INTERSIL

# IM6100 OPERATOR CONSOLE

INTERSIL CMOS/LSI

APPLICATION BULLETIN
M006

## INTRODUCTION

The operator console for the IM6100 Prototyping System, 6900-CONTRL, consists of an array of console switches and indicators to facilitate computer operation and maintenance. The programmer, or operator, may start and stop program execution, examine and modify the contents of main memory, modify and display internal processor information, select various modes of microprocessor operation, manually load and execute short machine language programs or load and execute programs via the Teletype or high speed tape reader. Since the microprocessor register and internal control signals are not available externally, the modification and display of internal processor information must be done under program control.

In this note, we discuss the architectural features that are incorporated in the IM6100 design to facilitate control panel operation, as well as the specific implementation of an operator console.

## MICROPROCESSOR—PANEL COMMUNICATION

The panel must communicate with the microprocessor. Since the panel operation is inherently asynchronous in nature, the microprocessor must be "interrupted" to carry out the panel operations. However, there are certain drawbacks associated with the panel communicating with the CPU through the normal interrupt channel. The conventional interrupt system is inactive when the CPU is halted. Even when the CPU is running, the interrupt system is not always enabled. The console interrupt must be recognized by the CPU even if the CPU is in the halt state. The CPU must go into the run state to execute the panel routine and restore the original run/hlt state when exiting from the panel program.

An interrupt request is granted by the CPU executing a "hardwired" subroutine call of the interrupt service routine. This capability is built into the CPU state sequencer. Since the control panel routines are normally stored in read-only memory, the entry point of the subroutine must be guaranteed to be a "ROM location".

It is clear that the panel-CPU communication link must satisfy certain special requirements. These requirements can best be met by the provision of a dedicated panel interrupt request (CPREQ) line to the CPU which is treated differently from the conventional device interrupt request (INTREQ) line.

## "TRANSPARENT" PANEL MEMORY

The panel routines require memory. If part of the main memory is used for the panel software, obviously that portion of the memory cannot be used by the user. It is desirable to make provisions for a control panel memory which is distinct from the main memory. This is an important consideration since the final version of the user system most probably will not have a full fledged control panel and the system designer would like to use the entire capacity of the main memory for the specific system application. One could then design a stand-alone completely independent and portable panel with the routines residing in the panel memory. The panel program may share the same addressing space as the main program and the console operations will be "transparent" to the system user.

In the IM6100, a panel request is acknowledged if the CPREQ line is active low. The panel request is granted irrespective of the run/hlt state of the CPU. The CPU is temporarily put in the run state for the duration of the panel routine and the CPU reverts back to its original state after executing the panel program. CPREQ also bypasses the interrupt enable system. An internal flip-flop (CNTRL FF) is set when the CPREQ is acknowledged which prevents further CPREQs from being granted.

As long as CNTRL FF is set, CPSEL, control panel memory select, becomes active low for memory references instead of MEMSEL, main memory select. CPSEL is used to distinguish between main memory and panel memory. The current contents of the program counter, PC, are deposited in location $0000_8$ of panel memory. The PC is then set to $7777_8$ and the CPU executes the panel program starting at location $7777_8$. The location $7777_8$ contains a jump instruction to the entry point of the panel routine. The panel memory system is expected to be organized with read-write memory in page 0 and read-only memory in the higher pages.

While the CPU is in the panel mode, MEMSEL becomes active during the final phase of "indirectly addressed" AND, TAD, ISZ, and DCA instructions. In indirectly addressed instructions, the address code refers to a memory location in which the 12-bit address of the operand is stored. The instructions are fetched from the panel memory. The indirect operand address pointer also comes from the panel memory. However, the final effective address refers to a location in the main memory. A main memory location may be examined by a TAD I instruction and modified by a DCA I instruction. Every location in the main memory is now accessible to the panel routine.

The CPU control structure is implemented by a microprogram stored in a programmable logic array (PLA). After the complete execution of an instruction, the sequencer scans power-on RESET, CPREQ, external RUN/HLT, Direct Memory Access Request (DMAREQ) and INTREQ, respectively. The request with the highest priority is granted. If there is no active request, the CPU fetches the next instruction in sequence. The CNTRL FF being set, masks off all requests except power-on RESET. The CNTRL FF also inhibits any program modification to the interrupt system. One of the interrupt system modifying instructions is then used to reset the CNTRL FF. When the CPU executes an ION, Interrupt On, instruction in the panel mode, it does not affect the interrupt system; instead, the CNTRL FF is reset after executing the next sequential instruction. One may exit from the panel routine by executing the following sequence: ION; JMP I $0000_8$. Location $0000_8$ of the panel memory contains either the original return address when the panel routine was entered or it may be a new "starting address" defined by the operator, for example, by activating the "LOAD PC switch". The CPU reverts to its original run/hlt state after executing the JMP I instruction, following an ION.

The control panel requests are usually triggered when the console function switches, LOAD PC, LOAD FLAGS, DEP MEM, EXAM, BIN BOOT, or USER FN, are activated. However, these function switches generate a CPREQ only if the CPU is in the halt state in 6900-CONTRL as shown in Figure 1. This is to prevent activation of these switches by accident while the user program is running. If the CPU information must be displayed in "real time", the CPREQ is generated by a timer at fixed intervals, for example, 30 times a second. The 6900-CONTRL timer requests are inhibited when the microprocessor has already granted an interrupt or DMA request. The

INTGNT, Device Interrupt Grant, and DMAGNT, DMA Grant, signals are used for this purpose. When a device interrupt is in progress, granting a control panel interrupt will interfere with the vectored interrupt priority scheme implemented in the Parallel Interface Element, PIE—IM6101. A control panel interrupt is not advisable when DMA operations are involved since it adversely affects the DMA response time as well as the DMA transfer rate.

