B RIGHT
3426  5201           JMP     MADD         /CONTINUE UNTILL S,C,=0
3427  4544  MULT21,  LOADO                /B TO 0
3430  0027           BREG
3431  1052           TAD     COMREG
3432  7710           SPA CLA              /DOUBLE PRECISION?
3433  5236           JMP     .+3          /YES
3434  4565           DECOP                /DEC OR ADDR
3435  4565           DECOP
3436  4546           LOADB                /O TO B
3437  0021           OREG
3440  1023           TAD     OEXT         /O EXT TO B EXT
3441  3031           DCA     BEXT
3442  4573           CLRA                 /0 TO A
3443  1036           TAD     ACMSW        /
3444  7104           CLL RAL              /
3445  7200           CLA                  /PROG NEG?
3446  1073           TAD     FLAG1
3447  7530           SPA SZL
3450  7410           SKP
3451  5256           JMP     .+5          /NO-PROD IS POS
3452  7060           CMA CML
3453  7520           SMA SNL
3454  5256           JMP     .+2          /NO-PROD IS POS
3455  4577           AMBO                 /PROD IS NEG-A MINUS B TO 0
3456  1052           TAD     COMREG
3457  7710           SPA CLA              /DOUBLE PRECISION?
3460  5266           JMP     MULEND       /YES-GO TO DEPOSIT
```

```
3461  4563  MULT3,  FSTEP                 /STEP TO STATE 3
3462  4561           GETOP                /OUTBRK USING OP ADDR
3463  3033           DCA     MQLSW        /MB TO MQLSW
3464  4555           INCOP                /INC OP ADDR
3465  4555           INCOP
3466  5776'  MULEND, JMP     DEP          /GO TO DEPOSIT


3467  4564  PFDIV,  ENTER                 /ENTER DIVIDE
3470  7240           STA
3471  3004           DCA     MDFLAG       /SET MULT, DIV', FLAG

3472  4563  DIV0,   FSTEP                 /STEP TO STATE 0
3473  4561           GETOP                /OUTBRK USING OP ADDR
3474  3027           DCA     BMSW         /MB TO BMSW
3475  1027           TAD     BMSW
3476  7710           SPA CLA
3477  7040           CMA
3500  3073           DCA     FLAG1        /MB0 TO OP SIGN
3501  4555           INCOP                /INC OP ADDR
3502  1036           TAD     ACMSW        /GET FAC MSW
3503  7710           SPA CLA              /IS FAC LT 0?
3504  5310           JMP     .+4          /YES
3505  4550           LOADA                /NO-FAC TO A
3506  0036           FACFR
3507  5323           JMP     DIV0A
3510  1037           TAD     ACLSW        /COMPLIMENT FAC TO A
3511  7041           CIA                  /A+1 TO 0 AND 0 TO A IS
3512  3025           DCA     ALSW         /EQUAL TO THE 2S COMPLIMENT
3513  1036           TAD     ACMSW        /OF THE FAC TO A'.
3514  7040           CMA
3515  7430           SZL
3516  7001           IAC                  /THIS COMPLETES MINUS
3517  3024           DCA     AMSW
3520  3026           DCA     AEXT         /CLEAR A EXT
3521  4544           LOADO                /SYNC 0 REGISTERS
3522  0024           AREG
3523  1067  DIV0A,  TAD     SHFLAG
3524  7650           SNA CLA              /OPERATOR SELECTED SHIFT COUNT?
3525  1375           TAD     (34          /NO--34 TO SHIFT CNTR
3526  1070           TAD     SHFCNT       /YES--SHFCNT=0 IF NOT USED
3527  6567           LSHFT                /LOAD FPP SHIFT REG
3530  7041           CIA
3531  3071           DCA     SHREG
3532  3033           DCA     MQLSW        /CLEAR MQ REG
3533  3032           DCA     MQMSW
3534  3034           DCA     MQEXT
```

```
3535  4563  DIV1,  FSTEP                  /STEP TO STATE 1
3536  4561         GETOP                  /OUTBRK USING OP ADDR
3537  3030         DCA     BLSW           /MB TO BLSW
3540  3031         DCA     BEXT           /CLEAR B EXT
3541  4544         LOADO                  /B TO 0
3542  0027         BREG
3543  1021         TAD     OMSW           /O=0?
3544  7640         SZA CLA
3545  5360         JMP     .+13           /NO
3546  1022         TAD     OLSW           /O=0?
3547  7640         SZA CLA
3550  5360         JMP     .+10           /NO
3551  1053         TAD     PSTAT          /YES--SET DIVIDE BY 0 BIT
3552  0374         AND     (7377          /SAVE OTHER BITS
3553  1373         TAD     (400
3554  3053         DCA     PSTAT
3555  4525         TRACE                  /TRACING PROGRAM?
3556  2231         DIVZ                   /YES--DIVIDE BY ZERO
3557  5772         JMP     EXIT           /GO TO EXIT
3560  4544         LOADO                  /A TO 0
3561  0024         AREG

                   NPAGE

3562  5771         JMP I   (.+200&7600
3571  3600
3572  2427
3573  0400
3574  7377
3575  0034
3576  2052
3577  4251
      3600         PAGE
```

```
3650  4563  DIV2,   FSTEP                 /STEP TO STATE 2
3651  1073           TAD      FLAG1       /GET OP SIGN
3652  7710           SPA CLA              /OP SIGN=0?
3653  5214           JMP      DIVIDE      /NO
3654  1030           TAD      BLSW        /NEGATE B SO THAT AN
3655  7041           CIA                  /ADD WILL WORK IN BOTH
3656  3030           DCA      BLSW        /CASES SINCE THIS COMPUTER
3657  1027           TAD      BMSW        /DOES NOT HAVE A SUBTRACT
3660  7040           CMA                  /INSTRUCTION
3661  7430           SZL
3662  7001           IAC
3663  3027           DCA      BMSW
3664  1071  DIVIDE,  TAD      SHREG       /CHECK SHIFT REG FOR ZERO
3665  7650           SNA CLA
3666  5303           JMP      ENDDIV      /NO DIVIDE IF SH REG = 0
3667  7300           CLA CLL              /A MINUS OR PLUS B
3670  1025           TAD      ALSW        /TO TEMP REG FOR
3671  1030           TAD      BLSW        /TRIAL SUBTRACT
3672  3063           DCA      TLSW
3673  7430           SZL                  /DID LSW OVERFLOW?
3674  7001           IAC                  /YES--ADD LSW OVERFLOW
3675  7100           CLL
3676  1024           TAD      AMSW
3677  1027           TAD      BMSW
3700  3062           DCA      TMSW
3701  7012           RTR                  /SAVE CARRY OUT
3702  7012           RTR                  /IN BIT 27
3703  3103           DCA      T1          /SAVE CARRY OUT
3704  1034           TAD      MQEXT       /MQ(N) TO MQ(N+1)
3705  7004           RAL
3706  1103           TAD      T1          /CARRY OUT TO MQ(2?)
3707  3034           DCA      MQEXT
3710  1033           TAD      MQLSW
3711  7004           RAL
3712  3033           DCA      MQLSW
3713  1032           TAD      MQMSW
3714  7004           RAL
3715  3032           DCA      MQMSW       /END OF SHIFT MQ
3716  1103           TAD      T1
3717  7640           SZA CLA              /CARRY OUT=1?
3720  5253           JMP      .+3         /YES - LOAD 0
3721  4270           JMS      OLEFT       /NO - SHIFT 0
3722  5255           JMP      .+3         /END OF SHIFT 0
```

```
3653  4544          LOADO                 /CARRY OUT = 0
3654  0062          TREG                  /LOAD 0
3655  2071          ISZ     SHREG         /DEC SHFT CNTR
3656  7410          SKP                   /SHFT CNTR NOT = 0
3657  5303          JMP     ENDDIV        /S,C,=0 END OF DIVIDE
3660  1103          TAD     T1
3661  7640          SZA CLA               /CARRY OUT = 1 ?
3662  4270          JMS     OLEFT         /YES - SHIFT 0
3663  4550          LOADA                 /0 TO A
3664  0021          OREG
3665  1023          TAD     OEXT          /0 EXT TO A EXT
3666  3026          DCA     AEXT
3667  5214          JMP     DIVIDE        /CONT. DIVIDE UNTIL S,C,=0
```

```
3670  3000  OLEFT,  0
3671  1023           TAD     OEXT          /GET EXT
3672  7104           RAL CLL               /SHIFT LEFT AND INSERT ZERO
3673  3023           DCA     OEXT
3674  1022           TAD     OLSW          /GET LSW
3675  7004           RAL                   /SHIFT LEFT AND INSERT LINK
3676  3022           DCA     OLSW
3677  1021           TAD     OMSW          /GET MSW
3700  7004           RAL                   /SHIFT LEFT AND INSERT LINK
3701  3021           DCA     OMSW
3702  5670           JMP I   OLEFT         /RETURN

3703  4546  ENDDIV,  LOADB                 /YES - MQ TO B
3704  0032           MQREG
3705  1034           TAD     MQEXT         /LOAD EXTENSION
3706  3031           DCA     BEXT
3707  4573           CLRA                  /ZERO TO A
3710  3076           DCA     CARYIN        /CLEAR CARRY IN FF
3711  4544           LOADO                 /B TO 0
3712  0027           BREG
3713  1052           TAD     COMREG        /GET COMMAND REGISTER
3714  7700           SMA CLA               /FIXED POINT MODE ?
3715  5347           JMP     FLDIV         /NO - GO TO FLOATING DIVIDE
3716  1021           TAD     OMSW
3717  7700           SMA CLA               /0 LESS THAN 0?
3720  5331           JMP     DIVC          /NO--CONTINUE
3721  4335           JMS     QUONEG        /A - B IF QUOTIENT IS NEGATIVE
3722  1053           TAD     PSTAT         /GET STATUS REG
3723  0377           AND     (7577         /SAVE OTHER BITS
3724  1376           TAD     (200          /SET FRAC OVERFLOW BIT
3725  3053           DCA     PSTAT
3726  4525           TRACE                 /TRACING PROGRAM?
3727  2241           DIVOV                 /YES-DIVIDE F.P. OVERFLOW
3730  5775'          JMP     EXIT          /GO TO EXIT
3731  4544  DIVC,    LOADO                 /B TO 0 IF SIGNS ARE LIKE
3732  0027           BREG
3733  4335           JMS     QUONEG        /A - B IF QUOTIENT IS NEGATIVE
3734  5774'          JMP     DVEND         /GO TO DEPOSIT

3735  0000  QUONEG,  0
3736  1036           TAD     ACMSW         /GET FAC MSW
3737  7104           CLL RAL               /SAVE SIGN
3740  7200           CLA
3741  1073           TAD     FLAG1         /GET OP SIGN
3742  7530           SPA SZL               /OP SIGN = FAC(0)?
3743  7060           CMA CML
3744  7530           SPA SZL
3745  4577           AMBO                  /NO-A MINUS B TO 0
3746  5735           JMP I   QUONEG        /RETURN
```

```
3747  4565   FLDIV,  DECOP                  /DEC OP ADDRESS
3750  4565           DECOP                  /DEC OP ADDR
3751  1021           TAD     OMSW
3752  7730           SMA CLA                /O LT 0?
3753  5356           JMP     .+3            /NO
3754  4773'          JMS     RARB           /YES - SHIFT B RIGHT 1 PLACE
3755  2071           ISZ     SHREG          /INC SHFT CNTR
3756  4544           LOADO                  /B TO 0 IF SIGNS ARE LIKE
3757  0027           BREG
3760  1036           TAD     ACMSW          /GET FAC MSW
3761  7104           CLL RAL                /SAVE SIGN
3762  7200           CLA
3763  1073           TAD     FLAG1          /GET OP SIGN
3764  7530           SPA SZL
3765  7060           CMA CML                /OP SIGN=FAC(0)?
3766  7530           SPA SZL
3767  4577           AMBO                   /NO-A MINUS B TO 0
3770  5772'          JMP     DIV3           /GO TO STATE 3

3772  4000
3773  4251
3774  4007
3775  2427
3776  0200
3777  7577
      4000           PAGE
```

```
4000  4563    DIV3,   FSTEP                   /STEP TO STATE 3
4001  4561            GETOP                   /OUTBRK USING OP ADDR
4002  3033            DCA     MQLSW           /MB TO MQLSW
4003  3032            DCA     MQMSW           /CLEAR MQ MSW
4004  3034            DCA     MQEXT           /CLEAR MQ EXT
4005  4555            INCOP
4006  4555            INCOP                   /INC OP ADDR
4007  5777/   DVEND,  JMP     DEP             /GO TO DEPOSIT

4010  4564    FLDX,   ENTER                   /ENTER LDX
4011  4563    LDX1,   FSTEP                   /STEP TO STATE 1
4012  4573            CLRA                    /0 TO AMSW
4013  1051            TAD     PIR             /GET INSTRUCTION
4014  0376            AND     (7
4015  3025            DCA     ALSW
4016  4546            LOADB                   /X0 ADDR TO B
4017  0065            X0ADR
4020  4576            APB0                    /A PLUS B TO 0
4021  4543            LOADOP                  /0 TO OP ADDR
4022  0021            OREG
```

```
4023  4563    LDX2,   FSTEP                   /STEP TO STATE 2
4024  4560            GETPC                   /OUTBRK USING FPC
4025  3025            DCA     ALSW            /MB TO ALSW
4026  3024            DCA     AMSW            /0 TO AMSW
4027  4553            INCPC                   /INC FPC
4030  4544            LOADO                   /A TO O
4031  0024            AREG

4032  4563    LDX3,   FSTEP                   /STEP TO STATE 3
4033  1022            TAD     OLSW
4034  4537            PUTX                    /O LSW TO INDEX REG
4035  5775            JMP     FETCH           /GO TO FETCH


4036  4564    FALN,   ENTER                   /ENTER ALN
4037  4563    ALN1,   FSTEP                   /STEP TO STATE 1
4040  3024            DCA     AMSW            /0 TO AMSW
4041  1051            TAD     PIR             /GET INSTRUCTION
4042  0376            AND     (7
4043  3025            DCA     ALSW            /FIR 9-11 TO ALSW
4044  4546            LOADB                   /X0 ADDR TO B
4045  0065            X0ADR
4046  4576            APBO                    /A PLUS B TO O
4047  4543            LOADOP                  /O TO OP ADDR
4050  0021            OREG
4051  4572            CLRB                    /0 TO B
4052  1052            TAD     COMREG          /GET COMMAND REGISTER
4053  7710            SPA CLA                 /D, P, MODE ?
4054  5257            JMP     .+3             /YES = DO NOT STORE EXPONENT
4055  1035            TAD     ACEXP           /FAC EXP TO B LSW
4056  4527            STORB                   /SIGN EXTEND TO BMSW

4057  4563    ALN2,   FSTEP                   /STEP TO STATE 2
4060  1051            TAD     PIR             /GET INSTRUCTION
4061  0376            AND     (7
4062  7450            SNA                     /X = 0 ?
4063  5270            JMP     .+5             /YES = 27 TO A
4064  1374            TAD     (X0             /GET PROGRAM INDEX REG
4065  3103            DCA     T1              /INSTEAD OF OUTBRK
4066  1503            TAD I   T1              /CONTENTS OF X
4067  7410            SKP
4070  1101            TAD     D27+1           /GET OCTAL 27
4071  4530            STORA                   /MB OR 27 TO A
4072  1052            TAD     COMREG          /GET COMMAND REGISTER
4073  7710            SPA CLA                 /D, P, MODE
4074  5277            JMP     ALN22           /LEAVE EXP ALONE
4075  1025            TAD     ALSW
4076  3035            DCA     ACEXP           /MB TO FAC EXPONENT


4077  4577    ALN22,  AMBO                    /A MINUS B TO O
4100  1022            TAD     OLSW
4101  3071            DCA     SHREG           /O TO SHIFT CNTR
4102  4546            LOADB                   /O TO B
```

```
4173  0021          OREG
4174  1021          TAD     OMSW
4175  7710          SPA CLA
4176  7040          CMA
4177  3073          DCA     FLAG1           /00 TO SHIFT 0 FF
4110  1373          TAD     (27
4111  3025          DCA     ALSW            /12 TO ALSW
4112  3024          DCA     AMSW            /SIGN EXTEND TO AMSW
4113  1073          TAD     FLAG1
4114  7640          SZA CLA                 /SHFT 0 FF=1?
4115  5320          JMP     .+3             /YES
4116  4577          AMBO                    /NO--A MINUS B TO 0
4117  7410          SKP
4120  4576          APBO                    /A PLUS B TO 0

                    NPAGE

4121  5772          JMP I   (.+200&7600
4172  4200
4173  0027
4174  0200
4175  1236
4176  0007
4177  2052
      4200          PAGE
```

```
4200  4563  ALN3,  FSTEP                 /STEP TO STATE 3
4201  1021          TAD     OMSW
4202  7700          SMA CLA               /OO=1?
4203  5206          JMP     .+3           /NO
4204  4572          CLRB                  /YES = Ø TO B
4205  5210          JMP     .+3
4206  4546          LOADB                 /FAC FRAC TO B
4207  0036          FACFR
4210  4544          LOADO                 /B TO O
4211  0027          BREG
4212  4531          SQEZ                  /O=Ø?
4213  5215          JMP     ALN4          /NO-GO TO STATE 4
4214  3071          DCA     SHREG         /YES-Ø TO SHFT CNTR


4215  4563  ALN4,  FSTEP                 /STEP TO STATE 4
4216  1073          TAD     FLAG1
4217  7004          RAL                   /SAVE SHFT O FF
4220  7200          CLA
4221  1071          TAD     SHREG         /GET SHFT CNTR
4222  7420          SNL
4223  7041          CIA                   /ADJUST SHIFT CNTR FOR ISZ
4224  3071          DCA     SHREG
4225  1073          TAD     FLAG1
4226  7650          SNA CLA               /SHFT O FF=Ø?
4227  5232          JMP     .+3           /YES
4230  4532          SHFTO                 /NO-SHIFTO UNTIL S',C',=Ø
4231  5246          JMP     ALNEND
4232  4533          SHFTB                 /SHIFT B UNTIL S',C',=Ø
4233  1036          TAD     FACFR         /GET FAC FRACTION
4234  7710          SPA CLA               /O = O AND FAC FR NEG ?
4235  4531          SQEZ                  /O = Ø ?
4236  5244          JMP     ALNEND-2      /NO - B TO O
4237  7240          STA                   /YES - LOGICAL 1 TO O
4240  3021          DCA     OMSW          /LOGICAL 1 TO OMSW
4241  7240          STA
4242  3022          DCA     OLSW          /LOGICAL 1 TO OLSW
4243  5246          JMP     .+3           /DO NOT MOVE B TO O
4244  4544          LOADO                 /B TO O
4245  0027          BREG
4246  4547  ALNEND, LOADAC                /O TO FAC FRACTION
4247  0021          OREG
4250  5777/         JMP     FETCH         /GO TO FETCH
```

```
4251  0000  RARB,  0
4252  1027        TAD     BMSW
4253  7110        RAR CLL
4254  3027        DCA     BMSW
4255  1030        TAD     BLSW
4256  7010        RAR
4257  3030        DCA     BLSW
4260  1031        TAD     BEXT
4261  7010        RAR
4262  0376        AND     (7400       /KEEP IT 4 BITS LONG
4263  3031        DCA     BEXT
4264  5651        JMP I   RARB

4265  4564  FATX, ENTER               /ENTER ATX
4266  4563  ATX1, FSTEP               /STEP TO STATE 1
4267  3024        DCA     AMSW        /0 TO A MSW
4270  1051        TAD     PIR         /GET INSTRUCTION
4271  0375        AND     (7
4272  3025        DCA     ALSW        /PIR9-11 TO ALSW
4273  4546        LOADB               /X0 ADDR TO B
4274  0065        X0ADR
4275  4576        APB0                /A PLUS B TO 0
4276  4543        LOADOP              /0 TO OP ADDR
4277  0021        OREG
4300  1052        TAD     COMREG      /GET COMMAND REG
4301  7700        SMA CLA             /FLOATING POINT?
4302  5310        JMP     .+6         /YES
4303  4546        LOADB               /NO-FAC FRAC TO B
4304  0036        FACFR
4305  3071        DCA     SHREG       /0 TO SHIFT CNTR
4306  4774'       JMS     SETST3      /SET PROG STATE 3
4307  5773'       JMP     ATX4        /GO TO STATE 4
4310  1035        TAD     ACEXP       /FAC EXP TO B LSW
4311  4527        STORB               /SIGN EXTEND TO BMSW
```

```
4312  4563    ATX2,   FSTEP                   /STEP TO STATE 2
4313  1372            TAD     (27
4314  4530            STORA                   /27 TO ALSW - 0 TO AMSW
4315  4577            AMBO                    /A MINUS B TO 0
4316  1022            TAD     OLSW
4317  3071            DCA     SHREG           /0 TO SHIFT CNTR
4320  4546            LOADB                   /0 TO B
4321  0021            OREG
4322  1021            TAD     OMSW            /GET SIGN OF 0
4323  7710            SPA CLA
4324  7040            CMA
4325  3073            DCA     FLAG1           /00 TO SHFT 0 FF
4326  1372            TAD     (27
4327  3025            DCA     ALSW            /27 TO ALSW
4330  3024            DCA     AMSW            /0 TO AMSW
4331  1073            TAD     FLAG1
4332  7650            SNA CLA                 /SHFT 0 FF=1?
4333  5336            JMP     .+3             /NO
4334  4576            APBO                    /YES-A PLUS B TO 0
4335  7410            SKP
4336  4577            AMBO                    /A MINUS B TO 0


4337  4563    ATX3,   FSTEP                   /STEP TO STATE 3
4340  1021            TAD     OMSW
4341  7700            SMA CLA                 /00=1?
4342  5346            JMP     .+4             /NO
4343  3027            DCA     BMSW            /YES-0 TO B
4344  3030            DCA     BLSW
4345  5350            JMP     .+3
4346  4546            LOADB                   /FAC FRAC TO B
4347  0036            FACFR
4350  4544            LOADO                   /B TO 0
4351  0027            BREG
4352  4531            SQEZ                    /0=0?
4353  5773'           JMP     ATX4            /NO-GO TO STATE 4
4354  3071            DCA     SHREG           /YES--0 TO SHFT CNTR
4355  5773'           JMP     ATX4            /GO TO STATE 4

4372  0027
4373  4400
4374  6344
4375  0007
4376  7400
4377  1236
      4400            PAGE
```

```
4400  4563   ATX4,   FSTEP                  /STEP TO STATE 4
4401  1073           TAD     FLAG1          /GET SHIFT O FF
4402  7640           SZA CLA                /SHIFT O FF =1?
4403  5224           JMP     ATX5-1         /YES
4404  1071           TAD     SHREG
4405  7041           CIA                    /NEGATE SHIFT CNTR FOR ISZ
4406  3071           DCA     SHREG
4407  4533           SHFTB                  /SHIFT B UNTIL S.C.=0
4410  1036           TAD     FACFR          /GET FAC FRACTION
4411  7710           SPA CLA                /O = 0 & FAC FRAC NEG ?
4412  4531           SOEZ                   /O = 0 ?
4413  5221           JMP     ATX4A          /NO = B TO O
4414  7240           STA
4415  3021           DCA     OMSW           /LOGICAL 1 TO OMSW
4416  7240           STA
4417  3022           DCA     OLSW           /LOGICAL 1 TO OLSW
4420  5225           JMP     ATX5           /GO TO STATE 5
4421  4544   ATX4A,  LOADO                  /B TO O (STATE 4-1)
4422  0027           BREG
4423  7410           SKP
4424  4532           SHFTO                  /SHIFT O UNTIL S.C.=0

