E R R A T U M

Page 4, line 11. The information in the internal memo Teletype Control Addition by Stephen F. Lundstrom, 6 June 1967, is no longer correct. Correct information will be published shortly in a memo by D. Wood and R. Breuder.

MAW:mb

SECTION	SUBSECTION	SHEET	DATE 17 August 1967

CONCOMP

PROPOSED	APPROVED	IMPLEMENTED	RELEASED

TO: All PDP-7 Users
FROM: David E. Wood
SUBJECT: Teletype Control Addition*

STANDARD OPERATION

The standard teletype control on the PDP-7 operates in a half-duplex mode. That is, the keyboard is connected in a closed loop with the printer as well as an input to the PDP-7. Thus, characters typed on the keyboard are also printed on the printer with no program control.

ADDITIONAL FEATURE

A panel switch has been added to the teletype control which allows the teletype to be used either in half-duplex or full-duplex. Recall that in full-duplex mode, the keyboard is solely input to the PDP-7 and the printer is solely an output device of the PDP-7. In full-duplex mode, the printer will record only those characters specifically typed under program control.

USES

The full duplex mode can be useful in a number of situations. Some of these are listed below.

1. To be able to enter data into a program while previous results are being printed.
2. To be able to type in commands to the text editor which are not printed, thus allowing listings to

---

* This memo supercedes a memo of the same title by Stephen F. Lundstrom, issued 6 June 1967, and listed as a reference in PDP-8 Simulator, a Concomp Project memorandum by Stephen F. Lundstrom and Dianne Callan, July 1967.

TO: All PDP-7 Users
FROM: David E. Wood
SUBJECT: Teletype Control Addition (cont'd)

be generated without a copy of the editor commands being included.

3. To be able to use the PDP-8 Simulator conveniently. (The teletype on the PDP-8 is full-duplex.)

IMPLEMENTATION

The full-duplex/half duplex procedure is implemented by inserting a normally closed SPST toggle switch in the signal line running from A14H to B14D in the PDP-7 I/O Interface Section. In order to separate completely the operations of the keyboard and teleprinter it is further necessary to remove the synchronization between their two clocks which results from the reset signal to CSØ - CS3. This is implemented by removing the connection from C12T to C18F, C18F to C18R, and C18R to C19F. (Refer to PDP-7 Drawing D-KA71A-0-9, "Teleprinter Control.")

LOCATION

The switch, whose function has been described, is located on a panel behind the doors just to the right of the PDP-7 Control Panel.

T H E U N I V E R S I T Y O F M I C H I G A N

Memorandum

PDP-8 SIMULATOR

Stephen F. Lundstrom
Dianne Callan

CONCOMP: Research in Conversational Use of Computers
F. H. Westervelt, Director
ORA Project 07449

supported by:

DEPARTMENT OF DEFENSE
ADVANCED RESEARCH PROJECTS AGENCY
WASHINGTON, D.C.

CONTRACT NO. DA-49-083 OSA-3050
ARPA ORDER NO. 716

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

July 1967

ABSTRACT

This program, written in PDP-7 code, simulates a PDP-8 computer, i.e., it will execute a program written in PDP-8 binary code.

1. REQUIREMENTS

1.1 Storage

00000_8 - 11400_8

The PDP-8 program is stored in 0002_8 - 7777_8. PDP-8 locations 0 and 1 are placed in locations ADD0 and ADD1 respectively, since these locations are used for interrupt control.

The simulation (PDP-7) program is stored in 10005_8 - 11400_8.

1.2 Subprograms and Subroutines

Standard RIM and PDP-7 binary loaders

PDP-8 binary tape of the program to be simulated

2. USAGE

2.1 Loading

Normal binary tape loading procedure for main program.

The PDP-8 tape may be loaded automatically (when the program is started at RESET (10422) or with SW4=1.)

2.2 Calling Sequence

Not applicable

2.3 Switch Settings

SW0 = Continue
SW1 = Load Address from Switch Register
SW2 = Examine location in Memory
SW3 = Start(clears simulated AC and all flags)
SW4 = Load a new tape and start over
SW5 = Deposit the contents of the PDP-8 Switch Register. When the simulation is "running," a panel dump on JMP and JMS results.
SW6-17 = PDP-8 Switch Register.

