SECTION III

OPERATION AND TESTING

3.1 This section contains all control panel information and system testing information. The testing is divided into two parts : automatic testing via the control panel with the built-in hardware microdiagnostic, and test programs.

3.2 CONTROL PANEL

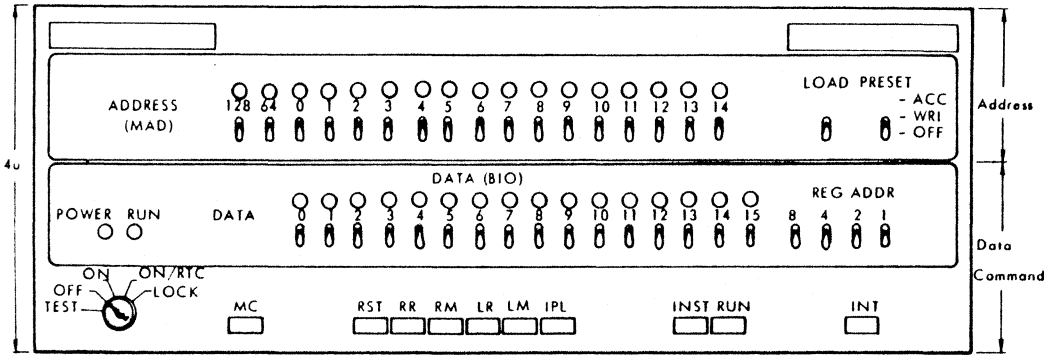
The P857M is provided with the Extended Control Panel (Figure 3-1). The top half of the panel contains ADDRESS controls; the bottom half, which is identical to the P852M/856M control panels, contains DATA controls and the CPU operating controls. When the computer is running, the BIO lines are displayed on the DATA lamps and the MAD lines are displayed on the ADDRESS lamps. When the computer stops, both the address and the contents of the next instruction are displayed. The ADDRESS half of the Extended Control Panel provides debugging facilities with the following functions :

- Display or load memory in the whole range of addresses (from 0 to 128K).
- Halt on a preset memory address.

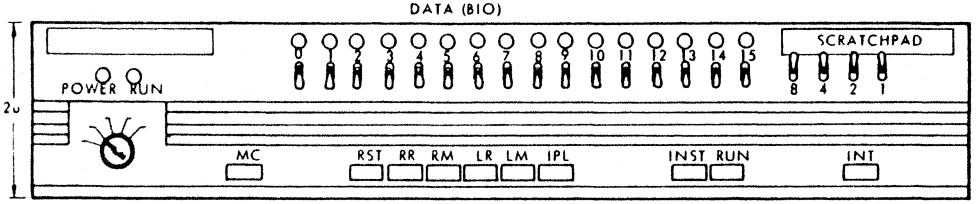
All controls and indicators on the panel are described in Table 3-1.

3.3 The P856M is provided with the Standard Control Panel (Figure 3-1). This panel is identical to the P852M panel, with the exception of the TEST position for the key switch. The TEST function operates the microdiagnostic test routines. The P856M Standard Control Panel lacks the ADDRESS controls and the halt-on-preset-address feature.

EXTENDED CONTROL PANEL (P857)



STANDARD CONTROL PANEL (P852, P856)



MINI CONTROL PANEL (OPTIONAL)

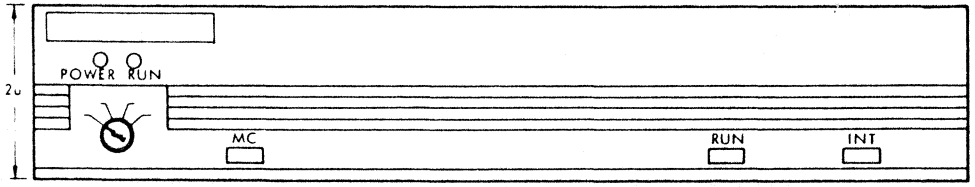


Figure 3-1 Control Panels

3.4 A mini control panel (Figure 3-1) is provided as an option for either the P856M or P857M systems.

3.5 Key and Command Operations

Control panel operations within the CPU operation sequence is shown in Figure 1-8 (Machine-State Pointer, Control Panel Operations). Each of the control-panel sequences shown on the flow diagram is described briefly in Table 3-1. The system is switched on by setting the key switch to ON, ON RTC, or LOCK. The LOCK position disables all control panel command switches except interrupt, which is still operable with the control panel locked. With the control panel switched ON or ON RTC, the CPU is set to the Run state by pressing RUN or IPL. Pressing any of the command switches MC, INST, RST, RR, RM, LR, LM will reset the Run state. When the key switch is set to TEST, the computer is set to Diagnostic mode, and the command switches then have different functions to those listed in Table 3-1.

3.6 Address Operations (Extended CP only)

The ADDRESS half of the control panel operates through an internal address register with incrementing logic. During normal running (PRESET at OFF), the ADDRESS lamps display the Bus MAD lines. When the computer stops, the ADDRESS lamps display the address of the next instruction, which is in the program counter (P), the address register (S), and on the MAD lines.

3.7 For either a read memory (RM) or load memory (LM) operation (Table 3-1), the ADDRESS lamps display the address placed in the control-panel address register by the panel switches. The address is first loaded by setting up the switches and then pressing LOAD ADDR. Thereafter, each time RM or LM is pressed, the address register is incremented and the ADDRESS lamps display the new address. The CPU program counter (P) is not used or affected by either of these operations.

Table 3-1 Control Panel Switches and Lamps

	Key Switch
OFF/ON	Main power switch connected directly to the power supply. The power is switched on (POWER lamp lighted) for positions ON, ON RTC, LOCK, and TEST.
ON	All panel controls are enabled.
ON RTC	All panel controls are enabled, and the Real Time Clock operates (Paragraph 5.6).
LOCK	All control-panel command switches except INT are disabled.
TEST	The automatic microdiagnostic test mode is selected (paragraph 3.16).
	Command Switches
MC	Master Clear : Clears or resets most hardware logic. Activates the GP Bus signal CLEARN, and the CPU signals MCL, MCLN.
RUN	Begins the program. The RUN switch sends the momentary signal START and the flip-flop signal RUNN to the CPU RUNF logic (paragraph 2.90).
INST	Instruction Step : Each time INST is pressed, the CPU performs the one instruction indicated by the program counter (P) and then halts. INST may be used to step the computer through a program (or part of one) instruction-by-instruction. The INST switch resets the control panel RUNN flip-flop, and sends a 113μsec START signal to the RUNF logic.
RST	Read Status. The contents of the program status word are displayed on the DATA lamps (paragraph 1.47).
RR	Read Register. The contents of the scratchpad register (A0-A15) selected by the SCRATCHPAD switches are displayed on the DATA lamps.
RM	Read Memory. The contents of memory are displayed on the DATA lamps. Consecutive words can be read by repeated pressing of the RM button.
	Std CP: memory address is selected by the program counter (P), and P is incremented with each RM. Ext CP: memory address is selected by the ADDRESS switches. The panel address register is automatically incremented; the program counter (P) is not used or affected.
LR	Load Register. The word code set on the DATA switches is loaded into the scratchpad register (A0-A15) specified by the SCRATCHPAD switches.
LM	Load Memory. The word code set on the DATA switches is loaded into memory. Consecutive words can be loaded by repeated pressing of LM. Std CP: memory address is selected by the program counter (P), and P is incremented with each LM. Ext CP: memory address is selected by the ADDRESS switches. The panel address register is automatically incremented; the program counter (P) is not used or affected.
IPL	Initial Program Loader. An initial bootstrap program located in a hardware read only memory is loaded into memory word locations 00 to 063_{10} (characters 00 to $7E_{hex}$).
INT	Interrupt. This button generates a level-1 Interrupt Request for the Operator's interrupt. The same interrupt can be set by the I/O console via the integral serial control unit. The interrupt may be used by the operator, for example, to change the running program with information supplied by the operator.
	Data
DATA	The 16 DATA switches are used to set a data word onto the Bus BIO lines during load register (LR) and load memory (LM) operations. For all computer operations, the DATA lamps display the contents of the Bus BIO lines. When a running computer stops, the DATA lamps display the contents of the next instruction. For RR and LR operations, the DATA lamps display the contents of the scratchpad register (A0-A15) selected by

	SCRATCHPAD. For RM and LM operations, the DATA lamps display the contents of the memory address selected by the panel address register (Ext CP) or program counter (Std CP)
	Address Section (Extended CP only)
ADDRESS	The ADDRESS switches are used to select an initial memory address for read memory (RM) and load memory (LM) operations. For all computer operations except RM and LM, the ADDRESS lamps display the contents of the Bus MAD lines, via the panel address register. When a running computer stops, the ADDRESS lamps display the address of the next instruction. For RM and LM operations, the ADDRESS lamps display the contents of the panel address register, which is loaded from the ADDRESS switches and incremented by the RM and LM operations. No control is provided for bit 15 (character selector) because the panel accesses only memory word addresses.
LOAD ADDR	When this button is pressed, the code set on the ADDRESS switches is immediately loaded into the panel address register. This address is incremented by successive RM or LM operations; the address register is reloaded from the MAD lines for any other operation.
PRESET	This switch is used to select a Stop On Address mode. The stop will occur when the MAD-line address, via the panel address register, compares with the code set on the ADDRESS switches. OFF Normal operation. Do not stop on address. ACC Stop On Address, Access. Stop when any memory operation accesses the address set on the ADDRESS switches. WRITE Stop On Address, Write. Stop when any memory operation writes at the location set on the ADDRESS switches.

	Scratchpad Registers
SCRATCHPAD	These four switches select one of the scratchpad registers (A0-A15) to be accessed by the read register (RR) or load register (LR) operation.

3.8 Halt on preset address is performed by setting the PRESET switch to ACC or to WRITE. The control-panel address register is loaded with the desired address set on the ADDRESS switches and LOAD ADDR pressed. In access mode (PRESET to ACC), the computer will halt whenever the memory address on the MAD lines is the same as the one set on the ADDRESS switches. In write mode (PRESET to WRITE), the computer will halt when the memory address on MAD compares with the ADDRESS switches during a memory write operation.

3.9 Interface Signals

The control panel interface signals are listed in Table 3-2, along with a brief description of each signal. All interface signals are also shown on the control panel block diagram, Figure 3-2. The control panel connector is included in the Wiring and Cabling part of Section III.

3.10 Control Panel Logic

A block diagram and a detailed schematic diagram of the control panel are provided in Figures 3-2 and 3-3.

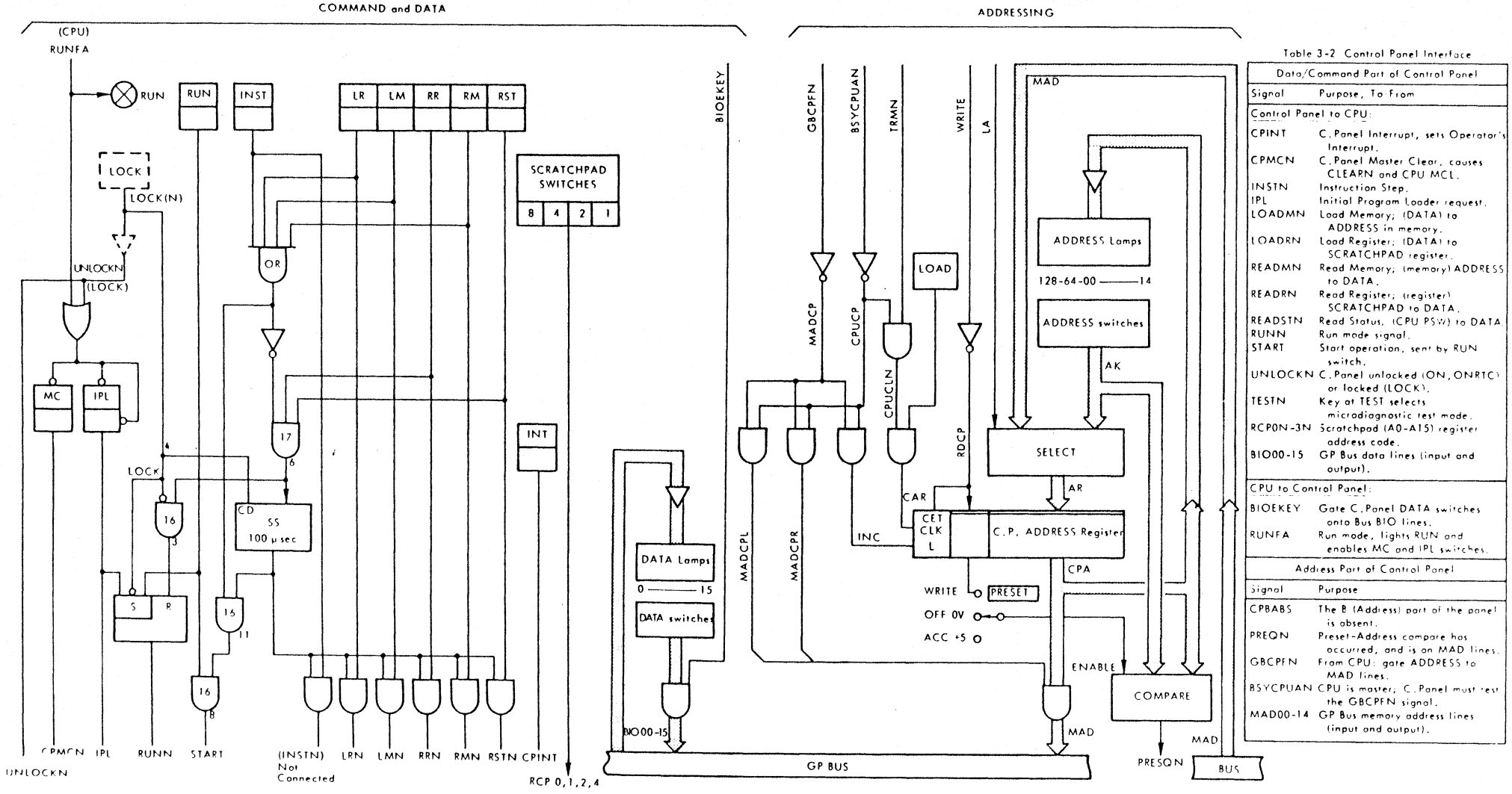


Table 3-2 Control Panel Interface

Data/Command Part of Control Panel	
Signal	Purpose, To-From
Control Panel to CPU:	
CPINT	C. Panel Interrupt, sets Operator's Interrupt.
CPMCN	C. Panel Master Clear, causes CLEARN and CPU MCL.
INSTN	Instruction Step.
IPL	Initial Program Loader request.
LOADMN	Load Memory; (DATA) to ADDRESS in memory.
LOADRN	Load Register; (DATA) to SCRATCHPAD register.
READMN	Read Memory; (memory) ADDRESS to DATA.
READRN	Read Register; (register) SCRATCHPAD to DATA.
READSTN	Read Status, (CPU PSW) to DATA
RUNN	Run mode signal.
START	Start operation, sent by RUN switch.
UNLOCKN	C. Panel unlocked (ON, ONRTC) or locked (LOCK).
TESTN	Key at TEST selects microdiagnostic test mode.
RCP0N-3N	Scratchpad (A0-A15) register address code.
BIO00-15	GP Bus data lines (input and output).
CPU to Control Panel:	
BIOEKEY	Gate C. Panel DATA switches onto Bus BIO lines.
RUNFA	Run mode, lights RUN and enables MC and IPL switches.
Address Part of Control Panel	
Signal	Purpose
CPBABS	The B (Address) part of the panel is absent.
PREQN	Preset-Address compare has occurred, and is on MAD lines.
GBCPFN	From CPU: gate ADDRESS to MAD lines.
BSYCPUAN	CPU is master, C. Panel must test the GBCPFN signal.
MAD00-14	GP Bus memory address lines (input and output).

Note 1 : With the key set to LOCK : the MC and IPL buttons are blocked ; INST, LR, LM, RR, RM, RST are blocked by the SS which is held reset ; and RUN is blocked from activating START by the reset SS.

Note 2 : Run state or Lock inhibit the MC and IPL buttons.