4425  4563   ATX5,   FSTEP                  /STEP TO STATE 5
4426  1022           TAD     OLSW           /SIMULATE INBRK USING
4427  3005           DCA     EMEM           /PROGRAM MEMORY
4430  1022           TAD     OLSW
4431  4537           PUTX                   /PUT OLSW IN PROG INDEX REG
4432  4570           CMEME                  /COMPARE MEMORY
4433  5777!          JMP     FETCH          /GO TO FETCH
```

```
4434  4564    FXTA,   ENTER                   /ENTER XTA
4435  4563    XTA1,   FSTEP                   /STEP TO STATE 1
4436  3024            DCA     AMSW            /0 TO AMSW
4437  1051            TAD     PIR             /GET INSTRUCTION
4440  0376            AND     (7
4441  3025            DCA     ALSW            /FIR9-11 TO ALSW
4442  4546            LOADB                   /X0 ADDR TO B
4443  0065            X0ADR
4444  4576            APBO                    /A PLUS B TO O
4445  4543            LOADOP                  /0 TO OP ADDR
4446  0021            OREG

4447  4563    XTA2,   FSTEP                   /STEP TO STATE 2
4450  1051            TAD     PIR             /GET INSTRUCTION
4451  0376            AND     (7
4452  1375            TAD     (X0             /COMPUTE INDEX REG ADDR
4453  3103            DCA     T1              /GET PROG INDEX REG INSTEAD
4454  1503            TAD I   T1              /OF OUTBRK
4455  4527            STORB                   /MB TO BLSW - SIGN EXTEND TO BMSW
4456  4544            LOADO                   /B TO O
4457  0027            BREG
4460  3071            DCA     SHREG           /0 TO SHIFT COUNTER
4461  1052            TAD     COMREG          /GET COMMAND REGISTER
4462  7710            SPA CLA                 /D, P, MODE ?
4463  5270            JMP     .+5             /YES - LEAVE EXPONENT ALONE
4464  1101            TAD     D27+1
4465  3035            DCA     ACEXP           /27 TO FAC EXP
4466  1035            TAD     ACEXP
4467  3033            DCA     MQLSW           /FAC EXP TO MQ LSW
4470  7240            STA                     /1 TO MULTIPLY - DIVIDE FLAG
4471  3004            DCA     MDFLAG          /FOR USE IN DEPOSIT
4472  5774/           JMP     DEP             /GO TO DEPOSIT
```

```
4473  4564  JXN,    ENTER                 /ENTER JXN
4474  4563  JXN1,   FSTEP                 /STEP TO STATE 1
4475  1051           TAD     PIR          /GET INSTRUCTION
4476  0373           AND     (70          /EXTRACT BITS 6-8
4477  7110           CLL RAR
4500  7012           RTR
4501  3025           DCA     ALSW         /FIR6-8 TO ALSW
4502  3024           DCA     AMSW
4503  4546           LOADB                /X0 ADDRESS TO B
4504  0065           X0ADR
4505  4576           APBO                 /A PLUS B TO 0
4506  4543           LOADOP               /O TO OP ADDR
4507  0021           OREG
4510  7040           CMA
4511  3021           DCA     OMSW         /LOGICAL 1 TO 0
4512  7040           CMA
4513  3022           DCA     OLSW
4514  5315           JMP     JXN2


4515  4563  JXN2,   FSTEP                 /STEP TO STATE 2
4516  1051           TAD     PIR          /GET INSTRUCTION
4517  0372           AND     (100         /EXTRACT BIT 5
4520  7640           SZA CLA              /FIR 5=1?
4521  5330           JMP     JXN2A        /YES
4522  4557           GETX                 /NO-OUTBRK USING OP ADDR
4523  3025           DCA     ALSW         /MB TO ALSW
4524  3024           DCA     AMSW         /0 TO AMSW
4525  4544           LOADO                /A TO O
4526  0024           AREG
4527  5334           JMP     .+5
4530  4551  JXN2A,  INCX                 /REQUEST INC BRK (X=X+1)
4531  4557           GETX                 /GET X REG
4532  7650           SNA CLA              /OVERFLOW?
4533  4571           CLRO                 /YES - LOGICAL 0 TO O
4534  4531           SOEZ                 /SKIP IF O=0
4535  5340           JMP     JXN3         /O NOT EQUAL 0
4536  4553           INCPC                /INC FPC
4537  5354           JMP     JXNEND       /GO TO FETCH
```

```
4540  4563    JXN3,   FSTEP                   /STEP TO STATE 3
4541  4560            GETPC                   /OUTBRK USING FPC
4542  3025            DCA     ALSW            /MB TO ALSW
4543  1051            TAD     PIR             /GET INSTRUCTION
4544  0376            AND     (7
4545  3024            DCA     AMSW            /FIR9-11 TO AMSW
4546  4544            LOADO                   /A TO O
4547  0024            AREG
4550  4543            LOADOP                  /O TO OP ADDR
4551  0021            OREG
4552  4542            LOADPC                  /OP ADDR TO FPC
4553  0044            OPADR

4554  5777/   JXNEND, JMP     FETCH           /GO TO FETCH

4572  0100
4573  0070
4574  2052
4575  0200
4576  0007
4577  1236
      4600            PAGE
```

```
4600  4564    JSR,    ENTER                 /ENTER JSR
4601  4563    JSR1,   FSTEP                 /STEP TO STATE 1
4602  4546            LOADB                 /P0 ADDR TO B
4603  2054            PBASE
4604  4544            LOADO                 /B TO O
4605  2027            BREG
4606  4543            LOADOP                /O TO OP ADDR
4607  2021            OREG
4610  4555            INCOP                 /INC OP ADDR

4611  4563    JSR2,   FSTEP                 /STEP TO STATE 2
4612  4560            GETPC                 /OUTBRK USING FPC
4613  3025            DCA     ALSW          /MB TO ALSW
4614  1051            TAD     PIR           /GET INSTRUCTION
4615  0377            AND     (7
4616  3024            DCA     AMSW          /FIR9-11 TO AMSW
4617  4553            INCPC                 /INC FPC
4620  4544            LOADO                 /A TO O
4621  0024            AREG

4622  4563    JSR3,   FSTEP                 /STEP TO STATE 3
4623  1041            TAD     PFPC          /GET FPC FIELD BITS
4624  0377            AND     (7
4625  1376            TAD     (1030         /SEE JMK STATE 3-1
4626  3006            DCA     MMEM          /SIMULATE INBRK
4627  4555            INCOP                 /INC OP ADDR

4630  4563    JSR4,   FSTEP                 /STEP TO STATE 4
4631  1042            TAD     PFPC+1        /SIMULATE INBRK
4632  3007            DCA     LMEM
4633  4565            DECOP                 /DEC OP ADDR FOR COMPARE
4634  4567            CMEMF                 /COMPARE MEMORY
4635  4543            LOADOP                /O TO OP ADDR
4636  0021            OREG
4637  4542            LOADPC                /OP ADR TO FPC
4640  0044            OPADR
4641  5775            JMP     FETCH         /GO TO FETCH
```

```
4642  4564    JSA,    ENTER                   /ENTER JSA
4643  4563    JSA2,   FSTEP                   /STEP TO STATE 2
4644  4560            GETPC                   /OUTBRK USING FPC
4645  3025            DCA     ALSW            /MB TO ALSW
4646  1051            TAD     PIR             /GET INSTRUCTION
4647  0377            AND     (7
4650  3024            DCA     AMSW            /FIR9-11 TO AMSW
4651  4553            INCPC                   /INC FPC
4652  4544            LOADO                   /A TO O
4653  0024            AREG
4654  4543            LOADOP                  /O TO OP ADDR
4655  0021            OREG

4656  4563    JSA3,   FSTEP                   /STEP TO STATE 3
4657  1041            TAD     PFPC            /SEE JSB STATE 3-1
4660  0377            AND     (7              /FPC F,B,
4661  1376            TAD     (1030           /JA= 1030 = JUMP ALWAYS
4662  3006            DCA     MMEM            /SIMULATE INBRK TO PROG MEMORY
4663  4555            INCOP                   /INC OP ADDR

4664  4563    JSA4,   FSTEP                   /STEP TO STATE 4
4665  1042            TAD     PFPC+1          /GET FPC
4666  3007            DCA     LMEM            /SIMULATE INBRK
4667  4565            DECOP                   /DEC OP ADDR FOR COMPARE
4670  4567            CMEMF                   /COMPARE MEMORY FRACTION
4671  4555            INCOP                   /RESTORE OP ADDR
4672  4555            INCOP                   /INC OP ADDR (STATE 4-2)
4673  4542            LOADPC                  /OP ADDR TO FPC
4674  0044            OPADR
4675  5775            JMP     FETCH           /GO TO FETCH


4775  1236
4776  1030
4777  0007
      5000            PAGE
```

```
5000  4536    JEQ,    SAEZ                    /FAC=0?
5001  5245            JMP     JFALSE          /NO
5002  5227            JMP     JTRUE           /YES

5003  4534    JGE,    SALZ                    /FAC GT OR=0?
5004  5227            JMP     JTRUE           /YES
5005  5245            JMP     JFALSE          /NO

5006  4535    JLE,    SAGZ                    /FAC LT OR=0?
5007  5227            JMP     JTRUE           /YES
5010  5245            JMP     JFALSE          /NO

5011  4536    JNE,    SAEZ                    /FAC=0?
5012  5227            JMP     JTRUE           /NO
5013  5245            JMP     JFALSE          /YES

5014  4534    JLT,    SALZ                    /FAC LT 0?
5015  5245            JMP     JFALSE          /NO
5016  5227            JMP     JTRUE           /YES

5017  4535    JGT,    SAGZ                    /FAC GT 0?
5020  5245            JMP     JFALSE          /NO
5021  5227            JMP     JTRUE           /YES

5022  7300    JAL,    CLA CLL
5023  1021            TAD     OMSW            /GET O SIGN
5024  7700            SMA CLA                 /IS O NEG
5025  5245            JMP     JFALSE          /NO
5026  5227            JMP     JTRUE           /YES


5027  4564    JTRUE,  ENTER                   /ENTER JMP CONDITION TRUE
5030  4563    JTRUE1, FSTEP                   /STEP TO STATE 1
5031  4560            GETPC                   /OUTBRK USING PFC
5032  3025            DCA     ALSW            /MB TO ALSW
5033  1051            TAD     PIR             /GET INSTRUCTION
5034  0377            AND     (7
5035  3024            DCA     AMSW            /FIR9-11 TO AMSW
5036  4544            LOADO                   /A TO O
5037  0024            AREG
5040  4543            LOADOP                  /O TO OP ADDR
5041  0021            OREG
5042  4542            LOADPC                  /OP ADDR TO FPC
5043  0044            OPADR
5044  5776'           JMP     FETCH           /GO TO FETCH
```

```
5045  4564  JFALSE, ENTER                 /ENTER JMP CONDITION FALSE
5046  4553          INCPC                 /INC FPC
5047  5776/         JMP     FETCH         /GO TO FETCH
5050  4564  JAC,    ENTER                 /ENTER JAC
5051  4563  JAC1,   FSTEP                 /STEP TO STATE 1
5052  4546          LOADB                 /FAC FRAC TO B
5053  0036          FACFR
5054  4544          LOADO                 /B TO O
5055  0027          BREG
5056  4543          LOADOP                /O TO OP ADDR
5057  0021          OREG
5060  4542          LOADPC                /OP ADDR TO FPC
5061  0044          OPADR
5062  5776/         JMP     FETCH         /GO TO FETCH

5063  4564  FSTF,   ENTER                 /ENTER STF
5064  1052          TAD     COMREG
5065  0375          AND     (3777         /RESET D',P', MODE FP
5066  3052          DCA     COMREG
5067  5776/         JMP     FETCH         /GO TO FETCH

5070  4564  FSTD,   ENTER                 /ENTER STD
5071  1052          TAD     COMREG
5072  7104          CLL RAL
5073  7130          STL RAR               /SET D',P', MODE FP
5074  3052          DCA     COMREG
5075  5776/         JMP     FETCH         /GO TO FETCH

5076  4564  FNEG,   ENTER                 /ENTER NEG
5077  4563  NEG1,   FSTEP                 /STEP TO STATE 1
5100  1036          TAD     ACMSW
5101  7040          CMA
5102  3024          DCA     AMSW          /COMPLIMENT FAC FRAC
5103  1037          TAD     ACLSW         /TO A
5104  7040          CMA
5105  3025          DCA     ALSW
5106  4544          LOADO                 /A TO O
5107  0024          AREG
5110  4554          INCOR                 /+1 TO O
5111  4547          LOADAC                /O TO FAC FRAC
5112  0021          OREG
5113  5776/         JMP     FETCH         /GO TO FETCH

5114  4564  FNORM,  ENTER                 /ENTER FNORM
5115  4563  NORM1,  FSTEP                 /STEP TO STATE 1
5116  4546          LOADB                 /FAC FRAC TO B
5117  0036          FACFR
5120  4544          LOADO                 /B TO O
5121  0027          BREG
5122  1035          TAD     ACEXP         /FAC EXP TO MQLSW
5123  3033          DCA     MQLSW
5124  3071          DCA     SHREG         /0 TO SHIFT CNTR
5125  5774/         JMP     DEP           /GO TO DEPOSIT
```

```
5126  4564   FPAUSE, ENTER                   /ENTER PAUSE
5127  4563   PAUS2,  FSTEP                   /STEP TO STATE 1
5130  4773'         JMS     TYP             /TYPE FPP PAUSE
5131  2161          TPAUSE
5132  0000          2
5133  1106          TAD     STEPSW          /SAVE SINGLE STEP SWITCH
5134  3103          DCA     T1
5135  7240          STA
5136  3106          DCA     STEPSW          /SET SINGLE STEP SW
5137  1105          TAD     CSTATE
5140  0372          AND     (377            /RESET PROG MAJOR STATE
5141  3105          DCA     CSTATE

5142  4563   PAUS1,  FSTEP                   /WAIT FOR OPERATOR
5143  1103          TAD     T1              /RESTORE SINGLE STEP SWITCH
5144  3106          DCA     STEPSW
5145  6555          FPST                    /RESTART FPP
5146  7000          NOP
5147  5776'         JMP     FETCH           /GO TO FETCH



5150  4564   FCLA,   ENTER                   /ENTER CLA
5151  3036          DCA     ACMSW           /0 TO FAC
5152  3037          DCA     ACLSW
5153  1052          TAD     COMREG          /GET THE COMMAND REGISTER
5154  7700          SMA CLA                 /D,P, MODE ?
5155  3035          DCA     ACEXP           /NO = CLEAR THE AC EXPONENT
5156  5776'         JMP     FETCH           /GO TO FETCH

5157  4564   SETB,   ENTER                   /ENTER SET BASE
5160  4563   SETB1,  FSTEP                   /STEP TO STATE 1
5161  4560          GETPC                   /OUTBRK USING FPC
5162  3055          DCA     PBASE+1         /MB TO P0 ADDR
5163  1051          TAD     PIR             /GET INSTRUCTION
5164  0377          AND     (7
5165  3054          DCA     PBASE           /FIR9-11 TO P0 F',B',
5166  4553          INCPC                   /INC FPC
5167  5776'         JMP     FETCH           /GO TO FETCH

5172  0377
5173  7313
5174  2052
5175  3777
5176  1236
5177  0007
      5200          PAGE
```

```
5200  4564    SETX,   ENTER                   /ENTER SET X
5201  4563    SETX1,  FSTEP                   /STEP TO STATE 1
5202  4560            GETPC                   /OUTBRK USING FPC
5203  3066            DCA     X0ADR+1         /MB TO X0 ADDR
5204  1051            TAD     PIR             /GET INSTRUCTION
5205  0377            AND     (7
5206  3065            DCA     X0ADR           /FIR9-11 TO X0 F.B.
5207  4553            INCPC                   /INC FPC
5210  4541            MOVEX                   /SET PROGRAM INDEX REGS
5211  5776            JMP     FETCH           /GO TO FETCH

5212  4564    ADDX,   ENTER                   /ENTER ADDX
5213  4563    ADDX1,  FSTEP                   /STEP TO STATE 1
5214  3024            DCA     AMSW            /0 TO AMSW
5215  1051            TAD     PIR             /GET INSTRUCTION
5216  0377            AND     (7
5217  3025            DCA     ALSW            /FIR9-11 TO ALSW
5220  4546            LOADB                   /X0 ADDR TO B
5221  0065            X0ADR
5222  4576            APBO                    /A PLUS B TO O
5223  4543            LOADOP                  /O TO OP ADDR
5224  0021            OREG

5225  4563    ADDX2,  FSTEP                   /STEP TO STATE 2
5226  1051            TAD     PIR
5227  0377            AND     (7              /GET PROGRAM INDEX REG INSTEAD
5230  1057            TAD     PXP             /OF OUTBRK USING OP ADDR
5231  3103            DCA     T1
5232  1503            TAD I   T1
5233  3030            DCA     BLSW            /MB TO BLSW
5234  3027            DCA     BMSW            /0 TO BMSW

5235  4563    ADDX3,  FSTEP                   /STEP TO STATE 3
5236  4560            GETPC                   /OUTBRK USING FPC
5237  3025            DCA     ALSW            /BM TO ALSW
5240  3024            DCA     AMSW            /0 TO AMSW
5241  4553            INCPC                   /INC PFC
5242  4576            APBO                    /A PLUS B TO O

5243  4563    ADDX4,  FSTEP                   /STEP TO STATE 4
5244  1022            TAD     OLSW
5245  4537            PUTX                    /O LSW TO X REG
5246  5776            JMP     FETCH           /GO TO FETCH
```

```
5247  4564    FSTA,   ENTER                   /TYPE ENTER FSTA
5250  4563    STA0,   FSTEP                   /STEP TO MAJOR STATE 0
5251  1052            TAD     COMREG          /GET COMMAND REGISTER
5252  7710            SPA CLA                 /D, P, MODE ?
5253  5260            JMP     STA1            /YES = GO TO STATE 1
5254  1035            TAD     ACEXP           /INBRK USING OP ADDR
5255  3005            DCA     EMEM            /FAC EXP TO MB
5256  4570            CMEME                   /COMPARE MEMORY EXPONENT
5257  4555            INCOP                   /INC OP ADDR

5260  4563    STA1,   FSTEP                   /STEP TO MAJOR STATE 1
5261  1036            TAD     ACMSW           /INBRK USING OP ADDR
5262  3006            DCA     MMEM            /FAC MSW TO MB
5263  4555            INCOP                   /INC OF ADDR

5264  4563    STA2,   FSTEP                   /STEP TO MAJOR STATE 2
5265  1037            TAD     ACLSW           /INBRK USING OP ADDR
5266  3007            DCA     LMEM            /FAC LSW TO MB
5267  4565            DECOP                   /DEC OP ADDR FOR COMPARE
5270  4567            CMEMF                   /COMPARE MEMORY
5271  4555            INCOP                   /RESTORE OP ADDR
5272  5776/           JMP     FETCH           /GO TO FETCH


5273  4564    FLDA,   ENTER                   /TYPE ENTER LDA
5274  4563    LDA0,   FSTEP                   /STEP TO  STATE 0
5275  1052            TAD     COMREG          /GET COMMAND REGISTER
5276  7710            SPA CLA                 /D, P, MODE ?
5277  5303            JMP     LDA1            /YES = GO TO STATE 1
5300  4561            GETOP                   /OUTBRK USING OP ADDR
5301  3035            DCA     ACEXP           /MB TO FAC EXP
5302  4555            INCOP

5303  4563    LDA1,   FSTEP                   /STEP TO MAJOR STATE 1
5304  4561            GETOP                   /OUTBRK USING OP ADDR
5305  3024            DCA     AMSW
5306  4555            INCOP

5307  4563    LDA2,   FSTEP                   /STEP TO MAJOR STATE 2
5310  4561            GETOP                   /OUTBRK USING OP ADDR
5311  3025            DCA     ALSW
5312  4544            LOADO                   /A TO O
5313  0024            AREG
5314  4547            LOADAC                  /O TO FAC FRAC
5315  0021            OREG
5316  5776/           JMP     FETCH


5376  1236
5377  0027
      5400            PAGE
```

```
                    / MINI ROUTINES (TR3)

5440  0000  FCLR,  0                        /CLEAR FPP AND PROGRAM REGS
5441  6552        FPICL                     /CLEAR FPP
5442  1377        TAD     (CLRT -1          /GET ADDRESS OF TABLE
5443  3010        DCA     10
5444  3105        DCA     CSTATE
5445  1410  FCLRA, TAD I  10                /GET ADDRESS
5446  7450        SNA                       /END OF TABLE?
5447  5600        JMP I   FCLR              /YES-EXIT
5450  3011        DCA     11                /NO-SAVE ADDRESS
5451  1410        TAD I   10                /GET COUNT
5452  3217        DCA     CLRCT             /SAVE COUNT
5453  3411        DCA I   11                /CLEAR LOCATION
5454  2217        ISZ     CLRCT             /END?
5455  5213        JMP     .-2               /NO
5456  5205        JMP     FCLRA             /YES-GET NEXT ADDR
5457  0000  CLRCT, 0
5460  0020  CLRT,  REGS                     /ADDR OF FPP REGS
5461  7760        -20
5462  0000        0
5463  0000  INC1,  0                        /INCREMENT OP ADDR
5464  7300        CLA CLL
5465  2045        ISZ     OPADR+1           /OP ADDR
5466  7410        SKP
5467  2044        ISZ     OPADR             /OP FLD
5470  7000        NOP
5471  5623        JMP I   INC1              /RETURN

5472  0000  INC2,  0                        /INCREMENT FPC
5473  7300        CLA CLL
5474  2042        ISZ     PFPC+1            /FPC ADDR
5475  7410        SKP
5476  2041        ISZ     PFPC              /FPC FIELD
5477  7000        NOP
5500  5632        JMP I   INC2              /RETURN

5501  0000  INC3,  0                        /INCREMENT ADRS
5502  7300        CLA CLL
5503  2050        ISZ     PAPT+1            /APT ADDRESS
5504  7410        SKP
5505  2047        ISZ     PAPT              /APT FIELD
5506  7000        NOP
5507  5641        JMP I   INC3              /RETURN
```

```
5450  0000    APTC,   0                       /COMPARE APT ADDRESS
5451  7200            CLA
5452  6565            RAPT                    /GET APT ADDR FROM FPP
5453  3301            DCA     SAVAPT          /SAVE FOR ERROR REPORT
5454  1301            TAD     SAVAPT          /RESTORE AC
5455  7041            CIA
5456  1050            TAD     PAPT+1          /COMPARE WITH PROGRAM ADDR
5457  7650            SNA CLA                 /APT ADDR OK?
5460  5650            JMP I   APTC            /RETURN
5461  1250            TAD     APTC            /GET RETURN ADDRESS
5462  4776            JMS     ERR             /TYPE ERROR DATA
5463  0005            ERROR   0005            /ADDRS ERROR CODE
5464  5650            JMP I   APTC            /RETURN TO MAIN PROGRAM

5465  0000    SPECFL, 0                       /SPECIAL OVERFLOW CONDITION
5466  1021            TAD     OMSW
5467  7500            SMA                     /IS O NEGATIVE ?
5470  5276            JMP     NOFLOW          /NO - RETURN + 1
5471  7104            CLL RAL                 /MOVE SIGN TO LINK
5472  7640            SZA CLA                 /ARE ALL OTHER BITS OF MSW 0 ?
5473  5276            JMP     NOFLOW          /NO - RETURN + 1
5474  1022            TAD     OLSW            /YES - CHECK LSW
5475  7640            SZA CLA                 /DOES O = 4000 0000 ?
5476  2265    NOFLOW, ISZ     SPECFL          /NO - RETURN + 1
5477  7300            CLA CLL
5500  5665            JMP I   SPECFL          /RETURN