2.4 Start-up and/or Entry

Starting address = 10422. When started at this location, low core is initialized and a new tape is loaded.

After loading this program and placing PDP-8 tape in the reader, press CONTINUE twice.

For a restart without automatically loading a tape, set PDP-7 Address Switches to location 10376 and press START.

2.5 Errors in Usage

The following message is printed when a checksum error occurs in the PDP-8 binary loader:

E #XXXXXX

where # is the error number, in this case 1, indicating a checksum error, and XXXXXX is the contents of the accumulator.

2.6 Recovery from such Errors

Reload the PDP-8-simulated tape.

3. RESTRICTIONS

3.1 IOT Instructions

Certain PDP-8 IOT instructions do not have exactly equivalent PDP-7 instructions, and therefore the PDP-7 instructions that are most similar are substituted. The instructions are:

PDP-8	PDP-7
KCC 6032	KRB followed by CLA
KRS 6034	KRB 700312
KRB 6036	KRB 700312
TPC 6044	TLS 700406

IOT and IOF are not executed by the simulator but are handled in an appropriate manner.

The remaining IOT's are mapped into PDP-7 IOT's as follows:

6XXY becomes 70XX0Y

3.2 Timing

The approximate simulation ratio is 100 to 1.

3.3 EAE

This simulation does not simulate the PDP-8 Extended Arithmetic Element.

3.4 Teletype

The PDP-7 teletype operates half duplex. The PDP-8 teletype operates full duplex and the input is commonly "echoed" on the printer. This echo may be suppressed by overrides in the object program, or by a simple hardware change to the PDP-7 that makes its teletype full duplex.

The full duplex/half duplex procedure was implemented by inserting a normally closed SPST switch in the signal line running from A14H to B14D in the PDP-7 I/O interface section.

Reference:

PDP-7 Drawing D-KA71A-0-9

CONCOMP Project

c/o Dr. B. Herzog
Industrial Engineering Dept.
The University of Michigan

An internal memo:

Re: Teletype Control Addition
From: Stephen F. Lundstrom
Date: 6 June 1967

4. DESCRIPTION

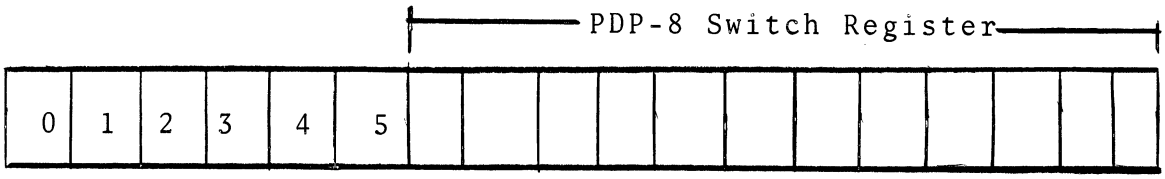
4.1 Discussion

Using the program and accumulator switches 0 - 5, one may load a PDP-8 binary tape, examine a given location in PDP-8 program memory, deposit PDP-8 instructions, etc., in a given location (0000_8 - 7777_8), or simulate a PDP-8 program.

On simulating a halt instruction, on simulating a JMP or JMS instruction (with SW5 = 1), or on a call to examine locations in memory (SW2 = 1), the program produces a dump consisting of a listing of the contents of the program counter, the memory buffer register, the memory address register, and the accumulator.

When "running" a PDP-8 program, the simulator program determines if each instruction is a basic memory reference, group 1 or 2 operate, or IOT instruction, and then executes the appropriate series of instructions which correspond to the PDP-8 instruction.

This program has been successfully run with all PDP-8 diagnostics, the PAL III assembler, the diagnostics for a 338 display control interfaced to the PDP-7, and numerous other PDP-8 object programs.



Switch No.	Function
0	Continue
1	Load Address from Switch Register
2	Examine Location in Memory
3	Start (clears simulated AC and all flags)
4	Load a new tape and start over
5	Deposit the contents of the PDP-8 Switch Register. When the simulation is "running", a panel dump on JMP and JMS results.
6 -17	PDP-8 Switch Register.