Figure 3-2 Control Panel Block Diagram

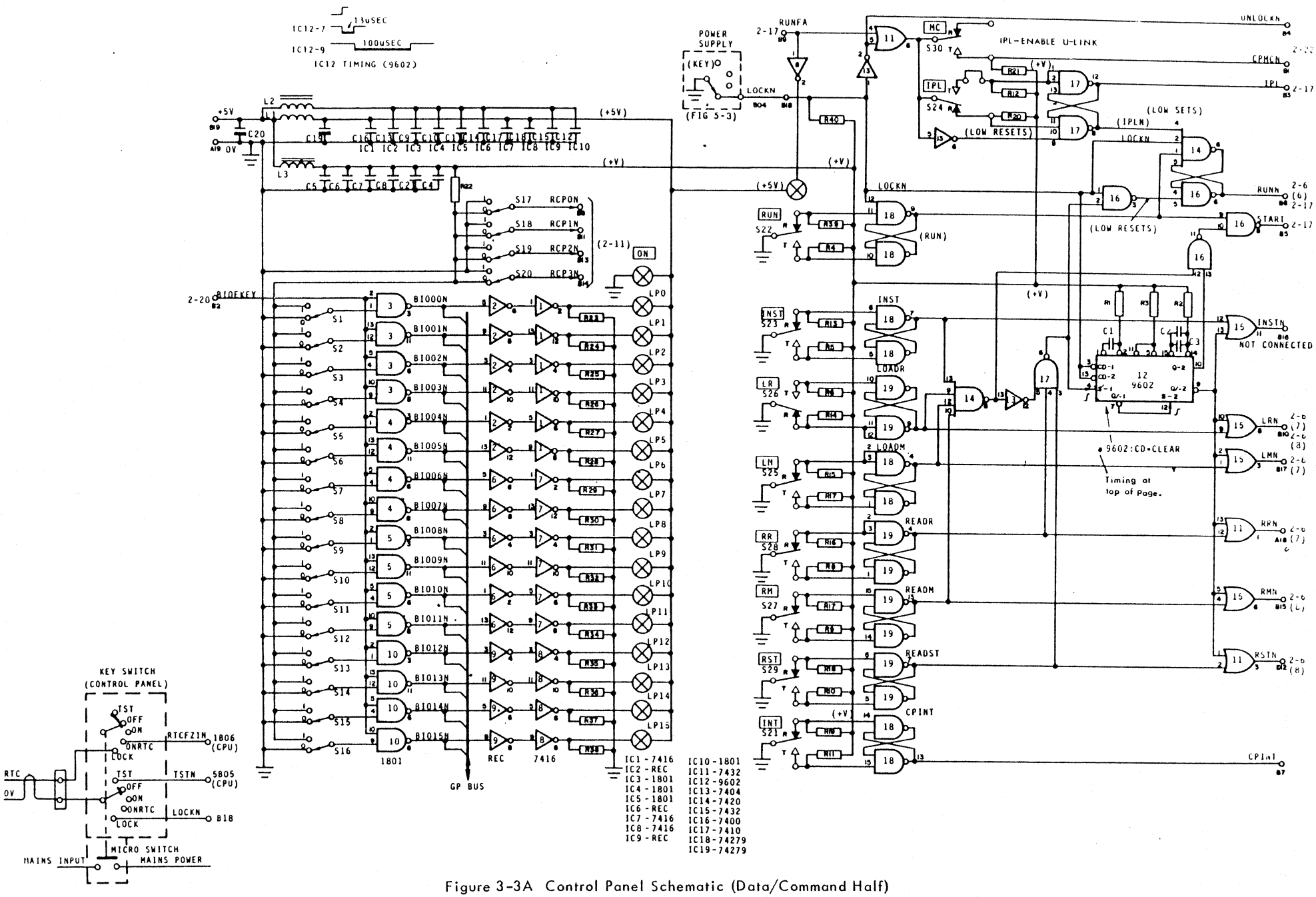


Figure 3-3A Control Panel Schematic (Data/Command Half)

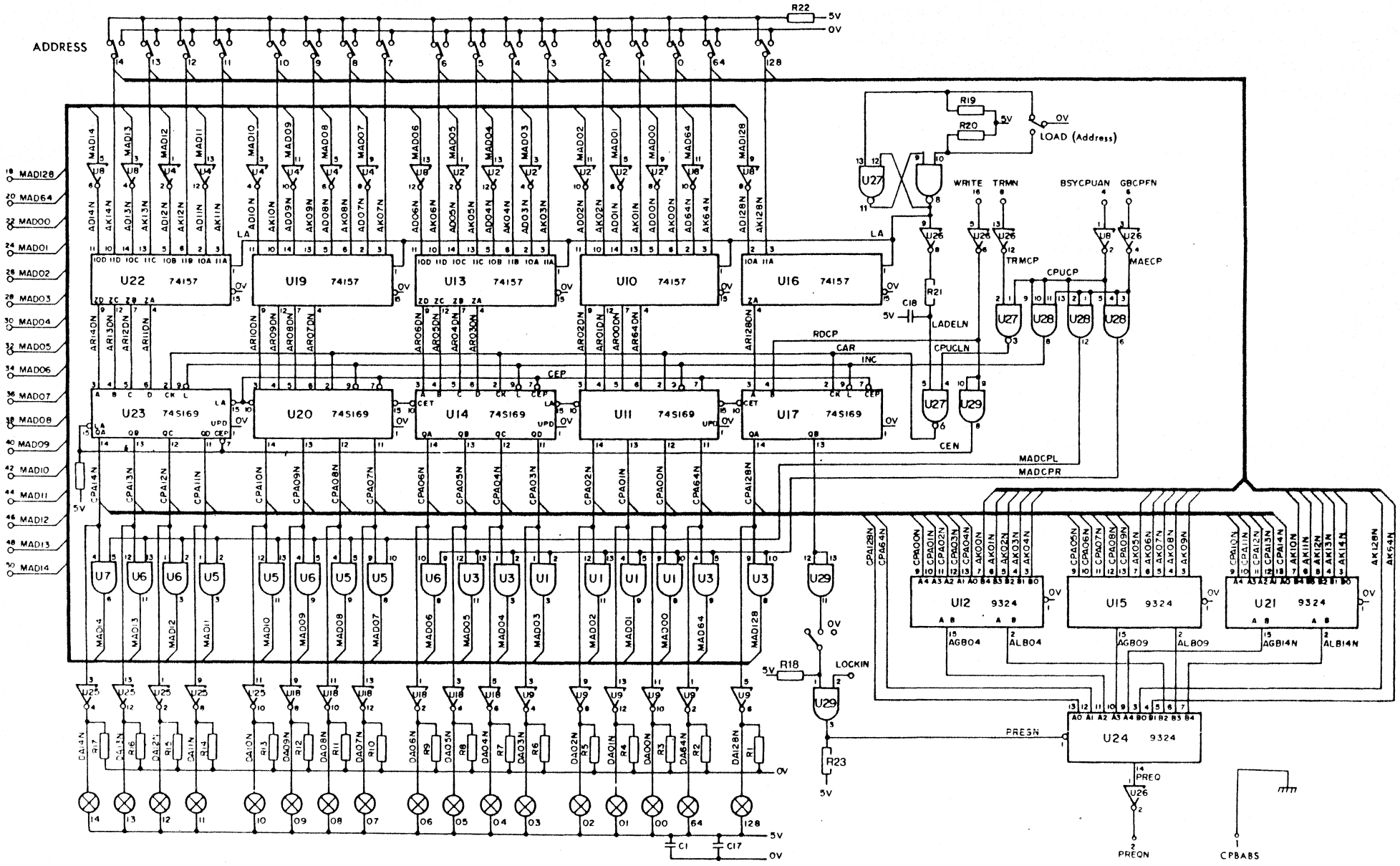


Figure 3-3B Control Panel Schematic (Address Half)

3.11 Key Switch. The Key Switch is shown on the power supply schematic, Figure 5-3. One contact of the switch controls the mains input power to the power supply. A second contact produces the enabling signal RTCFZIN for the real time clock logic. A third contact of the Key Switch produces the test signal TSTN and the control-panel enable/inhibit signal LOCKN. The test signal is described in the Diagnostic Testing part of Section III. The Lock signal is used to disable all the control-panel command switches except INT. The following table is a review of the Key Switch functions.

Key Switch	Power	Real Time Clock	UNLOCKN Signal	Conditioning for :
OFF	off	--	--	RUN, INST,
ON	on	--	low	LR, LM, RR, MCL,
ON RTC	on	on	low	RM, RST, IPL
LOCK	on	on	h	

3.12 Scratchpad. The four Scratchpad switches are used to address a scratchpad register (A0-A15). The switches produce the true-logic (+5V = logic 1; 0V = logic 0) signals RCP0-4.

3.13 Data. The DATA switches are used to set the 16 bits of a data word to be placed on the GP-Bus BIO lines. The DATA switches are gated onto the BIO lines by CPU signal BIOEKEY. BIOEKEY is produces by BUSFN and BOKFN in the Bus Controller logic (paragraph 2.10).

3.14 INITIAL PROGRAM LOADING

A hardware bootstrap is provided which consists of a 256x4 bit ROM and an IPL-microprogram routine. This hardware routine organizes the bootstrap-ROM contents into 64 sixteen-bit words and loads them into memory locations 00-63. The system then has a bootstrap loaded and is able to accept a software IPL followed by a program. A basic procedure is given here for loading a program from paper tape into the system. This procedure assumes that the tape has been assembled with the software IPL, followed by the program.

Program Load Routine

- Load IPL/Program tape into paper-tape reader.
- Set DATA switches for I/O Bus and paper tape reader :

0 0 0 1	0 0 0 0	0 0 1 0	0 0 0 0

(The codes are shown at the top of Table 2-10, Bootstrap Listing).

- Press INST, MC, IPL
  a. Hardware bootstrap organized by hardware IPL routine and loaded into memory 0-63 (/00 — /7E).
  b. The bootstrap routine is performed to load the software IPL low-core part into memory, starting at /80.
  c. The low-core IPL is performed (with access into the bootstrap) to load the IPL high-core part into memory, at the high end.
  d. The high-core IPL is performed to load the object program into memory, starting at /00 and overwriting the bootstrap and the low-core IPL. Interrupt start addresses are loaded at this time in locations /00 — /7E.

3.15 Bootstrap Test

The hardware bootstrap can be loaded into memory for testing (comparing with table 2-10) without being overwritten by the software IPL. This is done by performing the Program Load Routine without a device connected (Paper-tape reader off, disc off, etc.), as follows :

- Set DATA switches for I/O Bus and reader.
- Press INST, MC, IPL.
  a. Hardware bootstrap organized by hardware IPL routine and loaded into memory 00-63.
  b. CPU stops after bootstrap is loaded.
- Load highest memory address into P-register (DATA + reg=0).
- Press INST which increments P to 000.
- Begin pressing RM, and copy bootstrap from DATA lamps.

3.16 MICRODIAGNOSTIC TESTS

Diagnostic test routines are included in the CPU microinstruction control ROM. Approximately 100 words of the CPU control store are allocated for microdiagnostic routines. The complete microdiagnostic tests about 70% of the CPU hardware (Figure 3-4). These microdiagnostics are performed, with the system off line, by setting the control-panel key switch to TEST. The microdiagnostics are arranged sequentually from very basic data-path and register tests to complex instruction operations (Figure 3-5). The different methods of running the tests, in the suggested order, are :

- Automatic Go/No-Go Mode. The instruction-tests DLA and RB are executed and a final display is given. The memory and CPU/CU test is executed and a final display is given. Correct results in this mode indicate correct machine operation.
- Test-By-Test Mode. The operator steps sequentually through logic tests T1, T2, T3, the memory and CPU/CU tests M, and the instruction tests R, and obtains a display indication at each stage. The Chained Tests mode provides further exercise of the same sequence of tests.
- Basic Tests. The operator steps sequentually through basic tests of the control panel, L register, M register, and Q register, exercising all DATA switches and lamps for each test. This checks the basic registers and data paths.

3.17 Test Procedure, Basic Tests

- Key switch to TEST. (Loops through control-panel test.)
- Operate each DATA switch.

oo Corresponding DATA light should light.

- Press LR. (Loops through L-Register test.)
- Operate each DATA switch.

oo Corresponding DATA light should light.

- Press LR. (Loops through M-Register test.)
- Operate each DATA switch.

oo Corresponding DATA light should light.

- Press LR. (Loops through Q-Register test.)

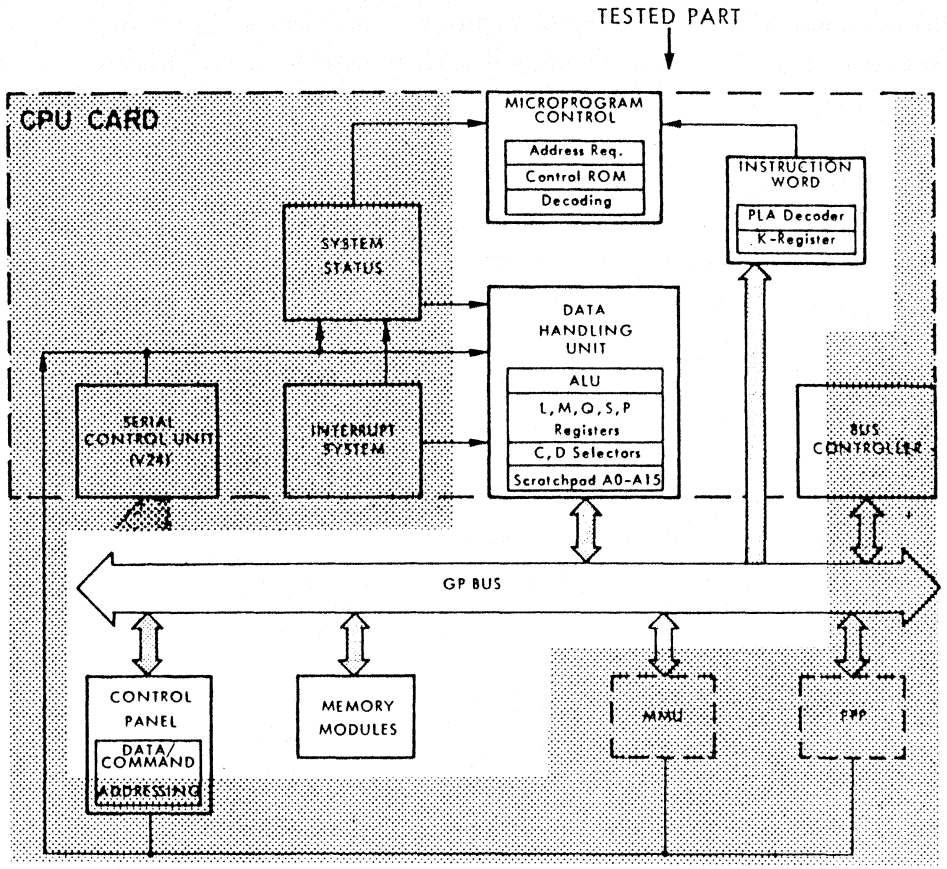


Figure 3-4 CPU Logic Tested by Microdiagnostics

- Operate each DATA switch.

oo Corresponding DATA light should light.

The test is now looping through the Q-Register test. To continue to the next series of tests, set the DATA switches as required (refer to Automatic Mode or Test-by-Test Mode) and press LR or RUN.

3.18 Test Procedure, Automatic Mode (Go/No-Go)

- Key switch to TEST. (Loops through control-panel test.)
- DATA-switch 0 to 0 position. (Enables final display.)
- RUN. (Instruction test performs DLA and RB and then loops at end-of-test display).

oo Wait for display 0000000000000000. If error, do Test-by-Test mode to diagnose fault. If correct, do Memory and CU test.

- Set address of existing CU device on DATA switches 02-07.
- Press LM. (Executes memory tests 1 and 2, and CPU/CU test and then loops at end-of-test display.)

oo Wait for display 0000000000000000. If error, do Test-by-Test mode to diagnose fault.

3.19 Test Procedure, Test-by-Test Mode

- Set address of existing CU device on DATA switches 02-07.
- Key switch to TEST. (Loops through control-panel test.)
- DATA-0 to 0 position. (Enables display at each stage.)
- DATA-15 to 1. (Selects test T1 first.)
- Press LR four times. (Steps through the basic tests, performs logic test T1, and loops at end-of-test display.)

oo Wait for display 0000000000000000.

- Press LR. (Executes test T2, loops at end-of-test display.)

oo Wait for display 0000000000000000.

- Press LR. (Executes test T3, loops at end-of-test display.)

oo Wait for display 0000000000000000.

- Press LR or RUN. (Executes instruction test R, loops at end-of-test display.)

oo Wait for display 0000000000000000.

- Press LR or LM. (Executes memory and CPU/CU test M, loops at end-of-test display.)

oo Wait for display 0000000000000000.

Chained tests performes the same tests in the same sequence as above, but without halt on display after each stage.

- Key switch to TEST.
- Address of existing CU device on DATA 02-07.
- DATA-0 to 1 (chained tests, no display loop).
- DATA-15 to 1 (selects test T1 first).
- Press LR four times (steps through basic tests, then loops through the five operating tests without stopping.)
- Set DATA-0 to 0 to stop the chained tests.

oo Wait for one of the five displays.