5501  0000    SAVAPT, 0



5576  7317
5577  5417
      5600            PAGE
```

```
5600  0000  STEP,  0
5601  6031       KSF
5602  7410       SKP
5603  4777'      JMS     KEYCK
5604  7240       STA
5605  1200       TAD     STEP          /GET EXIT ADDRESS
5606  7041       CIA
5607  1113       TAD     EXADDR        /COMPARE WITH RETURN ADDRESS
5610  7640       SZA CLA               /EXIT AT THIS ADDRESS ?
5611  5214       JMP     .+3           /NO
5612  4574       CLEAR                 /YES - CLEAR FPP AND REGISTERS
5613  5776'      JMP     INIT          /GO TO INIATE
5614  1117       TAD     CKO           /CHECK O FLAG
5615  7640       SZA CLA               /CHECK O REG ?
5616  4775'      JMS     COMPO         /YES
5617  7240       STA
5620  3117       DCA     CKO           /SET FLAG FOR NEXT STATE
5621  1120       TAD     CKOP          /CHECK OP ADDR FLAG
5622  7640       SZA CLA               /CHECK OP ADDR ?
5623  4774'      JMS     CKOPAD        /YES
5624  7240       STA
5625  3120       DCA     CKOP          /SET FLAG FOR NEXT STATE
5626  7604       LAS
5627  0373       AND     (400          /GET SWITCH 3
5630  7640       SZA CLA               /IS SWITCH 3 SET ?
5631  5250       JMP     SSTEP         /YES - GO TO SINGLE STEP
5632  1106       TAD     STEPSW
5633  7640       SZA CLA               /SINGLE STEP MODE
5634  5250       JMP     SSTEP         /YES
5635  7240       STA
5636  1200       TAD     STEP          /GET RETURN ADDRESS
5637  7041       CIA
5640  1114       TAD     ENTSTP        /COMPARE WITH ENTER STEP ADDR
5641  7640       SZA CLA               /ENTER STEP MODE ?
5642  5271       JMP     STEPGO        /NO
5643  7040       CMA
5644  3106       DCA     STEPSW        /YES - SET STEP SWITCH
5645  4772'      JMS     TYP
5646  2626       INSTEP
5647  0000       0
5650  4771'  SSTEP,  JMS     ASC       /CONVERT RETURN ADDR TO ASCII
5651  5600       STEP
5652  2652       ADDR
5653  4772'      JMS     TYP           /TYPE RETURN ADDRESS
5654  2652       ADDR
5655  0000       0
5656  4777'      JMS     KEYCK         /WAIT FOR TTY INPUT
```

```
5657  7240            STA
5660  1200            TAD     STEP            /GET RETURN ADDRESS
5661  7041            CIA
5662  1115            TAD     EXSTP           /EXIT STEP ADDRESS
5663  7640            SZA CLA                 /EXIT STEP MODE ?
5664  5271            JMP     STEPGO          /NO
5665  4772'           JMS     TYP
5666  2636            OUTSTP                  /EXIT STEP MODE
5667  0000            0
5670  3106            DCA     STEPSW          /YES - RESET STEP SWITCH
5671  1105    STEPGO, TAD     CSTATE
5672  1373            TAD     (400            /INC, CURRENT STATE REG
5673  3105            DCA     CSTATE
5674  4770'           JMS     MSTATE          /CHECK STATES
5675  7610            SKP CLA                 /STATE ERROR
5676  5302            JMP     .+4             /STATES OK
5677  1200            TAD     STEP            /GET RETURN ADDRESS
5700  4767'           JMS     ERR             /GO TO ERROR ROUTINE
5701  0000            ERROR   0000            /TIME STATE ERROR CODE
5702  6561            FMAINT                  /STEP FPP TO NEXT STATE
5703  7300            CLA CLL
5704  5600            JMP I   STEP
```

```
5705  1766'  TREXIT,  TAD     ASCCH          /GET ASCII CHARACTER
5706  0365           AND     (77            /STRIP TO 6 BITS
5707  1364           TAD     (-3
5710  7450           SNA                    /CNTRL C ?
5711  5763'          JMP     7700           /YES - RETURN TO DTA MONITOR
5712  1362           TAD     (-1
5713  7450           SNA                    /CNTRL D ?
5714  5321           JMP     .+5            /YES - SETUP TO READ IN DIAL
5715  1361           TAD     (-14
5716  7650           SNA CLA                /CNTRL P ?
5717  5760'          JMP     7600           /YES - RETURN TO PS-8 MONITOR
5720  5757'          JMP     DOTERR         /ILLEGAL INPUT
5721  1330           TAD     ER1            /MOVE DIAL LOADER TO
5722  3756'          DCA     4015           /4015 FOR TAPE READ
5723  1331           TAD     ER2            /INTO PROPER LOCATION
5724  3755'          DCA     4016           /AND AUTO START
5725  1332           TAD     ER3
5726  3754'          DCA     4017
5727  5756'          JMP     4015           /READ DIAL TAPE AND EXECUTE

5730  6141   ER1,    6141                   /LINC
5731  0701   ER2,    0701                   /RCG
5732  7300   ER3,    7300                   /BLOCK NUMBER



5754  4017
5755  4016
5756  4015
5757  0432
5760  7600
5761  7764
5762  7777
5763  7700
5764  7775
5765  0077
5766  0345
5767  7317
5770  6000
5771  7346
5772  7313
5773  0400
5774  6320
5775  6120
5776  1064
5777  0250
      6000           PAGE
```

```
                        /CHECK MAJOR STATE REGISTER

6000  0000    MSTATE, 0                       /GET PROGRAM STATE GENERATOR
6001  7300            CLA CLL
6002  1105            TAD     CSTATE
6003  0377            AND     (7774           /DELETE BITS 10 & 11
6004  3000            DCA     MT1             /SAVE FOR ERROR
6005  6562            RSTATE                  /READ STATE FROM FPP
6006  0377            AND     (7774           /DELETE BITS 10 & 11
6007  3001            DCA     MT2             /SAVE FOR ERROR
6010  1001            TAD     MT2
6011  7041            CIA
6012  1000            TAD     MT1             /COMPARE WITH PROGRAM STATE
6013  7650            SNA CLA                 /IS MAJOR STATE OK?
6014  2200            ISZ     MSTATE          /YES-RETURN+1
6015  5600            JMP I   MSTATE

6016  0000    TOPC,   0                       /LOAD THE FPC
6017  7240            STA
6020  1216            TAD     TOPC            /GET RETURN -1
6021  4261            JMS     LOAD
6022  0040            PFPC-1                  /ADDRESS OF FPC
6023  0000    TOAC,   0                       /LOAD THE FAC
6024  7240            STA
6025  1223            TAD     TOAC            /GET RETURN-1
6026  4261            JMS     LOAD
6027  0035            PFAC                    /ADDR OF FAC
6030  0000    TOO,    0                       /LOAD THE OREG
6031  7240            STA
6032  1230            TAD     TOO             /GET RETURN-1
6033  4261            JMS     LOAD
6034  4020            OMSW-1  LDEXT           /ADDR OF O
6035  0000    TOA,    0                       /LOAD THE A REG
6036  7240            STA
6037  1235            TAD     TOA             /GET RETURN-1
6040  4261            JMS     LOAD
6041  0023            AMSW-1                  /ADDR OF A
```

```
6042  0000  TOB,   0                    /LOAD THE A REG
6043  7240        STA
6044  1242        TAD     TOB           /GET RETURN-1
6045  4261        JMS     LOAD
6046  0026        BMSW-1                /ADDR OF B

6047  0000  TOMQ,  0
6050  7240        STA
6051  1247        TAD     TOMQ
6052  4261        JMS     LOAD
6053  0031        MQREG-1               /ADDR OF MQ

6054  0000  TOOP,  0                    /LOAD OP ADDR
6055  7240        STA
6056  1254        TAD     TOOP          /GET RETURN-1
6057  4261        JMS     LOAD
6060  0043        OPADR-1               /ADDR OF OP ADDR
6061  0000  LOAD,  0
6062  3010        DCA     10            /SAVE RETURN
6063  7360        STA STL
6064  1410        TAD I   10            /GET FROM ADDR
6065  3011        DCA     11            /SAVE FROM ADDR
6066  1661        TAD I   LOAD          /GET TO ADDR
6067  0376        AND     (3777
6070  3012        DCA     12
6071  1411        TAD I   11            /GET MSW
6072  3412        DCA I   12            /STORE MSW
6073  1411        TAD I   11            /GET LSW
6074  3412        DCA I   12            /STORE LSW
6075  1661        TAD I   LOAD
6076  7710        SPA CLA               /LOAD EXT ?
6077  1411        TAD I   11            /YES - GET THIRD WORD
6100  0375        AND     (7400         /SAVE 4 BITS ONLY
6101  3412        DCA I   12            /CLEAR OR LOAD EXT
6102  7300        CLA CLL
6103  5410        JMP I   10
```

```
6124  7000  APTPAC, 7
6125  1044        TAD     OPADR           /FIELD BITS OF OP ADDR
6126  7104        RAL CLL
6127  7006        RTL
6130  1054        TAD     PBASE           /FIELD BITS OF P0 ADDR
6131  7104        RAL CLL
6132  7006        RTL
6133  1065        TAD     X0ADR           /FIELD BITS OF X0 ADDR
6134  7104        RAL CLL
6135  7006        RTL
6136  1041        TAD     PFPC            /FIELD BITS OF FPC
6137  5704        JMP I   APTPAC          /RETURN

6140  0000  COMPO, 0
6141  7200        CLA
6142  6563        ROMSW                   /READ OMSW FROM FPP
6143  3002        DCA     SAVOM
6144  6564        ROLSW
6145  3003        DCA     SAVOL
6146  1002        TAD     SAVOM
6147  7041        CIA
6150  1021        TAD     OMSW            /COMPARE WITH PROG OMSW
6151  7640        SZA CLA                 /IS OMSW CORRECT?
6152  5340        JMP     ORERR           /NO - GO TO ERROR
6153  1003        TAD     SAVOL
6154  7041        CIA
6155  1022        TAD     OLSW            /COMPARE WITH PROG OLSW
6156  7650        SNA CLA                 /IS OLSW CORRECT?
6157  5740        JMP I   COMPO           /RETURN
6160  1774' ORERR, TAD    STEP
6161  4773' JMS   ERR
6162  0021        ERROR   0021
6163  5740        JMP I   COMPO
```

```
6144  7000    STRA,   0
6145  7100            CLL
6146  7510            SPA                     /CHECK SIGN
6147  7120            STL                     /SAVE SIGN IN LINK
6150  3025            DCA     ALSW            /STORE LSW
6151  7430            SZL                     /WAS LSW MINUS ?
6152  7240            STA                     /YES - MSW = 7777
6153  3024            DCA     AMSW
6154  3026            DCA     AEXT            /CLEAR EXT
6155  5744            JMP I   STRA            /RETURN

6156  0000    STRB,   0
6157  7100            CLL
6160  7510            SPA                     /CHECK SIGN
6161  7120            STL                     /SAVE SIGN IN LINK
6162  3030            DCA     BLSW            /STORE LSW
6163  7430            SZL                     /WAS SIGN MINUS ?
6164  7240            STA                     /YES - MSW = 7777
6165  3027            DCA     BMSW
6166  3031            DCA     BEXT            /CLEAR EXT
6167  5756            JMP I   STRB            /RETURN



6173  7317
6174  5600
6175  7400
6176  3777
6177  7774
      6200            PAGE
```

```
6200  0000  SETUP,  0
6201  7200        CLA
6202  1377        TAD     (SETTAB-1       /GET ADDRESS OF TABLE
6203  3010        DCA     10
6204  6211        CDF     10
6205  1410        TAD I   10              /GET MAJOR STATE WORD
6206  3105        DCA     CSTATE
6207  1410        TAD I   10              /GET TEXT ADDR
6210  3233        DCA     ENTTXT
6211  1410        TAD I   10
6212  7440        SZA
6213  5215        JMP     .+2
6214  5242        JMP     BADNWS          /SOMETHING IS WRONG
6215  7041        CIA
6216  1200        TAD     SETUP           /CHECK FOR PROPER ADDRESS
6217  7640        SZA CLA
6220  5204        JMP     SETUP+4
6221  6201        CDF     00              /RESET PROGRAM FIELD
6222  7604        LAS
6223  0376        AND     (100            /GET SW 5
6224  7640        SZA CLA                 /TRACING PROGRAM ?
6225  5232        JMP     ENTYP           /YES = TYPE ENTER
6226  7604        LAS
6227  0375        AND     (200            /GET SW 4
6230  7650        SNA CLA                 /STOP ON ENTER ?
6231  5600        JMP I   SETUP           /NO = RETURN
6232  4774  ENTYP,  JMS     TYP             /TYPE ENTER
6233  2740  ENTTXT, CRLF
6234  0000        0
6235  7604        LAS
6236  0375        AND     (200            /GET SW 4
6237  7640        SZA CLA                 /STOP ON ENTER ?
6240  4773        JMS     KEYCK           /YES = WAIT FOR KEYBOARD INPUT
6241  5600        JMP I   SETUP

6242  6201  BADNWS, CDF     00
6243  7402        HLT                     /PROGRAM MUST BE RE LOADED
6244  5243        JMP     .-1

6245  0000  ESTOP,  0
```

```
6246  0000    GAPT,   0                       /OUTBRK USING ADDRS
6247  4262            JMS     GET             /GET DATA
6250  0047            PAPT                    /DATA ADDRESS
6251  5646            JMP I   GAPT

6252  0000    GPC,    0                       /OUTBRK USING FPC
6253  4262            JMS     GET             /GET DATA
6254  0041            PFPC                    /ADDRESS OF DATA
6255  5652            JMP I   GPC

6256  0000    GOP,    0                       /OUTBRK USING OP ADDRESS
6257  4262            JMS     GET             /GET DATA
6260  0044            OPADR                   /DATA ADDRESS
6261  5656            JMP I   GOP

6262  0000    GET,    0                       /GET DATA
6263  7200            CLA
6264  1662            TAD I   GET             /GET ADDRESS REGISTER
6265  3103            DCA     T1              /SAVE
6266  2262            ISZ     GET             /INCREMENT RETURN
6267  1503            TAD I   T1              /GET FIELD BITS
6270  2103            ISZ     T1              /INC TO RIGHT HALF OF REGISTER
6271  0372            AND     (7              /3 BITS ONLY
6272  7106            CLL RTL                 /MOVE 3 BITS LEFT
6273  7004            RAL
6274  1371            TAD     (6201           /CREATE CDF INST
6275  3300            DCA     .+3
6276  1503            TAD I   T1              /GET ABSOLUTE ADDRESS
6277  3103            DCA     T1              /SAVE
6300  6201            CDF     00              /CHANGE TO CORRECT FIELD
6301  1503            TAD I   T1              /GET DATA
6302  6201            CDF     00              /RESTORE DATA FIELD
6303  5662            JMP I   GET             /RETURN
```

```
                        /CHECK SR05

6324  2000    TRSKP,    0
6325  7604              LAS
6326  0376              AND     (100          /GET SWITCH 5
6327  7650              SNA CLA               /IS SWITCH 5 SET ?
6330  5316              JMP     .+6           /NO - BYPASS TYPEOUT
6331  1704              TAD I   TRSKP         /GET TEXT ADDRESS
6332  3314              DCA     .+2
6333  4774'             JMS     TYP
6334  2000              0
6335  0000              0
6336  2304              ISZ     TRSKP
6337  5704              JMP I   TRSKP

                        /CHECK OP ADDRESS

6340  0000    CKOPAD,   0
6341  7200              CLA
6342  6566              RDOP                  /GET OP ADDR FROM FPP
6343  3121              DCA     SAVOP         /SAVE FOR TYPEOUT
6344  1045              TAD     OPADR+1       /GET PROGRAM OP ADDR
6345  7041              CIA
6346  1121              TAD     SAVOP         /COMPARE THE ADDRESSES
6347  7650              SNA CLA               /FPP OP ADDR CORRECT ?
6350  5740              JMP I   CKOPAD        /YES
6351  1770'             TAD     STEP          /NO - GET PC
6352  4767'             JMS     ERR           /GO TO ERROR ROUTINE
6353  0003              ERROR   0003          /OP ADDR ERROR CODE
6354  5740              JMP I   CKOPAD        /RETURN
```

```
6335  0000  SETST5, 0                    /SET PROG STATE 5
6336  7200        CLA
6337  1105        TAD     CSTATE
6340  0366        AND     (377
6341  1365        TAD     (2400
6342  3105        DCA     CSTATE
6343  5735        JMP I   SETST5
6344  0000  SETST3, 0                    /SET PROG STATE 3
6345  7200        CLA
6346  1105        TAD     CSTATE
6347  0366        AND     (377
6350  1364        TAD     (1400
6351  3105        DCA     CSTATE
6352  5744        JMP I   SETST3

6353  0000  SETST6, 0                    /SET PROG STATE 6
6354  7200        CLA
6355  1105        TAD     CSTATE
6356  0366        AND     (377
6357  1363        TAD     (3000
6360  3105        DCA     CSTATE
6361  5753        JMP I   SETST6


6363  3000
6364  1400
6365  2400
6366  0377
6367  7317
6370  5600
6371  6201
6372  0007
6373  0250
6374  7313
6375  0200
6376  0100
6377  1377
      6400        PAGE
```

```
                        /NORMALIZE THE O REGISTER

6400  0000    XNORM,  0
6401  7200            CLA
6402  1022            TAD     OLSW            /GET O REG LSW
6403  7640            SZA CLA                 /IS IT ZERO
6404  5212            JMP     .+6             /NO-NORMALIZE IT
6405  1021            TAD     OMSW            /GET O REG MSW
6406  7104            CLL RAL                 /REMOVE BITS 0 AND 1
6407  7104            CLL RAL
6410  7650            SNA CLA                 /IS O MSW ZERO
6411  5600            JMP I   XNORM           /YES-RETURN
6412  1021    XNCK,   TAD     OMSW            /GET OMSW
6413  7104            CLL RAL                 /SHIFT LEFT
6414  7530            SPA SZL                 /ARE BITS 0 AND 1 BOTH ZERO
6415  7410            SKP                     /NO
6416  5224            JMP     NXSHFT          /YES-NOT NORMALIZED
6417  7430            SZL                     /IS BIT 0=1
6420  7410            SKP                     /YES-CHECK BIT 1
6421  5244            JMP     XNEND           /NO-NO IS NORMALIZED
6422  7500            SMA                     /IS BIT 1=1
6423  5244            JMP     XNEND           /NO-NO IS NORMALIZED
6424  3021    NXSHFT, DCA     OMSW            /YES-NORMALIZE NUMBER
6425  1022            TAD     OLSW            /GET O REG LSW
6426  7104            CLL RAL
6427  3022            DCA     OLSW
6430  7430            SZL                     /SHIFT OUT?
6431  2021            ISZ     OMSW            /YES-INSERT INTO MSW
6432  1023            TAD     OEXT            /GET EXT
6433  0377            AND     (7400           /SAVE 4 BITS ONLY
6434  7104            RAL CLL                 /SHIFT LEFT
6435  3023            DCA     OEXT
6436  7430            SZL                     /SHIFT OUT ?
6437  2022            ISZ     OLSW            /YES - INSERT INTO LSW
6440  1071            TAD     SHREG
6441  1376            TAD     (-1             /DECREMENT SHIFT REG
6442  3071            DCA     SHREG
6443  5201            JMP     XNORM+1         /CHECK NO, AGAIN
6444  7010    XNEND,  RAR                     /RESTORE MSW
6445  3021            DCA     OMSW
6446  5600            JMP I   XNORM
```

```
6447  0000  XPUTX,  0                    /PUT DATA IN INDEX REG
6450  3104          DCA     T2           /SAVE DATA
6451  1051          TAD     PIR          /GET INSTRUCTION
6452  0375          AND     (7           /EXTRACT BITS 9-11
6453  1057          TAD     PXP          /ADD X0 ADDR
6454  3103          DCA     T1           /SAVE ADDR
6455  1104          TAD     T2           /GET DATA
6456  3503          DCA I   T1           /STORE DATA
6457  5647          JMP I   XPUTX        /RETURN

6460  0000  INDEX,  0                    /FIND INDEX REG SPECIFIED
6461  7300          CLA CLL              /BY BITS 6-8 OF FPP INSTRUCTION
6462  1051          TAD     PIR          /GET INSTRUCTION
6463  0374          AND     (70          /EXTRACT X BITS
6464  7010          RAR                  /RIGHT JUSTIFY
6465  7012          RTR
6466  1373          TAD     (X0          /ADD TO ADDRESS OF X0
6467  3103          DCA     T1           /SAVE X ADDR
6470  5660          JMP I   INDEX        /RETURN

6471  0000  XPLUS1, 0                    /INCREMENT X
6472  4260          JMS     INDEX        /GET ADDR OF X IN T1
6473  2503          ISZ I   T1           /INC X
6474  7000          NOP
6475  5671          JMP I   XPLUS1       /RETURN

6476  0000  XGETX,  0                    /GET INDEX REG
6477  4260          JMS     INDEX        /FIND ADDR OF X
6500  1503          TAD I   T1           /GET DATA IN X
6501  5676          JMP I   XGETX        /RETURN

6502  0000  SHIFTB, 0                    /SHIFT B UNTIL SHFT CNT=0
6503  7300          CLA CLL
6504  1071          TAD     SHREG        /GET SHIFT REG
6505  7650          SNA CLA              /SHFT CNT=0?
6506  5702          JMP I   SHIFTB       /YES-DO NOT SHIFT
6507  7100          CLL
6510  1027          TAD     BMSW         /NO-SHIFT BMSW
6511  7510          SPA
6512  7120          STL
6513  7010          RAR
6514  3027          DCA     BMSW
6515  1030          TAD     BLSW         /SHIFT B LSW
6516  7010          RAR
6517  3030          DCA     BLSW
6520  1031          TAD     BEXT         /SHIFT B EXTENTION
6521  7010          RAR
6522  0377          AND     (7400        /USE ONLY 4 BITS
6523  3031          DCA     BEXT
6524  2071          ISZ     SHREG        /SHIFT CNT=0?
6525  5307          JMP     SHIFTB+5     /NO-SHIFT AGAIN
6526  5702          JMP I   SHIFTB       /YES-RETURN
```

```
6527  0000  APTDEC, 0                        /DECREMENT PROGRAM
6530  7340           STA CLL                 /APT POINTER
6531  1050           TAD     PAPT+1          /12 BIT ABSOLUTE ADDR-1
6532  3050           DCA     PAPT+1
6533  7420           SNL                     /OVERFLOW?
6534  7040           CMA                     /YES
6535  1047           TAD     PAPT            /FIELD BITS
6536  3047           DCA     PAPT
6537  7300           CLA CLL
6540  5727           JMP I   APTDEC          /RETURN

6541  0000  CKSUB,  0                        /SKIP IF INSTRUCTION IS
6542  3352           DCA     CKST            /NOT A SUBTRACT
6543  1051           TAD     PIR             /GET INSTRUCTION
6544  7006           RTL
6545  7420           SNL                     /SUBTRACT?
6546  2341           ISZ     CKSUB           /NO-INCREMENT RETURN
6547  7300           CLA CLL
6550  1352           TAD     CKST            /RESTORE AC
6551  5741           JMP I   CKSUB           /RETURN