8 SIMULATOR

PIE	11400
AC	10005
ACOUT	11044
ADDPC	10047
ADDR	10006
ADD0	10007
ADD1	10010
AND8	10713
ASSEMB	10202
AUTIND	10676
BEGG	10063
BEGIN	10152
BEND	10141
CHEX	10171
CHKSUM	10011
CHR	10012
CMNT	10321
COMAC8	11156
COML8	10736
CORE	10547
CTR	10013
CTRR	10014
CYCLE	10446
C1	10015
DCA8	10760
DEP	10177
DPOSIT	10517
EAE8	11255
EFFADD	10644
ENDCOR	10556
ERROR	10236
ERR1	10221
EXAM	10501
GET0	10560
GET1	10562
GO	10156
GRP2	11257
HALTS	11356
HERE	10463
INCPC	10603
INTRPT	10421
INT8	10411
IOF8	11110
ION8	11105
IOP	10016
IOT8	11050
ISZ8	10744
JMPADD	10017
JMP8	11002
JMS8	10766
LABEL	10346

```
LDADD     10475
LDZRO     10020
LFTSFT    11223
LINK      10021
LOADR     10151
LOOP3     10325
LP3       10774
L1        10302
MA        10022
MAOUT     11034
MB        10023
MBOUT     11040
MEMTEM    10024
MINUS     10220
MSK       10025
MSKMB     11240
NONAUT    10707
NTBLK     10075
OCT       10274
OPR8      11116
OPS8      10613
ORAC      11365
ORIGIN    10026
ORSR      10055
PANEL     11006
PATCH     10533
PATCH2    10542
PC        10027
PCOUT     11031
PIE       11400
PRINT     10253
PTR       10030
PUT0      10575
PUT1      10600
RDTTY     10370
READ      10132
RESET     10422
RTSHFT    11173
RUN       10623
SAVAC     10031
SAVL      10032
SAV1      10033
SAV2      10034
SBEG      10065
SETLOW    10225
SGRP      11276
SHFTL     11214
SHFTR     11164
SIMUL     10375
SKIP      10035
STORE     10564
SWA       10036
SWITCH    10037
TAB       10356
TAD8      10722
TCR       10262
TEM1      10040
TEM8      10041
TSP       10270
TST1      10042
TST2      10043
TST3      10044
WORD1     10045
WORD2     10046
.KCC      11112
```

AC	10005
ADDR	10006
ADD0	10007
ADD1	10010
CHKSUM	10011
CHR	10012
CTR	10013
CTRR	10014
C1	10015
IOP	10016
JMPADD	10017
LDZRO	10020
LINK	10021
MA	10022
MB	10023
MEMTEM	10024
MSK	10025
ORIGIN	10026
PC	10027
PTR	10030
SAVAC	10031
SAVL	10032
SAV1	10033
SAV2	10034
SKIP	10035
SWA	10036
SWITCH	10037
TEM1	10040
TEM8	10041
TST1	10042
TST2	10043
TST3	10044
WORD1	10045
WORD2	10046
ADDPC	10047
ORSR	10055
BEGG	10063
SBEG	10065
NTBLK	10075
READ	10132
BEND	10141
LOADR	10151
BEGIN	10152
GO	10156
CHEX	10171
DEP	10177
ASSEMB	10202
MINUS	10220
ERR1	10221
SETLOW	10225
ERROR	10236
PRINT	10253
TCR	10262
TSP	10270
OCT	10274
L1	10302
CMNT	10321
LOOP3	10325
LABEL	10346
TAB	10356
RDTTY	10370
SIMUL	10375
INT8	10411
INTRPT	10421