## 6900-CONTRL

### SWITCH REGISTER

The switch register consists of 12 switches that enable the operator to load the program counter with a 12-bit address, to deposit a 12-bit data word in a selected memory location or to load the extended address bits if more than 4K of memory is used. The switch register can also be read by the user program with the OSR, OR the switch register, operate instruction.

### FUNCTION CONTROLS

**LOAD PC *Push Button***

The LOAD PC push button is used to load the Program Counter with a 12-bit address specified by the switch register. The PC information is displayed in the PC display. If the Rotary Switch is set to MD, then the DISPLAY will show the contents of the memory location immediately preceeding the location pointed to by the "new" PC.

**LOAD FLAGS *Push Button***

This push button is used to load switch register bits 6-11 into the Instruction Field, IF 0-2, and Data Field, DF 0-2, Registers if more than 4K of memory is used. If the display select rotary switch is set to FLAGS, the FLAG information is shown in the DISPLAY. If extended memory control is not used, the IF and DF information is not valid. Switch register bit 0 is loaded into the LINK.

**DEP MEM *Push Button***

If the operator wants to deposit data in a particular location of a specific memory field, the address must be loaded into the Program Counter by LOAD PC. Then the switch register switches must be set to correspond to the data to be deposited. The data is then deposited into the memory location specified by the content of the PC by activating DEP MEM. The PC is automatically incremented by one after the data is deposited to set up the next sequential memory address.

If the display select rotary switch was set to MD, the memory data, which was just deposited, is shown in the DISPLAY.

**EXAM *Push Button***

If the display select rotary switch is set to MD, the operator can cause the contents of the selected memory location to be brought from memory and displayed. In the memory examination mode, the Program Counter is incremented by one after the information is displayed to set up the next sequential memory address. Therefore, to modify the data in an examined location, the switch register and LOAD PC button must be used to return to the correct address.

If the Display Select rotary switch is set to AC, MQ or FLAGS, the DISPLAY will show the status of the Accumulator, MQ Register or extended address bits when the Exam button is activated. The PC is not affected.

**BIN BOOT *Push Button***

The BIN BOOT push button activates the bootstrap loader to read and store information contained in binary coded tapes, using ASR-33 Teletype paper tape reader or a high speed tape reader. For the bootstrap loader to function properly, the system must have a DEC PDP-8/E*compatible Teletype or high speed paper tape reader interface. The 6902-CPUTTY card provides for a DEC PDP-8/E*compatible ASR-33 Teletype Interface. Refer to Appendix I for a detailed description of the BIN BOOT operation.

**USER FN *Push Button***

This button may be activated to implement user defined routines. Refer to Appendix II for the PAL III listing of the control panel program. The panel memory is organized as 256 × 12 PROM and 16 × 12 RAM.

### MODE CONTROLS

**HLT *Switch* and CONT *Push Button***

The IM6100 can be stopped manually by flipping the HLT switch down. With the HLT switch up, the CONT push button is used to make the microprocessor execute programs starting at the location pointed to by the PC.

If the CONT push button is activated with the HLT switch down, the microprocessor will execute a single instruction pointed to by the PC and then halt. This single instruction capability is an extremely useful feature for debugging user programs.

**RESET *Push Button***

This button generates an initialize signal which is functionally equivalent to executing a CLEAR ALL FLAG, CAF $6007_8$, instruction. In the IM6100 prototyping system, they are wire-ANDed to generate the initialize signal.

The RESET and CAF will clear the AC, LINK, and all peripheral flags. The interrupt system is disabled. Refer to the application note entitled "Teletype Interface for the IM6100 CMOS Microprocessor" for the status of the Teletype flags on initialize.

The RESET differs from CAF in one important respect. The RESET will initialize all internal CPU flags, set the PC equal to $7777_8$ and then halt. Activating the RUN button will cause the CPU to execute the routine starting at location $7777_8$. RESET has the highest priority and pulsing the RESET while the CPU is executing a program in the panel memory will cause it to exit and go to the main memory location $7777_8$ and halt.

**FREE RUN *Switch* and SINGLE CLOCK *Push Button***

Since the IM6100 design is completely static, the CPU may be single clocked. If the FREE RUN switch is up, the microprocessor will run on the crystal oscillator on the 6902-CPUTTY card. The microprocessor can be single clocked by putting the FREE RUN switch down and then activating the SINGLE CLOCK. Two clock pulses are necessary for every state transition of the microprocessor since the microprocessor divides the input frequency by two for internal operation. Gating is provided on the CPUTTY board to ensure integral clocking and the SINGLE CLOCK is debounced to prevent false triggering in the single clock mode.

*Trademark of Digital Equipment Corporation, Maynard, MA.

**30Hz *Switch***

If the 30Hz switch is up, the panel will generate CPREQ's at 30Hz. While the CPU is executing the user program, this feature may be used to display processor state information in "real time". However, one must be careful to include these "window" timings to the actual execution time of the user task to calculate the overall time if timing considerations are critical. One must, therefore, be cautious while using the DMA and INT features of the IM6100 since the "real time" display would adversely affect the worst case response time and throughput of the microprocessor.

This is a very convenient feature in the "single instruction mode" since the processor state information will be "immediately" available in the displays after executing the instruction. Also, while 30Hz is active, it is not necessary to press EXAM to look at new state information by changing the display select rotary switch. The 30Hz program overhead is less than one percent of the CPU active time.

**3K ENABLE and 4K ENABLE *Switches***

The 4K EN switch, when it is down, disables writes into the 6901 M4KX12 CMOS memory plane. Since the state of the IM6100 is unknown as it is powered up, it is recommended that the memory is write protected when applying power to the system.

The 3K EN switch, when it is down, reconfigures the 4K × 12 memory into 1K × 12 RAM (locations $0000\text{-}1777_8$) and 3K × 12 ROM (locations $2000\text{-}7777_8$) to simulate a RAM-ROM system for user prototyping.