6552  0000  CKST,   0



6573  0200
6574  0070
6575  0007
6576  7777
6577  7400
      6600           PAGE
```

```
                        /CHECK DATA THE FPP STORED IN MEMORY

6600  0000  CMEM1,  2                       /COMPARE MEMORY EXPONENT
6601  7300          CLA CLL
6602  1377          TAD     (-1             /GET COUNT
6603  3103          DCA     T1              /SAVE COUNT
6604  1103          TAD     T1
6605  7041          CIA
6606  0376          AND     (2              /1 OR 2 WORD ERROR CODE
6607  7106          CLL RTL                 /MOVE COUNT TO BITS 6-8
6610  7004          RAL
6611  1376          TAD     (2              /MEMORY ERROR CODE
6612  3243          DCA     MCODE           /STORE FOR ERROR
6613  1103          TAD     T1              /GET WORD COUNT
6614  7041          CIA
6615  1375          TAD     (EMEM-2         /ADD COUNT TO BASE ADDR
6616  3774'         DCA     SAVMEM          /SAVE FOR ERROR REPORT
6617  1774'         TAD     SAVMEM
6620  3010          DCA     10
6621  7040          CMA
6622  1045          TAD     OPADR+1         /GET OP ADDRESS
6623  3011          DCA     11
6624  1044          TAD     OPADR           /GET FIELD BITS
6625  0373          AND     (7              /DELETE EXTRA BITS
6626  7104          CLL RAL
6627  7006          RTL                     /MOVE 3 BITS LEFT
6630  1372          TAD     (CDF            /CREATE CDF INSTRUCTION
6631  3232          DCA     .+1             /MODIFY PROGRAM
6632  6201  CMEM,   CDF     00              /USER FIELD
6633  1411          TAD I   11              /GET FPP DATA
6634  6201          CDF     00              /PROGRAM FIELD
6635  7041          CIA
6636  1410          TAD I   10              /PROGRAM DATA
6637  7650          SNA CLA                 /IS DATA OK ?
6640  5245          JMP     .+5             /YES
6641  1200          TAD     CMEM1           /NO
6642  4771'         JMS     ERR             /GO TO ERROR ROUTINE
6643  0002  MCODE,  ERROR   0002            /MEMORY ERROR CODE
6644  5247          JMP     .+3             /RETURN
6645  2103          ISZ     T1              /YES--END OF COMPARE?
6646  5232          JMP     CMEM            /NO--COMPARE NEXT WORD
6647  5600          JMP I   CMEM1           /RETURN

6650  0000  CMEM2,  0                       /COMPARE MEMORY FRACTION
6651  7300          CLA CLL
6652  1250          TAD     CMEM2           /GET RETURN
6653  3200          DCA     CMEM1           /SET RETURN IN EXP COMPARE
6654  7040          CMA                     /AC=-1
6655  5202          JMP     CMEM1+2         /GO TO EXPONENT COMPARE
```

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

6656  0000    PINDEX, 0                       /SET PROGRAM INDEX REGS
6657  7300            CLA CLL
6660  1065            TAD     X0ADR           /USER INDEX POINTER FIELD BITS
6661  7104            CLL RAL
6662  7006            RTL
6663  1372            TAD     (CDF            /CREATE CDF INST
6664  3274            DCA     PINCF           /MODIFY PROGRAM
6665  7240            STA
6666  1066            TAD     X0ADR+1         /USER INDEX POINTER
6667  3010            DCA     10
6670  1370            TAD     (X0-1           /ADDR OF PROG INDEX REGS
6671  3011            DCA     11
6672  1367            TAD     (-10            /COUNT
6673  3103            DCA     T1
6674  6201    PINCF,  CDF     00              /USER FIELD
6675  1410            TAD I   10
6676  6201            CDF     00              /PROGRAM FIELD
6677  3411            DCA I   11
6700  2103            ISZ     T1              /FINISHED?
6701  5274            JMP     PINCF           /NO
6702  5656            JMP I   PINDEX          /RETURN

                        /CLEAR THE A REGISTER

6703  0000    CLRAX,  0
6704  3024            DCA     AMSW
6705  3025            DCA     ALSW
6706  3026            DCA     AEXT
6707  5703            JMP I   CLRAX

                        /CLEAR THE B REGISTER

6710  0000    CLRBX,  0
6711  3030            DCA     BLSW
6712  3027            DCA     BMSW
6713  3031            DCA     BEXT
6714  5710            JMP I   CLRBX

                        /CLEAR THE O REGISTER

6715  0000    CLROX,  0
6716  3021            DCA     OMSW
6717  3022            DCA     OLSW
6720  3023            DCA     OEXT
6721  5715            JMP I   CLROX
```

```
                        /INCREMENT THE O REGISTER

6722  0000  OPLUS1, 0                       /INCOR=INC THE O REG
6723  7300           CLA CLL
6724  3111           DCA     OVFL           /CLEAR OVERFLOW
6725  2022           ISZ     OLSW           /INC LSW
6726  5722           JMP I   OPLUS1         /RETURN
6727  2021           ISZ     OMSW           /INC OMSW
6730  7000           NOP
6731  5722           JMP I   OPLUS1         /RETURN

                        /INCREMENT THE PROGRAM STATE GENERATOR

6732  0000  STINC, 0                       /INCST=INCREMENT THE
6733  7300           CLA CLL                /PROGRAM MAJOR STATE GEN,
6734  1105           TAD     CSTATE
6735  1366           TAD     (400
6736  3105           DCA     CSTATE
6737  5732           JMP I   STINC          /RETURN


                        /DOES INSTRUCTION STORE THE ANSWER IN MEMORY ?

6740  0000  MEMINS, 0                       /SKIP IF ANSWER IS NOT TO GO TO MEMORY
6741  7300           CLA CLL
6742  1051           TAD     PIR            /GET INSTRUCTION
6743  0365           AND     (7000          /EXTRACT OP CODE
6744  1364           TAD     (-5000
6745  7450           SNA                    /FADDM?
6746  5352           JMP     .+4            /YES-RETURN
6747  1363           TAD     (-2000
6750  7640           SZA CLA                /FMULM?
6751  2340           ISZ     MEMINS         /NO-INCREMENT RETURN
6752  7300           CLA CLL
6753  5740           JMP I   MEMINS         /RETURN


6763  6000
6764  3000
6765  7000
6766  0400
6767  7770
6770  0177
6771  7317
6772  6201
6773  0007
6774  0706
6775  0003
6776  0002
6777  7777
      7000           PAGE
```

```
                        /DECREMENT THE OP ADDRESS

7000  0000    OPDEC,    0                       /DECREMENT OP ADDRESS
7001  7340              CLL STA                 /AC=-1
7002  1045              TAD     OPADR+1         /12 BIT ABSOLUTE ADDR
7003  3045              DCA     OPADR+1
7004  7420              SNL
7005  7040              CMA
7006  1044              TAD     OPADR           /FIELD BITS
7007  3044              DCA     OPADR
7010  7300              CLA CLL
7011  5600              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

7012  0000  ADDAB,  0                       /APB0=A PLUST B TO O
7013  7300         CLA CLL
7014  1026         TAD     AEXT            /ADD EXTENSIONS
7015  1031         TAD     BEXT
7016  1076         TAD     CARYIN          /ADD CARRY INSERT
7017  3023         DCA     OEXT
7020  7004         RAL                     /INSERT CARRY OUT OF EXT
7021  1025         TAD     ALSW            /ADD LSW
7022  1030         TAD     BLSW
7023  3022         DCA     OLSW
7024  3076         DCA     CARYIN          /RESET CARRIN IN
7025  7004         RAL                     /INSERT CARRY OUT OF SLW
7026  1024         TAD     AMSW            /ADD MSW
7027  1027         TAD     BMSW
7030  3021         DCA     OMSW
7031  7010         RAR                     /CARRY OUT TO SIGN BIT
7032  3077         DCA     CAROUT          /SAVE CARRY OUT
7033  4522         JMS I   [SPECFL         /CHECK FOR 4000 0000 IN O
7034  5254         JMP     SETFL           /O = 4000 0000 SPECIAL OVERFLOW
7035  1024         TAD     AMSW            /CHECK OVERFLOW
7036  7104         RAL CLL                 /SIGN OF A IN LINK
7037  7200         CLA
7040  1027         TAD     BMSW            /GET B SIGN
7041  7530         SPA SZL                 /ARE BOTH SIGNS POS
7042  7060         CMA CML                 /NO-COMPLIMENT
7043  7730         SPA SZL CLA             /ARE BOTH SIGNS NEG
7044  5255         JMP     SETFL+1         /NO-NO OVERFLOW
7045  1024         TAD     AMSW            /GET SIGN OF OPERAND
7046  7104         RAL CLL                 /SAVE IN THE LINK
7047  7200         CLA
7050  1021         TAD     OMSW            /GET SIGN OF ANSWER
7051  7530         SPA SZL                 /ARE BOTH SIGNS POS?
7052  7060         CMA CML                 /COMPLIMENT
7053  7730         SPA SZL CLA             /ARE BOTH SIGNS NEG?
7054  7340  SETFL,  STA CLL                /NO-SET OVERFLOW
7055  3111         DCA     OVFL            /STORE OVERFLOW
7056  5612         JMP I   ADDAB           /RETURN
```

```
                        /SUBTRACT B FROM A AND STORE THE
                        /ANSWER IN THE 0 REGISTER

7057  2000    SUBAB,    0                       /AMB0=A MINUS B TO 0
7060  7300              CLA CLL                 /NEGATE B AND ADD
7061  1312              TAD     SUBSW           /GET PASS SWITCH
7062  7040              CMA                     /COMPLIMENT
7063  3312              DCA     SUBSW
7064  1031              TAD     BEXT            /GET EXTENSTION
7065  7040              CMA                     /NEGATE EXT
7066  1377              TAD     (400            /4 BITS ONLY
7067  0376              AND     (7400           /KEEP IT 4 BITS
7070  3031              DCA     BEXT
7071  1030              TAD     BLSW            /GET LSW
7072  7040              CMA                     /COMPLIMENT
7073  7430              SZL                     /CARRY OUT OF EXT?
7074  7101              IAC CLL                 /YES=ADD IN CARRY
7075  3030              DCA     BLSW
7076  1027              TAD     BMSW            /GET MSW
7077  7040              CMA                     /COMPLIMENT
7100  7430              SZL                     /CARRY OUT OF LSW
7101  7101              IAC CLL                 /YES=ADD IN CARRY
7102  3027              DCA     BMSW
7103  1312              TAD     SUBSW           /GET PASS SWITCH
7104  7650              SNA CLA                 /FIRST PASS?
7105  5310              JMP     .+3             /NO=SUBTRACT COMPLETE
7106  4576              APBO                    /ADD A AND MINUS B TO
7107  5260              JMP     SUBAB+1         /RESTORE B REG
7110  7300              CLA CLL
7111  5657              JMP I   SUBAB           /RETURN

7112  0000    SUBSW,    0


7113  0000    OEZ,      0                       /SKIP IF O=0
7114  7300              CLA CLL
7115  1021              TAD     OMSW
7116  7640              SZA CLA                 /O MSW=0?
7117  5713              JMP I   OEZ             /NO=DO NOT SKIP
7120  1022              TAD     OLSW            /YES=CHECK LSW
7121  7650              SNA CLA                 /OLSW=0?
7122  2313              ISZ     OEZ             /YES=INC RETURN
7123  5713              JMP I   OEZ

7124  0000    AEZ,      0                       /SKIP IF FAC=0
7125  7300              CLA CLL
7126  1036              TAD     ACMSW           /GET FAC MSW
7127  7640              SZA CLA                 /MSW=0?
7130  5724              JMP I   AEZ             /NO=RETURN
7131  1037              TAD     ACLSW           /GET FAC LSW
7132  7650              SNA CLA                 /LSW=0?
7133  2324              ISZ     AEZ             /YES=INC RETURN
7134  5724              JMP I   AEZ             /RETURN
```

```
7135  0000  ALZ,   0                    /SKIP IF FAC IS NEG
7136  7300         CLA CLL
7137  1036         TAD     ACMSW        /GET FAC MSW
7140  7710         SPA CLA              /IS FAC MINUS
7141  2335         ISZ     ALZ          /YES-INC RETURN
7142  5735         JMP I   ALZ          /RETURN


7143  0000  AGZ,   0                    /SKIP IF FAC IS GREATER THAN 0
7144  4335         JMS     ALZ          /IS FAC MINUS
7145  7410         SKP                  /NO-CHECK IF 0
7146  5743         JMP I   AGZ          /YES-DO NOT SKIP
7147  4324         JMS     AEZ          /IS FAC 0?
7150  2343         ISZ     AGZ          /NO-INC RETURN
7151  5743         JMP I   AGZ          /RETURN

7152  0000  SHIFTO, 0                    /SHIFT O LEFT UNTIL
7153  7300         CLA CLL              /SHIFT CNTR=0
7154  1071         TAD     SHREG
7155  7650         SNA CLA              /SHIFT CNT=0
7156  5752         JMP I   SHIFTO       /YES-NO SHIFT
7157  1023         TAD     OEXT         /GET EXTENTION
7160  7104         CLL RAL
7161  3023         DCA     OEXT
7162  1022         TAD     OLSW         /GET LSW
7163  7004         RAL
7164  3022         DCA     OLSW
7165  1021         TAD     OMSW         /GET MSW
7166  7004         RAL
7167  3021         DCA     OMSW
7170  2071         ISZ     SHREG        /SHIFT CNTR=0?
7171  5357         JMP     SHIFTO+5     /NO-SHIFT AGAIN
7172  5752         JMP I   SHIFTO       /YES RETURN


7176  7400
7177  0400
      7200         PAGE
```

```
                        /FPP-12 ARITHMETIC INSTRUCTIONS

7200  5273    INS0,   FLDA                    /FLDA
7201  3000            PFADD                   /F,P, ADD AND SUB
7202  3000            PFADD                   /F,P, ADD AND SUB
7203  3467            PFDIV                   /FDIV
7204  3257            FMULT                   /FMUL
7205  3000            PFADD                   /F,P, ADD AND SUB
7206  5247            FSTA                    /FSTA
7207  3257            FMULT                   /FMULM
7210  5273            FLDA                    /FLDA
7211  2650            DPADD                   /D,P, ADD AND SUB
7212  2650            DPADD                   /D,P, ADD AND SUB
7213  3467            PFDIV                   /FDIV
7214  3257            FMULT                   /FMULM
7215  2650            DPADD                   /D,P, ADD AND SUB
7216  5247            FSTA                    /FSTA
7217  3257            FMULT                   /FMULM
7220  5000    INS2I,  JEQ                     /TABLE OF INSTRUCTION
7221  5003            JGE                     /ADDRESSES FOR SPECIAL
7222  5006            JLE                     /FORMAT 2 OP CODE I
7223  5027            JTRUE                   /JUMP ALWAYS (JA)
7224  5011            JNE
7225  5014            JLT
7226  5017            JGT
7227  5022            JAL
7230  5200            SETX
7231  5157            SETB
7232  4642            JSA
7233  4600            JSR
7234  1653            SPEC21+3
7235  1653            SPEC21+3
7236  1653            SPEC21+3
7237  1653            SPEC21+3


7240  0400    CNTRLS, SA
7241  0502            RA
7242  0600            AS
7243  0506            RT
7244  0510            RF
7245  7352            TY
7246  0476            CM
7247  0436            EX
7250  0456            LDOP
7251  0663            SH
7252  0542            CL
7253  0452            EA
```

```
                        /SET RETURN FOR AUTO RESTART

7254  0000    SETRET, 0
7255  7300            CLA CLL
7256  1377            TAD     (STFPP+2        /GET REENTER ADDR OF NORMAL MODE
7257  3776'           DCA     RETURN          /SET REENTER ADDRESS
7260  5654            JMP I   SETRET          /RETURN

7261  0000    ASTER,  0                       /LINK TO WORD FROM FIELD 1
7262  4313            JMS     TYP
7263  2465            AST
7264  0000            0
7265  5661            JMP I   ASTER

7266  0000    TITLE,  0                       /TYPE TITLE ONCE ONLY
7267  7410            SKP                     /DELETED AFTER FIRST TYPEOUT
7270  5666            JMP I   TITLE           /THIS INST USED AFTER FIRST ENTRY
7271  1375            TAD     (7000           /GET A NOP
7272  3267            DCA     TITLE+1         /DELETE SKIP INSTRUCTION
7273  4313            JMS     TYP
7274  2672            MDEC                    /MAINDEC NUMBER
7275  0000            0
7276  4313            JMS     TYP
7277  2712            ECONO                   /ECO REVISION NUMBER
7300  0000            0
7301  5666            JMP I   TITLE

7302  0000    WORDL,  0
7303  4774'           JMS     WORD
7304  6212            CIF     10
7305  5702            JMP I   WORDL

7306  0000    KEYL1,  0                       /LINK TO KEYCK FROM FIELD 1
7307  4773'           JMS     KEYCK
7310  6211            CDF     10
7311  6212            CIF     10
7312  5706            JMP I   KEYL1

7313  0000    TYP,    0                       /LINK TO TYP10 IN FIELD 1
7314  6212            CIF     10
7315  5716            JMP I   .+1
7316  0025            TYPL

7317  0000    ERR,    0                       /LINK TO ERR10 IN FIELD 1
7320  6211            CDF     10
7321  6212            CIF     10
7322  4724            JMS I   .+2
7323  5717            JMP I   ERR
7324  1000            ERR10
```

```
7325  0000  STERR,  0                    /STATUS ERROR
7326  7300          CLA CLL
7327  6556          FPRST                /GET STATUS FROM FPP-12
7330  3772          DCA     STSAVE       /SAVE FOR ERROR TYPEOUT
7331  1325          TAD     STERR        /GET RETURN ADDRESS
7332  4317          JMS     ERR          /GO TO ERROR ROUTINE
7333  0004          ERROR   0004         /STATUS ERROR CODE
7334  5725          JMP I   STERR        /RETURN

7335  0000  TYPNCR,  0                    /TYPE WITH NO CARRIAGE RETURN
7336  7300          CLA CLL
7337  1335          TAD     TYPNCR       /GET RETURN ADDRESS
7340  3313          DCA     TYP          /SET TYP ENTRY FOR RETURN
7341  7240          STA                  /SET THE AC = 7777
7342  6211          CDF     10           /FIELD 1
7343  3771          DCA I   (CRSW        /SET NO RETURN SWITCH
7344  6201          CDF     00           /RESTORE FIELD
7345  5314          JMP     TYP+1        /GO TO TYPE ROUTINE

7346  0000  ASC,    0                    /LINK TO ASCII0 IN FIELD 1
7347  6212          CIF     10
7350  5751          JMP I   .+1
7351  0072          ASCL

7352  6212  TY,     CIF     10           /GO TO TYPE ROUTINE IN FIELD 1
7353  5754          JMP I   .+1
7354  0200          TY10
```

```
7371  0024
7372  0351
7373  0250
7374  0261
7375  7000
7376  0350
7377  1020
0122  5465
0123  0400
0124  0243
0125  6304
0126  6740
0127  6156
0130  6144
0131  7113
0132  7152
0133  6502
0134  7135
0135  7143
0136  7124
0137  6447
0140  6400
0141  6656
0142  6016
0143  6054
0144  6030
0145  6047
0146  6042
0147  6023
0150  6035
0151  6471
0152  6732
0153  5432
0154  6722
0155  5423
0156  5441
0157  6476
0160  6252
0161  6256
0162  6246
0163  5600
0164  6200
0165  7000
0166  6527
0167  6650
0170  6600
0171  6715
0172  6710
0173  6703
0174  5400
0175  5450
0176  7012
```

```
7177  7057
      0001          FIELD 1
```

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

0200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0300  11111111  11111111  11111111  11111111  11111111  11000111  11111111  11111111

0400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0500  11111111  11111111  11111111  11111111  11110000  00001111  11111111  11111111

0600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0700  11111110  00000000  00000000  00000000  00000000  00011111  11111111  11111111


1000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1100  11111111  11111111  11111111  11111111  11111111  11101111  11111111  11111111

1200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1300  11111111  11111111  11111111  11111111  11111111  11111111  00000111  11111111

1400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1500  11111111  11111111  11111111  11110000  00000000  00000000  00000001  11111111

1600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1700  11111111  11111111  00000000  00000000  00000000  00001111  11111111  11111111


2000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2100  11111111  11111111  11111111  11111111  11000000  00000000  01111111  11111111

2200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2300  11111111  11111110  00000000  00000000  00000000  00000000  00000000  00111111

2400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2500  11111111  11111111  11111111  11111111  11111100  00000000  00000011  11111111

2600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2700  11000000  00000000  00000000  00000000  00000000  00000000  00001111  11111111


3000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
3100  11111111  11111111  11111111  11111000  00000000  00000000  00000000  00011111

3200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
3300  11111111  11111111  11111111  11111111  10000000  00000000  00000000  00001111

3400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
3500  11111111  11111111  11111111  11111111  11111111  11111111  11100000  01111111

3600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
3700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  10111111
```

```
4000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4100  11111111  11111111  11000000  00000000  00000000  00000000  00000000  00111111

4200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4300  11111111  11111111  11111111  11111111  11111111  11111100  00000000  00111111

4400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4500  11111111  11111111  11111111  11111111  11111111  11111000  00000000  00111111

4600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111100
4700  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000111

5000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  00111111

5200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5300  11111111  11111110  00000000  00000000  00000000  00000000  00000000  00000011

5400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5500  11000000  00000000  00000000  00000000  00000000  00000000  00000000  00000011

5600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5700  11111111  11111111  11111111  11100000  00000000  00001111  11111111  11111111

6000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  00011111

6200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6300  11111111  11111111  11111111  11111111  11111111  11111111  11011111  11111111

6400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6500  11111111  11111111  11111111  11111111  11111111  11100000  00000000  00011111

6600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6700  11111111  11111111  11111111  11111111  11111111  11110000  00011111  11111111

7000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
7100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11100011

7200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
7300  11111111  11111111  11111111  11111111  11111111  11111000  00000000  01111111

7400
7500

7600
7700
```

```
      0000    *0
0000  7243    BEGIN


      0020    *20
0020  0000    T10,    0
0021  0000    T20,    0
0022  0000    ASCWD0, 0
0023  0000    OCTWD0, 0
0024  0000    CRSW,   0
```

```
                /LINK TYP IN FIELD 0 TO TYP10 IN FIELD 1

0025  7240  TYPL,   STA                     /AC = -1
0026  1577          TAD I   (TYP            /SUBTRACT 1 FROM RETURN ADDRESS
0027  6211          CDF     10              /DATA FIELD 1
0030  3010          DCA     10              /SAVE RETURN ADDRESS
0031  1176          TAD     (-6             /GET TEXT COUNT
0032  3051          DCA     TYPLX           /SET LOOP COUNTER
0033  1175          TAD     (TYPLD-1        /TEXT TABLE ADDRESS
0034  3011          DCA     11
0035  3411          DCA I   11              /CLEAR TEXT TABLE
0036  2051          ISZ     TYPLX
0037  5035          JMP     .-2
0040  1175          TAD     (TYPLD-1        /TEXT TABLE ADDRESS
0041  3011          DCA     11
0042  6201          CDF     00              /MOVE TEXT TABLE
0043  1410          TAD I   10              /FROM FIELD 0 TO FIELD 1
0044  6211          CDF     10
0045  7450          SNA                     /END OF TEXT TABLE ?
0046  5052          JMP     .+4             /YES
0047  3411          DCA I   11              /NO - SAVE TEXT ADDRESS
0050  5042          JMP     .-6             /GET NEXT ADDRESS
0051  0000  TYPLX,  0