RESET	10422
CYCLE	10446
HERE	10463
LDADD	10475
EXAM	10501
DPOSIT	10517
PATCH	10533
PATCH2	10542
CORE	10547
ENDCOR	10556
GET0	10560
GET1	10562
STORE	10564
PUT0	10575
PUT1	10600
INCPC	10603
OPS8	10613
RUN	10623
EFFADD	10644
AUTIND	10676
NONAUT	10707
AND8	10713
TAD8	10722
COML8	10736
ISZ8	10744
DCA8	10760
JMS8	10766
LP3	10774
JMP8	11002
PANEL	11006
PCOUT	11031
MAOUT	11034
MBOUT	11040
ACOUT	11044
IOT8	11050
ION8	11105
IOF8	11110
.KCC	11112
OPR8	11116
COMAC8	11156
SHFTR	11164
RTSHFT	11173
SHFTL	11214
LFTSFT	11223
MSKMB	11240
EAE8	11255
GRP2	11257
SGRP	11276
HALTS	11356
ORAC	11365
PIE	11400

```
/
/    VARIABLE ASSIGNMENTS
/
10005/
AC,             0
ADDR,           0
ADD0,           0
ADD1,           0
CHKSUM,         0
CHR,            0
CTR,            0
CTRR,           0
C1,             0
IOP,            0
JMPADD,         0
LDZRO,          0
LINK,           0
MA,             0
MB,             0
MEMTEM,         0
MSK,            0
ORIGIN,         0
PC,             0
PTR,            0
SAVAC,          0
SAVL,           0
SAV1,           0
SAV2,           0
SKIP,           0
SWA,            0
SWITCH,         0
TEM1,           0
TEM8,           0
TST1,           0
TST2,           0
TST3,           0
WORD1,          0
WORD2,          0
/
/    ADD TO PROGRAM COUNTER
/
ADDPC,          0
                LAC PC
                TAD (1
                AND (7777
                DAC PC
                JMP I ADDPC
/
/    OR WITH SWITCH REGISTER
/
ORSR,           0
                CLA!OAS
                AND (7777
                DAC TEM8
                JMS ORAC
                JMP I ORSR
```

TO

T1 1 of 2

```
/
/           THIS PROGRAM LOADS PDP-8 BINARY TAPES INTO
/           THE LOW ORDER BITS OF THE PDP-7
/
/
/
/           EXTRACT ERRORS, FIELD, L/T
/
BEGG,       0
            DZM SWITCH  /SET SWITCH
SBEG,       JMS READ    /GET A CHARACTER
            LAC SWA
            SZA
            JMP NTBLK
            LAC CHR
            SAD (0)
            JMP SBEG
            ISZ SWA
NTBLK,      LAC CHR
            TAD MINUS   /TEST FOR 377
            SPA!SNA!CLA
            JMP .+4     /NO
            ISZ SWITCH  /YES: COMPLEMENT SWITCH
            CMA
            JMP BEGG+1
            TAD SWITCH  /NOT 377
            SZA!CLA    /IS SWITCH SET$
            JMP BEGG+2  /YES -IGNORE
            TAD CHR     /NO - TEST FOR CODE
            AND (300)   /TYPES
            TAD (-177)
            SPA
            ISZ BEGG    /DATA OR ORIGIN
            SPA!SNA!CLA
            JMP I BEGG  /DATA, ORIGIN OR L/T
            TAD CHR     /FIELD SETTING
            RTR
            RTR
            RTR
            RAR
            AND (70000)
            DAC MEMTEM
            TAD (7777)
            AND ORIGIN
            ADD MEMTEM
            DAC ORIGIN
            JMP BEGG+2  /CONTINUE INPUT
/
/           READ ROUTINE
/
READ,       0
            RSA
            RSF
            JMP .-1
            RRB
```

```
            DAC CHR
            JMP I READ
/
/           TRAILER CODE SEEN                                    T1   2of2
/
BEND,       JMS ASSEMB
            CMA
            TAD (1)
            TAD CHKSUM
            AND (7777)
            SZA
            JMS ERR1
            JMP I LOADR
LOADR,      0           /START ENTRY IS HERE
BEGIN,      DZM SWA
            JMS BEGG    /GET CHARACTER
            JMP .-1     /IGNORE LEADER
            DZM CHKSUM  /NOW HAVE TAPE INFOR - ZERO CHKSUM
GO,         LAC CHR
            DAC WORD1
            JMS READ
            DAC WORD2
            JMS BEGG    /LOOK AHEAD
            JMP BEND    /TRAILER, END
            JMS ASSEMB
            SNL
            JMP DEP
            TAD MEMTEM
            DAC ORIGIN
CHEX,       LAC WORD1
            TAD WORD2
            TAD CHKSUM
            AND (7777)
            DAC CHKSUM
            JMP GO
DEP,        DAC I ORIGIN
            ISZ ORIGIN
            JMP CHEX
/
/           ASSEMBLE WORD AND SET LINK APPROPRIATELY
/
ASSEMB,     0
            LAC WORD1
            CLL!RTL
            RTL
            RTL
            TAD WORD2
            AND (7777)
            DAC TEM1
            LAC WORD1
            AND (100)
            SZA!CLA!CLL
            CML
            LAC TEM1
            JMP I ASSEMB
MINUS,      777402
ERR1,       0
            JMS ERROR
            1
            JMP I ERR1
```