## DISPLAY

**Program Counter (PC)**

The PC displays the contents of the Program Counter in the 30Hz mode. If the 30Hz option is inactive, the operator must press the EXAM push button with the ROT SW set to AC, MQ or FLAGS. If the ROT SW is set to MD, the PC will be updated by one to point to the next sequential address. When operating on memory while the machine is halted, the PC displays the memory address.

In the single clock mode, the PC displays the instruction address.

**Display**

In the Free Run mode, the Display contains information which is selected by the Display Rotary Switch, AC, MQ, FLAGS or MD.

In the single clock mode, the Display follows the DX bus. Note that the Display information is valid only when DX contains valid information, i.e., the Display is undefined when the DX bus is tristated.

**RUN, XTA and IFETCH**

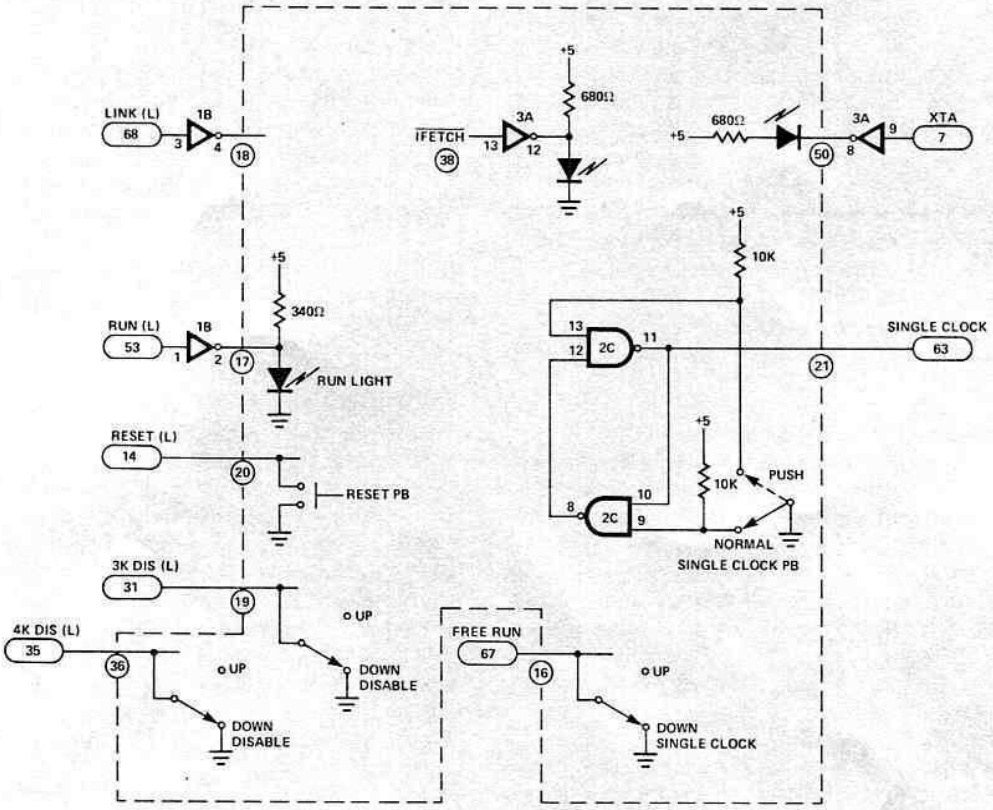
These discrete LEDs show the active state of the RUN, XTA and IFETCH lines of the IM6100.

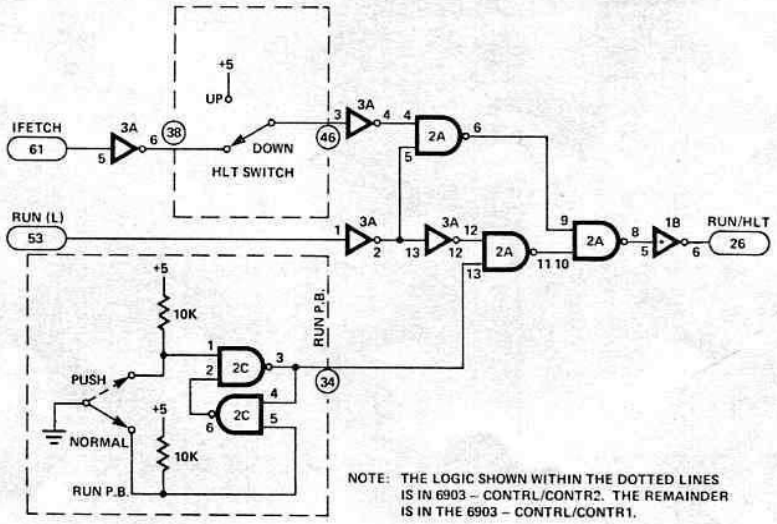
## CONCLUSION

The IM6100 is an excellent example of what can be provided in a microprocessor to facilitate panel functions. Undoubtedly, there are other ways of implementing these features. The IM6100 approach is interesting since it is extremely simple and straightforward. It requires only two additional pins and a minimum of internal logic, one additional line in the internal PLA. This approach does not require any new instructions and does not change the processor state. There are a number of panel options which can greatly increase the usefulness and flexibility of the microprocessor based system. For example, the panel can be used as a maintenance tool by storing test and exercise programs in the panel memory. Any additional feature is incorporated just by increasing the size of the panel memory to handle more software. The 6903-CONTRL may, therefore, be looked at as a completely independent, self-contained, standalone device which can be plugged into the system whenever panel functions are needed, and disconnected without disturbing any part of the user system.


Picture hb9aik

CONTROL PANEL




NOTE: THE LOGIC SHOWN WITHIN THE DOTTED LINES IS IN 6903 – CONTRL/CONTR2. THE REMAINDER IS IN THE 6903 – CONTRL/CONTR1.