0052  1174          TAD     (TYPLD          /GET DATA ADDRESS
0053  3773'         DCA     TYP10           /SET TYP RETURN
0054  1024          TAD     CRSW            /GET CARRAIGE RETURN SWITCH
0055  7650          SNA CLA                 /RETURN WANTED ?
0056  5772'         JMP     TYP10+1         /YES
0057  3024          DCA     CRSW            /RESET CARRAIGE RETURN SWITCH
0060  5771'         JMP     TYP10+5         /BYBASS CARRAIGE RETURN

0061  0000  TYPLD,  0
0062  0000          0
0063  0000          0
0064  0000          0
0065  0000          0
0066  0000          0
0067  6201          CDF     00
0070  6202          CIF     00
0071  5410          JMP I   10              /RETURN TO FIELD 0
```

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

 2072  7240   ASCL,    STA                     /AC = -1
 2073  1570            TAD I   (ASC            /RETURN ADDRESS -1
 2074  3010            DCA     10              /DATA ADDRESS
 2075  1410            TAD I   10              /GET FIRST WORD FRO FIELD 0
 2076  3112            DCA     ASCDAT          /SAVE
 2077  1410            TAD I   10              /GET SECOND WORD FROM FIELD 0
 2100  3106            DCA     ASCL2           /SAVE
 2101  1512            TAD I   ASCDAT          /SET DATA FIELD 1
 2102  3112            DCA     ASCDAT          /CONVERT TO ASCII
 2103  6211            CDF     10
 2104  4767'           JMS     ASC10
 2105  2112   ASCL1,   ASCDAT
 2106  0000   ASCL2,   0
 2107  6201            CDF     00
 2110  6202            CIF     00
 2111  5410            JMP I   10
 2112  0000   ASCDAT,  0


                        /LINK TO WORD IN FIELD 1

 2113  0000   WORD0,   0
 2114  6201            CDF     00              /DATA FIELD 0
 2115  6202            CIF     00              /INSTRUCTION FIELD 0
 2116  4566            JMS I   (WORDL          /JMS TO WORD LINK IN FIELD 0
 2117  1565            TAD I   (ASCWD          /GET THE ASCII WORD FROM FIELD 0
 2120  3022            DCA     ASCWD0          /SAVE IN FIELD 1
 2121  1564            TAD I   (OCTWD          /GET THE OCTAL WORD FROM FIELD 0
 2122  6211            CDF     10              /DATA FIELD 1
 2123  3023            DCA     OCTWD0          /RETURN
 2124  5513            JMP I   WORD0

                        /LINK TO KEYCK IN FIELD 0

 2125  0000   KEYCKL,  0
 2126  6201            CDF     00              /CHANGE TO FIELD 1
 2127  6202            CIF     00
 2130  4563            JMS I   (KEYL1          /JMS TO KEYCK LINK IN FIELD 0
 2131  5525            JMP I   KEYCKL          /RETURN

 2132  6201   KEYCK0,  CDF     00              /GO TO KEYCK+4 IN FIELD 0
 2133  6202            CIF     00
 2134  5562            JMP I   (KEYCK+4

       0200             *200
```

```
                /ROUTINE TO HANDLE ,TY COMMAND

0200  4113  TY10,   JMS      WORD0          /ALLOW TTY INPUT
0201  1377          TAD      (TWD-1         /ADDRESS OF ARGUMENT TABLE
0202  3010          DCA      10             /SAVE
0203  1376          TAD      (ROUT-1        /TEXT ADDRESS
0204  3011          DCA      11
0205  1375          TAD      (-25           /TEXT IS 25 WORDS LONG
0206  3020          DCA      T10
0207  1374          TAD      (4040          /FILL TEXT BUFFER WITH SPACES
0210  3411          DCA I    11
0211  2020          ISZ      T10            /FINISHED ?
0212  5207          JMP      .-3            /NO
0213  3411          DCA I    11             /ZERO LAST WORD OF TEXT BUFFER

0214  1410  TYA,    TAD I    10             /GET ARGUMENT FROM TABLE
0215  7450          SNA                     /END OF TABLE ?
0216  5330          JMP      TYALL          /YES - CHECK REQUEST FOR ALL
0217  7041          CIA                     /NO
0220  1022          TAD      ASCWD0         /COMPARE WITH INPUT ARGUMENT
0221  7650          SNA CLA                 /MATCH ?
0222  5225          JMP      .+3            /YES - TYPE REGISTER
0223  2010          ISZ      10             /NO - SKIP OVER ADDRESS
0224  5214          JMP      TYA            /GET NEXT ARGUMENT

0225  1410          TAD I    10             /GET REGISTER ADDRESS
0226  3020          DCA      T10            /SAVE
0227  1020          TAD      T10
0230  0373          AND      (7000          /EXTRACT WORD COUNT
0231  7106          RTL CLL                 /MOVE WORD COUNT TO
0232  7006          RTL                     /BITS 9-11
0233  7040          CMA
0234  3021          DCA      T20            /SAVE MINUS WORD COUNT
0235  1020          TAD      T10            /GET REGISTER ADDRESS
0236  0372          AND      (777           /EXTRACT ADDRESS BITS
0237  3243          DCA      TYT1           /SAVE ADDRESS
0240  1371          TAD      (ROUT          /ADDRESS OF TEXT BUFFER
0241  3244          DCA      TYT2           /SAVE

0242  4266  TYB,    JMS      GETREG         /GET DATA FROM REGISTER
0243  0021  TYT1,   OREG                    /REGISTER ADDRESS
0244  2466  TYT2,   ROUT                    /TEXT BUFFER ADDRESS
0245  2243          ISZ      TYT1           /INCREMENT REGISTER ADDRESS
0246  2244          ISZ      TYT2
0247  2244          ISZ      TYT2           /INCREMENT TEXT BUFFER ADDRESS +3
0250  2244          ISZ      TYT2
0251  2021          ISZ      T20            /END OF WORD COUNT
0252  5242          JMP      TYB            /NO - GET NEXT WORD
0253  3644          DCA I    TYT2           /ZERO LAST WORD OF TEXT BUFFER
```

```
0254  1370    TYC,    TAD     (,+3            /GET RETURN ADDRESS
0255  3767/           DCA     TYP10           /TO BYPASS CARRAIGE RETURN
0256  5766/           JMP     TYP10+5         /TYPE DATA
0257  2466            ROUT
0260  0000            0
0261  4767/           JMS     TYP10           /TYPE CARRAIGE RETURN
0262  2740            CRLF
0263  0000            0
0264  5305            JMP     TNEXT           /GET NEXT REGISTER

0265  0000    TYCONT, 0

0266  0000    GETREG, 0                       /GET REGISTER DATA
0267  1666            TAD I   GETREG          /GET REGISTER ADDRESS
0270  3020            DCA     T10             /SAVE
0271  6201            CDF     00              /DATA IS IN FIELD 0
0272  1420            TAD I   T10             /GET DATA
0273  6211            CDF     10              /RESTORE DATA FIELD
0274  3020            DCA     T10             /SAVE DATA
0275  2266            ISZ     GETREG          /INC RETURN ADDRESS
0276  1666            TAD I   GETREG          /GET ADDRESS OF TEXT STRING
0277  3302            DCA     .+3
0300  4765/           JMS     ASC10           /CONVERT DATA TO ASCII
0301  0020            T10                     /ADDRESS OF DATA
0302  2466            ROUT                    /ADDRESS OF TEXT STRING
0303  2266            ISZ     GETREG          /INC RETURN
0304  5666            JMP I   GETREG



0305  1265    TNEXT,  TAD     TYCONT          /GET CONTINUE SWITCH
0306  7640            SZA CLA                 /IS IT SET ?
0307  5313            JMP     .+4             /YES
0310  3265            DCA     TYCONT          /NO - RESET SWITCH
0311  5132            JMP     KEYCK0          /RETURN TO TTY MONITOR
0312  5201            JMP     TYI0+1          /TYPE NEXT ARGUMENT
0313  1413            TAD I   13              /GET NEXT REGISTER
0314  7450            SNA                     /END OF TABLE ?
0315  5310            JMP     .-5             /YES - EXIT
0316  3322            DCA     .+4             /NO - TYPE TITLE
0317  1364            TAD     (,+3
0320  3767/           DCA     TYP10
0321  5766/           JMP     TYP10+5         /BYPASS CARRAIGE RETURN
0322  2516            POEQ                    /TYPE REGISTER NAME
0323  0000            0
0324  1412            TAD I   12              /GET NEXT WORD FROM ARGUMENT TABLE
0325  3022            DCA     ASCWD0          /FAKE OUT TYI0 ROUTINE
0326  2012            ISZ     12              /SKIP OVER ADDRESS
0327  5201            JMP     TYI0+1          /TYPE REGISTER DATA
```

```
0330  1022    TYALL,  TAD     ASCWD0          /GET INPUT ARGUMENT
0331  1363            TAD     (-0114          /COMPARE WITH "AL"
0332  7640            SZA CLA                 /MATCH ?
0333  5762/           JMP     TYDATA          /NO - TYPE VARIABLE DATA
0334  7240            STA                     /YES
0335  3265            DCA     TYCONT          /SET THE CONTINUE SWITCH
0336  1377            TAD     (TWD-1          /GET ADDRESS OF ARGUMENT TABLE
0337  3012            DCA     12
0340  1361            TAD     (REGEQ-1        /ADDRESS OF REG NAME TABLE
0341  3013            DCA     13
0342  4767/           JMS     TYP10           /TYPE A CARRAIGE RETURN
0343  2740            CRLF
0344  0000            0
0345  5305            JMP     TNEXT           /TYPE CONTENTS OF ALL REGISTERS

0361  0577
0362  0400
0363  7664
0364  0322
0365  0620
0366  0672
0367  0665
0370  0257
0371  2466
0372  0777
0373  7000
0374  4040
0375  7753
0376  2465
0377  0456
      0400            PAGE
```

```
0400  1023  TYDATA, TAD   OCTWD0        /CHECK FOR FIELD BITS
0401  7510        SPA                   /IS NUMBER MINUS ?
0402  5777'       JMP   DOTER0          /YES - INPUT ERROR
0403  1376        TAD   (-10            /SUBTRACT 8
0404  7700        SMA CLA               /IS NUMBER G.T. 7 ?
0405  5777'       JMP   DOTER0          /YES - INPUT ERROR
0406  1023        TAD   OCTWD0          /NO - GET FIELD BITS
0407  7104        RAL CLL
0410  7006        RTL                   /MOVE TO BITS 6 - 8
0411  1375        TAD   (CDF            /CREATE CORRECT CDF INST.
0412  3226        DCA   TDFLD           /MODIFY PROGRAM
0413  4113        JMS   WORD0           /GET ADDRESS
0414  1023        TAD   OCTWD0          /OCTAL ADDRESS
0415  3256        DCA   TYADD           /STORE ADDRESS IN POINTER
0416  4113        JMS   WORD0           /GET WORD COUNT
0417  1023        TAD   OCTWD0
0420  7040        CMA
0421  3255        DCA   TYCNT           /STORE COUNT IN COUNTER

0422  2255  TYD,   ISZ   TYCNT          /END OF WORD COUNT ?
0423  5226        JMP   TDFLD           /NO GET NEXT WORD
0424  5132        JMP   KEYCK0
0425  5774'       JMP   TYI0+1

0426  6201  TDFLD, CDF   00             /TYPEOUT DATA FIELD
0427  1656        TAD I TYADD           /GET DATA WORD
0430  6211        CDF   10
0431  3020        DCA   T10             /STORE DTAT WORD
0432  4773'       JMS   ASC10           /CONVERT ADDRESS TO ASCII
0433  0456        TYADD
0434  2466        ROUT
0435  4773'       JMS   ASC10           /CONVERT DATA TO ASCII
0436  0020        T10
0437  2471        ROUT+3
0440  3772'       DCA   ROUT+5          /TERMINATE TEXT STRING
0441  4771'       JMS   TYP10           /TYPE ADDR AND DATA
0442  2466        ROUT
0443  2740        CRLF
0444  0000        0
0445  2256        ISZ   TYADD           /INCREMENT DATA ADDRESS
0446  5222        JMP   TYD             /RETURN TO CHECK FOR END
0447  1226        TAD   TDFLD           /OVERFLOW TO NEXT FIELD
0450  1370        TAD   (10             /INCREMENT DATA FIELD BITS
0451  0367        AND   (70
0452  1375        TAD   (CDF
0453  3226        DCA   TDFLD           /STORE NEW CDF INSTRUCTION
0454  5222        JMP   TYD             /RETURN TO CHECK FOR END

0455  0000  TYCNT, 0
0456  0000  TYADD, 0
```

```
                /,TY COMMAND ARGUMENT TABLE

2457  2017      TWD,    2017            /P0 - PROGRAM 0 REG
2460  2021              OREG    2000
2461  0617              0617            /F0 - FPP 0 REGISTER
2462  1002              SAVOM   1000
2463  0001              0001            /A
2464  2024              AREG    2000
2465  0002              0002            /B
2466  2027              BREG    2000
2467  1521              1521            /MQ
2470  2032              MQREG   2000
2471  0103              0103            /AC
2472  2035              PFAC    2000
2473  2003              2003            /PC - PROGRAM FPC
2474  1041              PFPC    1000
2475  1122              1122            /IR
2476  0051              PIR     0000
2477  1720              1720            /OP - OP ADDRESS
2500  1044              OPADR   1000
2501  0104              0104            /AD - ADRS
2502  1047              PAPT    1000
2503  2324              2324            /ST - STATUS
2504  0053              PSTAT   0000
2505  2023              2023            /PS - PROGRAM STATES
2506  0105              CSTATE  0000
2507  0623              0623            /FS - READ FPP STATES
2510  0001              MT2
2511  0120              0120            /AP - APT
2512  7210              APT     7000
2513  0030              0030            /X - INDEX REGISTERS
2514  7200              X0      7000
2515  2310              2310            /SH - SHIFT REGISTER
2516  0071              SHREG
2517  0000              0000


2567  0070
2570  0010
2571  0665
2572  2473
2573  0620
2574  0201
2575  6201
2576  7770
2577  1113
      0600              PAGE
```

```
                /REGISTER NAME TABLE

0600  2516    REGEQ,  PCEQ
0601  2522            FCEQ
0602  2526            AEQ
0603  2532            BEQ
0604  2536            MQEQ
0605  2542            ACEQ
0606  2611            PCEQ
0607  2622            PIREQ
0610  2546            OPEQ
0611  2616            ADEQ
0612  2553            STATEQ
0613  2560            PSEQ
0614  2602            RSEQ
0615  2570            APEQ
0616  2574            XEQ
0617  0000            0
```

```
                    /CONVERT OCTAL WORD TO 6 BIT ASCII

0620  0000    ASC10,  0
0621  1620            TAD I   ASC10           /GET ADDRESS OF DATA WORD
0622  3262            DCA     ASC4            /SAVE
0623  2220            ISZ     ASC10           /INCREMENT RETURN
0624  1620            TAD I   ASC10           /GET TEXT ADDRESS
0625  6211            CDF     10              /DATA FIELD 1
0626  3261            DCA     ASC3            /SAVE TEXT ADDRESS
0627  2220            ISZ     ASC10           /INCREMENT RETURN
0630  1264            TAD     ASC77           /GET MASK
0631  7040            CMA                     /LEFT HALF
0632  0662            AND I   ASC4            /EXTRACT LEFT HALF OF DATA
0633  7112            CLL RTR                 /MOVE TO RIGHT HALF
0634  7012            RTR
0635  7012            RTR
0636  4244            JMS     ASCB            /CONVERT LEFT HALF
0637  2261            ISZ     ASC3            /INCREMENT TEXT ADDRESS
0640  1264            TAD     ASC77           /GET MASK
0641  0662            AND I   ASC4            /EXTRACT RIGHT HALF OF DATA WORD
0642  4244            JMS     ASCB            /CONVERT RIGHT HALF
0643  5620            JMP I   ASC10           /EXIT

0644  0000    ASCB,   0                       /CONVERT 2 OCTAL DIGITS
0645  3263            DCA     ASC5            /SAVE DATA
0646  1263            TAD     ASC5            /RESTORE DATA
0647  7006            RTL                     /MOVE DATA 1 DIGIT LEFT
0650  7004            RAL
0651  0257            AND     ASC1            /DELETE RIGHT DIGIT
0652  1263            TAD     ASC5            /GET CORRECT RIGHT DIGIT
0653  0257            AND     ASC1            /SAVE ONLY 2 CORRECT DIGITS
0654  1260            TAD     ASC2            /INSERT ASCII MODIFIER
0655  3661            DCA I   ASC3            /STORE CONVERTED DATA
0656  5644            JMP I   ASCB            /RETURN

0657  0707    ASC1,   0707
0660  6060    ASC2,   6060
0661  0000    ASC3,   0
0662  0000    ASC4,   0
0663  0000    ASC5,   0
0664  0077    ASC77,  77
```

```
                        /TYPE 6 BIT ASCII TEXT STRINGS

0665  0000  TYP10,  0
0666  1335          TAD     T215            /GET CARRAIGE RETURN
0667  4324          JMS     TTY             /TYPE IT
0670  1336          TAD     T212            /GET LINE FEED
0671  4324          JMS     TTY             /TYPE IT
0672  1665          TAD I   TYP10           /GET TEXT ADDRESS
0673  6211          CDF     10              /DATA FIELD 1
0674  2265          ISZ     TYP10           /INCREMENT RETURN
0675  7450          SNA                     /END OF TEXT TABLE ?
0676  5665          JMP I   TYP10           /YES - RETURN
0677  3323          DCA     TMES            /NO - SAVE ADDRESS
0700  1723          TAD I   TMES            /GET TEXT WORD
0701  7112          CLL RTR                 /MOVE LEFT CHARACTER TO RIGHT HALF
0702  7012          RTR
0703  7012          RTR
0704  4311          JMS     TYPA            /CONVERT AND TYPE LEFT CHARACTER
0705  1723          TAD I   TMES            /GET TEXT WORD
0706  2323          ISZ     TMES            /INCREMENT TEXT ADDRESS
0707  4311          JMS     TYPA            /CONVERT AND TYPE RIGHT HALF
0710  5300          JMP     .-10            /DO IT AGAIN

0711  0000  TYPA,   0
0712  0342          AND     T77             /EXTRACT RIHGT HALF
0713  7450          SNA                     /END OF TEXT ?
0714  5272          JMP     TYP10+5         /YES - GET NEXT ADDRESS
0715  1337          TAD     TM40            /SUBTRACT 40
0716  7510          SPA                     /LESS THAN 40 ?
0717  1340          TAD     T100            /YES - 300 SERIES CHARACTER - ADD 100
0720  1341          TAD     T240            /ADD ORIGINAL 40 + 200
0721  4324          JMS     TTY             /TYPE CHARACTER
0722  5711          JMP I   TYPA            /RETURN
0723  0000  TMES,   0



0724  0000  TTY,    0
0725  6046          TLS                     /OUTPUT CHARACTER TO TTY
0726  6041          TSF                     /WAIT FOR FLAG
0727  5326          JMP     .-1
0730  6042          TCF                     /CLEAR FLAG
0731  7200          CLA
0732  6031          KSF                     /INPUT FROM KEYBOARD ?
0733  5724          JMP I   TTY             /NO - RETURN
0734  5132          JMP     KEYCK0          /YES - ANSWER KEYBOARD REQUEST
0735  0215  T215,   215
0736  0212  T212,   212
0737  7740  TM40,   -40
0740  0100  T100,   100
0741  0240  T240,   240
0742  0077  T77,    77

      1000          PAGE
```

```
1000  0000    ERR10,  0
1001  3311            DCA     ERRET           /SAVE RETURN ADDR
1002  7604            LAS
1003  7004            RAL
1004  7710            SPA CLA                 /IS SW 1 SET ?
1005  5303            JMP     NOREP           /YES = NO REPORT
1006  4777'           JMS     ASC10           /CONV RETURN ADDR TO ASCII
1007  1111            ERRET
1010  2335            EROUT+3
1011  4776'   DATERR, JMS     CODE            /GET ERROR CODE
1012  0375            AND     (7              /TYPE OF ERROR
1013  7104            RAL CLL
1014  1374            TAD     (DATTXT         /ADDRESS OF TEXT TABLE
1015  3312            DCA     ERRT1
1016  4316            JMS     GETT1           /GET ADDR FROM FIELD 1
1017  3276            DCA     ETXT1           /STORE IN TYPE ADDR STRING
1020  2312            ISZ     ERRT1
1021  4316            JMS     GETT1           /GET SECOND ADDR
1022  3300            DCA     ETXT2           /STORE IN TYPE ADDR STRING
1023  4776'           JMS     CODE            /GET ERROR CODE
1024  0375            AND     (7
1025  7004            RAL                     /MULTIPLY BY 2
1026  1373            TAD     (DATABL         /GET ADDR OF DATA TABLE
1027  3312            DCA     ERRT1
1030  4316            JMS     GETT1           /GET FIRST WORD
1031  7510            SPA
1032  4321            JMS     INDIR           /INDIRECT ADDRESS
1033  3236            DCA     DATC1           /SAVE DATA ADDRESS
1034  4772'           JMS     SPACE           /PUT SPACES IN TEXT
1035  4771'           JMS     GETREG          /CONVERT TO ASCII
1036  0000    DATC1,  0
1037  1727            DTXT1
1040  4776'           JMS     CODE            /GET ERROR CODE
1041  0370            AND     (70             /WORD COUNT
1042  7650            SNA CLA
1043  5252            JMP     WORD2           /FINISHED FIRST WORD
1044  2236            ISZ     DATC1           /INCREMENT DATA ADDRESS
1045  1236            TAD     DATC1
1046  3250            DCA     .+2
1047  4771'           JMS     GETREG          /CONVERT SECOND DATA WORD
1050  0000            0
1051  1732            DTXT1+3
```

```
1052  2312    WORD2,  ISZ     ERRT1
1053  4316            JMS     GETT1           /GET SECOND WORD
1054  7510            SPA
1055  4321            JMS     INDIR           /INDIRECT ADDRESS
1056  3260            DCA     DATC2
1057  4771'           JMS     GETREG          /CONVERT LSW
1060  0000    DATC2,  0
1061  1736            DTXT2
1062  4776'           JMS     CODE            /GET ERROR CODE
1063  0370            AND     (70             /WORD COUNT
1064  7650            SNA CLA
1065  5274            JMP     ERRTP           /TYPE DATA
1066  2260            ISZ     DATC2           /GET ADDR OF LSW
1067  1260            TAD     DATC2
1070  3272            DCA     .+2
1071  4771'           JMS     GETREG          /CONVERT LSW
1072  0000            0
1073  1741            DTXT2+3

1074  4767'   ERRTP,  JMS     TYP10           /TYPE DATA
1075  2332            EROUT
1076  2740    ETXT1,  CRLF
1077  1727            DTXT1
1100  2740    ETXT2,  CRLF
1101  1736            DTXT2
1102  0000            0
1103  7604    NOREP,  LAS
1104  7700            SMA CLA                 /SW0 SET ?
1105  4125            JMS     KEYCKL          /NO - WAIT FOR KEYBOARD INPUT
1106  6201            CDF     00
1107  6202            CIF     00              /GO BACK TO FIELD 0
1110  5600            JMP I   ERR10           /EXIT

1111  0000    ERRET,  0
1112  0000    ERRT1,  0

1113  6201    DOTER0, CDF     00              /LINK TO DOTERR IN FIELD 0
1114  6202            CIF     00
1115  5766            JMP I   (DOTERR


1116  0000    GETT1,  0                       /GET DATA WORD
1117  1712            TAD I   ERRT1
1120  5716            JMP I   GETT1
```

```
                /INDIRECT DATA - USED FOR MEMORY ERROR

1121  0000  INDIR,  0
1122  0365          AND      (777          /EXTRACT REGISTER ADDRESS
1123  3355          DCA      EDAT2         /SAVE ADDRESS
1124  6201          CDF      00            /GET REGISTER DATA
1125  1755          TAD I    EDAT2         /FROM FIELD 0
1126  2355          ISZ      EDAT2         /INCREMENT REGISTER ADDRESS
1127  0375          AND      (7            /EXTRACT FIELD BITS
1130  7104          RAL CLL                /MOVE TO BITS 6-8
1131  7006          RTL
1132  1364          TAD      (CDF          /MAKE CDF INSTRUCTION
1133  3336          DCA      .+3           /MODIFY PROGRAM
1134  1755          TAD I    EDAT2         /GET NEXT REGISTER DATA
1135  3355          DCA      EDAT2         /SAVE
1136  6211          CDF      10            /CHANGE TO DATA MEMORY FIELD
1137  1755          TAD I    EDAT2         /GET DATA
1140  2355          ISZ      EDAT2         /INCREMENT ADDRESS
1141  6201          CDF      00            /DATA FIELD 0
1142  3763          DCA I    (T1           /STORE DATA IN FIELD 0
1143  1336          TAD      .-5           /GET CDF INSTRUCTION
1144  3345          DCA      .+1           /MODIFY PROGRAM
1145  6211          CDF      10            /CHANGE TO DATA MEMORY FIELD
1146  1755          TAD I    EDAT2         /GET SECOND DATA WORD
1147  6201          CDF      00
1150  3762          DCA I    (T2           /STORE IN FIELD 0
1151  6211          CDF      10            /RESTORE DATA FIELD
1152  1363          TAD      (T1           /GET FIELD 0 DATA ADDRESS