T2

```
/
/    INITIALIZE LOW CORE
/
SETLOW,      0
             DZM PTR
             LAC (-10000+1
             DAC CTR       /INITIALIZE COUNTER
             DZM I PTR     /ZERO WORD
             ISZ PTR       /MOVE POINTER
             ISZ CTR       /CHECK COUNT
             JMP .-3       /DO MORE
             JMP I SETLOW              /ALL DONE
/
/    ERROR PRINT OUT
/
ERROR,       0
             DAC SAV1      /SAVE ACCUMULATOR
             JMS TCR       /START NEW LINE
             LAW 305       /TYPE AN E
             JMS PRINT
             LAC I ERROR
             ISZ ERROR
             JMS OCT       /AND ERROR NUMBER
             JMS TSP       /FOLLOWED BY
             LAC SAV1
             JMS OCT       /ACCUMULATION CONTENTS
             JMS TCR       /AND GO TO NEXT LINE BEFORE
             JMP I ERROR /EXITING
/
/    BASIC IO ROUTINES
/
/
/    PRINT ROUTINE
/
PRINT,       0
             TLS
             TSF
             JMP .-1
             TCF
             CLA
             JMP I PRINT
/
/    CARRIAGE RETURN AND LINE FEED
/
TCR,         0
             LAW 215       /CARRIAGE RETURN
             JMS PRINT
             LAW 212       /LINE FEED
             JMS PRINT
             JMP I TCR
/
/    TYPE A SPACE
/
TSP,         0
             LAW 200
             JMS PRINT
             JMP I TSP
```

T3

```
/
/    OCTAL TYPEOUT OF THE 18 BIT NUMBERIN AC
/
OCT,        0
            DAC SAV2      /SAVE NUMBER
            LAC (-20)     /SET TO LEADING BLANKS
            DAC LDZRO
            LAC (-6+1     /SET DIGIT COUNTER
            DAC CTR
L1,         LAC SAV2      /GET NEXT DIGIT
            RAL
            RTL
            DAC SAV2      /SAVE THE REST
            RAL
            AND (7        /MASK OFF THIS ONE
            SZA
            DZM LDZRO     /NOT A ZERO
            ADD LDZRO
            ADD (260      /CONVERT TO ASCII
            JMS PRINT
            ISZ CTR       /DIGITS ALL DONE YET
            JMP L1        /NO
            CLA           /YES, LEAVE
            JMP I OCT
```

T4

```
/
/   TYPES OUT COMMENTS,  LETTERS PACKED TWO TO A WORD
/
CMNT,       0
            LAC I CMNT
            ISZ CMNT
            DAC PTR
LOOP3,      LAC I PTR
            RTR
            RTR
            RTR
            RTR
            RAR
            AND (377
            SNA
            JMP I CMNT
            JMS PRINT
            LAC I PTR
            AND (377
            SNA
            JMP I CMNT
            JMS PRINT
            ISZ PTR
            JMP LOOP3
```

```
/    ALLOW THE OPERATOR TO TYPE A LABEL
/
LABEL,      0
            JMS RDTTY
            DAC C1
            JMS PRINT
            LAC (375
            SAD C1
            JMP I LABEL
            JMP LABEL+1
/
/    PROGRAM TO TABULATE THE INDICATED NUMBER OF SPACES
/
TAB,        0
            LAC I TAB
            ISZ TAB
            CMA
            DAC CTRR
            ISZ CTRR
            JMP .+2
            JMP I TAB
            JMS TSP
            JMP .-4
/
/    READ IN FROM TELETYPE
/
RDTTY,      0
            KSF
            JMP .-1
            KRB
            JMP I RDTTY
/
/    PDP-8 SIMULATION
/
SIMUL,      0
            LAC (JMP INTRPT
            DAC 1
            JMS CYCLE
            JMS RUN
            LAC PIE
            SZA
            ION
            NOP
            NOP
            IOF
            JMP SIMUL+1
/
/    PDP-8 INTERRUPT CONTROL
/
INT8,       LAC PC
            DAC MB
            DZM MA
            JMS STORE
            LAC (1
            DAC PC
            DZM PIE
            JMP SIMUL+1
/
/    PDP-7 INTERRUPT CONTROL
/
INTRPT,     JMP INT8
```