6903–CONTRL LOGIC

# APPENDIX I

## INTRODUCTION

The BIN BOOT accepts tapes prepared with Digital Equipment Corporation PAL III, PAL D, PAL 8 or MACRO 8 assemblers and Intersil's FORTRAN/PAL III Cross Assembler. Diagnostic messages may be included on tapes. The BIN BOOT is functionally identical to the DEC BINARY LOADER described in the DEC Utility Routine Manual, DEC-81-RZPA-D, and the "Introduction to Programming" handbook. However, unlike the DEC BIN LOADER, the BIN BOOT does not use any locations in the main memory and hence all of main memory is available for user programs.

## EXTERNAL TAPE FORMAT

Tapes to be read by the BIN BOOT must be in binary-coded format and have about one foot of leader-trailer code (any code with channel 8 punched; preferably code 200). The first two characters represent the initial address or origin. The initial character of the origin has no punch in channel 8, while channel 7 is punched. The second character designating the origin has no punches in either channel 8 or 7. Data characters have no punches in channel 8 ro 7. A 12-bit binary word is represented by two 6-bit characters on the tape in channels 6 through 1, channel 6 of the initial character being the most significant bit. The data characters are loaded into sequential locations following the origin set up. If more than 4K of memory is used, the assembler outputs a "field-setting" command of the form 11 XXX 000 (channel 8-1) to indicate the memory field into which the following data is to be loaded. If for example, XXX were 101, all data following the field designator should be loaded into memory field five. Trailer tape is similar to the leader. A concluding 2-character group before the trailer represents the checksum and has no punches in channel 8 or 7.

## CHECKSUM

When any of the PDP-8 assemblers are used to produce a binary tape, a checksum is automatically punched at the end of the binary tape. This is the sum of all data on the tape including the origin but excluding diagnostic messages, leader/trailer code and field settings. The sum is accumulated character by character and not word by word. Carry out of the Accumulator, AC, is ignored.

If the checksum accumulated while using the BIN BOOT does not agree with the last two characters on tape (i.e., the checksum on the tape calculated and placed there by the assembler), an error in loading has occurred.

The microprocessor will halt after the tape has been loaded and the AC will be unequal to zero if a checksum error has occurred.

If the tape was started before the leader, the microprocessor will halt at the leader with AC equal to 7600 or 0000, depending on the number of blank characters read before the microprocessor halts.

## MEMORY EXTENSION USAGE

The BIN BOOT may be used to load the binary tape into any valid memory field. If the memory extension controller is not used, the extended memory field instructions of the BIN BOOT are treated as "don't cares".

## BIN BOOT PROGRAM

Refer to Appendix II for the PAL III listing of the BIN BOOT program.

The Program proceeds as follows: The incoming character is tested to see if it is a "rub-out" (all eight tape channels punched). If this is the case, all subsequent information coming from the reader is ignored until another rub-out is detected. This is the mechanism by which the assembler diagnostic messages are detected. They are preceded and followed by a single rub-out character. Within the diagnostic message any character is valid except, of course, a single rub-out character which would prematurely conclude the diagnostic message. Note that two consecutive rub outs within the diagnostic message would, in effect, be ignored.

Next the character is tested to see if it is leader or field setting. Leader information is ignored. The "change data field" routine is executed if the character is in the field format.

If the character is not part of the diagnostic message, leader or field setting, then it is part of the origin address, contains part of the data word and is part of the checksum and the appropriate course is followed. The BIN BOOT always "looks ahead" by one character to see if trailer follows the character just read. If it does, then the two characters read before the trailer is the checksum.