1153  5721          JMP I    INDIR         /RETURN

1154  0000  EDAT1,  0
1155  0000  EDAT2,  0

1162  0104
1163  0103
1164  6201
1165  0777
1166  0432
1167  0665
1170  0070
1171  0266
1172  1200
1173  1551
1174  1532
1175  0007
1176  1216
1177  0620
      1200          PAGE
```

```
                /FILL TEXT BUFFER WITH SPACES

1200  0000  SPACE,  0
1201  1377         TAD      (DTXT1-1        /ADDRESS OF FIRST BUFFER
1202  3016         DCA      16              /SAVE
1203  1376         TAD      (DTXT2-1        /ADDRESS OF SECOND BUFFER
1204  3017         DCA      17              /SAVE
1205  1375         TAD      (-5             /WORD COUNT
1206  3020         DCA      T10             /SET COUNTER
1207  1374         TAD      (4040           /ASCII SPACES
1210  3416         DCA I    16              /STORE IN FIRST BUFFER
1211  1374         TAD      (4040           /ASCII SPACES
1212  3417         DCA I    17              /STORE IN SECOND BUFFER
1213  2020         ISZ      T10             /INCREMENT COUNTER
1214  5207         JMP      .-9             /STORE NEXT WORD
1215  5600         JMP I    SPACE           /RETURN

1216  0000  CODE,   0
1217  6201         CDF      00              /GET CODE FROM FIELD 0
1220  1773         TAD I    (ERR            /GET ADDRESS OF ERROR CODE
1221  3020         DCA      T10             /SAVE
1222  1420         TAD I    T10             /GET ERROR CODE
1223  6211         CDF      10              /RESTORE DATA FIELD
1224  5616         JMP I    CODE


1373  7317
1374  4040
1375  7773
1376  1735
1377  1726
      1400         PAGE
```

```
1400  0010    SETTAB, 0010                    /INIATE
1401  2004            TINIT
1402  1076            ENINIT

1403  7500            7500                    /FETCH
1404  2010            TFETCH
1405  1237            FETCH+1

1406  0004            0004                    /TRAPPED INSTRUCTIONS
1407  2016            TTRAP
1410  1623            TRAP1

1411  5200            5200                    /DEPOSIT
1412  2030            TDEP
1413  2053            DEP11

1414  7420            7420                    /FEXIT
1415  2013            TEXIT
1416  2430            EXIT0

1417  7440            7440                    /D, P, ADD - SUBTRACT
1420  2034            TDPAS
1421  2651            DPADD0

1422  7440            7440                    /FLOATING PT, ADD & SUBTRACT
1423  2047            TFADD
1424  3001            FADD0

1425  7440            7440                    /MULTIPLY
1426  2063            TMULT
1427  3260            FMULT+1

1430  7440            7440                    /DIVIDE
1431  2070            TDIV
1432  3470            PFDIV+1

1433  0004            0004                    /LDX
1434  2074            TLDX
1435  4011            LDX1

1436  0004            0004                    /ALN
1437  2076            TALN
1440  4037            ALN1

1441  0004            0004                    /ATX
1442  2100            TATX
1443  4266            ATX1
```

```
1444  0004          0004          /XTA
1445  2102          TXTA
1446  4435          XTA1

1447  0004          0004          /JXN
1450  2104          TJXN
1451  4474          JXN1

1452  0004          0004          /JSR
1453  2106          TJSR
1454  4601          JSR1

1455  0404          0404          /JSA
1456  2110          TJSA
1457  4643          JSA2

1460  0004          0004          /JAC
1461  2112          TJAC
1462  5051          JAC1

1463  0004          0004          /JMP TRUE
1464  2114          TTRUE
1465  5030          JTRUE1

1466  0004          0004          /JMP FALSE
1467  2126          TFALSE
1470  5046          JFALSE+1

1471  0004          0004          /CLA
1472  2140          TCLA
1473  5151          FCLA+1

1474  0004          0004          /STF
1475  2143          TSTF
1476  5064          FSTF+1

1477  0004          0004          /STD
1500  2147          TSTD
1501  5071          FSTD+1

1502  0004          0004          /FNEG
1503  2153          TNEG
1504  5077          NEG1

1505  0004          0004          /NORM
1506  2156          TNORM
1507  5115          NORM1

1510  0004          0004          /PAUSE
1511  2740          CRLF
1512  5127          FPAUSE+1
```

```
1513  0004          0004                    /SETB
1514  2203          TSETB
1515  5160          SETB1

1516  0004          0004                    /SETX
1517  2206          TSETX
1520  5201          SETX1

1521  0004          0004                    /ADDX
1522  2211          TADDX
1523  5213          ADDX1

1524  7440          7440                    /STA
1525  2214          TSTA
1526  5250          FSTA+1

1527  7440          7440                    /LDA
1530  2217          TLDA
1531  5274          FLDA+1


1532  1610  DATTXT, FSIS
1533  1622          PSIS
1534  1657          FOIS
1535  1667          POIS
1536  1674          FMIS
1537  1704          PMIS
1540  1711          FOPIS
1541  1722          POPIS
1542  1572          FSTIS
1543  1602          PSTIS
1544  1627          FAPIS
1545  1637          PAPIS
1546  2740          CRLF
1547  2740          CRLF
1550  2740          CRLF
```

```
1551  0001  DATABL, MT2                  /TIME STATE - CODE 0000
1552  0000        MT1
1553  0002        SAVOM                 /O REGISTER - CODE 0001
1554  0021        OMSW
1555  4044        OPADR   4000          /MEMORY - CODE 0002
1556  4705        MFLD    4000
1557  0121        SAVOP                 /OP ADDRESS - CODE 0003
1560  0045        OPADR+1
1561  0351        STSAVE                /STATUS - CODE 0004
1562  0053        PSTAT
1563  5501        SAVAPT                /ADDRS - CODE 0005
1564  0050        PAPT+1
1565  2740        CRLF
1566  2740        CRLF
1567  2740        CRLF
1570  2740        CRLF
1571  2740        CRLF
```

```
1572  4023    FSTIS,  TEXT    " STATUS * FPP "
1573  2401
1574  2425
1575  2340
1576  5240
1577  0620
1600  2040

1601  0000    0
1602  4052    PSTIS,  TEXT    " * PROG "
1603  4020
1604  2217
1605  0740

1606  0000    0
1607  0000            0
1610  2411    FSIS,   TEXT    "TIME STATE *  FPP "
1611  1505
1612  4023
1613  2401
1614  2405
1615  4052
1616  4040
1617  0620
1620  2040

1621  0000            0
1622  4052    PSIS,   TEXT    " * PROG "
1623  4020
1624  2217
1625  0740

1626  0000            0
1627  4001    FAPIS,  TEXT    " ADDRS * FPP "
1630  0404
1631  2223
1632  4052
1633  4006
1634  2020
1635  4000
1636  0000    0
1637  4052    PAPIS,  TEXT    " * PROG "
1640  4020
1641  2217
1642  0740

1643  0000    0
1644  0620    STRTER, TEXT    "FPP-12 DID NOT START"
1645  2055
1646  6162
1647  4004
1650  1104
1651  4016
```

```
1652  1724
1653  4023
1654  2401
1655  2224

1656  0000     0
1657  1740     FOIS,   TEXT    "0 REG *  FPP "
1660  2205
1661  0740
1662  5240
1663  4006
1664  2020
1665  4000
1666  0000             0
1667  4052     POIS,   TEXT    " * PROG "
1670  4020
1671  2217
1672  0740

1673  0000             0
1674  1505     FMIS,   TEXT    "MEMORY *  FPP "
1675  1517
1676  2231
1677  4052
1700  4040
1701  0620
1702  2040

1703  0000             0
1704  4052     PMIS,   TEXT    " * PROG "
1705  4020
1706  2217
1707  0740

1710  0000             0
1711  1720     FOPIS,  TEXT    "OP ADDR *  FPP "
1712  4001
1713  0404
1714  2240
1715  5240
1716  4006
1717  2020
1720  4000
1721  0000             0
1722  4052     POPIS,  TEXT    " * PROG "
1723  4020
1724  2217
1725  0740

1726  0000             0
1727  7777     DTXT1,  TEXT    "????  ???? "
1730  7777
1731  4040
1732  7777
1733  7777
```

```
1734  4000
1735  0000          0
1736  7777    DTXT2,  TEXT    "????  ???? "
1737  7777
1740  4040
1741  7777
1742  7777
1743  4000
1744  0000          0



      2000          PAGE

2000  0516    ENT,    TEXT    "ENTER  "
2001  2405
2002  2240
2003  4000
2004  1116    TINIT,  TEXT    "INIATE "
2005  1101
2006  2405
2007  4000
2010  0605    TFETCH, TEXT    "FETCH"
2011  2403
2012  1000
2013  0605    TEXIT,  TEXT    "FEXIT"
2014  3011
2015  2400
2016  2422    TTRAP,  TEXT    "TRAPPED INSTRUCTION"
2017  0120
2020  2005
2021  0440
2022  1116
2023  2324
2024  2225
2025  0324
2026  1117
2027  1600
2030  0405    TDEP,   TEXT    "DEPOSIT"
2031  2017
2032  2311
2033  2400
2034  0456    TDPAS,  TEXT    "D', P', ADD - SUBTRACT "
2035  4020
2036  5640
2037  0104
2040  0440
2041  5540
2042  2325
2043  0224
2044  2201
2045  0324
```

```
2046  4000
2047  3614    TFADD,  TEXT    "FLOATING PT, ADD - SUB "
2050  1701
2051  2411
2052  1607
2053  4020
2054  2456
2055  4001
2056  0404
2057  4055
2060  4023
2061  2502
2062  4000
2063  1525    TMULT,  TEXT    "MULTIPLY "
2064  1424
2065  1120
2066  1431
2067  4000
2070  0411    TDIV,   TEXT    "DIVIDE "
2071  2611
2072  0405
2073  4000
2074  1404    TLDX,   TEXT    "LDX"
2075  3000
2076  0114    TALN,   TEXT    "ALN"
2077  1600
2100  0124    TATX,   TEXT    "ATX"
2101  3000
2102  3024    TXTA,   TEXT    "XTA"
2103  0100
2104  1230    TJXN,   TEXT    "JXN"
2105  1600
2106  1223    TJSR,   TEXT    "JSR"
2107  2200
2110  1223    TJSA,   TEXT    "JSA"
2111  0100

2112  1201    TJAC,   TEXT    "JAC"
2113  0300

2114  1215    TTRUE,  TEXT    "JMP CONDITION TRUE "
2115  2040
2116  0317
2117  1604
2120  1124
2121  1117
2122  1640
2123  2422
2124  2505
2125  4000
2126  1215    TFALSE, TEXT    "JMP CONDITION FALSE"
2127  2040
2130  0317
2131  1604
2132  1124
```

```
2133  1117
2134  1640
2135  0601
2136  1423
2137  0500
2140  0603    TCLA,   TEXT    "FCLA "
2141  1401
2142  4000
2143  2324    TSTF,   TEXT    "START F"
2144  0122
2145  2440
2146  0600
2147  2324    TSTD,   TEXT    "START D"
2150  0122
2151  2440
2152  0400
2153  0616    TNEG,   TEXT    "FNEG "
2154  0507
2155  4000
2156  0616    TNORM,  TEXT    "FNORM"
2157  1722
2160  1500
2161  0620    TPAUSE, TEXT    "FPP PAUSE,,,,;ALT MODE TO CONTINUE "
2162  2040
2163  2001
2164  2523
2165  0556
2166  5656
2167  5656
2170  0114
2171  2440
2172  1517
2173  0405
2174  4024
2175  1740
2176  0317
2177  1624
2200  1116
2201  2505
2202  4000
2203  2305    TSETB,  TEXT    "SETB "
2204  2402
2205  4000
2206  2305    TSETX,  TEXT    "SETX "
2207  2430
2210  4000
2211  0104    TADDX,  TEXT    "ADDX "
2212  0430
2213  4000
2214  0623    TSTA,   TEXT    "FSTA "
2215  2401
2216  4000
2217  0614    TLDA,   TEXT    "FLDA "
2220  0401
2221  4000
```

```
2222  0516    ENDMUL, TEXT    "END MULTIPLY "
2223  0440
2224  1525
2225  1424
2226  1120
2227  1431
2230  4000
2231  0411    DIVZ,   TEXT    "DIVIDE BY ZERO "
2232  2611
2233  0405
2234  4002
2235  3140
2236  3205
2237  2217
2240  4000
2241  0411    DIVOV,  TEXT    "DIVIDE FIXED POINT OVERFLOW"
2242  2611
2243  0405
2244  4006
2245  1130
2246  0504
2247  4020
2250  1711
2251  1624
2252  4017
2253  2605
2254  2206
2255  1417
2256  2700
2257  0516    DIVOK,  TEXT    "END DIVIDE "
2260  0440
2261  0411
2262  2611
2263  0405
2264  4000
2265  0516    LDXOK,  TEXT    "END LDX"
2266  0440
2267  1404
2270  3000
2271  0516    ALNOK,  TEXT    "END ALN"
2272  0440
2273  0114
2274  1600
2275  0516    ATXOK,  TEXT    "END ATX"
2276  0440
2277  0124
2300  3000
2301  0516    XTAOK,  TEXT    "END XTA"
2302  0440
2303  3024
2304  0100
2305  0516    JXNOK,  TEXT    "END JXN"
```

```
2326  0440
2327  1230
2330  1600
2331  0516    JSROK,  TEXT    "END JSR"
2332  0440
2333  1223
2334  2200
2335  0516    JSAOK,  TEXT    "END JSA"
2336  0440
2337  1223
2340  0100
2341  0516    ADXOK,  TEXT    "END ADX"
2342  0440
2343  0104
2344  3000
2345  0516    STAOK,  TEXT    "END FSTA "
2346  0440
2347  0623
2350  2401
2351  4000


2352  0522    EROUT,  TEXT    "ERROR ???? * "
2353  2217
2354  2240
2355  7777
2356  7777
2357  4052
2360  4000

2361  0516    DEPOK,  TEXT    "END DEPOSIT"
2362  0440
2363  0405
2364  2017
2365  2311
2366  2400

2367  2516    VFER,   TEXT    "UNDERFLOW"
2370  0405
2371  2206
2372  1417
2373  2700

2374  0516    EOP,    TEXT    "END TRACE"
2375  0440
2376  2422
2377  0103
2400  0500

2401  0516    DPASOK, TEXT    "END DOUBLE PRECISION ADD-SUBTRACT"
2402  0440
2403  0417
2404  2502
2405  1405
2406  4020
```

```
2367  2205
2370  2311
2371  2311
2372  1716
2373  4001
2374  0404
2375  5523
2376  2502
2377  2422
2400  0103
2401  2400

2402  0456    DPFLOW, TEXT    "D',P', ADD-SUBTRACT OVERFLOW "
2403  2056
2404  4001
2405  0404
2406  5523
2407  2502
2410  2422
2411  0103
2412  2440
2413  1726
2414  0522
2415  0614
2416  1727
2417  4000

2420  0516    ENDFAD, TEXT    "END FLOATING PNT, ADD-SUBTRACT "
2421  0440
2422  0614
2423  1701
2424  2411
2425  1607
2426  4020
2427  1624
2430  5640
2431  0104
2432  0455
2433  2325
2434  0224
2435  2201
2436  0324
2437  4000

2440  0516    FSTAOK, TEXT    "END FSTA "
2441  0440
2442  0623
2443  2401
2444  4000

2445  0516    LDAOK,  TEXT    "END FLDA "
2446  0440
2447  0614
2450  0401
2451  4000
```

```
2452  2516    INOK,   TEXT    "END INIATE "
2453  2440
2454  1116
2455  1101
2456  2405
2457  4000

2460  0516    SFOK,   TEXT    "END FETCH"
2461  0440
2462  0605
2463  2403
2464  1000

2465  5200    AST,    TEXT    "*"


2466  7777    ROUT,   TEXT    "????  ????  ????  ????  ????  ????  ????  ???? "
2467  7777
2470  4040
2471  7777
2472  7777
2473  4040
2474  7777
2475  7777
2476  4040
2477  7777
2500  7777
2501  4040
2502  7777
2503  7777
2504  4040
2505  7777
2506  7777
2507  4040
2510  7777
2511  7777
2512  4040
2513  7777
2514  7777
2515  4000

2516  2017    POEQ,   TEXT    "PO =   "
2517  4075
2520  4040
2521  4000

2522  0617    FOEQ,   TEXT    "FO =   "
2523  4075
2524  4040
2525  4000

2526  0140    AEQ,    TEXT    "A =    "
2527  7540
2530  4040
```

```
2531  4000

2532  0240  BEQ,    TEXT   "B =    "
2533  7540
2534  4040
2535  4000

2536  1521  MQEQ,   TEXT   "MQ =   "
2537  4075
2540  4040
2541  4000

2542  0601  ACEQ,   TEXT   "FAC =  "
2543  0340
2544  7540
2545  4000

2546  1720  OPEQ,   TEXT   "OP ADDR ="
2547  4001
2550  0404
2551  2240
2552  7500

2553  2324  STATEQ, TEXT   "STATUS = "
2554  0124
2555  2523
2556  4075
2557  4000

2560  2022  RSEQ,   TEXT   "PROG STATE IS  "
2561  1707
2562  4023
2563  2401
2564  2405
2565  4011
2566  2340
2567  4000

2570  0120  APEQ,   TEXT   "APT =  "
2571  2440
2572  7540
2573  4000

2574  1116  XEQ,    TEXT   "INDEX REGS "
2575  0405
2576  3040
2577  2205
2600  0723
2601  4000

2602  0620  RSEQ,   TEXT   "FPP STATE IS "
2603  2040
2604  2324
2605  0124
2606  0540
```

```
2607  1123
2610  4000

2611  0620    PCEQ,   TEXT    "PC =    "
2612  0340
2613  7540
2614  4040
2615  4000

2616  0104    ADEQ,   TEXT    "ADRS = "
2617  2223
2620  4075
2621  4000

2622  1122    PIREQ,  TEXT    "IR  =  "
2623  4040
2624  7540
2625  4000

2626  0516    INSTEP, TEXT    "ENTER STEP MODE"
2627  2405
2630  2240
2631  2324
2632  0520
2633  4015
2634  1704
2635  0500

2636  0530    OUTSTP, TEXT    "EXIT STEP MODE "
2637  1124
2640  4023
2641  2405
2642  2040
2643  1517
2644  0405
2645  4000


2646  7777    ASADDR, TEXT    "????   "
2647  7777
2650  4040
2651  4000
2652  7777    ADDR,   TEXT    "???? "
2653  7777
2654  4000

2655  0522    APTERR, TEXT    "ERROR SAVING APT IN EXIT"
2656  2217
2657  2240
2660  2301
2661  2611
2662  1607
2663  4001
2664  2024
2665  4011
```

```
2666  1640
2667  0530
2670  1124

2671  0000    0
2672  5252    MDEC,   TEXT    "*** TRACE ***  MAINDEC 12-D0LC"
2673  5240
2674  2422
2675  0103
2676  0540
2677  5252
2700  5240
2701  4015
2702  0111
2703  1604
2704  0503
2705  4061
2706  6255
2707  0460
2710  1403

2711  0000    0
2712  0114    ECONO,  TEXT    "ALL ECO'S THROUGH #9 MUST BE INSTALLED"
2713  1440
2714  0503
2715  1747
2716  2340
2717  2410
2720  2217
2721  2507
2722  1040
2723  4371
2724  4015
2725  2523
2726  2440
2727  0205
2730  4011
2731  1623
2732  2401
2733  1414
2734  0504

2735  0000    0
2736  7777    QMK,    TEXT    "?? "
2737  4000

2740  0000    CRLF,   0
      3000            *,&7000+1000
```

```
                /FLOATING POINT EXERCISER
                /PDP-8 CODE STARTING LOCATION 200
                /
                /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
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
4001    5700            BASE
4002    1101            ESETX
4003    5016            XREG1
4004    0006            ESTRTD
4005    0031            EXTA!1                  /LOAD X1
4006    0023            EATX!3                  /RESET X3 FOR REENTRY
4007    1031            EJA
4010    5041            LINK+2
4011    0040            EFNOP
4012    0040            EFNOP
4013    1131    FPP1,   EJSR
4014    4476            FSET            /SETUP FOR TST
4015    1131            EJSR
4016    4734            RANGEN
4017    1121            EJSA
4020    5037            LINK
4021    1031            EJA
4022    4040            FPP2
4023    0005            ESTRTF
4024    0511    FPP1R,  EFLDA!511       /LOAD FROM TABLE1
4025    5775            TABLE1+3
4026    6531            EFSTA!531       /STORE IN TABLE 2
4027    6375            TABLE2+3
4030    1131            EJSR            /COMPARE TABLE1 WITH TABLE2
4031    4536            FCMPR
4032    2101            EJXN!100                /COMPLETED BLOCK
4033    4024            FPP1R
4034    0002            EFCLA           /YES
4035    1210            EFADD!210       /LOCK ONTO TST
4036    1041            EJNE            /IF OFFSET 10 IS NON-ZERO
4037    4013            FPP1
```

```
                        /
                        /EXECUTE IN DOUBLE PRECISION MODE
                        /MOVE RANDOM ARG TABLE3 TO TABLE4
                        /
4040  1131      FPP2,   EJSR
4041  4643              DPSET           /SETUP FOR TST
4042  1121              EJSA
4043  5037              LINK
4044  1031              EJA
4045  4063              FPP3
4046  0006              ESTRTD
4047  0511      FPP2R,  EFLDA!511       /LOAD RANDOM ARG
4050  6776              TABLE3-2        /FROM TABLE3
4051  6531              EFSTA!531       /STORE SAME
4052  7376              TABLE4-2        /IN TABLE4
4053  1131              EJSR            /JMP-SAVE RETURN
4054  4716              DPCMPR          /TO COMPARE SUB
4055  2101              EJXN!100                /COMPLETED BLOCK
4056  4047              FPP2R
4057  0002              EFCLA           /YES
4060  1210              EFADD!210       /LOCK ONTO TST
4061  1041              EJNE            /IF OFFSET 10 IS NON-ZERO
4062  4040              FPP2
                        /
                        /EXECUTE IN FLOATING POINT MODE
                        /ADD SUBTRACT TEST
                        /
4063  1131      FPP3,   EJSR            /JMP-SAVE RETURN
4064  4476              FSET            /SETUP SUB
4065  1121              EJSA
4066  5037              LINK
4067  1031              EJA
4070  4112              FPP4
4071  0005              ESTRTF
4072  0511      FPP3R,  EFLDA!511       /LOAD RANDOM ARG
4073  5775              TABLE1-3        /FROM TABLE1
4074  1411              EFADD!411       /A+A
4075  5775              TABLE1-3
4076  2411              EFSUB!411       /A+A-A
4077  5775              TABLE1-3
4100  6531              EFSTA!531       /STORE IN TABLE2
4101  6375              TABLE2-3
4102  1131              EJSR            /JMP-SAVE RETURN
4103  4560              FCMPAS          /COMPARE SUB
4104  2101              EJXN!100                /COMPLETED BLOCK
4105  4072              FPP3R
4106  0002              EFCLA           /YES
4107  1210              EFADD!210       /LOCK ONTO TST
4110  1041              EJNE            /IF OFFSET 10 IS NON-ZERO
4111  4063              FPP3
```

```
                        /
                        /EXECUTE IN DOUBLE PRECISION MODE
                        /ADD-SUBTRACT TEST
                        /
4112  1131      FPP4,   EJSR            /JMP-SAVE RETURN
4113  4643              DPSET           /SETUP SUB
4114  1121              EJSA
4115  5037              LINK
4116  1031              EJA
4117  4141              FPP5
4120  0006              ESTRTD
4121  0511      FPP4R,  EFLDA!511       /LOAD RANDOM ARG
4122  6776              TABLE3-2        /FROM TABLE3
4123  1411              EFADD!411       /A+A
4124  6776              TABLE3-2
4125  2411              EFSUB!411       /A+A-A
4126  6776              TABLE3-2
4127  6531              EFSTA!531       /STORE RESULT
4130  7376              TABLE4-2        /IN TABLE4
4131  1131              EJSR            /JMP-SAVE RETURN
4132  4716              DPCMPR          /COMPARE SUB
4133  2101              EJXN!100                /COMPLETED BLOCK
4134  4121              FPP4R
4135  0002              EFCLA           /YES
4136  1210              EFADD!210       /LOCK ONTO TEST
4137  1041              EJNE            /IF OFFSET 10 IS NON-ZERO
4140  4112              FPP4
```

```
                    /
                    /EXECUTE IN FLOATING POINT MODE
                    /MULTIPLY-DIVIDE TEST
                    /
4141  1131    FPP5,   EJSR            /JMP-SAVE RETERN
4142  4476            FSET            /SETUP SUB
4143  1121            EJSA
4144  5037            LINK
4145  1031            EJA
4146  4172            FPP6
4147  0005            ESTRTF
4150  0511    FPP5R,  EFLDA!511       /LOAD RANDOM ARG
4151  5775            TABLE1-3        /FROM TABLE1
4152  0004            EFNORM          /NORMALIZE
4153  6211            EFSTA!211       /STORE IN OFFSET 11
4154  4411            EFMUL!411       /A*A
4155  5775            TABLE1-3
4156  6212            EFSTA!212       /STORE IN OFFSET 12
4157  3211            EFDIV!211       /A*A/A
4160  6531            EFSTA!531       /STORE RESULT
4161  6375            TABLE2-3        /IN TABLE2
4162  1131            EJSR            /JMP-SAVE RETURN
4163  4572            FCMPMD          /COMPARE SUB
4164  2101            EJXN!100                /COMPLETED BLOCK
4165  4150            FPP5R
4166  0002            EFCLA           /YES
4167  1210            EFADD!210       /LOCK ONTO TST
4170  1041            EJNE            /IF OFFSET 10 IS NON-ZERO
4171  4141            FPP5
```

```
                /
                /EXECUTE IN DOUBLE PRECISION MODE
                /MULTIPLY DIVIDE TEST
                /
4172  1131      FPP6,   EJSR            /JMP-SAVE RETURN
4173  4643              DPSET           /SETUP ROUTINE
4174  1121              EJSA
4175  5037              LINK
4176  1031              EJA
4177  4222              FPP7
4200  0006              ESTRTD
4201  0511      FPP6R,  EFLDA!511       /LOAD RANDOM ARG
4202  6776              TABLE3-2        /FROM TABLE3
4203  4411              EFMUL!411       /A*A
4204  6776              TABLE3-2
4205  6212              EFSTA!212       /STORE IN OFFSET 12
4206  3411              EFDIV!411       /A*A/A
4207  6776              TABLE3-2
4210  6531              EFSTA!531       /STORE RESULT
4211  7376              TABLE4-2        /IN TABLE4
4212  1131              EJSR            /JMP-SAVE RETURN
4213  4716              DPCMPR          /COMPARE SUB
4214  2101              EJXN!100                /COMPLETED BLOCK
4215  4201              FPP6R
4216  0002              EFCLA           /YES
4217  1210              EFADD!210       /LOCK ONTO TST
4220  1041              EJNE            /IF OFFSET 10 IS NON-ZERO
4221  4172              FPP6
```

```
                        /