- Press LR to step through the tests and obtain the display for each.
- Set DATA-0 to 1 to restart the chained-tests loop.

To restart the tests at the beginning, do :

- DATA-0 to 1.
- Key switch away from TEST (OFF is best), then back to TEST.

To Start a test out of sequence (for example, to start test T3 after an error indication for test T2) :

- Key to TEST; CU address on DATA 02-07; DATA-0 to 0.
- Set DATA 08-15 to select desired test :

  00000001 (/01) for logic test T1.
  00000010 (/02) for logic test T2.
  00000100 (/04) for logic test T3.
  00001000 (/08) for instruction test R.
  00010000 (/10) for memory, CU test M.

3.20 Analysing Tested Functions

3.21 Basic Tests. In the control-panel test (Figure 3-6), the operator verifies the operation of all control-panel DATA switches and lamps and the BIO

Note: 0 = lamp on
● = lamp off

data path to and from the GP Bus. In the L-register test, the data path through the CPU D-selector and L-register is added to the control-panel test. The M-register test adds the C-selector, M-register, and ALU to the data-path loop. The Q-register test adds the Q-register to the data-path loop.

3.22 These tests exercise all bit positions of the control-panel switches and lamps and the CPU L, M, and Q registers as well as the data path. Not all the functions of the registers, the C and D selectors, or the ALU are tested.

3.23 Q-Register Test. This test is not only a part of the basic tests, but also the start and the error output of the following logic, memory, and instruction tests. While looping through the Q-register test, the DATA switches content are loaded into the CPU registers to select the type of Test-by-Test operation and the number of the next test. (Test selection and incrementing is shown in Figure 3-7.) If a test has been performed and an error occured, the test branches back to the Q-test loop and the DATA lamps display the setting of the switches rather than the correct display.

3.24 Logic Test T1. The tested functions (Figure 3-8) are :

- Q-reg. shifted left
- A-Bus selection
- Constant Two
- Qo
- ALU=0
- A OR B
- A + B
- $\overline{B}$

3.25 Logic Test T2. The tested functions (Figure 3-8) are :

- Q-reg. shifted right
- Constant Ten
- ALUZERO
- P-register
- P - 2
- A - B
- A + B
- crossed A

3.26 Logic Test T3. The tested functions (Figure 3-8) are :

- A-operand shifted right
- 4 X A
- Reading and writing scratchpad, address incrementation by CT counter (particular addresses A=A2 and A=A15).

3.27 Instruction Tests R. The tested functions (Figure 3-10) are :

- The K-register is loaded with the instruction code for the DLA instruction.
- Scratchpad registers A1 and A2 are loaded with known values.
- Branch to the DLA microprogram and execute DLA instruction.
- Return to Test R via CPU Idle loop and verify A1 and A2. (The CR is verified by the next instruction.)
- The K-register is loaded with the instruction code for the RB instruction.
- Address the RB microprogram with the PLA, and execute RB instruction.
- Return to Test R via CPU Idle loop and verify P-register.
- A failure in CR positioning or in CV (condition verified) circuit, or PLA addressing function should cause wrong value of P counter.

3.28 Memory Test M1. The tested functions (Figure 3-9) are :

- Bit-15 is set to 1 in all addresses of a 16k block (4k for P856).
- The block is read and verified.
- The pattern is shifted left 1 position (to bit 14) and the block is read and verified.
- Shifting the pattern and reading the block is continued until all bit positions have been read and verified.

3.29 Memory Test M2. The tested functions (Figure 3-9) are:

- Every word of a 16k memory block (4k for P856) is written with its own address value.
- Each word is read and verified.

3.30 CPU/CU Dialogue Test M3 (part of Memory test loop). The tested functions (Figure 3-9) are:

- An existing CU device address (from DATA switches 02-07) and TMP are sent on the GP-Bus MAD lines.
- The CU response TPM is verified by means of condition register (CR $\neq$ 3).

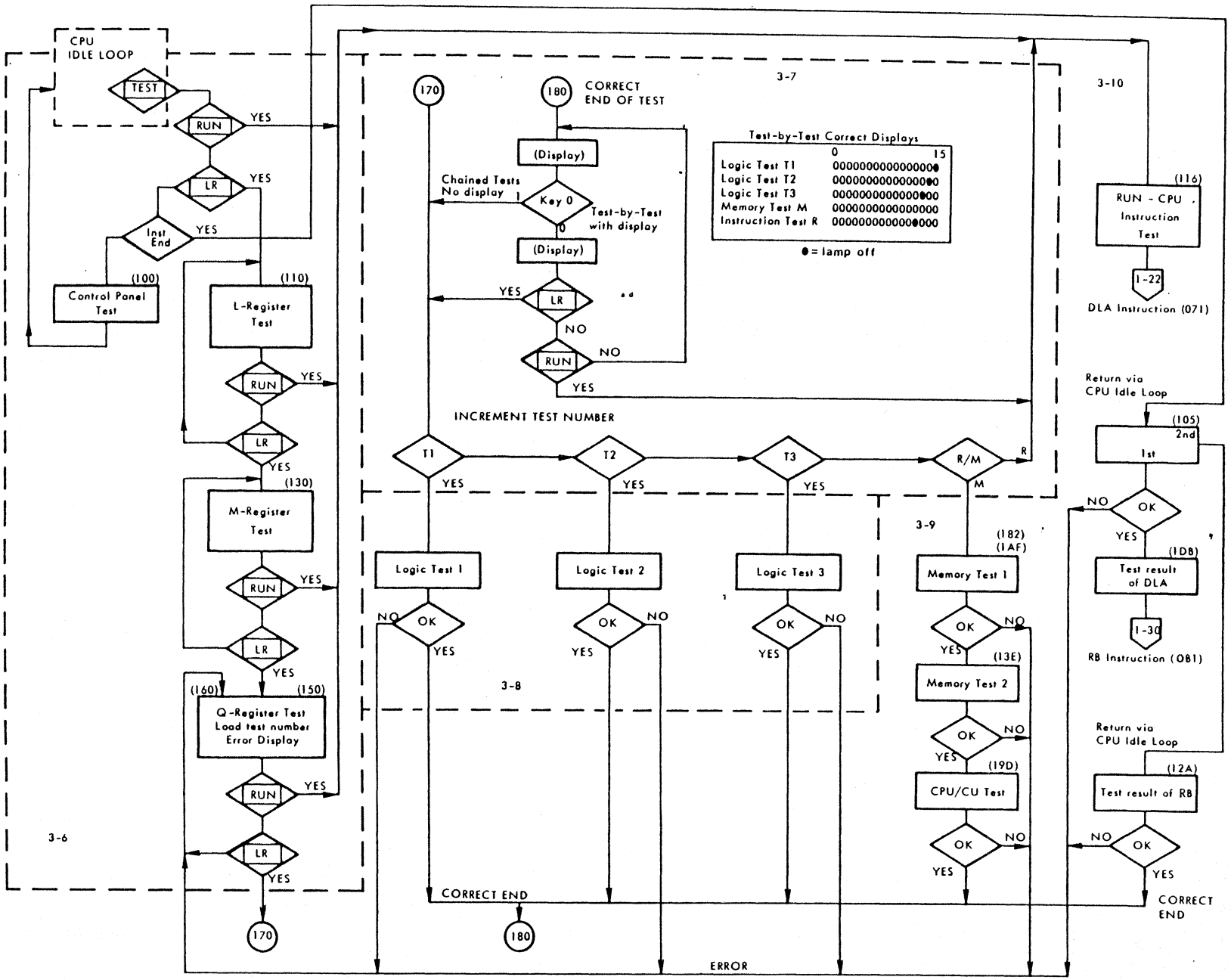


Figure 3-5 Microdiagnostics Block Diagram

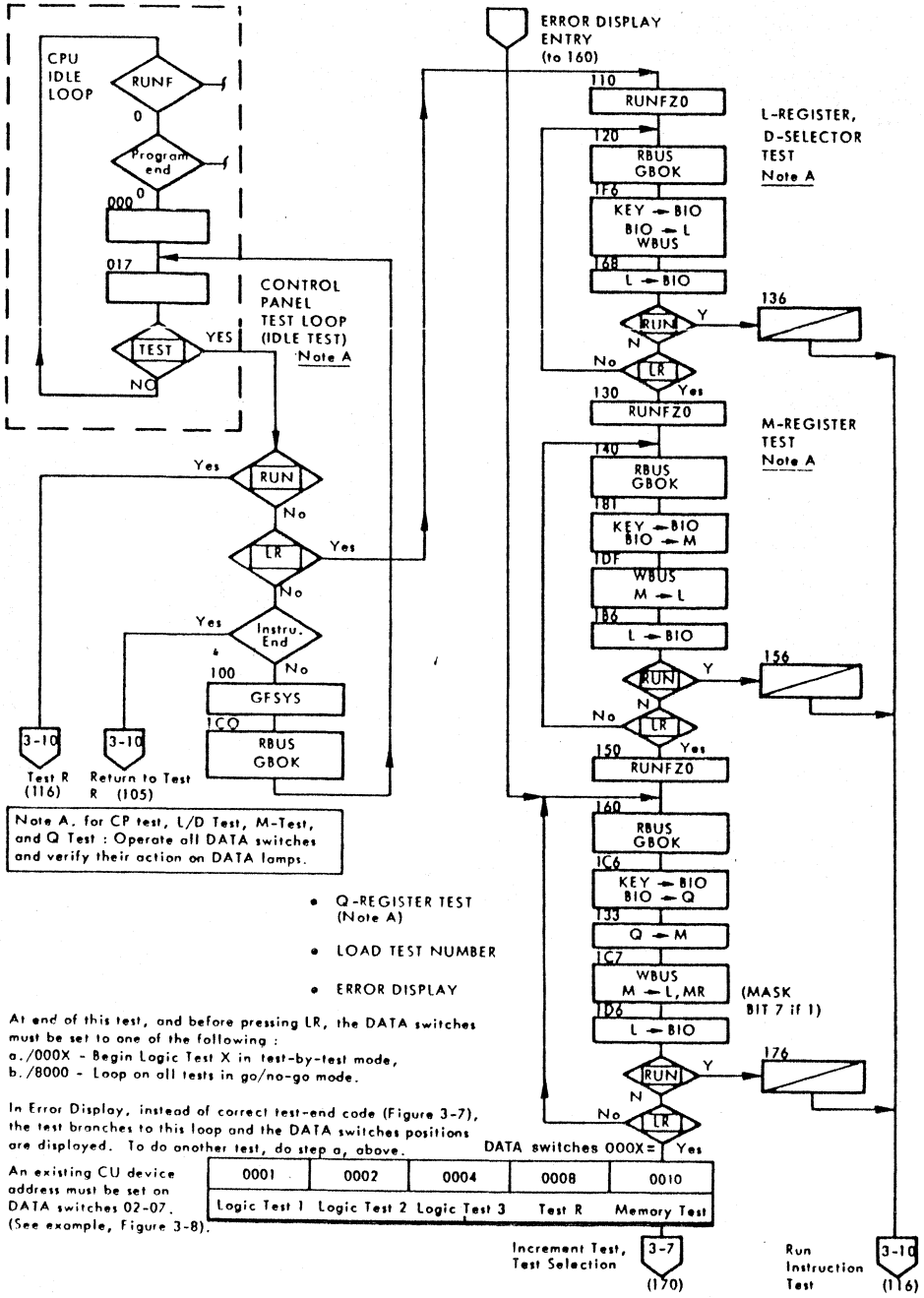


Figure 3-6 Start and Basic Tests

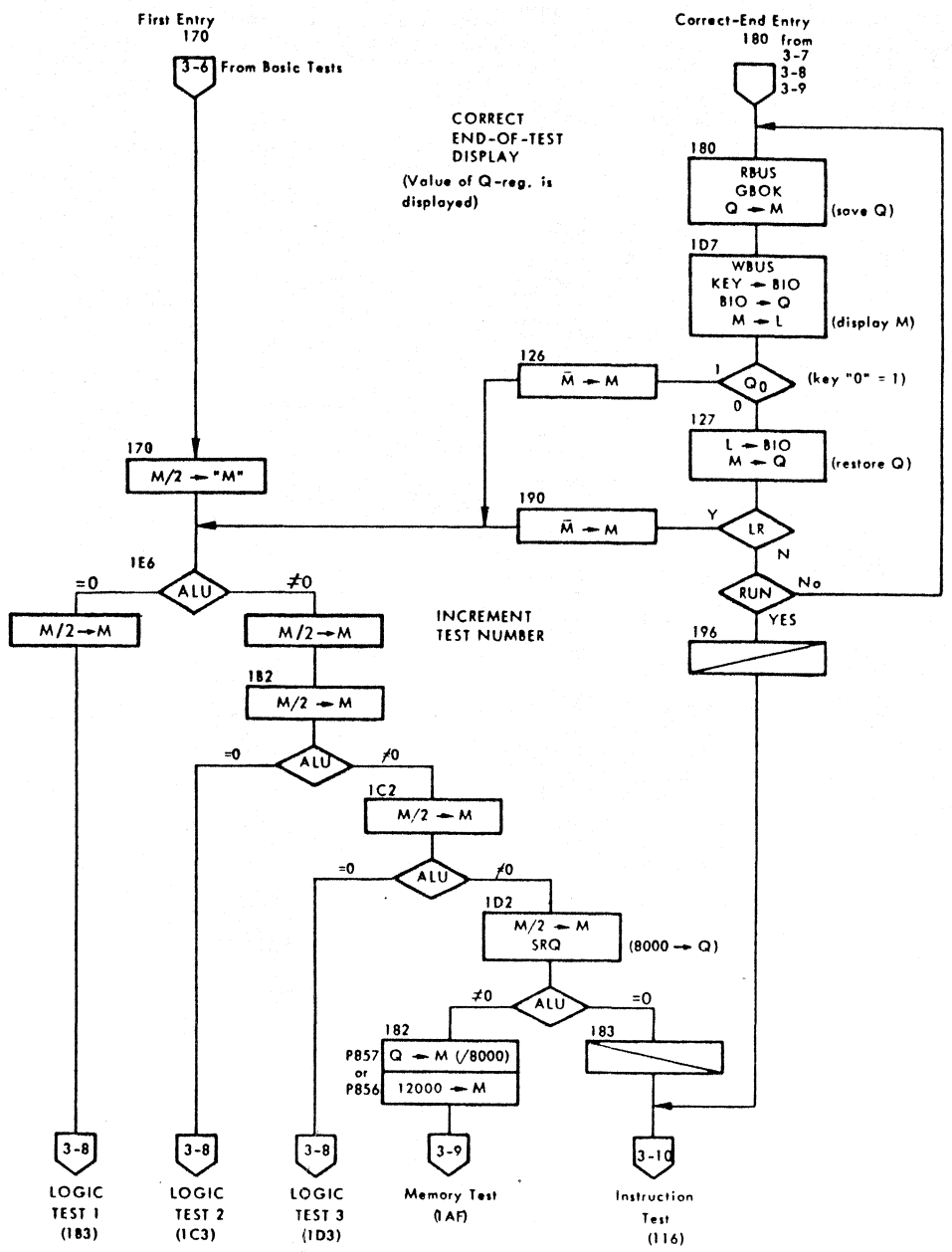


Figure 3-7 Increment Test, Test-End Display

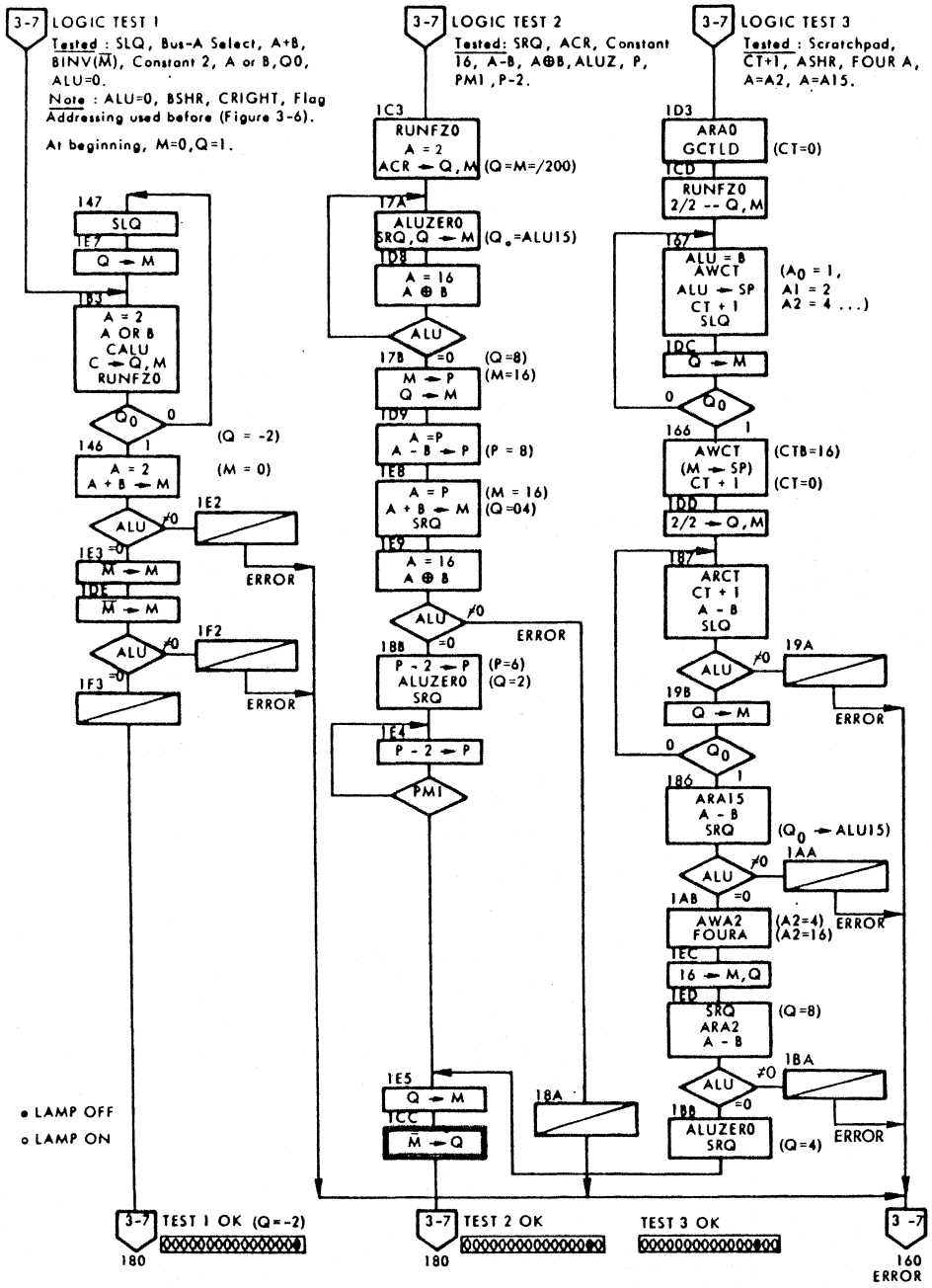


Figure 3-8 Logic Tests (T1, T2, T3)