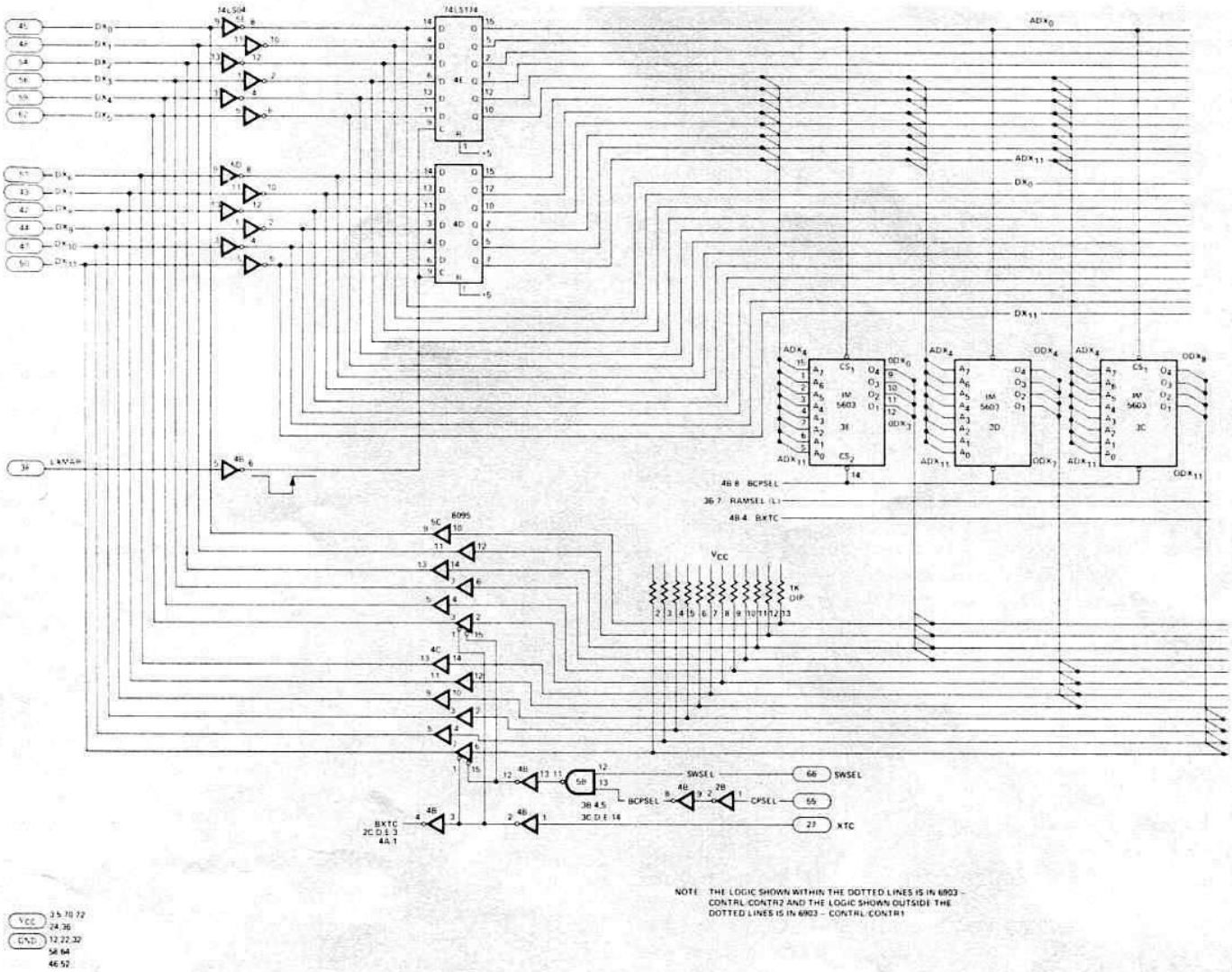
## BIN BOOT LOADING PROCEDURE

1. *Halt the machine.*
2. Place the tape to be loaded (tape must be in binary format) in the Teletype or High Speed reader. Make sure that the reader is "on line". The leader portion of the tape must be over the read head.
3. In the Data Field register, place the field into which the program is to be loaded with the LD FLAGS push button and the switch register.
4. When using the high speed reader switch register bit 0 must be 0. *When using the TTY reader, the switch register bit 0 must be 1.*
5. Press BIN BOOT.
6. The CPU halts after reading in the tape.
7. Examine Accumulator by pressing EXAM with the rotary switch set to AC. If the switch was set to AC while the tape was being read in, DISPLAY will automatically show the contents of AC. The AC must be equal to 0000, if the tape was properly read in. While the tape is being read, the PC shows the current origin and the Display shows the data being loaded into memory.

## STARTING THE PROGRAM

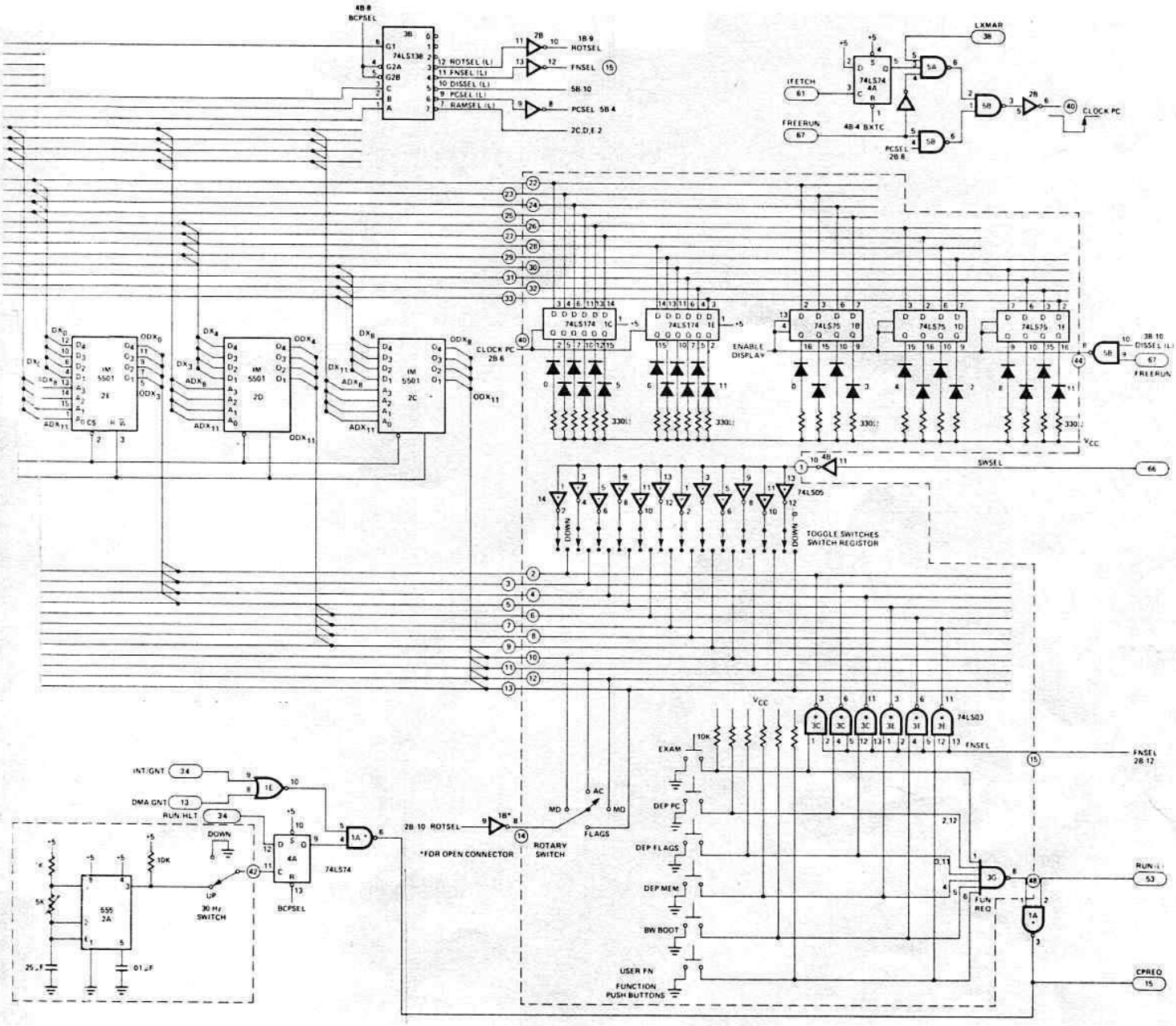
1. Press RESET to initialize the processor system, if required.
2. After the program has been successfully loaded, place starting address of the program in Switch register. Press LOAD PC.
3. Place the field where program exists in the Instruction Field register with the Switch register and LOAD FLAGS.
4. Bring the Halt switch to the run state (up).
5. Press CONTINUE.