                        /EXECUTED IN FLOATING POINT MODE
                        /NORMALIZE-ALIGN TEST
                        /
4222  1131      FPP7,   EJSR            /JMP-SAVE RETURN
4223  4476              FSET            /SETUP SUB
4224  1121              EJSA
4225  5037              LINK
4226  1031              EJA
4227  4264              FPP10
4230  0005              ESTRTF
4231  0411      FPP7R,  EFLDA!411       /LOAD RANDOM ARG
4232  5776              TABLE1-2        /FROM TABLE1
4233  0006              ESTRTD
4234  0022              EATX!2          /STORE IN IR 2
4235  0005              ESTRTF
4236  0511              EFLDA!511       /LOAD RANDOM ARG
4237  5775              TABLE1-3        /FROM TABLE1
4240  0004              EFNORM          /NORMALIZE
4241  0012              EALN!2          /ALIGN ON IR 2
4242  0004              EFNORM          /ETC
4243  0012              EALN!2
4244  0004              EFNORM
4245  0012              EALN!2
4246  0004              EFNORM
4247  0012              EALN!2
4250  0004              EFNORM
4251  0012              EALN!2
4252  6531              EFSTA!531       /STORE RESULT
4253  6375              TABLE2-3        /IN TABLE2
4254  1131              EJSR            /JMP-SAVE RETURN
4255  4536              FCMPR           /COMPARE SUB
4256  2101              EJXN!100                /COMPLED BLOCK
4257  4231              FPP7R
4260  0002              EFCLA           /YES
4261  1210              EFADD!210       /LOCK ONTO TST
4262  1041              EJNE            /IF OFFSET 10 IS NONZERO
4263  4222              FPP7
```

```
                    /
                    /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
                    /
4264  1131  FPP10,  EJSR             /JMP-SAVE RETURN
4265  4654          DPSET1           /SETUP SUB
4266  1121          EJSA
4267  5037          LINK
4270  1031          EJA              /GO TO NEXT TEST
4271  4317          FPP11
4272  0006          ESTRTD
4273  0712  FPP10R, EFLDA!712        /LOAD RAN ARG FROM TABLE 3
4274  0014          EALN!4           /SHIFT VIA IR 4
4275  0015          EALN!5           /SHIFT VIA IR 5
4276  0014          EALN!4           /ETC
4277  0015          EALN!5
4300  0014          EALN!4
4301  0015          EALN!5
4302  0014          EALN!4
4303  0015          EALN!5
4304  0014          EALN!4
4305  0015          EALN!5
4306  6731          EFSTA!731        /STORE RESULT IN TABLE4
4307  1131          EJSR             /JMP-SAVE RETURN
4310  4726          DPCPR1           /COMPARE ROUTINE
4311  2101          EJXN!100                 /COMPLETED BLOCK
4312  4273          FPP10R
4313  0002          EFCLA            /YES
4314  1210          EFADD!210        /LOCK ONTO TST
4315  1041          EJNE             /IF OFFSET 10 IS NON-ZERO
4316  4264          FPP10
```

```
                        /
                        /EXECUTED IN FLOATING POINT MODE
                        /ADD TO MEMORY-SUBTRACT TEST
                        /PRELIMINARY COMPARE OF FADDM AND FADD RESULTS MADE
                        /THEY SHOULD BE EQUAL
                        /
4317  1131      FPP11,  EJSR            /JMP-SAVE RETURN
4320  4510              FSET1           /SETUP SUB
4321  1121              EJSA
4322  5037              LINK
4323  1031              EJA
4324  4352              FPP12
4325  0005              ESTRTF
4326  0713      FPP11R, EFLDA!713       /LOAD RAN ARG FROM TABLE1
4327  1734              EFADD!734       /A+A
4330  6222              EFSTA!222       /STORE IN OFFSET 22
4331  0613              EFLDA!613       /LOAD RAN ARG FROM TABLE1
4332  5634              EFADDM!634      /A+A TO MEMORY
4333  0634              EFLDA!634       /LOAD A+A
4334  2222              EFSUB!222       /SUBTRACT OFFSET 22
4335  1001              EJEQ            /SHOULD EQUAL ZERO
4336  4340              GO11            /YES
4337  0000              EFEXIT          /FADDM AND FADD RESULTS DIFFER
4340  0613      GO11,   EFLDA!613       /LOAD ARG IN TABLE1
4341  6634              EFSTA!634       /STORE BACK IN TABLE2
4342  1131              EJSR            /JMP-SAVE RETURN
4343  4536              FCMPR           /COMPARE
4344  2101              EJXN!100                /COMPLETED BLOCK
4345  4326              FPP11R
4346  0002              EFCLA           /YES
4347  1210              EFADD!210       /LOCK ONTO TST
4350  1041              EJNE            /IF OFFSET 10 IS NON-ZERO
4351  4317              FPP11
```

```
                        /
                        /EXECUTED IN DOUBLE PRECISION MODE
                        /ADD TO MEMORY-SUBTRACT TEST
                        /PRELIMINARY COMPARE OF FADDM AND FADD RESULTS ARE MADE
                        /THEY SHOULD BE EQUAL
                        /
4352  1131      FPP12,  EJSR            /JMP-SAVE RETURN
4353  4671              DPSET2          /SETUP SUB
4354  1121              EJSA
4355  5037              LINK
4356  1031              EJA
4357  4405              FPP13
4360  0006              ESTRTD
4361  0712      FPP12R, EFLDA!712       /LOAD RAN ARG FROM TABLE3
4362  1731              EFADD!731       /A*A
4363  6222              EFSTA!222       /STORE IN OFFSET 22
4364  0612              EFLDA!612       /LOAD RAN ARG FROM TABLE3
4365  5631              EFADDM!631      /A*A TO MEMORY
4366  0631              EFLDA!631       /LOAD RESULT A*A
4367  2222              EFSUB!222       /SUBTRACT OFFSET 22
4370  1001              EJEQ            /SHOULD EQUAL ZERO
4371  4373              GO12            /YES
4372  0000              EFEXIT          /FADDM AND FADD RESULTS DIFFER
4373  0612      GO12,   EFLDA!612       /LOAD ARG IN TABLE3
4374  6631              EFSTA!631       /STORE BACK IN TABLE4
4375  1131              EJSR            /JMP-SAVE RETURN
4376  4716              DPCMPR          /COMPARE SUB
4377  2101              EJXN!100                /COMPLETED BLOCK
4400  4361              FPP12R
4401  0002              EFCLA           /YES
4402  1210              EFADD!210       /LOCK ONTO TST
4403  1041              EJNE            /IF OFFSET 10 IS NON-ZERO
4404  4352              FPP12
```

```
                        /
                        /EXECUTED IN FLOATING POINT MODE
                        /MULTIPLY TO MEMORY DIVIDE TEST
                        /PRELIMINARY COMPARE OF FMULM AND FMUL RESULTS ARE MADE
                        /THEY SHOULD BE EQUAL
                        /
4405  1131      FPP13,  EJSR              /JMP-SAVE RETURN
4406  4510              FSET1             /SETUP SUB
4407  1121              EJSA
4410  5037              LINK
4411  1031              EJA
4412  4442              FPP14
4413  0005              ESTRTF
4414  0713      FPP13R, EFLDA!713         /LOAD RAN ARG FROM TABLE1
4415  0004              EFNORM            /NORMALIZE
4416  6211              EFSTA!211         /STORE IN OFFSET 11
4417  4734              EFMUL!734         /A*A
4420  6222              EFSTA!222         /STORE IN OFFSET 22
4421  0211              EFLDA!211         /LOAD OFFSET 11
4422  7634              EFMULM!634        /A*A TO MEMORY
4423  0634              EFLDA!634         /LOAD RESULT A*A
4424  2222              EFSUB!222         /SUBTRACT OFFSET 22
4425  1001              EJEQ              /SHOULD EQUAL ZERO
4426  4430              GO13              /YES
4427  0000              EFEXIT            /FMULM AND FMUL RESULTS DIFFER
4430  0613      GO13,   EFLDA!613         /GET ARG IN TABLE1
4431  6634              EFSTA!634         /STORE BACK IN TABLE2
4432  1131              EJSR              /JMP-SAVE RETURN
4433  4536              FCMPR             /COMPARE
4434  2101              EJXN!100                  /COMPLETED BLOCK
4435  4414              FPP13R
4436  0002              EFCLA             /YES
4437  1210              EFADD!210         /LOCK ONTO TST
4440  1041              EJNE              /IF OFFSET 10 IS NON-ZERO
4441  4405              FPP13
```

```
                    /
                    /EXECUTED IN DOUBLE PRECISION MODE
                    /MULTIPLY TO MEMORY-DIVIDE TEST
                    /PRELIMINARY COMPARE OF FMULM AND FMUL RESULTS ARE MADE
                    /THEY SHOULD BE EQUAL
                    /
4442  1131      FPP14,  EJSR              /JMP-SAVE RETURN
4443  4671              DPSET2            /SETUP SUB
4444  1121              EJSA
4445  5037              LINK
4446  1031              EJA
4447  4013              FPP1
4450  0006              ESTRTD
4451  0712      FPP14R, EFLDA!712         /LOAD RAN ARG FROM TABLE3
4452  4731              EFMUL!731         /A*A
4453  6222              EFSTA!222         /STORE IN OFFSET 22
4454  0612              EFLDA!612         /LOAD RAN ARG FROM TABLE3
4455  7631              EFMULM!631        /A*A TO MEMORY
4456  0631              EFLDA!631         /LOAD RESULT A*A
4457  2222              EFSUB!222         /SUBTRACT OFFSET 22
4460  1001              EJEQ              /SHOULD EQUAL ZERO
4461  4463              GO14              /YES
4462  0000              EFEXIT            /FMULM AND FMUL RESULT DIFFERS
4463  0612      GO14,   EFLDA!612         /LOAD ARG IN TABLE3
4464  6631              EFSTA!631         /PUT BACK IN TABLE4
4465  1131              EJSR              /JMP-SAVE RETURN
4466  4716              DPCMPR            /COMPARE SUB
4467  2101              EJXN!100          /COMPLETED BLOCK
4470  4451              FPP14R
4471  0002              EFCLA             /YES
4472  1210              EFADD!210         /LOCK ONTO TST
4473  1031              EJA
4474  4013              FPP1
                    /
                    /DUMMY POINTER
                    /END OF FPP TESTS WAS FPP14
                    /
4475  0000      FPP15,  0
                    /
                    /ENTERED ONLY IN FLOATING POINT MODE
                    /SET UP OF FPP INDEX REGISTERS
                    /
4476  0005      FSET,   ESTRTF
4477  0002              EFCLA             /CLEAR THE FAC
4500  0100              FLDX!0            /LOAD IR 0
4501  7653              7653              /WITH -125
4502  0101              ELDX!1            /LOAD IR 1
4503  0000              0                 /WITH 0
4504  0103              ELDX!3            /LOAD IR 3
4505  0000              0                 /WITH 0
4506  1031              EJA               /JMP ALWAYS
4507  5701              BASE-1            /OFFSET 0 IN BASE REG TABLE
```

```
                        /
                        /ENTERED ONLY IN FLOATING POINT MODE
                        /SETS UP FPP INDEX REGISTERS
                        /DUPLICATES TABLE1 INTO TABLE2
                        /
4510  0005      FSETI,  ESTRTF
4511  0002              EFCLA           /CLEAR THE FAC
4512  0100              ELDX!0          /LOAD IR 0
4513  7653              7653            /WITH -125
4514  0101              ELDX!1          /LOAD IR 1
4515  0000              0               /WITH 0
4516  0102              ELDX!2          /LOAD IR 2
4517  7653              7653            /WITH -125
4520  0103              ELDX!3          /LOAD IR 3
4521  0000              0               /WITH 0
4522  0104              ELDX!4          /LOAD IR 4
4523  0000              0               /WITH 0
4524  0105              ELDX!5          /LOAD IR 5
4525  0000              0               /WITH 0
4526  0541      MORE,   EFLDA!541       /LOAD RAN ARG
4527  5775              TABLE1-3        /IN TABLE1
4530  6551              EFSTA!551       /STORE RAN ARG
4531  6375              TABLE2-3        /IN TABLE2
4532  2121              EJXN!120                /DONE 125 TIMES
4533  4526              MORE            /NO-DO IT AGAIN
4534  1031              EJA             /JMP ALWAYS
4535  5701              BASE+1          /OFFSET 0 IN BASE REG TABLE
```

```
                    /
                    /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
                    /
4536  0411      FCMPR,  EFLDA!411       /LOAD RAN ARG
4537  5775              TABLE1-3        /IN TABLE1
4540  2431              EFSUB!431       /SUBTRACT RESULT
4541  6375              TABLE2-3        /IN TABLE2
4542  1001              EJEQ .          /JMP IF FAC=0
4543  4545              ERET            /RET ROUTINE TO REENTER MAIN FPP PROG
4544  0040              EFNOP
4545  0216      ERET,   EFLDA!216       /LOAD OFFSET 16 IN BASE REG TABLE
4546  1001              EJEQ            /JMP IF FAC=0--IE, DONT LOCK ONTO TEST SEQ
4547  4556              RETINC          /RETURN TO MAIN FPP PROG
4550  0110              EADDX!0         /ADD TO IR 0
4551  7777              7777            /-1
4552  0111              EADDX!1         /ADD TO IR 1
4553  7777              7777            /-1
4554  0113              EADDX!3         /ADD TO IR 3
4555  7777              7777            /-1
4556  1031      RETINC, EJA             /JMP ALWAYS
4557  5701              BASE+1          /OFFSET OF 0 IN BASE REG
```

```
                        /
                        /ENTER ONLY IN FLOATING POINT MODE
                        /FROM AN ADDITION-SUBTRACTION TEST
                        /COMPARE TABLE1 WITH TABLE2
                        /THEY SHOULD BE EQUAL
                        /
4560  0411    FCMPAS, EFLDA!411       /LOAD RAN ARG
4561  5775            TABLE1-3        /IN TABLE1
4562  2431            EFSUB!431       /SUBTRACT RESULT
4563  6375            TABLE2-3        /IN TABLE2
4564  1041            EJNE            /JMP IF FAC NOT 0
4565  4570            FASCK           /ADD-SUBTRACT DATA ERROR
4566  1031            EJA             /JMP ALWAYS
4567  4545            ERET            /RET ROUTINE TO REENTER MAIN FPP PROG
                        /
                        /FLOATING POINT ADD-SUB FAILED
                        /
4570  1031    FASCK,  EJA             /DATA ERROR EXIT
                        /
4571  4545            ERET
                        /ENTERED ONLY IN FLOATING POINT MODE
                        /FROM ROUTINE DOING A MULTIPLY-DIVIDE TEST
                        /THIS ROUTINE COMPARES TABLE1 WITH TABLE2
                        /THEY SHOULD BE EQUAL
                        /
4572  0411    FCMPMD, EFLDA!411       /LOAD RAN ARG
4573  5775            TABLE1-3        /IN TABLE1
4574  2431            EFSUB!431       /SUBTRACT RESULT
4575  6375            TABLE2-3        /IN TABLE2
4576  1041            EJNE            /JMP IF FAC NOT 0 TO FEMD
4577  4602            FMDCK           /CHECK FOR LEGAL UNDERFLOW
4600  1031            EJA             /JMP ALWAYS
4601  4545            ERET            /RET ROUTINE TO REENTER MAIN FPP PROG
```

```
                        /
                        /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
                        /
4602  0431      FMDCK,  EFLDA!431       /LOAD RESULT
4603  6375              TABLE2-3        /FROM TABLE2
4604  1001              EJEQ            /JMP IF FAC=0 TO
4605  4632              FZMD            /ROUTINE TO TEST FOR LEGAL UNDERFLOW
4606  0006              ESTRTD
4607  1214              EFADD!214       /SUBTRACT 1 CONTAINED IN OFFSET 14 OF BASE REG
4610  0005              ESTRTF
4611  6215              EFSTA!215       /STORE IN OFFSET 15 OF BASE REG
4612  0215              EFLDA!215       /LOAD OFFSET 15 IN BASE REG
4613  2411              EFSUB!411       /SUBTRACT RAN ARG
4614  5775              TABLE1-3        /IN TABLE1
4615  1001              EJEQ            /JMP IF FAC=0
4616  4545              ERET            /RET ROUTINE TO REENTER MAIN FPP PROG
4617  0431              EFLDA!431       /LOAD WITH RESULT
4620  6375              TABLE2-3        /IN TABLE2
4621  0006              ESTRTD
4622  1217              EFADD!217       /ADD 1 CONTAINED IN OFFSET 17 OF BASE REG
4623  0005              ESTRTF
4624  6221              EFSTA!221       /STORE IN OFFSET 21 BASE REG
4625  0221              EFLDA!221       /LOAD OFFSET 21 IN BASE REG
4626  2411              EFSUB!411       /SUBTRACT RAN ARG
4627  5775              TABLE1-3        /IN TABLE1
4630  1031              EJA
4631  4545              ERET            /RET ROUTINE TO RE-ENTER MAIN FPP PROG
                        /
                        /ENTERED ONLY IN FLOATING POINT MODE
                        /WHEN RESULT HAS ZERO MANTISSA
                        /ROUTINE TESTS FOR A LEGAL UNDERFLOW
                        /
4632  0411      FZMD,   EFLDA!411       /LOAD RANDOM ARG
4633  5775              TABLE1-3        /FROM TABLE1
4634  4213              EFMUL!213       /MULTIPLY BY CONSTANT IN OFFSET 13 OF BASE REG
4635  1001              EJEQ            /JMP IF FAC=0
4636  4545              ERET            /RET ROUTINE TO MAIN FPP PROG
4637  0002              EFCLA           /CLEAR FAC
4640  6223              EFSTA!223       /FLAG 8 UNDERFLOW NOT LEGAL
4641  1031              EJA
4642  4545              ERET
```

```
                        /
                        /SET UP OF FPP INDEX REGISTERS
                        /FOR OPERATION IN DOUBLE PRECISION MODE
                        /
4643  0006      DPSET,  ESTRTD
4644  0100              ELDX!0          /LOAD IR 0
4645  7600              7600            /WITH -200
4646  0101              ELDX!1          /LOAD IR 1
4647  0000              0               /WITH 0
4650  0103              ELDX!3          /LOAD IR 3
4651  0000              0               /WITH 0
4652  1031              EJA             /JMP ALWAYS
4653  5701              BASE+1          /OFFSET 0 IN BASE REG TABLE
                        /
                        /SET UP OF FPP INDEX REGISTERS ON PAGE 0
                        /FOR OPERATION IN DOUBLE PRECISION MODE
                        /ENTERED FROM FPP10 TEST
                        /
4654  0006      DPSET1, ESTRTD
4655  0100              ELDX!0          /LOAD IR 0
4656  7600              7600            /WITH -200
4657  0101              ELDX!1          /LOAD IR 1
4660  0000              0               /WITH 0
4661  0103              ELDX!3          /LOAD IR 3
4662  0000              0               /WITH 0
4663  0104              ELDX!4          /LOAD IR 4
4664  0014              14              /WITH 14
4665  0105              ELDX!5          /LOAD IR 5
4666  7764              7764            /WITH -200
4667  1031              EJA             /JMP ALWAYS
4670  5701              BASE+1          /OFFSET 0 IN BASE REG TABLE
```

```
                        /
                        /SET UP OF FPP INDEX REGISTERS ON PAGE 0
                        /DUPLICATION OF TABLE3 INTO TABLE4
                        /
4671  0006      DPSET2, ESTRTD
4672  0100              ELDX!0          /LOAD IR 0
4673  7600              7600            /WITH -200
4674  0101              ELDX!1          /LOAD IR 1
4675  0000              0               /WITH 0
4676  0102              ELDX!2          /LOAD IR 2
4677  7600              7600            /WITH -200
4700  0103              ELDX!3          /LOAD IR 3
4701  0000              0               /WITH 0
4702  0104              ELDX!4          /LOAD IR 4
4703  0000              0               /WITH 0
4704  0105              ELDX!5          /LOAD IR 5
4705  0000              0               /WITH 0
4706  0541      MOR,    EFLDA!541       /LOAD RAN ARG
4707  6776              TABLE3-2        /FROM TABLE3
4710  6551              EFSTA!551       /STORE IT
4711  7376              TABLE4-2        /IN TABLE4
4712  2121              EJXN!120        /DONE 200 TIMES
4713  4706              MOR             /NO-DO IT AGAIN
4714  1031              EJA             /JMP ALWAYS
4715  5701              BASE+1          /OFFSET 0 IN BASE REG TABLE


                        /
                        /COMPARES DOUBLE PRECISION NUMBERS
                        /TABLE3 WITH TABLE4
                        /USING DOUBLE WORD DIRECT REFERENCE INSTRUCTIONS
                        /
4716  0411      DPCMPR, EFLDA!411       /LOAD RANDOM ARG
4717  6776              TABLE3-2        /FROM TABLE3
4720  2431              EFSUB!431       /SUBTRACT RESULT
4721  7376              TABLE4-2        /FROM TABLE4
4722  1001              EJEQ            /IF THEY ARE EQUAL
4723  4545              ERET            /JMP TO RET
4724  1031              EJA
4725  4545              ERET
                        /
                        /COMPARES DOUBLE PRECISION NUMBERS
                        /TABLE3 WITH TABLE4
                        /USING SINGLE WORD INDIRECT REFERENCE INSTRUCTIONS
                        /
4726  0631      DPCPR1, EFLDA!631       /LOAD RESULT FROM TABLE4
4727  2612              EFSUB!612       /SUBTRACT RAN ARG FROM TABLE3
4730  1001              EJEQ            /IF THEY ARE EQUAL
4731  4545              ERET            /JMP TO RET
4732  1031              EJA
4733  4545              ERET
                        /
```

```
4734  1101    RANGEN, ESETX
4735  5026            XREG2
4736  1111            ESETB
4737  5002            RBASE
4740  0101            ELDX!1
4741  0000            0000
4742  0100            ELDX!0
4743  7600            -200
4744  0006            ESTRTD
4745  0202            EFLDA!202
4746  1051            EJLT!1
4747  4756            RCONST
4750  2203            EFSUB!203
4751  1051            EJLT
4752  4756            RCONST
4753  0202            EFLDA!202
4754  2200            EFSUB!200
4755  6202            EFSTA!202
4756  0202    RCONST, EFLDA!202
4757  6411            EFSTA!411
4760  6000            TABLE1
4761  0411    RGENA,  EFLDA!411
4762  6000            TABLE1
4763  1201            EFADD!201
4764  6411            EFSTA!411
4765  7000            TABLE3
4766  1201            EFADD!201
4767  6511            EFSTA!511
4770  6000            TABLE1
4771  2101            EJXN!100
4772  4761            RGENA
4773  6202            EFSTA!202
4774  1101            ESETX
4775  5016            XREG1
4776  1111            ESETB
4777  5700            BASE
5000  1031            EJA!1
5001  5701            BASE+1
```

```
5002  0000    RBASE,  2000
5003  2657            2657
5004  1234            1234
5005  0000            2000
5006  0005            0005
5007  0011            0011
5010  0000            0000
5011  2200            2200
5012  0000            0000
5013  0000            0000
5014  1373            1373
5015  0000            2000

5016  0000    XREG1,  0
5017  0000            0
5020  0000            0
5021  0000            0
5022  0000            0
5023  0000            0
5024  0000            0
5025  0000            0
5026  0000    XREG2,  0
5027  0000            0
5030  0000            0
5031  0000            0
5032  0000            0
5033  0000            0
5034  0000            0
5035  0000            0
```

```
5036  0000    EEND,   0
                /
                /
5037  1031    LINK,   EJA
5040  4011            FPP1-2
5041  2101            FJXN!100
5042  5045            .+3
5043  1031            EJA
5044  5037            LINK
5045  0006            ESTRTD
5046  0401            EFLDA!401
5047  5037            LINK
5050  1401            EFADD!401
5051  5053            TWO
5052  0007            EJAC

5053  0000    TWO,    0000
5054  0002            0002


5055  0000    XREG,   0

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