T5 1 of 2

T5 2of2

```
/
/   RESET SYSTEM AND RESTART
/
RESET,      HLT
            JMS TCR
            JMS SETLOW   /INITIALIZE LOW CORE
            JMS LOADR    /NOW LOAD THE TAPE
            LAC Ø        /SET UP INTERRUPTS
            DAC ADDØ
            LAC 1
            DAC ADD1
            LAS
            AND (20000)
            SZA
            JMP .-3
            JMS SIMUL    /NOW START THE PDP-8 AND DISPLAY SIMULATION
            NOP          /ALL DONE WITH THAT
            JMS TCR
            JMS TCR
            JMS TCR
            JMS TCR
            NOP
            JMP RESET    /GO GET READY TO DO IT AGAIN
```

```
/
/   CHECK SR STATUS
/
CYCLE,      0
            OAS!CLA
            RAL
            SZL
            JMP I CYCLE /RUN - GO EXECUTE
            RAL
            SZL
            JMP LDADD   /LOAD ADDRESS
            RAL
            SZL
            JMP EXAM    /EXAMINE SUCCESSIVE LOCATIONS
            RAL
            SZL
HERE,       JMP PATCH   /PDP-8 START-CLEAR FLAGS AND ZERO AC
            RAL
            SNL
            JMP .+3
            DZM BEGG
            JMP RESET+1 /LOAD A NEW TAPE AND START OVER
            RAL
            SNL
            JMP CYCLE+1 /NOTHING SET, WAIT TILL IT IS
            JMP DPOSIT
LDADD,      OAS!CLA
            AND (7777
            DAC PC
            JMP CYCLE+1
EXAM,       LAC PC
            DAC MA
            JMS INCPC
            JMS CORE
            LAW 252
            JMS PRINT
            LAC MA
            JMS OCT
            JMS TAB
            3
            LAC MB
            JMS OCT
            JMS TCR
            JMP CYCLE+1
DPOSIT,     CLA!OAS    /LOAD AC WITH SWITCH REGISTER
            AND (7777)
            DAC MB
            LAC PC
            DAC MA
            JMS STORE
            ISZ PC
            LAS
            AND (10000)
            SZA
            JMP .-3
            JMP CYCLE+1
PATCH,      DZM AC
            CAF
            LAC (JMP I CYCLE)
            DAC HERE
```

```
            LAC (JMP PATCH2)
            DAC HERE+1
            JMP I CYCLE
PATCH2,     LAC (JMP PATCH)
            DAC HERE
            LAC (RAL)
            DAC HERE+1
            JMP CYCLE+1
/
/   CORE CYCLE
/
CORE,       0
            LAC MA
            SAD (0
            JMP GET0
            SAD (1
            JMP GET1
            LAC I MA
ENDCOR,     DAC MB
            JMP I CORE
GET0,       LAC ADD0
            JMP ENDCOR
GET1,       LAC ADD1
            JMP ENDCOR
/
/  STORE CYCLE
/
STORE,      0
            LAC MA
            SAD (0
            JMP PUT0
            SAD (1
            JMP PUT1
            LAC MB
            DAC I MA
            JMP I STORE
PUT0,       LAC MB
            DAC ADD0
            JMP I STORE
PUT1,       LAC MB
            DAC ADD1
            JMP I STORE
/
/   INCREMENT PROGRAM COUNTER
/
INCPC,      0
            DAC ORIGIN  /SAVE ACCUMULATOR FOR LATER RESTORATION
            LAC PC
            TAD (1
            AND (7777
            DAC PC
            LAC ORIGIN
            JMP I INCPC
/
/   PDP-8 OPERATION DUISPATCHER
/
OPS8,       AND8
            TAD8
            ISZ8
            DCA8
```

```
            JMS8
            JMP8
            IOT8
            OPR8
/
/   RUN FOR ONE COMPLETE CYCLE
/
RUN,        0
            LAC PC
            DAC MA
            JMS INCPC
            JMS CORE    /GET INSTUCTION
            LAC MB      /DECODE IT
            RTR
            RTR
            RTR
            RTR
            RAR
            AND (7