INTERSIL




NOTE: THE LOGIC SHOWN WITHIN THE DOTTED LINES IS IN 6903 – CONTRL/CONTR2 AND THE LOGIC SHOWN OUTSIDE THE DOTTED LINES IS IN 6903 – CONTRL/CONTR1

6903–CONTRL LOGIC




ADDRESSES

| | | | | |
|---|---|---|---|---|
| 1XX | XX•• | ••• | ••• | PROM |
| 0XX | XX0 | 00• | ••• | RAM |
| 0XX | XX0 | 01X | XXX | PC |
| 0XX | XX0 | 10X | XXX | DISPLAY |
| 0XX | XX1 | 00X | XXX | FN |
| 0XX | XX1 | 00X | XXX | ROT |

# APPENDIX II

Control Panel Program Listing:

```
                /INTERSIL 6903-CONTRL ROUTINE
                /PROM LOCATIONS 7400-7777.  THE PROM ADDRESS IS
                        /COMPLIMENTED-  SEE LOGIC DIAGRAM
                /RAM LOCATIONS 0000-0017
                /PC DISPLAY IN 0020
                /DISPLAY IN 0040
                /FUNCTION SWITCHES IN 0060
                        /EXAM-DX0 DEP PC -DX1 DEP FLAGS- DX2
                        /DEP MEM- DX3 BINBOOT- DX4 USER -DX5
                /ROTARY SWITCH IN 0100
                        /MD-DX8 AC-DX9 MQ-DX10 FLAGS-DX11

                /RAM 0004-0017 MAY BE USED FOR USER FN

                /48 PROM LOCATIONS ARE AVAILABLE TO
                 /IMPLEMENT USER FUNCTION. CURRENTLY
                 /THE  "USER FN " DECREMENTS PC BY 1 TO
                 /RESTORE PC FOR A "EXAM AND MODIFY"
                 /MEMORY FUNCTION.

                / THE TIME REQUIRED TO SERVICE A 30HZ
                 /REQUEST IS 200 MICROSECONDS AT 4MHZ.
                 /FOR CP ROUTINES TO FUNCTION PROPERLY
                 /WHEN THE HLT SW IS DOWN, AN EVEN NUMBER
                 /OF INSTRUCTIONS MUST BE EXECUTED IN
                 /THE CONTROL PANEL PROGRAM. "NOP" IS
                 /USED IN CERTAIN ROUTINES TO ENSURE THIS.

                /THE PROGRAM DEBOUNCES A FN SW CLOSURE AND
                 /RELEASE BY 63.5 MS AT 4 MHZ

                        PCLEDS=0020
                       DISLEDS=0040
                          FNSW=0060
                         ROTSW=0100

                        *0000
                /RAM LOCATIONS

0000  0000  PC,     0000
0001  0000  AC,     0000
0002  0000  FLAGS,  0000
0003  0000  THREE,  0000

0004  0000          0000
0005  0000  EXEC,   0000
0006  0000          0000

0007  0000  RDRSEL, 0000
0010  0000  BEGSW,  0000
0011  0000  RUBSW,  0000
0012  0000  RDRSW,  0000
0013  0000  CHAR,   0000
0014  0000  WORD1,  0000
0015  0000  WORD2,  0000
0016  0000  CHKSUM, 0000

                        /LOCATIONS 004-0017 AV TO USER

                /PROM PROGRAM

                        *7777
7777  5776          JMP I 7776          /CPREQ ENTRY
                        *7776
7776  7400          START               /CP ROUTINE ENTRY
                        *7400
7400  3001  START,  DCA AC

7401  6004          GTF
7402  3002          DCA FLAGS           /SAVE AC AND FLAGS

7403  3004          DCA EXEC-1      /INIT TO 0000 TO COUNT
                                      /DEBOUNCE DELAY

7404  1060          TAD FNSW
7405  0245          AND K7700
7406  3003          DCA THREE
7407  1003          TAD THREE
7410  7450          SNA
7411  5234          JMP HZ30                /CPREQ GENERATED BY
                                            /TIMER
7412  2004          ISZ EXEC-1              /63.5 MS DELAY
7413  5212          JMP .-1
7414  7000          NOP                     /SYNC

                                            /ACTIVE FNSW WILL BE 1
7415  7004          RAL
7416  7430          SZL
7417  5304          JMP EXAM
7420  7510          SPA
7421  5316          JMP DEPPC

7422  7006          RTL
7423  7430          SZL
7424  5633          JMP I XDEP
7425  7510          SPA
7426  5321          JMP DEPMEM

7427  7006          RTL
7430  7630          SZL CLA
7431  5725          JMP I BINBOOT
7432  5326          JMP USER
7433  7607  XDEP,   DEPFLAGS

7434  1000  HZ30,   TAD PC
7435  3020          DCA PCLEDS

7436  1100          TAD ROTSW
7437  7012          RTR
7440  7500          SMA
7441  5275          JMP FLDIS
7442  7420          SNL
7443  5300          JMP MQDIS

7444  7012          RTR
7445  7700  K7700,  SMA CLA
7446  5302          JMP ACDIS

7447  7000  MDDIS,  NOP     /FOR EVEN INSTRUCTION SYNC
7450  7240          CLA CMA
7451  1000          TAD PC
7452  3004          DCA EXEC-1
7453  1404          TAD I EXEC-1
7454  3040          DCA DISLEDS     /INSTR THAT WAS JUST EXECUTED
                                        /OR DATA THAT WAS JUST DEPOSITED

7455  1003  EXIT,   TAD THREE
7456  7650          SNA CLA
7457  5267          JMP .+10

7460  1060          TAD FNSW
7461  0245          AND K7700
7462  7640          SZA CLA
7463  5260          JMP .-3                 /IF FN SW SPREQ WAIT TIL
                                            /SW IS RELEASED
7464  3004          DCA EXEC-1