```
                5700                    *5700
5700            0000    BASE,           2               /OFFSET OF 0
5701            0000                    2
5702            0000                    2
5703            0000                    0               /OFFSET OF 1
5704            0001                    0001
5705            7376                    7376
5706            0000                    0               /OFFSET OF 2
5707            0001                    0001
5710            6776                    6776
5711            0000                    0               /OFFSET OF 3
5712            0001                    0001
5713            5775                    5775
5714            0000                    0               /OFFSET OF 4
5715            0001                    0001
5716            6375                    6375
5717            0000                    0               /OFFSET OF 5
5720            0000                    0
5721            0000                    0
5722            0000                    0               /OFFSET OF 6
5723            0000                    0
5724            0000                    0
5725            0000                    0               /OFFSET OF 7
5726            0000                    0
5727            0000                    0
5730            0000                    0               /OFFSET OF 10
5731            0000                    0
```

```
5732  0000    LOKTST, 0
5733  0000            0               /OFFSET OF 11
5734  0000            0
5735  0000            0
5736  0000            0               /OFFSET OF 12
5737  0000            0
5740  0000            0
5741  6000            6000            /OFFSET OF 13
5742  3777            3777
5743  7777            7777
5744  0000            0000            /OFFSET OF 14
5745  7777            7777
5746  7777            7777
5747  0000            0               /OFFSET OF 15
5750  0000            0
5751  0000            0
5752  0000            0               /OFFSET OF 16
5753  0000            0
5754  0000    LOKSEG, 0
5755  0000            0               /OFFSET OF 17
5756  0000            0
5757  0001            0001
5760  0000            0               /OFFSET OF 20
5761  0000            0
5762  0000            0
5763  0000            0               /OFFSET OF 21
5764  0000            0
5765  0000            0
5766  0000    RESLT,  0               /OFFSET OF 22
5767  0000            0
5770  0000            0
5771  0000            0000            /OFFSET OF 23
5772  0000            0000
5773  7777    UFLO,   7777
```

```
                        /
                        /
                        /
                        /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////////////
                        ////////////////////////////////////
```

@

```
0162  0254
0163  7336
0164  0347
0165  0346
0166  7302
0167  0620
0170  7346
0171  0672
0172  0666
0173  0665
0174  0061
0175  0060
0176  7772
0177  7313
```

```
0000  10000000  00000000  11111111  11111111  11111111  11111111  11111111  11111111
0100  11111111  11111111  11111111  11111000  00000000  00000000  00111111  11111111

0200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0300  11111111  11111111  11111111  11111111  11111100  00000000  01111111  11111111

0400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0500  11111111  11111111  00000000  00000000  00000000  00000000  00000001  11111111

0600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0700  11111111  11111111  11111111  11111111  11100000  00000000  00000000  00000000


1000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1100  11111111  11111111  11111111  11111111  11111111  11111100  00111111  11111111

1200  11111111  11111111  11111000  00000000  00000000  00000000  00000000  00000000
1300  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00011111

1400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

1600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1700  11111111  11111111  11111111  11111111  11111000  00000000  00000000  00000000


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  10000000  00000000  00000000  00000000


3000
3100

3200
3300

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  11111100  00000000  00000000
5100  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

5200
5300

5400
5500

5600  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
5700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11110000

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    0107
ACBIT   0541
ACEQ    2542
ACEXP   0035
ACLSW   0037
ACMSW   0036
ACNEZ   2247
ADDAB   7012
ADDR    2652
ADDX    5212
ADDX1   5213
ADDX2   5225
ADDX3   5235
ADDX4   5243
ADEQ    2616
ADXOK   2321
AEQ     2526
AEXT    0026
AEZ     7124
AGZ     7143
ALN1    4037
ALN2    4057
ALN22   4077
ALN3    4200
ALN4    4215
ALNEND  4246
ALNOK   2271
ALSW    0025
ALZ     7135
AMBO    4577
AMSW    0024
APBO    4576
APEQ    2570
APT     0210
APTC    5450
APTDEC  6527
APTERR  2655
APTPAC  6104
APTSAV  1044
AREG    0024
ARITH   2030
AS      0600
ASADDR  2646
ASC     7346
ASC1    0657
ASC10   0620
ASC2    0660
ASC3    0661
ASC4    0662
ASC5    0663
ASC77   0664
ASC8    0644
ASCCH   0345
ASCDAT  0112
ASCL    0072
ASCL1   0105
ASCL2   0106
ASCWD   0346
ASCWD0  0022
ASFLD   0633
ASGET   0616
ASLOC   0673
ASPNTR  0662
AST     2465
ASTER   7261
ATX1    4266
ATX2    4312
ATX3    4337
ATX4    4400
ATX4A   4421
ATX5    4425
ATXOK   2275
BADNWS  6242
BASE    5700
BEGIN   0243
BEQ     2532
BEXT    0031
BLSW    0030
BMSW    0027
BRANCH  1046
BREG    0027
CAPT    4575
CAROUT  0077
CARYIN  0076
CKMEM   2131
CKO     0117
CKOP    0120
CKOPAD  6320
CKST    6552
CKSUB   6541
CL      0542
CLEAR   4574
CLRA    4573
CLRAX   6703
CLRB    4572
CLRBX   6710
CLRCT   5417
CLRO    4571
CLROX   6715
CLRT    5420
CM      0476
CMEM    6632
CMEM1   6600
CMEM2   6650
CMEME   4570
CMEMF   4567
CNTRLS  7240
CODE    1216
COMPO   6120
COMREG  0052
CRLF    2740
CRSW    0024
CSTATE  0105
D27     0100
DATABL  1551
DATC1   1036
DATC2   1060
DATERR  1011
DATTXT  1532
DECAPT  4566
DECOP   4565
DEP     2052
DEP11   2053
DEP12   2116
DEP13   2135
DEP14   2200
DEP144  2240
DEP15   2255
DEPEND  2415
DEPOK   2341
DIV0    3472
DIV0A   3523
DIV1    3535
DIV2    3600
DIV3    4000
DIVC    3731
DIVIDE  3614
DIVOK   2257
DIVOV   2241
DIVZ    2231
DOT     0413
DOTCK   0414
DOTER0  1113
DOTERR  0432
DOTEX   0245
DPADD   2650
DPADD0  2651
DPADD1  2657
DPASOK  2361
DPCMPR  4716
DPCPR1  4726
DPFLOW  2402
DPOVFL  2675
DPSET   4643
DPSET1  4654
DPSET2  4671
DTXT1   1727
DTXT2   1736
DVEND   4007
EA      0452
EADDX   0110
EALN    0010
EATX    0020
ECONO   2712
EDAT1   1154
EDAT2   1155
EEND    5036
EFADD   1000
EFADDM  5000
EFCLA   0002
EFDIV   3000
EFEXIT  0000
EFLDA   0000
EFMUL   4000
EFMULM  7000
EFNEG   0003
EFNOP   0040
EFNORM  0004
EFSTA   6000
EFSUB   2000
EJA     1031
EJAC    0007
EJEQ    1001
EJGT    1061
EJLT    1051
EJNE    1041
EJSA    1121
EJSR    1131
EJXN    2001
ELDX    0100
EMEM    0005
END15   2315
ENDDIV  3703
ENDFAD  2420
ENDMUL  2222
ENINIT  1076
ENT     2000
ENTER   4564
ENTSTP  0114
ENTTXT  6233
ENTYP   6232
EOP     2354
ER1     5730
ER2     5731
ER3     5732
ERET    4545
EROUT   2332
ERR     7317
```

ERR10	1000	FECH1	1300	FPP4	4112	INIT1	1137
ERRET	1111	FECH2	1320	FPP4R	4121	INIT2	1200
ERROR	0000	FECH24	1350	FPP5	4141	INIT3	1205
ERRT1	1112	FECH3	1400	FPP5R	4150	INIT4	1213
ERPTP	1074	FECH34	1406	FPP6	4172	INIT5	1220
ESETB	1111	FECH4	1415	FPP6R	4201	INIT6	1225
ESETX	1101	FECH42	1424	FPP7	4222	INOK	2452
ESTART	4000	FECH5	1446	FPP7R	4231	INS0	7200
ESTOP	6245	FECH6	1454	FPPRUN	1045	INS21	7220
ESTRTD	0006	FECH7	1471	FPRST	6556	INSTEP	2626
ESTRTF	0005	FECH72	1512	FPST	6555	JAC	5050
ETXT1	1076	FECH73	1532	FSET	4476	JAC1	5051
ETXT2	1100	FEND	1533	FSET1	4510	JAL	5022
EX	0436	FETCH	1236	FSIS	1610	JEQ	5000
EXADDR	0113	FLAG1	0073	FSTA	5247	JFALSE	5045
EXCOM	2627	FLAG2	0074	FSTAOK	2440	JGE	5003
EXEC	1600	FLAG3	0075	FSTD	5070	JGT	5017
EXIT	2427	FLDA	5273	FSTEP	4563	JLE	5006
EXIT0	2430	FLDIV	3747	FSTF	5063	JLT	5014
EXIT1	2470	FLDX	4010	FSTIS	1572	JNE	5011
EXIT2	2501	FMAINT	6561	FXTA	4434	JSA	4642
EXIT3	2512	FMDCK	4602	FXMD	4632	JSA2	4643
EXIT4	2523	FMIS	1674	GAPT	6246	JSA3	4656
EXIT5	2534	FMULT	3257	GET	6262	JSA4	4664
EXIT6	2600	FNEG	5076	GETAPT	4562	JSAOK	2313
EXIT7	2605	FNORM	5114	GETOP	4561	JSR	4600
EXITSW	0072	FOEQ	2522	GETPC	4560	JSR1	4601
EXITWD	0260	FOIS	1657	GETREG	0266	JSR2	4611
EXSAV	2457	FOPIS	1711	GETT1	1116	JSR3	4622
EXSTP	0115	FPAUSE	5126	GETX	4557	JSR4	4630
EXTA	0030	FPCOM	6553	GO11	4340	JSROK	2311
EXWD	0110	FPHLT	6554	GO12	4373	JTRUE	5027
FACFR	0036	FPICL	6552	GO13	4430	JTRUE1	5030
FAD01A	3127	FPINT	6551	GO14	4463	JXN	4473
FADD0	3001	FPIST	6557	GOP	6256	JXN1	4474
FADD01	3105	FPP1	4013	GPC	6252	JXN2	4515
FADD02	3200	FPP10	4264	HALT	7402	JXN2A	4530
FADD1	3026	FPP10R	4273	IGNFL	0112	JXN3	4540
FADD2	3041	FPP11	4317	INC1	5423	JXNEND	4554
FADD3	3213	FPP11R	4326	INC2	5432	JXNOK	2305
FADEND	3252	FPP12	4352	INC3	5441	KEYCK	0250
FALN	4036	FPP12R	4361	INCAPT	4556	KEYCK0	0132
FAPIS	1627	FPP13	4405	INCOP	4555	KEYCKL	0125
FASCK	4570	FPP13R	4414	INCOR	4554	KEYL1	7306
FATX	4265	FPP14	4442	INCPC	4553	LDA0	5274
FCLA	5150	FPP14R	4451	INCST	4552	LDA1	5303
FCLR	5400	FPP15	4475	INCX	4551	LDA2	5307
FCLRA	5405	FPP1R	4024	INDEX	6460	LDAC	0650
FCMPAS	4560	FPP2	4040	INDIR	1121	LDAOK	2445
FCMPMD	4572	FPP2R	4047	INEND	1234	LDEXT	4000
FCMPR	4536	FPP3	4063	INIT	1064	LDOP	0456
FECH?	1250	FPP3R	4072	INIT0	1105	LDX1	4011

LDX2	4023	OEZ	7113	RARB	4251	SHREG	0071
LDX3	4032	OLEFT	3670	RBASE	5002	SOEZ	4531
LDXOK	2265	OLSW	0022	RCONST	4756	SPACE	1200
LINK	5037	OMSW	0021	RDOP	6566	SPEC1	1620
LMEM	0007	OPADR	0044	READ	0265	SPEC2	1637
LOAD	6061	OPDEC	7000	REGEQ	0600	SPEC20	1674
LOADA	4550	OPEQ	2546	REGS	0020	SPEC21	1650
LOADAC	4547	OPERND	4100	RESLT	5766	SPEC3	2000
LOADB	4546	OPLUS1	6722	RET	0404	SPECFL	5465
LOADMQ	4545	OPOS	2400	RETINC	4556	SSTEP	5650
LOADO	4544	OREG	0021	RETURN	0350	STA0	5250
LOADOP	4543	ORERR	6140	RF	0510	STA1	5260
LOADPC	4542	OUTSTP	2636	RFA	0517	STA2	5264
LODX	0466	OVFL	0111	RFB	0526	STAOK	2325
LOKSEG	5754	PAPIS	1637	RGENA	4761	START	0230
LOKTST	5732	PAPT	0047	ROLSW	6564	STATEQ	2553
LSHFT	6567	PAUS0	5127	ROMSW	6563	STEP	5600
MADD	3401	PAUS1	5142	ROUT	2466	STEPGO	5671
MCNT	0116	PBASE	0054	RSEQ	2602	STEPSW	0106
MCODE	6643	PCEQ	2611	RSTATE	6562	STERR	7325
MDEC	2672	PFAC	0035	RT	0506	STFPP	1016
MDFLAG	0004	PFADD	3000	RUBOUT	1000	STINC	6732
MEMINS	6740	PFDIV	3467	RUTXT	1012	STORA	4530
MFLD	0705	PFPC	0041	SA	0400	STORB	4527
MMEM	0006	PINCF	6674	SAEZ	4536	STRA	6144
MOR	4706	PINDEX	6656	SAGZ	4535	STRB	6156
MORE	4526	PIR	0051	SALZ	4534	STRTER	1644
MOVEX	4541	PIREQ	2622	SAVAPT	5501	STSAVE	0351
MQEQ	2536	PMIS	1704	SAVMEM	0706	SUBAB	7057
MQEXT	0034	POEQ	2516	SAVOL	0003	SUBSW	7112
MQLSW	0033	POIS	1667	SAVOM	0002	T1	0103
MQMSW	0032	POPIS	1722	SAVOP	0121	T10	0020
MQREG	0032	PROCES	1613	SETB	5157	T100	0740
MSTATE	6000	PSEQ	2560	SETB1	5160	T2	0104
MT1	0000	PSIS	1622	SETFL	7054	T20	0021
MT2	0001	PSTAT	0053	SETRET	7254	T212	0736
MULEND	3466	PSTIS	1602	SETST3	6344	T215	0735
MULT0	3262	PUTX	4537	SETST5	6335	T240	0741
MULT1	3306	PX0	0220	SETST6	6353	T77	0742
MULT1A	3337	PX1	0221	SETTAB	1400	TABLE1	6000
MULT2	3400	PX2	0222	SETUP	6200	TABLE2	6400
MULT21	3427	PX3	0223	SETX	5200	TABLE3	7000
MULT3	3461	PX4	0224	SETX1	5201	TABLE4	7400
NEG1	5077	PX5	0225	SFOK	2460	TADDX	2211
NOFLOW	5476	PX6	0226	SH	0663	TALN	2076
NOREP	1103	PX7	0227	SHFCNT	0070	TATX	2100
NORM	4540	PXP	0057	SHFLAG	0067	TCLA	2140
NORM1	5115	QMK	2736	SHFOP	3073	TDEP	2030
NXSHFT	6424	QUONEG	3735	SHFTB	4533	TDFLD	0426
OCTWD	0347	RA	0502	SHFTO	4532	TDIV	2070
OCTWD0	0023	RANGEN	4734	SHIFTB	6502	TDPAS	2034
OEXT	0023	RAPT	6565	SHIFTO	7152	TEST	4000

TEXIT	2013	TYB	0242
TFADD	2047	TYC	0254
TFALSE	2126	TYCNT	0455
TFETCH	2010	TYCONT	0265
TINIT	2004	TYD	0422
TITLE	7266	TYDATA	0400
TJAC	2112	TYP	7313
TJSA	2110	TYP10	0665
TJSR	2106	TYPA	0711
TJXN	2104	TYPL	0025
TLDA	2217	TYPLD	0061
TLDX	2074	TYPLX	0051
TLSW	0063	TYPNCR	7335
TM40	0737	TYT1	0243
TMES	0723	TYT2	0244
TMSW	0062	UFLO	5773
TMULT	2063	VFER	2347
TNEG	2153	WORD	0261
TNEXT	0305	WORD0	0113
TNORM	2156	WORD2	1052
TOA	6035	WORDL	7302
TOAC	6023	X0	0200
TOB	6042	X0ADR	0065
TOFECH	2454	X1	0201
TOMEM	4526	X2	0202
TOMQ	6047	X3	0203
TOO	6030	X4	0204
TOOP	6054	X5	0205
TOPC	6016	X6	0206
TPAUSE	2161	X7	0207
TRACE	4525	XEQ	2574
TRAP1	1623	XGETX	6476
TRAPED	1622	XNCK	6412
TREG	0062	XNEND	6444
TREXIT	5705	XNORM	6400
TRSKP	6304	XPLUS1	6471
TSETB	2203	XPUTX	6447
TSETX	2206	XREG	5055
TSTA	2214	XREG1	5016
TSTD	2147	XREG2	5026
TSTF	2143	XTA1	4435
TTRAP	2016	XTA2	4447
TTRUE	2114	XTAOK	2301
TTY	0724		
TWD	0457		
TWO	5053		
TXTA	2102		
TY	7352		
TY10	0200		
TYA	0214		
TYADD	0456		
TYALL	0330		

ERRORS DETECTED: 0

LINKS GENERATED: 233

RUN-TIME: 48 SECONDS

4K CORE USED