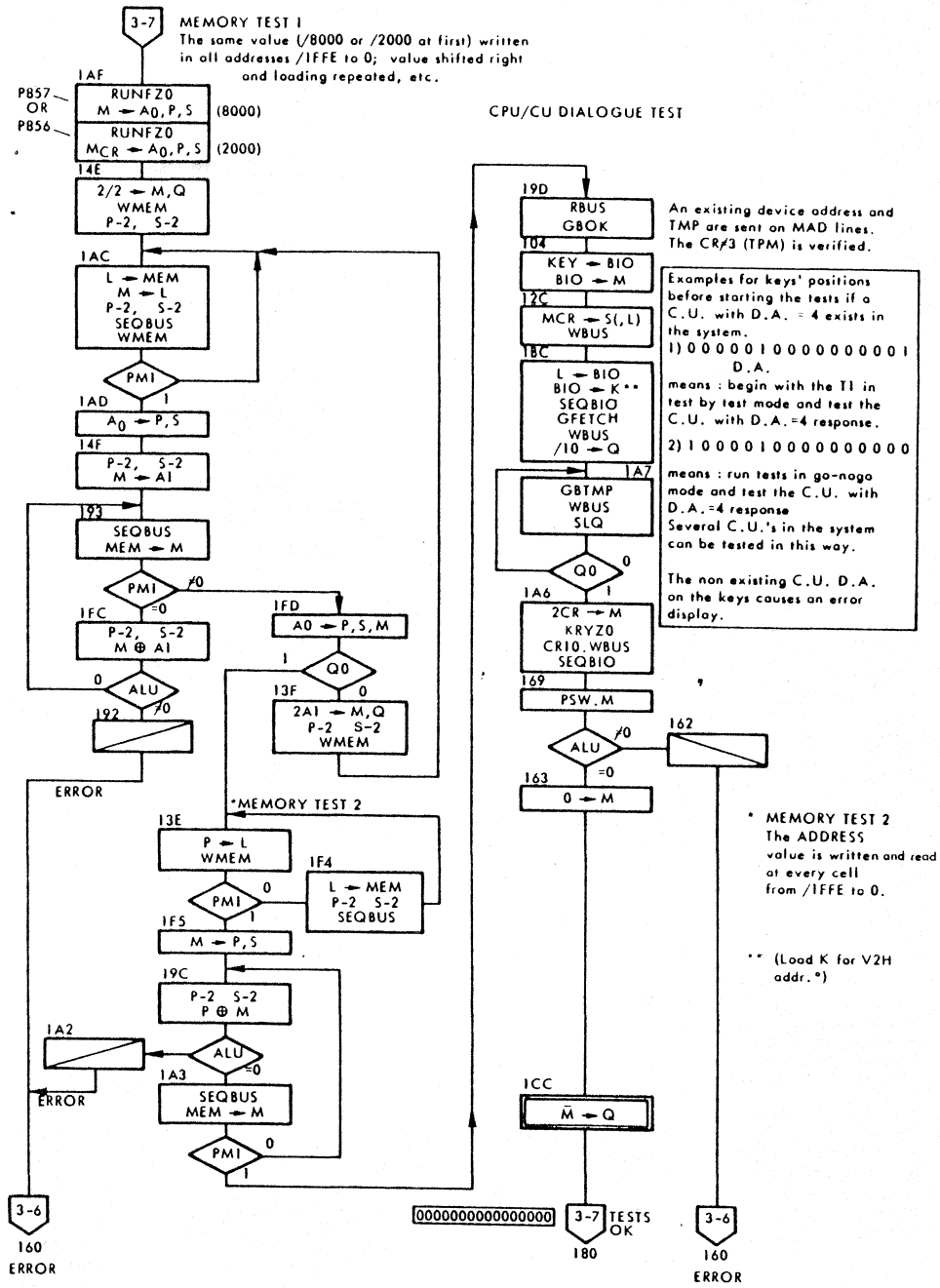


Figure 3-9 Memory and CPU/CU Dialogue Tests (Test M)

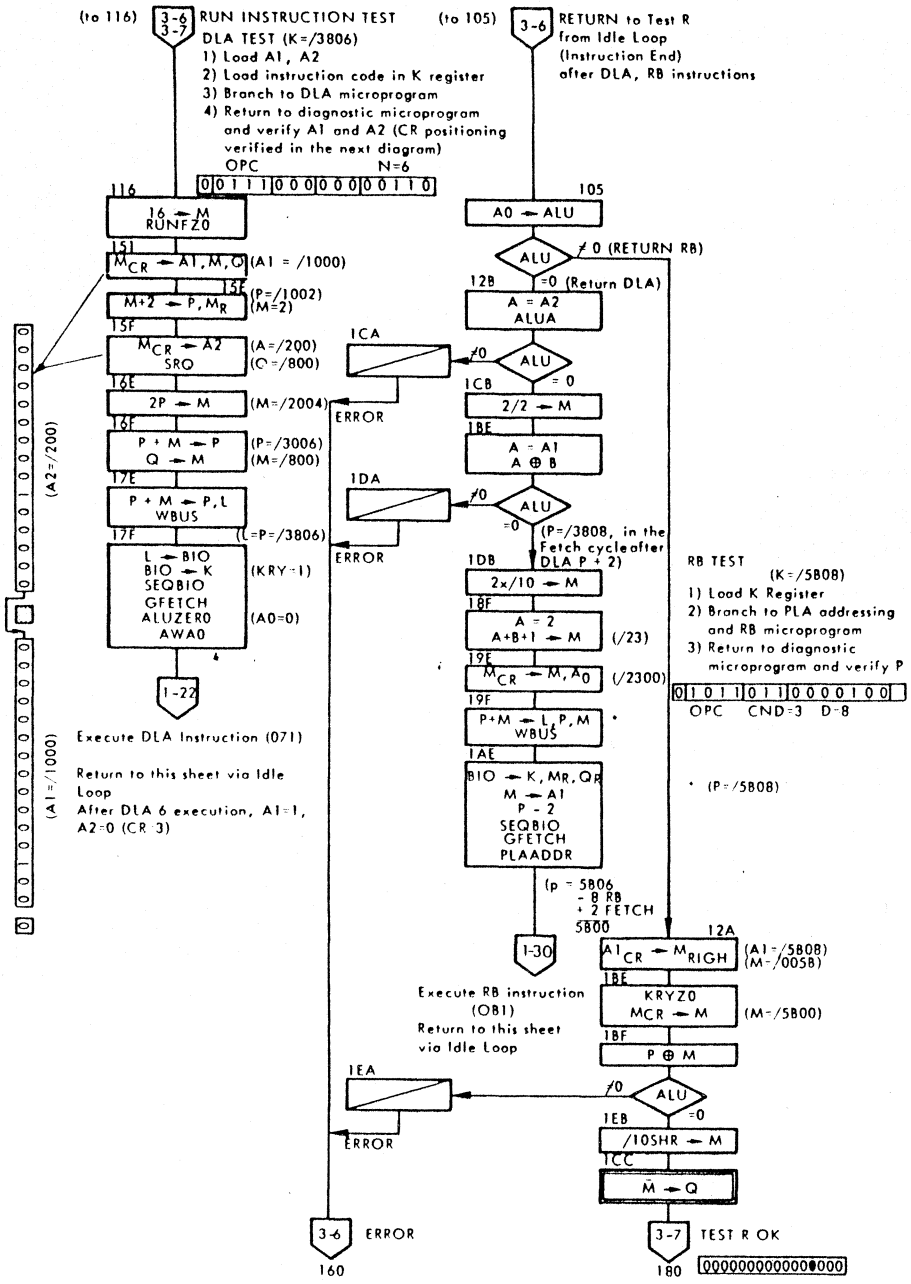


Figure 3-10 Run CPU Instruction Tests (Test R)

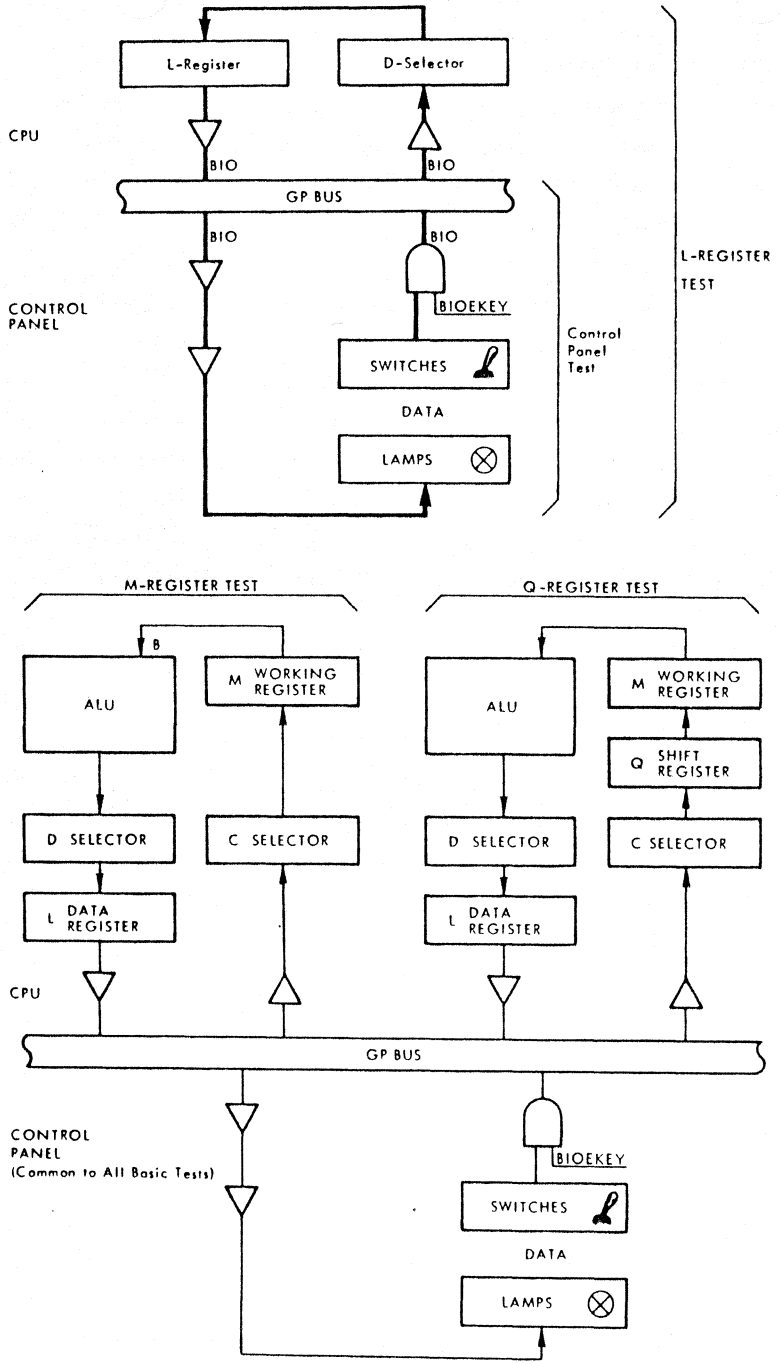


Figure 3-11 Logic Tested by Basic Tests

3.31 TROUBLESHOOTING

Figure 3-12 shows a flowchart of a fault finding procedure covering the most probable areas of a system breakdown assuming that the fault is continuous. When a decision point is reached on the flowchart the engineer must choose the path most appropriate to the fault symptoms. Where the choice is not clear, first take the path that most easily isolates and tests one part of the system and then gradually include other parts of the system until the fault is isolated. For intermittant faults the same procedure can be used but must be repeated each time the symptoms occur (if it is impossible to simulate the fault). It should also be remembered that software not debugged properly can cause faults, which to the programmer appear to be hardware oriented. Therefore, before starting on the hardware, check with the operator if the fault occurs with all programs or with only certain programs, and if these programs are new to the system. If the standard test programs run correctly but the customer's programs do not run correctly, the Software Product Support group should be contacted.

3.32 Microdiagnostics

The microdiagnostic test will enable you to isolate the faulty area in the system and these should be run first before you try any of the other tests suggested in this section. Quite often what appears to be the faulty part of the system may in fact be serviceable, but due to another part of the system the wrong responses are being received. For example, a CU and its device may appear faulty when in fact the cause is operation of the Condition Register.

3.33 Bus Failures

If a GP Bus failure is indicated check the following signals, with no program running, which should all be high if the Bus is operable: RSLN, TMRN, TRMN, TMPN, TPMN, TMEN, MSN, and BSYN.

3.34 RSLN (3B17). This signal is used by the CPU card, the CU cards, and the memories and when it is low it indicates a failure in the power supply. If the power supply is correct, the most probable cause is either the CPU card or one of the memory cards. Disconnect the memory cards one at a time and then the CPU card remembering that the signal RSLN comes from a transistor with an open collector, so for this test connect a 1k ohm resistor between pins 3B17 and 3B19 (+5V).

3.35 TMRN (3A29). This signal is generated by either the CPU card or the IOP and is only used by the IOP or the memories and does not leave the CPU chassis. If this signal is low it indicates a fault in one of the memories, the IOP, or the CPU card. These should be disconnected one at a time from the system until the fault disappears.

3.36 TRMN (3A28). This signal is generated by either the IOP or the memories and is used by these during an exchange with a peripheral. If this signal is low it indicates a fault in one of the memories, the IOP, or the CPU card. These should be disconnected one at a time from the system until the fault disappears.

3.37 TMPN (3A31). This signal is generated by either the CPU or the IOP and is used by all the controllers on the Bus. If this signal is low it indicates that the fault is in the IOP, the CPU, or one of the CU's. The last one removed contains the faulty logic element.

3.38 TPMN (3A32). This signal is generated by each CU and is used by either the CPU or the IOP. If the signal is low it indicates that the fault is either in one of the CU's or in one of the I/O cables. Isolate the CU's and I/O cables one at a time until the fault disappears. The fault will be found in the last item removed.

3.39 TMEN (3A30). This signal is generated by the CPU to validate the address of the external register of either the IOP or MMU. If the signal is low check the CPU, IOP, MMU and the backpanel wiring.