7465  2004          ISZ EXEC-1
7466  5265          JMP .-1                 /THEN DEBOUNCE FOR 63.5 MS

7467  1002          TAD FLAGS
7470  7004          RAL                     /RESTORE LINK
7471  7200          CLA
7472  1001          TAD AC          /RESTORE AC
7473  6001          ION             /RESET CP INT FF
7474  5400          JMP I 0000      /EXIT
```

```
7475  7300  FLDIS,  CLA CLL
7476  1002          TAD FLAGS
7477  5254          JMP EXIT-1

7500  7701  MQDIS,  CLA MQA
7501  5254          JMP EXIT-1

7502  1001  ACDIS,  TAD AC
7503  5254          JMP EXIT-1


7504  7300  EXAM,   CLA CLL
7505  1100          TAD ROTSW

7506  7012          RTR
7507  7012          RTR
7510  7620          SNL CLA
7511  7410          SKP
7512  5234          JMP HZ30          /EXIT FOR AC,MQ & FLAGS
7513  2000  ISZPC,  ISZ PC                    /MD EXAM, INCREMENT PC
7514  5234          JMP HZ30
7515  5234          JMP HZ30

7516  7604  DEPPC,  CLA OSR
7517  3000          DCA PC
7520  5234          JMP HZ30

7521  7000  DEPMEM, NOP
7522  7604          CLA OSR
7523  3400          DCA I PC
7524  5313          JMP ISZPC

7525  7631  BINBOOT,BEGIN

7526  0000  USER,   0000     /START USER DEFINED LOCATIONS
                    *7602    /END USER FN
7602  7240          CLA CMA           /DECREMENT PC ROUTINE
7603  1000          TAD PC
7604  3000          DCA PC
7605  5770          JMP I XHZ30       /EXIT
7606  0000          0000              /SYNC

                          /RTF CANNOT BE USED TO RESTORE FLAGS
                          /SINCE IT WILL CAUSE CPINTFF TO BE RESET
7607  7604  DEPFLA, CLA OSR
7610  3002          DCA FLAGS
7611  1375          TAD RET
7612  3006          DCA EXEC+1

7613  1002          TAD FLAGS
7614  0374          AND K70
7615  1230          TAD C6202         /CIF
7616  3005          DCA EXEC          /EXECUTE CIF BY JMS
7617  4004          JMS EXEC-1

7620  1002          TAD FLAGS
7621  7004          RAL
7622  7006          RTL
7623  0374          AND K70
7624  1372          TAD K6201

7625  3005          DCA EXEC
7626  4004          JMS EXEC-1        /EXECUTE CDF
7627  5770          JMP I XHZ30       /EXIT

7630  6202  C6202,  CIF
```

```
/INTERSIL ABSOLUTE BIN LOADER. PROGRAM COMPATIBLE
/WITH DEC BIN
/IF THE USER TAPE IS PROPRLY LOADED CPU WILL STOP
/UITH AC=0000.IF THE USER TAPE IS STARTED BEFORE
/LEADER THEN THE PROGRAM WILL STOP AT THE LEADER
/WITH AC=0000 OR 7600.


/SW(0)=1 FOR TTY ENTRY
/SW(0)=0 FOR HS RDR

/DEC PDP-8/E COMPATIBLE TTY AND HS RDR
/MNEMONICS
 /KCC     -SET TTY RDR RUN
 /RFC     -SET HS RDR RUN
 /KSF     -SKP IF TTY CHAR RDY
 /RSF     -SKP IF HSRDR CHAR RDY
 /KRB     -AC(4-11) GETS TTY CHAR
                  /SET TTY RDR RUN
 /RRB RFC-AC(4-11) GETS HSRDR CHAR
                  /SET HSRDR RUN
```

```
7631  6032  BEGIN,  KCC              /INIT TTY READER
7632  6014          RFC              /INIT HS READER
7633  6214          RDF              /GET DATA FIELD
7634  1372          TAD K6201        /FORM CDF INSTRUCTION
7635  3005          DCA EXEC         /CDF SUBROUTINE INIT.
7636  7604          CLA OSR          /READ SW REG
7637  3007          DCA RDRSEL       /READER SELECT
7640  1375          TAD RET
7641  3006          DCA EXEC+1       /INIT CDF SUBROUTINE.

                        /THE "BEGG" ROUTINE MAY BE ENTERED
                        /FROM "BEGIN" OR "GO" LOOP. BEGSW=7777
                        /IF FROM BEGIN AND =0000 IF FROM GO.
7642  7040          CMA
7643  3010          DCA BEGSW

7644  3011  BEGG,   DCA RUBSW        /RUBSW=7777 FOR DIAGNOSTIC
                                     /MESSAGES PUNCHED ON BIN TAPE.


                        /THE "READ" MAY BE ENTERED FROM "BEGG"
                        /OR "GO". RDRSW=7777 IF FROM BEGG ELSE
                        /IT IS = 0000.
7645  7040          CMA
7646  3012          DCA RDRSW

7647  1007  READ,   TAD RDRSEL
7650  7710          SPA CLA
7651  5256          JMP LO           /SW(0)=1 FOR TTY RDR

                        /HS RDR ENTRY
7652  6011  HI,     RSF
7653  5252          JMP .-1
7654  6016          RRB RFC
7655  5261          JMP SAV

                        /TTY RDR ENTRY
7656  6031  LO,     KSF
7657  5256          JMP .-1
7660  6036          KRB

                        /SAVE CHARACTER
7661  3013  SAV,    DCA CHAR
7662  1013          TAD CHAR

                        /CHECK SW FOR PROPER EXIT
7663  2012          ISZ RDRSW
7664  5325          JMP GO+5         /SW=0000 RETURN TO GO LOOP
```

```
                        /CONTINUE BEGG LOOP
                        /CHECK FOR RUB OUT
7665  1364          TAD M376
7666  7750          SPA SNA CLA     /AC=0001 FOR RUBOUT; SKIP
7667  5273          JMP NORUB
                        /RUB OUT ENTRY