            TAD (OPS8
            TAD (JMP I 0)           /BY COMPUTING A JUMP
            DAC .+1
            HLT         /HALT IF IT DIDN'T WORK.
            HLT
/
/   MEMORY REFERENCE, GET EFFECTIVE ADDRESS
/
EFFADD,     0
            LAC MA
            DAC TEM8
            LAC MB
            AND (177
            DAC MA
            LAC MB
            AND (200
            SNA
            JMP .+5
            LAC TEM8
            AND (7600
            TAD MA
            DAC MA
            LAC MB
            AND (400)   /CHECK FOR INDIRECT ADDRESSING
            SNA
            JMP I EFFADD
            LAC MA
            AND (0010)  /CHECK FOR AUTO-INDEX REGISTERS
            SNA
            JMP NONAUT
            LAC MA
            TAD (-20)
            SMA
            JMP NONAUT
AUTIND,     JMS CORE
            LAC MB
            TAD (1)
            AND (7777)
            DAC MB
            JMS STORE
            LAC MB
```

```
                DAC MA
                JMP I EFFADD
NONAUT,         JMS CORE
                LAC MB
                DAC MA
                JMP I EFFADD
/
/   LOGICAL AND
/
AND8,           JMS EFFADD
                JMS CORE
                LAC MB
                AND AC
                AND (7777
                DAC AC
                JMP I RUN
/

/   TWO'S COMPLEMENT ADD
/
TAD8,           JMS EFFADD
                JMS CORE
                LAC MB
                TAD AC
                DAC TEM8
                AND (7777
                DAC AC
                LAC TEM8
                AND (10000
                SZA
                JMS COML8
                JMP I RUN
/
/   COMPLEMENT THE LINK
/
COML8,          0
                LAC LINK
                CMA
                AND (1
                DAC LINK
                JMP I COML8
/
/   INDEX
/
ISZ8,           JMS EFFADD
                JMS CORE
                ISZ MB
                LAC MB
                AND (7777
                DAC MB
                JMS STORE
                LAC MB
                SZA
                JMP I RUN
                JMS INCPC
                JMP I RUN
/
/   DEPOSIT
/
DCA8,           JMS EFFADD
                LAC AC
```

```
                DAC MB
                JMS STORE
                DZM AC
                JMP I RUN
/
/   JMS
/
JMS8,           JMS EFFADD
                LAC PC
                DAC I MA
                LAC MA
                DAC PC
                JMS INCPC
LP3,            OAS!CLA
                AND (10000
                SNA
                JMP I RUN
                JMS PANEL
                JMP I RUN
/
/   JMP
/
JMP8,           JMS EFFADD
                LAC MA
                DAC PC
                JMP LP3
/
/   PANEL DUMP
/
PANEL,          0
                JMS CMNT
                PCOUT
                LAC PC
                JMS OCT
                JMS CMNT
                MAOUT
                LAC MA
                JMS OCT
                JMS CMNT
                MBOUT
                LAC MB
                JMS OCT
                JMS CMNT
                ACOUT
                LAC AC
                JMS OCT
                JMS TCR
                JMP I PANEL
PCOUT,          252320
                303240
                275000
MAOUT,          240240
                240315
                301240
                275000
MBOUT,          240240
                240315
                302240
                275000
ACOUT,          240240
                240301
                303240
                275000
```