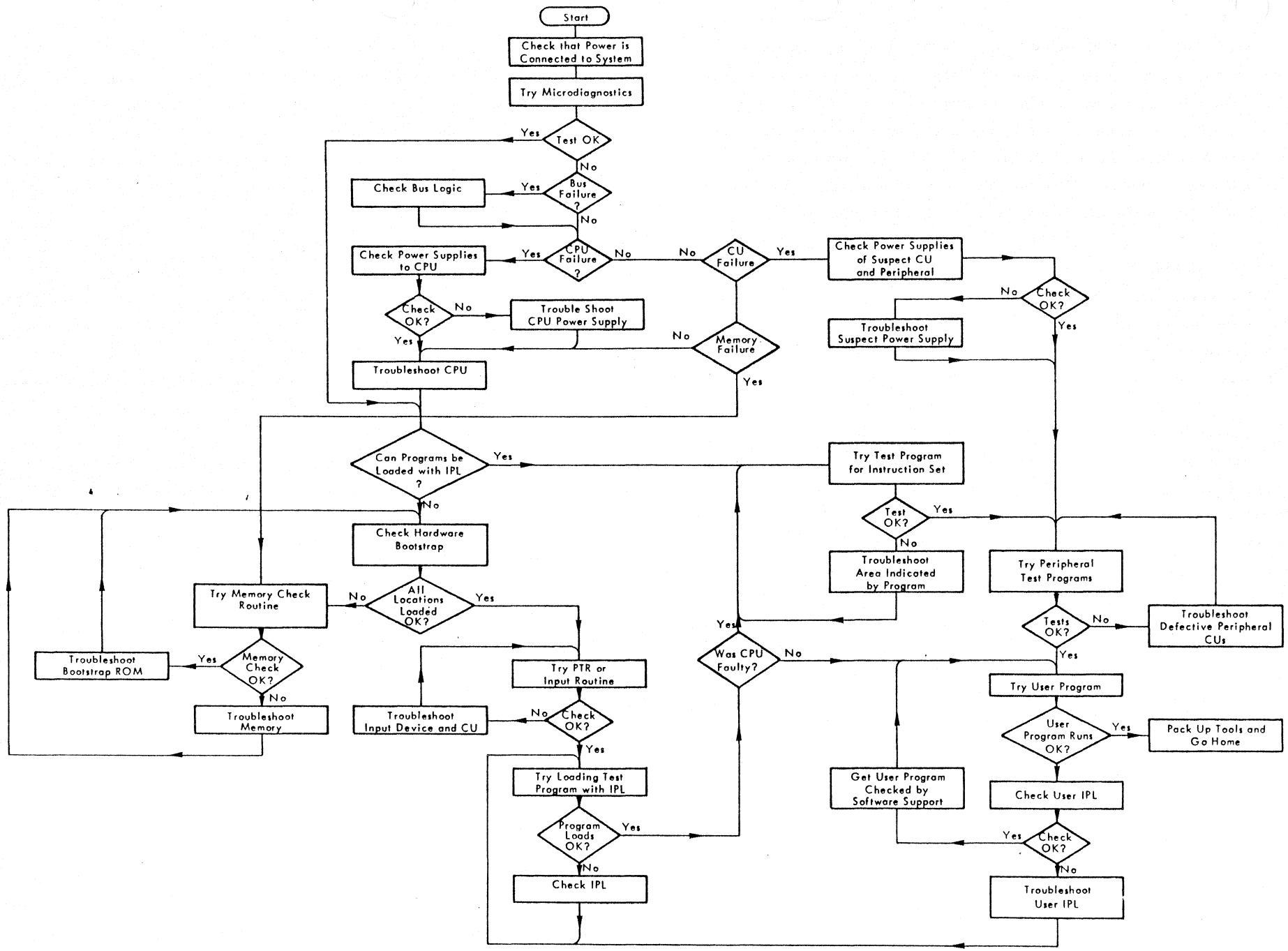


Figure 3-12 Fault Finding Flow Chart

3.40 MSN (3A37). This signal is generated by all the masters connected to the Bus and is used by the masters and the CPU, and it does not leave the CPU chassis. When this signal is low it stops any other master from accessing the Bus and stops the CPU from displaying information on the indicator lamps of the control panel. To find the faulty element disconnect the masters from the Bus one at a time starting with the IOP(s). After these have been eliminated disconnect the masters of the CPU. If the fault still exists check the back panel wiring.

3.41 BSYN (3A38). This signal is generated by all the masters and the CPU, and it does not leave the CPU chassis. When this signal is low it indicates a faulty master and the masters should be isolated from the Bus one at a time until the fault disappears. The fault is either in the last master to be disconnected or, if all the masters have been eliminated, check the back panel wiring.

3.42 Control Panel

In principle, operation of the READ REGISTER and READ STATUS buttons of the control panel is not able to block the Bus. Operation of the LOAD REGISTER, READ MEMORY, and LOAD MEMORY buttons is able to block the Bus. If the control panel is suspect check the signals from the control panel and in particular the signals RUN and START. The IPL operation can also cause the Bus to block especially with an IOP peripheral. In this case check that the signals OKO, OKI, and SPYC are sent correctly; if they are correct the fault will be found in the IOP.

3.43 Bus Blocks Whilst Program is Running

These faults are almost certainly caused by a timing error of one of the Bus signals. Try to find out if the fault occurs with the Programmed Channel peripherals or the IOP Channel peripherals. If the fault is intermittent it will be necessary to try first one channel and then the other using the appropriate test program. In either case check all Bus signals before changing any major components.

3.44 Condition Register Check

If the Condition Register is suspect try the following routines to check that it is being set to the correct state for each arithmetic operation:

- Load any memory address in register A0 (Program Counter).
- Set the hexadecimal pattern 1101 on the DATA switches and push the LM button to load into memory (this loads the instruction ADK A1,1 into the selected memory location).
- Set the hexadecimal pattern 207F on the DATA switches and push the LM button (this loads a Halt Instruction into memory).
- Set the hexadecimal pattern FFFF on the DATA switches, set the 8 4 2 1 switches to select register A1, then push the LR to load the pattern into the register.
- Load the memory address that contains the ADK instruction into register A0.
- Push the RUN button. The routine will execute the instruction and stop at the halt instruction.
- Push the RST button and read the state of the Condition Register (bits 6 and 7). For the pattern above the state should be 00 indication a Zero result in register A1.
- Repeat the routine using the hexadecimal pattern 80 — the result should be 01 indicating a positive Sign bit in register A1.
- Repeat the routine using the hexadecimal pattern FFFE — the result should be 10 indicating a negative Sign bit in register A1.
- Repeat the routine using the hexadecimal pattern 7FFF — the result should be 11 indicating an Overflow condition in register A1.

3.45 If the expected result is not obtained for any of the patterns, repeat that pattern using a Data Probe to check the cause of the fault.

3.46 ALU Data Path Check

Use the following routine to check the data path logic of the ALU:

- Select a memory address on the DATA switches and load it into register A0.
- Set the DATA switches to hexadecimal pattern 3941 and push the LM button. This loads the instruction SLL A1,1 into memory.

- Set the DATA switches to hexadecimal pattern 207F and push the LM button to load the Halt instruction.
- Set the DATA switches to hexadecimal pattern 5F06 and push the LM button to load the RB instruction.
- Set the DATA switches to hexadecimal pattern 0001. Set the 8 4 2 1 switches to address register A1 and push the LR button to load the pattern into the register.
- Load the memory address that contains the SLL instruction into register A0.
- Set the 8 4 2 1 switches to address register A1.
- Push the RUN button. The routine will execute the instruction and stop at the Halt instruction.
- Push the RR button and check that the contents of register A1 have been shifted one place left.
- Alternately push the RUN and RR buttons and check that the contents of register A1 are shifted one place left for each operation of the buttons until the register contains zeros.
- Repeat the routine using an SRL instruction hexadecimal pattern 3961 and test pattern 1000 in register A1.
- Repeat the routine using a DLC instruction hexadecimal pattern 38C1 and test pattern 0001 loaded in register A2. For this test the contents of both register A1 and A2 should be checked for left shifts.
- Repeat the routine using a DRC instruction hexadecimal pattern 38E1 and test pattern 0001 loaded in register A2. For this test the contents of both register A1 and A2 should be checked for right shifts.

3.47 Control Unit Fault

If the microdiagnostics indicate a CU failure check that the CU address on the DATA switches corresponds to the address set up by the U links on that particular CU card. If the two addresses correspond use the following routine and check the state of the MAD lines and timing signals:

- Load any memory address in register A0.
- Set the DATA switches to either of two hexadecimal patterns: 4980 plus the device address for a TST instruction, or 4180 for a CIO Halt instruction (these two instructions are always accepted by the CU irrespective of the sequensor state); then push the LM button to load the instruction in memory.
- Set the DATA switches to hexadecimal pattern 5F04 then push the LM button to load the RB instruction into memory.
- Load the memory address that contains the I/O instruction into register A0.
- Push the RUN button. The routine will execute the instruction repeatedly (because of the RB instruction) until stopped by pushing the INST button.

Whilst the routine is running check the state of the MAD lines and timing signals both on the Backpanel and on the CU card using a Data Probe. If the fault cannot be found on the CU card addressed by the Bootstrap ROM, remove the other CU cards one at a time until the fault disappears then troubleshoot the last card removed. If the fault persists with all other CU cards removed check the printed circuit backpanel and the operation of the CPU Condition Register.

3.48 SOFTWARE IPL

The type of IPL used by the customer will depend on the system configuration, but all IPLs perform a similar function in that they load programs from a chosed device. The three IPL listings given are all used by the Standard Test Programs: the IPL52S is a Stand Alone program loader which prints out the program IDENT, the IPL52H is similar except it has a Halt feature, and the IPL52N is the same as the IPL52S except it does not print out the program IDENT. Only the IPL52S will be described here and like all IPLs it is in two parts, the Low Core IPL and the High Core IPL.

3.49 Low Core IPL

The Bootstrap loads 80 characters then jumps to location hexadecimal 84 which is the start address of the Low Core IPL. The Low Core IPL has three functions:

- It computes the size of memory and subtracts 400 words which gives the first address of the High Core IPL.
- It modifies the contents of some of the Bootstrap instructions then jumps back to Bootstrap (as necessary) to load the High Core IPL.
- When loading is complete it jumps to the start address of the High Core IPL.

3.50 High Core IPL

Once loaded by the Low Core IPL and Bootstrap the High Core IPL loads the Object Program and enables:

- The peripheral, whose address is on the DATA switches (and has been stored in register A15 by the Bootstrap), to be used as the input device.
- The IDENT of the program to be printed out on Operator's peripheral.
- The program to be loaded into its designated place in memory (overwriting the Bootstrap and Low Core IPL).
- The program to be started either automatically or manually.

3.51 Verification of the Low Core IPL

This can only be carried out if loading stops before the message IDENT 'Name' has been typed out on the Operator's peripheral. To verify that loading is taking place (even if the input device is apparently operating, the Bootstrap may not actually be loading the IPL data) try the following routine:

- Stop the CPU by pushing the INST button.
- Load the hexadecimal value 84 into location 84 (the start address of the Low Core IPL) and try to load the IPL again. If the CPU loops on location 84, the Bootstrap is not loading the IPL and the status of peripheral will be found in register A7.
- If loading takes place beyond location 84 but still stops before the Ident is printed, load hexadecimal 80 into register A0.
- Push the INST button and check the contents of memory by comparing the data displayed against the IPL listing. Repeat until each location of the Low Core IPL has been checked or the faulty location found.

3.52 Verification of the High Core IPL

This can only be carried out if the messages EOS, EOF have not been typed out on the Operator's peripheral. If the program loops in the High Core IPL the loading address +2 will be found in register A11, otherwise compute the loading address by subtracting hexadecimal 400 from the memory size, then use the following routine to find where the loading aborts:

- Stop the CPU by pushing the INST button.
- Stop the CPU by pushing the INST button.
- Load the start address of the High Core IPL into register A0.
- Push the INST button and compare the displayed memory contents with the IPL listing.

3.53 If both the Low Core and High Core IPLs and the program appear to be loading correctly but the program does not start, check the contents of register A0 to find out where the CPU has stopped. This check should indicate whether the program has tried to start or if the fault is due to the IPL. Another possible fault that points to a malfunction of the IPL is if the Operator's peripheral or its CU is malfunctioning.