7670  2011  RUB,     ISZ RUBSW       /FIRST OR SECOND RUB
7671  7040          CMA             /FIRST
7672  5244          JMP BEGG        /SET RUBSW AND FETCH NEXT CHAR
                        /DATA ENTRY
7673  1011  NORUB,   TAD RUBSW
7674  7640          SZA CLA         /IGNORE DATA IF SW=7777
7675  5245          JMP BEGG+1      /LEAVE RUBSW SET AND LOOK
                                      /FOR NEXT RUBOUT
                        /VALID DATA ENTRY POINT
7676  1013          TAD CHAR
7677  0373          AND K300        /CHANNEL 8 AND 7
7700  1344          TAD M200        /AC<0 IF DATA OR ORIGIN
7701  7510          SPA             /SKIP IF L/T OR FIELD
7702  5315          JMP DAORG
7703  7750          SPA SNA CLA     /SKIP IF FIELD. IF L/T AC=0000
7704  5312          JMP LT

                        /FIELD ENTRY POINT
7705  1013          TAD CHAR
7706  0374          AND K70
7707  1372          TAD K6201
7710  3005          DCA EXEC        /UPDATE EXEC SUBROUTINE
7711  5245          JMP BEGG+1      /FETCH NEXT CHARACTER

                        /LT EXIT
7712  2010  LT,      ISZ BEGSW
7713  5356          JMP END         /BEG ENTERED FROM GO AND HENCE
                                      /TRAILER. IBIN EXIT
7714  5242          JMP BEGG-2      /BEGG ENTERED FROM BEGIN AND
                                      /HENCE LEADER.GO FETCH NEXT CHAR
                        /DATA OR ORIGIN EXIT
7715  7200  DAORG,   CLA
7716  2010          ISZ BEGSW
7717  5330          JMP GO+10       /ENTERED FROM GO; RETURN

                        /CONTINUE BEGIN ENTRY

7720  3016  GO,      DCA CHKSUM      /CHECK SUM CLEARED IF INITIAL
                                      /ENTRY.
7721  1013          TAD CHAR
7722  3014          DCA WORD1       /SATE CHAR IN WORD1

7723  3012          DCA RDRSW       /SET UP RDR SW FOR ENTRY
                                      /FROM GO
7724  5247          JMP READ        /PSEUDO JMS TO READ
                        /RETURN FROM READ
7725  3015          DCA WORD2

7726  3010          DCA BEGSW       /SET UP BEGG SW FOR
                                      /ENTRY FROM GO
7727  5244          JMP BEGG        /PSEUDO JMS TO BEGG
                        /CHARACTER LOOK AHEAD RETURN
                        /FROM BEGG IF NEXT CHAR IS NOT
                        /TRAILER.
7730  1014          TAD WORD1
7731  7106          CLL RTL
7732  7006          RTL
7733  7006          RTL
7734  1015          TAD WORD2
7735  7430          SZL             /L=0 IF DATA;=1 FOR ORIGIN
7736  5347          JMP ORIGIN
```

```
                        /DATA ENTRY
7737  4004          JMS EXEC-1      /EXECUTE CDF
7740  3400          DCA I 0000      /DEPOSIT DATA IN MEM

                        /DIPLAY CODE
7741  1400          TAD I 0000
7742  3040          DCA DISLEDS     /DISPLAY DATA THAT
                                      /WAS JUST DEPOSITED

7743  2000          ISZ 0000        /UPDATE POINTER
7744  7600  M200,   7600            /GROUP 2 CLA ALSO -200 CONSTANT
7745  7200          CLA             /IF POINTER WRAPS AROUND FROM 7777
7746  7410          SKP
                        /ORIGIN ENTRY
7747  3000  ORIGIN, DCA 0000        /UPDATE ORIGIN- NEW
                        /DISPLAY CODE
7750  1000          TAD 0000
7751  3020          DCA PCLEDS      /DISPLAY PC
                        /CHECKSUM CALCULATION
7752  1014  CHEX,    TAD WORD1
7753  1015          TAD WORD2
7754  1016          TAD CHKSUM
7755  5320          JMP GO          /UPDATE CHECK SUM AND CONTINUE
                                      /ON GO LOOP

                        /RETURN FROM BEGG IF NEXT CHARACTER
                        /ON LOOK AHEAD IS TRAILER
7756  1014  END,     TAD WORD1
7757  7002          BSW
7760  1015          TAD WORD2
7761  7041          CIA             /FETCH AND NEGATE CHECKSUM
                                      /FROM TAPE
7762  1016          TAD CHKSUM      /AND ADD TO CALCULATED
                                      /CHECKSUM. AC=0000 IF OK

                        /BINBOOT EXIT
7763  3001          DCA AC
7764  7402  M376,    HLT             /PSEUDO HLT TO GUARANTEE THAT THE
                                      /PROGRAM WILL STOP ON EXIT SINCE
                                      /IT WILL EXECUTE AN EVEN NUMBER OF
                                      /INSTRUCTIONS FROM HERE ON. ALSO
                                      /THE CONSTANT -376
7765  6004          GTF
7766  3002          DCA FLAGS
7767  5770          JMP I .+1
7770  7434  XHZ30,   HZ30
7771  0010  L0010,   0010

7772  6201  K6201,   6201
7773  0300  K300,    300
7774  0070  K70,     70
7775  5404  RET,     JMP I EXEC-1    /RETURN FROM CDF SUBROUTINE

                        /OCT 31 1975        T.THOMAS
```