T7 1of5

```
/
/   IOT'S
/
IOT8,       LAC MB
            AND (7
            DAC IOP
            LAC MB
            RAL
            RTL
            AND (7700
            TAD IOP
            TAD (700000
            SAD (700001)
            JMP ION8
            SAD (700002)
            JMP IOF8
            SAD (700302)
            LAC (JMS .KCC)
            SAD (700304)
            LAC (KRB)
            SAD (700306)
            LAC (KRB)
            SAD (700404)
            LAC (TLS)
            DAC .+2
            LAC AC
            HLT

            JMP .+2
            JMS INCPC
            DAC AC
            NOP          /PREVIOUSLY JMS CHKIO FOR DISPLAY
            JMP I RUN

ION8,       LAC (1)
            DAC PIE
            JMP RUN+1

IOF8,       DZM PIE
            JMP I RUN

.KCC,       0
            KRB
            CLA
            JMP I .KCC
```

T7 2 of 5

```
/
/   OPERATES
/
OPR8,       LAC (200
            DAC MSK
            LAC MB
            AND (400
            SZA
            JMP GRP2
            JMS MSKMB
            DZM AC
            JMS MSKMB
            DZM LINK
            JMS MSKMB
            JMS C0MAC8
            JMS MSKMB
            JMS COML8
            JMS MSKMB
            JMS SHFTR
            JMS MSKMB
            JMS SHFTL
            LAC MB
            AND (1
            SNA
            JMP I RUN
            ISZ AC
            LAC AC
            AND (10000
            SNA
            JMP I RUN
            LAC AC
            AND (7777
            DAC AC
            JMS C0ML8
            JMP I RUN
C0MAC8,     0
            LAC AC
            CMA
            AND (7777
            DAC AC
            JMP I COMAC8
SHFTR,      0
            JMS RTSHFT
            LAC MB
            AND (2
            SZA
            JMS RTSHFT
            JMP I SHFTR
RTSHFT,     0
            LAC LINK
            RTL!CLL
            RTL
```

```
                RTL
                RTL
                RTL
                RTL
                TAD AC
                RAR
                AND (7777
                DAC AC
                CLA
                SZL
                TAD (1
                DAC LINK
                JMP I RTSHFT
SHFTL,          0
                JMS LFTSFT
                LAC MB
                AND (2
                SZA
                JMS LFTSFT
                JMP I SHFTL
LFTSFT,         0
                LAC AC
                RAL!CLL
                TAD LINK
                DAC TEM8
                AND (7777
                DAC AC
                LAC TEM8
                AND (10000
                SZA!CLA
                TAD (1
                DAC LINK
                JMP I LFTSFT
MSKMB,          0
                DZM TEM8
                LAC MB
                AND MSK
                SZA
                ISZ TEM8
                LAC MSK
                RAR
                DAC MSK
                LAC TEM8
                SNA
                ISZ MSKMB
                JMP I MSKMB
EAE8,           NOP
                JMP I RUN
GRP2,           LAC MB
                AND (1)
                SZA
                JMP EAE8
                LAC MB
                AND (100)    /TEST BIT 5 (SMA)
                SNA
                JMP SGRP
                LAC (1)
                DAC TST1
                LAC AC
                RTL
```

T7 4of5

```
        RTL
        RTL
        SMA
SGRP,   DZM TST1
        LAC MB
        AND (40)      /TEST BIT 6 (SZA)
        SNA
        JMP .+5
        LAC (1)
        DAC TST2
        LAC AC
        SZA
        DZM TST2
        LAC MB
        AND (20)      /TEST BIT 7 (SNL)
        SNA
        JMP .+5
        LAC (1)
        DAC TST3
        LAC LINK
        SNA
        DZM TST3
        LAC TST1
        TAD TST2
        TAD TST3
        DAC SKIP
        LAC MB
        AND (10)      /CHECK THE REVERSE BIT
        SNA
        JMP .+6
        LAC SKIP
        SZA
        LAW 17777     /-1 IN AC
        TAD (1)
        DAC SKIP
        LAC SKIP
        SZA
        JMS ADDPC
        LAC MB
        AND (200)
        SZA
        DZM AC        /CLEAR AC
        LAC MB
        AND (4)
        SZA
        JMS ORSR      /LOAD SWITCH REG FROM AC
        LAC MB
        AND (2)
        SZA
        JMS HALTS
        JMP I RUN
```

T7 5 of 5

```
HALTS,      0
            JMS PANEL
            OAS!CLA
            AND (440000)
            SNA
            JMP I HALTS
            JMP .-4
ORAC,       0
            LAC TEM8
            CMA
            DAC TEM8
            LAC AC
            CMA
            AND TEM8
            CMA
            AND (7777)
            DAC AC
            JMP I ORAC
```