IPL52S

```
00000                                  IDENT   IPL528
00001                                  ENTRY   IPL88
00002                                  ENTRY   IPL44
00003                          *
00004                          *                                RELIER IPL GENERATION
00005          000C            NBREC   EQU     12
00006                          *
00007          0000            BIPLPR  EQU     *
00008          0000            IPL88   EQU     *
00009  0000    010C                    LDK     A1,NBREC
00010  0002    8220                    LDK.L   A2,BIPL-80
       0004    00B0  R
00011  0006    0785                    LDK     A7,/85                   BASIC WRITE
00012  0008    80A0                    LDKL    A8,DECB
       000A    001E  R
00013          000C            PUNCH   EQU     *
00014  000C    1250                    ADK     A2,80
00015  000E    8241                    ST      A2,DECBUF                SET BUFF ADDR
       0010    0020  R
00016  0012    2804                    LKM                              PUNCH ONE RECORD
00017  0014    0001                    DATA    1
00018  0016    1901                    SUK     A1,1                     COUNT DONE ?
00019  0018    590E                    RB(1)   PUNCH                   NO
00020  001A    2804                    LKM                              YES
00021  001C    0003                    DATA    3                        EXIT
00022                          *
00023          001E            DECB    EQU     *
00024  001E    0003                    DATA    3                        FILE CODE
00025  0020                    DECBUF  RES     1
00026  0022    0050                    DATA    80
00027  0024    0000                    DATA    0
00028  0026    0000                    DATA    0
00029  0028    0000                    DATA    0
00030                                  EJECT
00031                                  RORG    BIPLPR+/100
00032                          *
00033          0100            BIPL    EQU     *
00034                          *                        1ST RECORD = LOW CORE IPL
00035                          *                        LOW CORE IPL,1 RECORD,LOADED AT /80 AND
00036                          *                        STARTED AT /84
00037                          *
00038  0100    FFFF            LDFLG   DATA    /FFFF                    LDFLG = -1 IF JUST LOADED
00039  0102    0000                    DATA    0                        NOT USED
00040          0104            IPL     EQU     *
00041                          *                        ADDR OF IPL = BASE ADDR
00042  0104    82A0                    LDKL    A10,/84
       0106    0084
00043                          *                        CHECK IF JUST LOADED OR NOT
00044  0108    9048                    IM      LDFLG-IPL,A10
       010A    FFFC
00045  010C    511E                    RF(1)   IPL100                   NO,NOT THE 1ST TIME
```

```
00046                    *
00047                    *
00048                    *                                   YES,COMPUTE HIGH CORE LIMIT
00049                    *
00050  010E  8720                LDK.L    A7,/5555            PATTERN
       0110  5555
00051  0112  81A0                LDKL     A9,/FC00            HIGH CORE -/400
       0114  FC00
00052        0116      IPL010    EQU      *
00053  0116  8727                STR      A7,A9
00054  0118  FF26                CWR*     A7,A9               IF THE LOCATION (A9) EXISTS,(A7)=((A9))
00055  011A  5006                RF(0)    IPL020              MATCH
00056  011C  99A0                SUKL     A9,/2000            NO,NEXT LOWER BLOCK OF 4K
       011E  2000
00057  0120  5F0C                RB       IPL010
00058        0122      IPL020    EQU      *
00059  0122  8386                LDR      A11,A9              SAVE BASE ADDR OF HIGH CORE IPL
00060  0124  93A0                ADKL     A11,2               START ADD.
       0126  0002
00061  0128  8486                LDR      A12,A9              SAVE LOAD ADD.
00062  012A  5704                RF       IPL110
00063        012C      IPL100    EQU      *
00064  012C  94A0                ADKL     A12,80
       012E  0050
00065                    * LOAD ADDRESS OF NEXT RECORD
00066        0130      IPL110    EQU      *
00067  0130  871F                LDR      A7,A15              RESTORE CIO INST IN BOOT
00068  0132  273F                ANK      A7,/3F
00069  0134  9720                ADK.L    A7,/41C0            CIO INSTRUCTION
       0136  41C0
00070  0138  872D                STR      A7,A3
00071                    *
00072  013A  0557                LDK      A5,/57              CHANGE LOC. /5A OF BOOT IN ORDER TO
00073  013C  E541                SC       A5,/5A              CANCEL LEADING CHARACTER FLAG
       013E  005A
00074                    *                                   RE INITIALIZE A5,A6
00075  0140  0550                LDK      A5,80               # OF CHARACTERS
00076  0142  8612                LDR      A6,A12              LOAD ADDR.
00077                    *                                  CHECK IF COUNT DONE
00078  0144  904B                IM       CNTFLG=IPL,A10
       0146  0048
00079  0148  890E                ABR(1)   A11
00080  014A  0F42                AB       /42                 START BOOT ABAIN
00081        014C      CNTFLG    EQU      *                   = NUMBER OF RECORDS OF HIGH CORE IPL
00082  014C  FFF5                DATA     1=NBREC
00083  014E  0000                DATA     0
00084  0150  0000                DATA     0
00085  0152  0000                DATA     0
00086  0154  0000                DATA     0
00087  0156  0000                DATA     0
00088  0158  0000                DATA     0
00089  015A  0000                DATA     0
00090  015C  0000                DATA     0
00091  015E  0000                DATA     0
00092  0160  0000                DATA     0
00093  0162  0000                DATA     0
00094  0164  0000                DATA     0
00095  0166  0000                DATA     0
```

```
00096  0168  0000          DATA    0
00097  016A  0000          DATA    0
00098  016C  0000          DATA    0
00099  016E  0000          DATA    0
00100  0170  0000          DATA    0
00101  0172  0000          DATA    0
00102  0174  0000          DATA    0
00103  0176  0000          DATA    0
00104  0178  0000          DATA    0
00105  017A  0000          DATA    0
00106  017C  0000          DATA    0
00107  017E  0000          DATA    0
00108  0180  0000          DATA    0
00109  0182  0000          DATA    0
00110  0184  0000          DATA    0
00111  0186  0000          DATA    0
00112  0188  0000          DATA    0
00113  018A  0000          DATA    0
00114  018C  0000          DATA    0
00115  018E  0000          DATA    0
00116  0190  0000          DATA    0
00117  0192  0000          DATA    0
00118  0194  0000          DATA    0
00119  0196  0000          DATA    0
00120  0198  0000          DATA    0
00121  019A  0000          DATA    0
00122  019C  0000          DATA    0
00123  019E  0000          DATA    0
00124  01A0  0000          DATA    0
00125  01A2  0000          DATA    0
00126  01A4  0000          DATA    0
00127  01A6  0000          DATA    0
00128  01A8  0000          DATA    0
00129  01AA  0000          DATA    0
00130  01AC  0000          DATA    0
00131  01AE  0000          DATA    0
00132  01B0  0000          DATA    0
00133  01B2  0000          DATA    0
00134  01B4  0000          DATA    0
00135  01B6  0000          DATA    0
00136  01B8  0000          DATA    0
00137  01BA  0000          DATA    0
00138  01BC  0000          DATA    0
00139  01BE  0000          DATA    0
00140  01C0  0000          DATA    0
00141  01C2  0000          DATA    0
00142  01C4  0000          DATA    0
00143  01C6  0000          DATA    0
00144  01C8  0000          DATA    0
00145  01CA  0000          DATA    0
00146  01CC  0000          DATA    0
00147  01CE  0000          DATA    0
00148  01D0  0000          DATA    0
00149  01D2  0000          DATA    0
00150  01D4  0000          DATA    0
00151  01D6  0000          DATA    0
00152  01D8  0000          DATA    0
00153  01DA  0000          DATA    0
```

```
00154  01DC  0000              DATA    0
00155  01DE  0000              DATA    0
00156  01E0  0000              DATA    0
00157  01E2  0000              DATA    0
00158  01E4  0000              DATA    0
00159  01E6  0000              DATA    0
00160  01E8  0000              DATA    0
00161  01EA  0000              DATA    0
00162  01EC  0000              DATA    0
00163  01EE  0000              DATA    0
00164  01F0  0000              DATA    0
00165  01F2  0000              DATA    0
00166  01F4  0000              DATA    0
00167  01F6  0000              DATA    0
00168  01F8  0000              DATA    0
00169  01FA  0000              DATA    0
00170  01FC  0000              DATA    0
00171  01FE  0000              DATA    0
00172  0200  0000              DATA    0
00173  0202  0000              DATA    0
00174  0204  0000              DATA    0
00175                          EJECT
00176                                                  HIGH CORE IPL
00177        0000    PTR       EQU     0             DEVICE ADDR WILL BE INITIALIZED BY HCIPL
00178        0010    ASR       EQU     /10
00179        0001    S         EQU     1
00180        0000    H         EQU     0
00181        0001    ROM       EQU     1
00182        0000    SA        EQU     0
00183                *
00184        0000    MODE      EQU     SA            FOR MESSAGE  OUTPUT
00185                          BORG    BIPL+80       1ST RECORD OF HIGH CORE IPL
00186        0150    HCIPL1    EQU     *             1ST WORD OF THE CURRENT RECORD
00187  0150  FEFE              DATA    /FEFE
00188        0152    HCIPL     EQU     *
00189  0152  85A0              LDKL    A13,OUTMSG-HCIPL
       0154  00EA
00190  0156  958E              ADR     A13,A11       LOAD A13 WITH ADDR OF OUTMSG ROUTINE
00191  0158  868E              LDR     A14,A11
00192  015A  96A0              ADKL    A14,CPZON-HCIPL
       015C  00E8
00193                          *       ADDR OF STACK
00194  015E  871E              LDR     A7,A15        GET DEVICE ADDR
00195  0160  87CF              ST      A15,SAVA15-HCIPL,A11  SAVE  4*4  FLAG
       0162  0354
00196  0164  9606              RF(6)   *+8
00197  0166  904F              IM      COREND+2-HCIPL,A11
       0168  0358
00198  016A  9C06              RB(4)   *-4
00199  016C  273F              ANK     A7,/3F
00200                          *                     INIT IO INSTRUCTIONS
00201  016E  974F              ADS     A7,CIOPTR-HCIPL,A11
       0170  0142
00202  0172  974F              ADS     A7,INP2-HCIPL,A11
       0174  0188
00203  0176  974F              ADS     A7,END2-HCIPL,A11
       0178  01CF
00204  017A  974F              ADS     A7,END1-HCIPL,A11
```

```
        017C  0208
00205   017E  974F              ADS     A7,INR04-HCIPL,A11
        0180  01A2
00206   0182  811E              LDR     A1,A15
00207   0184  39C8              SLC     A1,8                    TEST MULTI-SINGLE DEVICE CONT.
00208   0186  5602              BF(6)   HCIPL2
00209   0188  270F              ANK     A7,/F                   MULTI DEVICE CONTROLLER
00210         018A      HCIPL2  EQU     *
00211   018A  974F              AD.S    A7,SSTMLX-HCIPL,A11
        018C  014E
00212   018E  974F              ADS     A7,SSTPR2-HCIPL,A11
        0190  018C
00213   0192  974F              ADS     A7,SSTPTR-HCIPL,A11
        0194  01D0
00214   0196  974F              ADS     A7,SSTPR1-HCIPL,A11
        0198  020A
00215   019A  3F41              SLL     A7,1
00216   019C  974F              AD.S    A7,WER1-HCIPL,A11       INIT WER/RER INST
        019E  013A
00217   01A0  974F              AD.S    A7,WER2-HCIPL,A11
        01A2  013C
00218   01A4  974F              AD.S    A7,RER-HCIPL,A11
        01A6  017C
00219   01A8  57AC              RF      CKOK
00220   01AA  4F42      SYSMSG  DATA    'OBJCT TAPE ON RF'
        01AC  4A43
        01AE  5420
        01B0  5441
        01B2  5045
        01B4  204F
        01B6  4E20
        01B8  5245
00221   01BA  4144              DATA    'ADER, THINK OF B'
        01BC  4552
        01BE  2E20
        01C0  5448
        01C2  494E
        01C4  4B20
        01C6  4F46
        01C8  2042
00222   01CA  4153              DATA    'ASE !'
        01CC  4520
        01CE  2120
00223   01D0  0D0A              DATA    /0D0A
00224         01D2      STAD    EQU     *
00225   01D2  4543      ECMSG   DATA    'EC'
00226   01D4  0D0A              DATA    X'0D0A'
00227   01D6  4F56      OFLMSG  DATA    'OVFR'
        01D8  4652
00228   01DA  0D0A              DATA    /0D0A
00229   01DC            BUFF    RES     39
00230   022A  FFFF              DATA    /FFFF
00231   022C  0000      MASTFG  DATA    0
00232   022E  0000      SAVBAS  DATA    0
00233   0230                    RES     6                       * STACK  AREA
00234         023A      CFZON   EQU     *-2
00235         023C      OUTMSG  EQU     *
```

```
00236  023C  0304            LDK     A3,4
00237  023E  4300            CIO     A3,S,ASR
00238  0240  F324            LCR     A3,A1
00239  0242  4310    OTR     OTR     A3,0,ASR
00240  0244  5C04            RB(4)   *-2
00241  0246  1101            ADK     A1,1
00242  0248  1A01            SUK     A2,1
00243  024A  5C0C            RB(4)   OTR-2
00244  024C  0300            LDK     A3,0
00245  024E  4390            CIO     A3,H,ASR
00246  0250  4B00            SST     A3,ASR
00247  0252  5C04            RB(4)   *-2
00248  0254  F03A            RTN     A14
00249                 *
00250                 *
00251         0256    CKOK    EQU     *
00252  0256  81A0            LDK.L   A9,0                    SET LOADING ADDRESS
       0258  0000
00253                 *
00254                 *
00255                 *                                 GO LOAD SYSTEM
00256  025A  84A0            LDK.L   A12,MNLD-HCIPL
       025C  029C
00257  025E  948F            ADR     A12,A11
00258  0260  F693            CFR     A14,A12
00259                 *
00260                 *
00261  0262  810A            LDR     A1,A10                  TEST START ADDRESS
00262  0264  8C04            ABR(4)  A1                      EOS  OR EOF HAS BEEN READ
00263  0266  207F            HLT                             NO START ADDRESS
00264  0268  0000    BADDR   DATA    0
00265  026A  0300    RAFL    LDK     A3,0
00266  026C  0700            LDK     A7,0
00267  026E  8520            LDK.L   A5,BUFF-HCIPL
       0270  008A
00268  0272  950F            ADR     A5,A11
00269  0274  80CE            LD      A8,SAVA15-HCIPL,A11
       0276  0354
00270  0278  5610            RF(6)   INPA
00271  027A  0604            LDK     A6,4                    4*4 SO ASR
00272  027C  0111            LDK     A1,/11                  SEND X-ON
00273  027E  4600            CIO     A6,S,ASR
00274  0280  4110            OTR     A1,0,ASR
00275  0282  5C04            RB(4)   *-2
00276  0284  4290            CIO     A2,H,ASR
00277  0286  4A00            SST     A2,ASR
00278  0288  5C04            RB(4)   *-2
00279         028A    INPA    EQU     *
00280  028A  0650            LDK     A6,80                   LENGTH
00281  028C  7600    WER1    WER     A6,0                    INIT MLX WHATEVER
00282  028E  7501    WER2    WER     A5,1                    CHANNEL IS
00283  0290  8602            LDR     A6,A8
00284  0292  3E68            SRL     A6,8
00285         0294    CIOPTR  EQU     *
00286  0294  4600            CIO     A6,S,PTR                AVAILABLE ON THE PAPER READER
00287  0296  5C04            RB(4)   *-2
00288  0298  824F            LD      A2,SAVA15-HCIPL,A11
       029A  0354
```

```
00289  029C  3AC3          SLC     A2,3            MLX ?
00290  029E  525A          RF(2)   INP2            NO
00291        02A0  SSTMLX   EQU     *
00292  02A0  49C0          SST     A1,PTR
00293  02A2  5C04          RB(4)   *-2
00294        02A4  TMTEST   EQU     *
00295  02A4  A120          ANKL    A1,/1007        TAPE MARK OR FATAL ERROR ?
       02A6  1007
00296  02A8  501A          RF(0)   MLX1            NO
00297  02AA  2107          ANK     A1,/7           FATAL ERROR ?
00298  02AC  5486          RF(4)   ECCL8           YES
00299                      *
00300  02AE  8120          LDK.L   A1,'E'          PUT EOF IN BUFF
       02B0  3A45
00301  02B2  8135          STR     A1,A5
00302  02B4  8120          LDK.L   A1,'OF'
       02B6  4F46
00303  02B8  8155          ST      A1,2,A5
       02BA  0002
00304  02BC  0704          LDK     A7,4            UPDATA POINTRERS
00305  02BE  951C          ADR     A5,A7
00306  02C0  579E          RF      PRASCI
00307                      *
00308  02C2  5F5A  RAFL3   RB      RAFL            RELAY TOWARDS RAFL
00309                      *
00310        02C4  MLX1     EQU     *
00311  02C4  E134          LCR     A1,A5
00312  02C6  E920          CWK     A1,/18          IS IT ASCII
       02C8  0018
00313  02CA  55EA          RF(5)   RELAY           NO
00314  02CC  0750          LDK     A7,80           YES
00315  02CE  7E00  RER     RER     A6,0            READ REMAINING LENGTH
00316  02D0  A620          ANK.L   A6,/FFF
       02D2  0FFF
00317  02D4  9F18          SUR     A7,A6           COMPUTE TRANSMITTED LGT
00318  02D6  951C          ADR     A5,A7           UPDATE BUFF POINTER
00319  02D8  5786          RF      PRASCI
00320  02DA  4A00  INP2    INR     A2,0,PTR
00321  02DC  50DA          RF(0)   SWITCH          INR HAS BEEN ACCEPTED ;
00322                      *                       PROCESS THE CHARTER
00323  02DE  4AC0  SSTPR2  SST     A2,PTR          TRY A SST
00324  02E0  5C08          RB(4)   INP2            TRY AGAIN INR WHEN SST REFUSED
00325  02E2  8108          LDR     A1,A2
00326  02E4  1300          ADK     A3,0            CHECK WETHER ASCII OR OBJECT
00327  02E6  5278          RF(2)   PRASCI          RECORD WAS ASCII ; PRINT IT
00328  02E8  0600          LDK     A6,0            STORE A NON ASCII CHARACTER AT
00329  02EA  8635          STR     A6,A5           THE END OF BUFFER ,
00330                      *                       TO BE USED IN MX1
00331  02EC  5F4A          RB      TMTEST
00332        02EE  OBJINP   EQU     *
00333  02EE  8082          LDR     A8,A8
00334  02F0  560C          RF(6)   EIGHT           8-8 DO NOT INPUT 2ND CHARACTER
00335  02F2  220F          ANK     A2,/F
00336  02F4  4E00  INR44   INR     A6,0,PTR
00337  02F6  5C04          RB(4)   *-2
00338  02F8  260F          ANK     A6,/F
00339  02FA  3A44          SLL     A2,4
00340  02FC  9218          ADR     A2,A6           JOIN THE TWO HALF CHARACTERS
```

```
00341          02FE    EIGHT   EQU     *
00342  02FE    EF04            CWR     A7,A1
00343  0300    501F            RF(0)   END2
00344  0302    E235            SCR     A2,A5
00345  0304    B408            XRR     A4,A2
00346  0306    1501            ADK     A5,1
00347  0308    EF20            CWK     A7,1
       030A    0001
00348  030C    5408            RF(4)   OBJIN1
00349  030E    0400            LDK     A4,0
00350  0310    8108            LDR     A1,A2
00351  0312    9108            ADR     A1,A2
00352  0314    1103            ADK     A1,3
00353  0316    1701    OBJIN1  ADK     A7,1
00354  0318    5F40    INP2R   RB      INP2
00355  031A    2207    FIRST   ANK     A2,/7
00356  031C    0150            LDK     A1,80
00357  031E    5F22            RB(7)   EIGHT
00358  0320    4280    END2    CIO     A2,H,PTR
00359          0322    SSTPTR  EQU     *
00360  0322    49C0            SST     A1,PTR
00361  0324    5C04            RB(4)   *+2
00362  0326    8082            LDR     A8,A8                   IF ASR WAIT
00363  0328    5606            RF(6)   *+8
00364  032A    904F            TM      COREND+2-HCIPL,A11      WAIT
       032C    0358
00365  032E    5C06            RB(4)   *+4
00366  0330    24FF            ANK     A4,/FF
00367  0332    5004            RF(0)   PROLOI+2
00368          0334    ECCLS   EQU     *
00369  0334    0180            LDK     A1,ECMSG-HCIPL
00370  0336    910F            ADR     A1,A11
00371  0338    0204            LDK     A2,4
00372  033A    84A0            LDK.L   A12,OUTMSG-HCIPL
       033C    00FA
00373  033E    948E            ADR     A12,A11
00374  0340    F693            CFR     A14,A12
00375  0342    207F    STOP    HLT
00376  0344    5FDC            RB(7)   RAFL
00377  0346    EA20    ASCINP  CWK     A2,/0D
       0348    000D
00378  034A    500E            RF(0)   END1
00379  034C    EF20            CWK     A7,68
       034E    0044
00380  0350    5878            RB(0)   INP2
00381  0352    E235            SCR     A2,A5
00382  0354    1501            ADK     A5,1
00383  0356    1701            ADK     A7,1
00384  0358    5F80            RB(7)   INP2
00385  035A    4280    END1    CIO     A2,H,PTR
00386          035C    SSTPR1  EQU     *
00387  035C    4AC0            SST     A2,PTR
00388  035E    5C04            RB(4)   *+2
00389          0360    PRASCI  EQU     *
00390  0360    018A            LDK     A1,BUFF-HCIPL
00391  0362    910E            ADR     A1,A11
00392  0364    821C            LDR     A2,A7
00393  0366    0320            LDK     A3,/20
```

```
0368  E335           SCR      A3,A5
036A  1501           ADK      A5,1
036C  8320           LDK.L    A3,/0D0A
036E  000A
0370  8335           STR      A3,A5
0372  1203           ADK      A2,3
0374  3A61           SRL      A2,1
0376  3A41           SLL      A2,1
0378  8082           LDR      A8,A8                 IF ASR DO NOT OUTPUT ASCII
037A  5608           RF(6)    PRASCA                A SECOND TIME (ALREADY PRINTED WHEN READ
037C  904F           IM       COREND+2-HCIPL,A11    BUT WAIT ABIT
037E  0358
0380  5C06           RB(4)    *-4
0382  5702           RF       PRASCB
      0384    PRASCA EQU      *
0384  F697           CFR      A14,A13
      0386    PRASCB EQU      *
0386  028A           LDK      A2,BUFF-HCIPL
0388  920E           ADR      A2,A11
038A  8328           LDR*     A3,A2
038C  EB20           CWK      A3,/3A45              * :E
038E  3A45
0390  5CD0           RB(4)    RAFL3
0392  1202           ADK      A2,2
0394  8328           LDR*     A3,A2
0396  EB20           CWK      A3,/4F46              *OF
0398  4F46
039A  5CDA           RB(4)    RAFL3
              *
              *
              *                         SET RETURN PARAMETERS
              *
039C  82CE           LD       A10,COREND-HCIPL,A11  START ADDRESS IN A10
039E  0356
03A0  84A0           LDK.L    A12,MNLD-HCIPL
03A2  029C
03A4  948E           ADR      A12,A11               MNLD ADDRESS  IN A12
03A6  81CE           LD       A9,SAVBAS-HCIPL,A11   LOADING BASE  IN A9
03A8  000C
03AA  80CE           LD       A8,BADDR-HCIPL,A11    ENDING ADDRESS IN A8
03AC  0116
03AE  874E           LD       A7,MASTFG-HCIPL,A11   MASTER FLAG IN A7
03B0  000A
              *
              *
03B2  F03A           RTN      A14                   RETURN TO CALLING
              *
03B4  5FF4    RAFL2  RB       RAFL3                 * RELAY TOWARDS RAFL
03B6  5750    RELAY  RF       PROL01+2
              *
03B8  1300    SWITCH ADK      A3,0
03BA  59CE           RB(1)    OBJINP
03BC  227F           ANK      A2,/7F
03BE  58E6           RB(0)    INP2
03C0  FA20           CWK      A2,/7F
03C2  007F
03C4  58EC           RB(0)    INP2
03C6  1300           ADK      A3,0
```

```
00442  03C8  5A84              RB(2)    ASCINP
00443  03CA  FA20              CWK      A2,/1F
       03CC  001F
00444  03CE  5116              RF(1)    ASCII
00445  03D0  FA20              CWK      A2,/14
       03D2  0014
00446  03D4  510C              RF(1)    OBJEC
00447  03D6  EA20              CWK      A2,8
       03D8  0008
00448  03DA  5206              RF(2)    OBJEC
00449  03DC  EA20              CWK      A2,/10
       03DE  0010
00450  03F0  5CCA              RB(4)    INP2R
00451  03E2  1301    OBJEC    ADK      A3,1
00452  03E4  5FCC              RB(7)    FIRST
00453  03E6  1801    ASCII    SUK      A3,1
00454  03E8  5FA4              RB(7)    ASCINP
00455                          EJECT
00456                 *
00457                 *
00458                 *  MAIN  LOADING  PART          ENTRY PARAMETERS :
00459                 *
00460                 *                                 A1 = MASTER FLAG
00461                 *                                 A9 = BASE ADDRESS
00462                 *
00463                 *
00464                 *
00465                 *                 1- OUTPUT MSG REQUESTING  TAPE ON READER
00466                 *                 2- PERFORMS A 'HALT' - THE USER HAS THEN THE
00467                 *                    POSSIBILITY TO ALTER THE BASE ADDRESS (REG,A9)
00468                 *                                AND  THE MASTER FLAG  (REG,A1)
00469                 *                                          (A1=0 MASTER)
00470                 *                                          (A1=1 USER)
00471                 *                 3- LOADING PROCESS STARTS  WHEN USER DEPPRESS START
00472                 *                    BUTTON
00473                 *
00474                 *
00475                 *
00476                 *
00477       03EA      RAFL1    EQU      *
00478  03EA  5F38              RB       RAFL2
00479                 *
00480                 *
00481                 *                              ERRONNOUS CLUSTER,PRINT ERR MESSAGE
00482       03EC      CLC01    EQU      *
00483  03EC  5FBA              RB       ECCLS
00484                 *
00485                 *
00486                 *
00487                 *
00488       03EF      MNLD     EQU      *
00489  03EE  8404              LDR      A4,A1                  SAVE  MASTER FLAG
00490  03F0  0200              LDK      A2,0
00491  03F2  824F              ST       A2,COREND-HCIPL,A11    RA7  START  ADDRESS
       03F4  0356
00492                          TFT      MODE=BOM
00504                          XIF
```

```
00505  03F6  81CF              ST      A9,SAVBAS-HCIPL,A11
       03F8  00DC
00506                  *
00507                  *
00508                  *
00509                  * PROCESS LOADING : THIS  MODULE READ A  CLUSTER
00510                  *          AND  BRANCH ACCORDING TO THE CLUSTER TYPE
00511                  *          ON EXIT   A1 = BUFF  ADDRESS +1
00512                  *                    A2 = WORD  COUNT
00513                  *                    A3 = TYPE
00514                  *          THE TYPE MUST BE    3,4,7   IF NOT THIS ;HALT
00515  03FA  82A0      PROLO   LDK.L   A10,STAD-HCIPL          END ADDRESS
       03FC  0080
00516  03FE  928F              ADR     A10,A11
00517        0000      ABA     EQU     0
00518  0400  81CF              ST      A9,BADDR-HCIPL,A11      BADDR =BASE ADDRESS
       0402  0116
00519  0404  20BF      PROGLD  INH
00520        0406      PROL01  EQU     *
00521  0406  3F1E              RB      RAFL1
00522  0408  8120              LDK.L   A1,BUFF-HCIPL
       040A  008A
00523  040C  910E              ADR     A1,A11
00524  040F  0401              LDK     A4,1
00525  0410  0300              LDK     A3,0
00526  0412  E324              LCR     A3,A1                   A3 = TYPE
00527  0414  1101              ADK     A1,1
00528  0416  0200              LDK     A2,0
00529  0418  E224              LCR     A2,A1                   A2 = WORD COUNT
00530  041A  1101              ADK     A1,1
00531  041C  EB20              CWK     A3,3
       041E  0003
00532  0420  500F              RF(0)   CLCODF                  BRANCH  ON CLUSTER CODE
00533  0422  EB20              CWK     A3,4
       0424  0004
00534  0426  503A              RF(0)   CLIMOD                 INTERNAL MODIFICATION
00535  0428  EB20              CWK     A3,7
       042A  0007
00536  042C  505E              RF(0)   CLEND                  END/START
00537  042E  9F2A              RB(7)   PROL01
00538                  ************************************
00539                  * CLUSTER CODE  TYPE 3
00540                  *          UPON ENTRY: A1=ADDRESS OF BUFF+1 (RBK
00541                  *                      A2=WORD COUNT
00542                  *                      A9=BADDRESS
00543                  *                      A10=ENDADDRESS
00544                  ********************************************
00545  0430  834F      CLCODF  LD      A3,BUFF+6-HCIPL,A11
       0432  0090
00546  0434  5C30              RB(4)   PROL01                  EMBK SET SKIP THE CLUSTER
00547  0436  834E      CLC01A  LD      A3,BUFF+4-HCIPL,A11
       0438  00AF
00548  043A  A311              TM      A3,A4                   IS IT  RELOCATABLE SECTION
00549  043C  5004              RF(0)   CLC04
00550  043E  3301              XRK     A3,1
00551  0440  9306              ADR     A3,A9
00552  0442  8524      CLC04   LDR*    A5,A1                   A5=(RBK)
```

```
00553  0444  1106            ADK     A1,6                A1= ADDRESS OF ST CODE WORD IN BUFF
00554  0446  1A03            SUK     A2,3                A2= NUMBER OF CODE WORD
00555                  *                                A3= STORAGE ADDRESS
00556                  *                                A4= MASK FOR RBK
00557                  *                                A6= CODE WORD
00558  0448  3CF1      CLC05   SRC     A4,1
00559  044A  8624              LDR*    A6,A1
00560  044C  EB0A              CWR     A3,A10              COMPARE LOAD ADDRESS WITH AD OF IPL
00561  044E  5864              RB(0)   CLC01
00562  0450  A511              TM      A5,A4
00563  0452  5002              RF(0)   CLC07
00564  0454  9606              ADR     A6,A9
00565  0456  8620      CLC07   STR     A6,A3               STORE CODE WORDS
00566  0458  1102              ADK     A1,2
00567  045A  1302              ADK     A3,2
00568  045C  1A01              SUK     A2,1
00569  045E  5C18              RB(4)   CLC05
00570  0460  5F68              RB(7)   PROLO
00571                  *
00572                  ****************************************************************
00573                  * INTERNAL MODIFICATION CLUSTER
00574                  ***********************************************
00575  0462  0701      CLIMOD  LDK     A7,1                A7= MASK FOR ADDRESS
00576  0464  8524              LDR*    A5,A1               A5=(RBK)
00577  0466  1A01              SUK     A2,1
00578  0468  3CE1      CLIM1   SRC     A4,1
00579  046A  1102              ADK     A1,2
00580  046C  8324              LDR*    A3,A1               A3=ADDRESS
00581  046E  A310              TM      A3,A7               IS IT RELOCATABLE
00582  0470  5008              RF(0)   CLIM2               NO
00583  0472  3301              XRK     A3,1
00584  0474  9306              ADR     A3,A9               YES AD BASE
00585  0476  EB0A              CWR     A3,A10              ADDRESS OK
00586  0478  588F              RB(0)   CLC01
00587  047A  1102      CLIM2   ADK     A1,2
00588  047C  8624              LDR*    A6,A1               TAKE CODE WORD
00589  047E  A511              TM      A5,A4               IS IT RELOCATABLE
00590  0480  5002              RF(0)   CLIM3
00591  0482  9606              ADR     A6,A9               AD BASE
00592  0484  8620      CLIM3   STR     A6,A3               STORE CODE WORD
00593  0486  1A02              SUK     A2,2
00594  0488  5C22              RB(4)   CLIM1
00595  048A  5F92              RB(7)   PROLO
00596                  ****************************************
00597                  *  CLUSTER  END/START
00598                  ****************************************
00599  048C  8324      CLEND   LDR*    A3,A1
00600  048E  500A              RF(0)   CLEN3A              FINISH WO START
00601  0490  A311              TM      A3,A4
00602  0492  5002              RF(0)   CLEN1
00603  0494  9306              ADR     A3,A9
00604                  *
00605                  *
00606       0496       CLEN1   EQU     *
00607  0496  834F              ST      A3,COREND-HCIPL,A11
       0498  0356
00608  049A  814F      CLEN3A  LD      A1,BUFF+6-HCIPL,A11 UPDATE BASE ADDRESS
       049C  0090
```

```
00609  049E  914F            AD.S    A1,BADDR=HCIPL,A11
       04A0  0116
00610  04A2  9184            ADR     A9,A1
00611  04A4  5FAC            RB      PROLO
00612                  *
00613                  *
00614  04A6  0000      SAVA15  DATA    0
00615                  *
00616                  ********************************************************************************
00617                  *
00618  04A8  0000      COREND  DATA    0
00619                  *
00620                  *
00621                  ********************************************************************************
00622  04AA  0000              DATA    0
00623  04AC  0000              DATA    0
00624  04AE  0000              DATA    0
00625  04B0  0000              DATA    0
00626  04B2  0000              DATA    0
00627  04B4  0000              DATA    0
00628  04B6  0000              DATA    0
00629  04B8  0000              DATA    0
00630  04BA  0000              DATA    0
00631  04BC  0000              DATA    0
00632  04BE  0000              DATA    0
00633  04C0  0000              DATA    0
00634  04C2  0000              DATA    0
00635  04C4  0000              DATA    0
00636  04C6  0000              DATA    0
00637  04C8  0000              DATA    0
00638  04CA  0000              DATA    0
00639  04CC  0000              DATA    0
00640  04CE  0000              DATA    0
00641  04D0  0000              DATA    0
00642  04D2  0000              DATA    0
00643  04D4  0000              DATA    0
00644  04D6  0000              DATA    0
00645  04D8  0000              DATA    0
00646  04DA  0000              DATA    0
00647  04DC  0000              DATA    0
00648  04DE  0000              DATA    0
00649  04E0  0000              DATA    0
00650  04E2  0000              DATA    0
00651  04E4  0000              DATA    0
00652  04E6  0000              DATA    0
00653  04E8  0000              DATA    0
00654  04EA  0000              DATA    0
00655  04EC  0000              DATA    0
00656  04EE  0000              DATA    0
00657  04F0  0000              DATA    0
00658  04F2  0000              DATA    0
00659  04F4  0000              DATA    0
00660  04F6  0000              DATA    0
00661  04F8  0000              DATA    0
00662  04FA  0000              DATA    0
00663  04FC  0000              DATA    0
00664  04FE  0000              DATA    0
00665  0500  0000              DATA    0
```

```
00666  0502  0000                    DATA    0
00667  0504  0000                    DATA    0
00668  0506  0000                    DATA    0
00669  0508  0000                    DATA    0
00670  050A  0000                    DATA    0
00671  050C  0000                    DATA    0
00672  050E  0000                    DATA    0
00673  0510  0000                    DATA    0
00674  0512  0000                    DATA    0
00675  0514  0000                    DATA    0
00676  0516  0000                    DATA    0
00677  0518  0000                    DATA    0
00678  051A  0000                    DATA    0
00679  051C  0000                    DATA    0
00680  051E  0000                    DATA    0
00681  0520  0000                    DATA    0
00682                                EJECT
00683        0522        IPL44       EQU     *
00684  0522  030C                    LDK     A3,NAREC
00685  0524  8420                    LDKL    A4,BTPL
       0526  0100  R
00686  0528  0785                    LDK     A7,/A5
00687                          * BASIC WRITE
00688  052A  80A0                    LDKI    A8,DECB44
       052C  0582  R
00689        052E        PUNCH4      EQU     *
00690  052E  0550                    LDK     A5,80
00691  0530  8620                    LDKL    A6,BUF44
       0532  058E  R
00692        0534        PCH400      EQU     *
00693  0534  0100                    LDK     A1,0
00694  0536  0200                    LDK     A2,0
00695  0538  E130                    LCR     A1,A4
00696  053A  3964                    SRL     A1,4
00697  053C  E921                    CCK     A1,0
       053E  0000
00698  0540  5006                    RF(0)   PCH405
00699  0542  E921                    CCK     A1,/500
       0544  0500
00700  0546  5202                    RF(2)   PCH410          LESS THAN 5
00701        0548        PCH405      EQU     *
00702  0548  2910                    ORK     A1,/10
00703        054A        PCH410      EQU     *
00704  054A  3948                    SLL     A1,8
00705  054C  E130                    LCR     A1,A4
00706  054E  A120                    ANKL    A1,/FF0F
       0550  FF0F
00707  0552  E921                    CCK     A1,0
       0554  0000
00708  0556  5006                    RF(0)   PCH415
00709  0558  E921                    CCK     A1,/500
       055A  0500
00710  055C  5202                    RF(2)   PCH420
00711        055E        PCH415      EQU     *
00712  055E  2910                    ORK     A1,/10
00713        0560        PCH420      EQU     *
00714  0560  8139                    STR     A1,A6
00715                          * STORE WORD
```

```
00716  0562  1602              ADK     A6,2                NEXT WORD IN BUFFER
00717  0564  1401              ADK     A4,1                NEXT CHZR, IN IPL
00718  0566  1D01              SUK     A5,1
00719                     *
00720  0568  5936              RB(1)   PCH400
00721  056A  EB20              CWK     A3,1                LAST RECORD ?
       056C  0001
00722  056E  5406              RF(4)   PCH430              NO
00723  0570  0113              LDK     A1,/13              XOFF
00724  0572  8141              ST      A1,BUF44+158
       0574  062C  R
00725        0576        PCH430  EQU     *
00726  0576  2804              LKM
00727  0578  0001              DATA    1
00728  057A  1801              SUK     A3,1
00729  057C  5950              RB(1)   PUNCH4
00730                     *
00731  057E  2804              LKM
00732  0580  0003              DATA    3
00733                     *
00734        0582        DECB44  EQU     *
00735  0582  0003              DATA    3
00736  0584  058E  R           DATA    BUF44
00737  0586  00A0              DATA    160
00738  0588  0000              DATA    0
00739  058A  0000              DATA    0
00740  058C  0000              DATA    0
00741  058E              BUF44   RES     80
00742                     *
00743                     *
00744                     *
00745                             END     BIPLPR
```

SYMBOL TABLE

```
ABA     0000  A   ASCII   03F6  R   ASCINP  0346  R   ASR     0010  A
BADDR   0268  R   BIPL    0100  R   BIPLPR  0000  R   BOM     0001  A
BUF44   058E  R   BUFF    01DC  R   CFZON   023A  R   CIOPTR  0294  R
CKOK    0256  R   CLCO1   03EC  R   CLCO1A  0436  R   CLCO4   0442  R
CLCO5   0448  R   CLCO7   0456  R   CLCODE  0430  R   CLEN1   0496  R
CLEN3A  049A  R   CLEND   048C  R   CLIM1   0468  R   CLIM2   047A  R
CLIM3   0484  R   CLIMOD  0462  R   CNTFLG  014C  R   COREND  04A8  R
DECB    001E  R   DECB44  0582  R   DECBUF  0020  R   ECCL8   0334  R
ECM8G   01D2  R   EIGHT   02FE  R   END1    035A  R   END2    0320  R
FIRST   031A  R   H       0000  A   HCIPL   0152  R   HCIPL1  0150  R
HCIPL2  018A  R   INP2    02DA  R   INP2R   0318  R   INPA    028A  R
INR44   02F4  R   IPL     0104  R   IPL010  0116  R   IPL020  0122  R
IPL100  012C  R   IPL110  0130  R   IPL44   0522  R   IPL88   0000  R
LDFLG   0100  R   MASTFG  022C  R   MLX1    02C4  R   MNLD    03EE  R
MODE    0000  A   NBREC   000C  A   OBJEC   03E2  R   OBJIN1  0316  R
OBJINP  02EE  R   OFLMSG  01D6  R   OTR     0242  R   OUTM8G  023C  R
PCH400  0534  R   PCH405  0548  R   PCH410  054A  R   PCH415  055E  R
PCH420  0560  R   PCH430  0576  R   PRASCA  0384  R   PRASCB  0386  R
PRASCI  0360  R   PROGLD  0404  R   PROLO   03FA  R   PROLO1  0406  R
PTR     0000  A   PUNCH   000C  R   PUNCH4  052E  R   RAFL    026A  R
RAFL1   03EA  R   RAFL2   0384  R   RAFL3   02C2  R   RELAY   03B6  R
RER     02CE  R   S       0001  A   SA      0000  A   SAVA15  04A6  R
SAVBAS  022E  R   SSTMLX  02A0  R   SSTPR1  039C  R   SSTPR2  02DE  R
SSTPTR  0322  R   STAD    01D2  R   STOP    0342  R   SWITCH  03B8  R
SYSMSG  01AA  R   TMTEST  02A4  R   WER1    028C  R   WER2    028E  R
```

IPL52H

```
00000                          IDENT   IPL52H
00001                          ENTRY   IPL88
00002                          ENTRY   IPL44
00003                  *
00004                  *                       BFI IER IPL GFNERATION
00005          000C    NBREC   EQU     12
00006                  *
00007          0000    BIPLPR  EQU     *
00008          0000    IPL88   EQU     *
00009  0000    010C            LDK     A1,NBREC
00010  0002    8220            LDK.L   A2,BIPL-80
       0004    0080  R
00011  0006    0785            LDK     A7,/85          BASIC WRITF
00012  0008    80A0            LDKL    A8,DECB
       000A    001E  R
00013          000C    PUNCH   EQU     *
00014  000C    1250            ADK     A2,80
00015  000E    8241            ST      A2,DECBUF       SFT BUFF ADDR
       0010    0020  R
00016  0012    2804            LKM                     PUNCH ONF RECORD
00017  0014    0001            DATA    1
00018  0016    1901            SUK     A1,1            COUNT DONE ?
00019  0018    590F            RB(1)   PUNCH          NO
00020  001A    2804            LKM                     YES
00021  001C    0003            DATA    3               EXIT
00022                  *
00023          001F    DECB    EQU     *
00024  001F    0003            DATA    3               FILE CODF
00025  0020            DECBUF  RFS     1
00026  0022    0050            DATA    80
00027  0024    0000            DATA    0
00028  0026    0000            DATA    0
00029  0028    0000            DATA    0
00030                          EJECT
00031                          RORG    BIPLPR+/100
00032                  *
00033          0100    BIPL    EQU     *
00034                  *                       1ST RECORD = LOW CORE IPL
00035                  *                       LOW CORE IPL,1 RECORD,LOADED AT /80 AND
00036                  *                       STARTED AT /84
00037                  *
00038  0100    FFFF    LDFLG   DATA    /FFFF           LDFLG = -1 IF JUST LOADED
00039  0102    0000            DATA    0               NOT USED
00040          0104    IPL     EQU     *
00041                  *                       ADDR OF IPL = BASE ADDR
00042  0104    82A0            LDKL    A10,/84
       0106    0084
00043                  *                       CHFCK IF JUST LOADED OR NOT
00044  0108    904B            IM      LDFLG-IPL,A10
       010A    FFFC
00045  010C    511E            RF(1)   IPL100          NO,NOT THE 1ST TIME
```

```
00046                          *
00047                          *
00048                          *                                YES,COMPUTE HIGH CORE LIMIT
00049                          *
00050  010E  8720                     LDK.L    A7,/5555              PATTERN
       0110  5555
00051  0112  81A0                     LDKL     A9,/FC00              HIGH CORE =/400
       0114  FC00
00052        0116  IPL010    EQU      *
00053  0116  8727                     STR      A7,A9
00054  0118  FF26                     CWR*     A7,A9                 IF THE LOCATION (A9) EXISTS,(A7)=((A9))
00055  011A  5006                     RF(0)    IPL020                MATCH
00056  011C  99A0                     SUKL     A9,/2000              NO,NEXT LOWER BLOCK OF 4K
       011E  2000
00057  0120  5FOC                     RB       IPL010
00058        0122  IPL020    EQU      *
00059  0122  8386                     LDR      A11,A9                SAVE BASE ADDR OF HIGH CORE IPL
00060  0124  93A0                     ADKL     A11,2                 START ADD.
       0126  0002
00061  0128  8486                     LDR      A12,A9                SAVE LOAD ADD.
00062  012A  5704                     RF       IPL110
00063        012C  IPL100    EQU      *
00064  012C  94A0                     ADKL     A12,80
       012E  0050
00065                          * LOAD ADDRESS OF NEXT RECORD
00066        0130  IPL110    EQU      *
00067  0130  871E                     LDR      A7,A15                RESTORE CIO INST IN BOOT
00068  0132  273F                     ANK      A7,/3F
00069  0134  9720                     ADK.L    A7,/41C0              CIO INSTRUCTION
       0136  41C0
00070  0138  872D                     STR      A7,A3
00071                          *
00072  013A  0557                     LDK      A5,/57                CHANGE LOC. /5A OF BOOT IN ORDER TO
00073  013C  F541                     SC       A5,/5A                CANCEL LEADING CHARACTER FLAG
       013E  005A
00074                          *                                RE INITIALIZE A5,A6
00075  0140  0550                     LDK      A5,80                 # OF CHARACTERS
00076  0142  8612                     LDR      A6,A12                LOAD ADDR.
00077                          *                                CHECK IF COUNT DONE
00078  0144  9048                     IM       CNTFLG-IPL,A10
       0146  0048
00079  0148  890E                     ABR(1)   A11
00080  014A  0F42                     AB       /42                   START BOOT ABAIN
00081        014C  CNTFLG    EQU      *                              = NUMBER OF RECORDS OF HIGH CORE IPL
00082  014C  FFF5                     DATA     1-NBREC
00083  014E  0000                     DATA     0
00084  0150  0000                     DATA     0
00085  0152  0000                     DATA     0
00086  0154  0000                     DATA     0
00087  0156  0000                     DATA     0
00088  0158  0000                     DATA     0
00089  015A  0000                     DATA     0
00090  015C  0000                     DATA     0
00091  015E  0000                     DATA     0
00092  0160  0000                     DATA     0
00093  0162  0000                     DATA     0
00094  0164  0000                     DATA     0
00095  0166  0000                     DATA     0
```

```
00096  0168  0000          DATA    0
00097  016A  0000          DATA    0
00098  016C  0000          DATA    0
00099  016E  0000          DATA    0
00100  0170  0000          DATA    0
00101  0172  0000          DATA    0
00102  0174  0000          DATA    0
00103  0176  0000          DATA    0
00104  0178  0000          DATA    0
00105  017A  0000          DATA    0
00106  017C  0000          DATA    0
00107  017E  0000          DATA    0
00108  0180  0000          DATA    0
00109  0182  0000          DATA    0
00110  0184  0000          DATA    0
00111  0186  0000          DATA    0
00112  0188  0000          DATA    0
00113  018A  0000          DATA    0
00114  018C  0000          DATA    0
00115  018E  0000          DATA    0
00116  0190  0000          DATA    0
00117  0192  0000          DATA    0
00118  0194  0000          DATA    0
00119  0196  0000          DATA    0
00120  0198  0000          DATA    0
00121  019A  0000          DATA    0
00122  019C  0000          DATA    0
00123  019E  0000          DATA    0
00124  01A0  0000          DATA    0
00125  01A2  0000          DATA    0
00126  01A4  0000          DATA    0
00127  01A6  0000          DATA    0
00128  01A8  0000          DATA    0
00129  01AA  0000          DATA    0
00130  01AC  0000          DATA    0
00131  01AE  0000          DATA    0
00132  01B0  0000          DATA    0
00133  01B2  0000          DATA    0
00134  01B4  0000          DATA    0
00135  01B6  0000          DATA    0
00136  01B8  0000          DATA    0
00137  01BA  0000          DATA    0
00138  01BC  0000          DATA    0
00139  01BE  0000          DATA    0
00140  01C0  0000          DATA    0
00141  01C2  0000          DATA    0
00142  01C4  0000          DATA    0
00143  01C6  0000          DATA    0
00144  01C8  0000          DATA    0
00145  01CA  0000          DATA    0
00146  01CC  0000          DATA    0
00147  01CE  0000          DATA    0
00148  01D0  0000          DATA    0
00149  01D2  0000          DATA    0
00150  01D4  0000          DATA    0
00151  01D6  0000          DATA    0
00152  01D8  0000          DATA    0
00153  01DA  0000          DATA    0
```

```
00154  01DC  0000              DATA    0
00155  01DE  0000              DATA    0
00156  01E0  0000              DATA    0
00157  01E2  0000              DATA    0
00158  01E4  0000              DATA    0
00159  01E6  0000              DATA    0
00160  01E8  0000              DATA    0
00161  01EA  0000              DATA    0
00162  01EC  0000              DATA    0
00163  01EE  0000              DATA    0
00164  01F0  0000              DATA    0
00165  01F2  0000              DATA    0
00166  01F4  0000              DATA    0
00167  01F6  0000              DATA    0
00168  01F8  0000              DATA    0
00169  01FA  0000              DATA    0
00170  01FC  0000              DATA    0
00171  01FE  0000              DATA    0
00172  0200  0000              DATA    0
00173  0202  0000              DATA    0
00174  0204  0000              DATA    0
00175                          EJECT
00176                                                  HIGH CORE IPL
00177        0000  PTR         EQU     0               DEVICE ADDR WILL BE INITIALIZED BY HCIPL
00178        0010  ASR         EQU     /10
00179        0001  S           EQU     1
00180        0000  H           EQU     0
00181        0001  BOM         EQU     1
00182        0000  SA          EQU     0
00183                          *
00184        0001  MODE        EQU     BOM             FOR MESSAGE  OUTPUT
00185                          RORG    BIPL+80         1ST RECORD OF HIGH CORE IPL
00186        0150  HCIPL1      EQU     *               1ST WORD OF THE CURRENT RECORD
00187  0150  FEFE              DATA    /FEFE
00188        0152  HCIPL       EQU     *
00189  0152  85A0              LDKL    A13,OUTMSG-HCIPL
       0154  00EA
00190  0156  958E              ADR     A13,A11         LOAD A13 WITH ADDR OF OUTMSG ROUTINE
00191  0158  868E              LDR     A14,A11
00192  015A  96A0              ADKL    A14,CFZON-HCIPL
       015C  00F8
00193                          *                       ADDR OF STACK
00194  015E  871E              LDR     A7,A15          GET DEVICE ADDR
00195  0160  87CF              ST      A15,SAVA15-HCIPL,A11  SAVE  4*4  FLAG
       0162  0366
00196  0164  5606              RF(6)   *+8
00197  0166  904F              IM      COREND+2-HCIPL,A11
       0168  036A
00198  016A  5C06              RB(4)   *-4
00199  016C  273F              ANK     A7,/3F
00200                          *                       INIT IO INSTRUCTIONS
00201  016E  974F              ADS     A7,CIOPTR-HCIPL,A11
       0170  0142
00202  0172  974F              ADS     A7,INP2-HCIPL,A11
       0174  0188
00203  0176  974F              ADS     A7,END2-HCIPL,A11
       0178  01CF
```

```
00204  017A  974F            ADS     A7,END1-HCIPL,A11
       017C  0208
00205  017E  974F            ADS     A7,INR44-HCIPL,A11
       0180  01A2
00206  0182  811E            LDR     A1,A15
00207  0184  39C8            SLC     A1,8                    TEST MULTI-SINGLE DEVICE CONT.
00208  0186  5602            RF(6)   HCIPL2
00209  0188  270F            ANK     A7,/F                   MULTI DEVICE CONTROLLER
00210         018A  HCIPL2    EQU     *
00211  018A  974F            AD.8    A7,SSTMLX-HCIPL,A11
       018C  014E
00212  018E  974F            ADS     A7,SSTPR2-HCIPL,A11
       0190  01BC
00213  0192  974F            ADS     A7,SSTPTR-HCIPL,A11
       0194  01D0
00214  0196  974F            ADS     A7,SSTPR1-HCIPL,A11
       0198  020A
00215  019A  3F41            SLL     A7,1
00216  019C  974F            AD.8    A7,WFR1-HCIPL,A11       INIT WER/RFR INST
       019E  013A
00217  01A0  974F            AD.8    A7,WER2-HCIPL,A11
       01A2  013C
00218  01A4  974F            AD.8    A7,RFR-HCIPL,A11
       01A6  017C
00219  01A8  57AC            RF      CKOK
00220  01AA  4F42  SYSMSG    DATA    'OBJCT TAPE ON RF'
       01AC  4A43
       01AE  5420
       01B0  5441
       01B2  5045
       01B4  204F
       01B6  4E20
       01B8  5245
00221  01BA  4144            DATA    'ADER, THINK OF B'
       01BC  4552
       01BE  2E20
       01C0  5448
       01C2  494F
       01C4  4820
       01C6  4F46
       01C8  2042
00222  01CA  4153            DATA    'ASE !'
       01CC  4520
       01CE  2120
00223  01D0  0D0A            DATA    /0D0A
00224         01D2  STAD      EQU     *
00225  01D2  4543  ECMSG     DATA    'EC'
00226  01D4  0D0A            DATA    X'0D0A'
00227  01D6  4F56  OFLMSG    DATA    'OVFR'
       01D8  4652
00228  01DA  0D0A            DATA    /0D0A
00229  01DC        BUFF      RES     39
00230  022A  FFFF            DATA    /FFFF
00231  022C  0000  MASTFG    DATA    0
00232  022E  0000  SAVBAS    DATA    0
00233  0230                  RES     6                       * STACK  AREA
00234         023A  CFZON     EQU     *-2
00235         023C  OUTMSG    EQU     *
```

```
00236  023C  0304            LDK       A3,4
00237  023E  43D0            CIO       A3,S,ASR
00238  0240  F324            LCR       A3,A1
00239  0242  4310  OTR       OTR       A3,0,ASR
00240  0244  5C04            RB(4)     *+2
00241  0246  1101            ADK       A1,1
00242  0248  1A01            SUK       A2,1
00243  024A  5C0C            RB(4)     OTR-2
00244  024C  0300            LDK       A3,0
00245  024E  4390            CIO       A3,H,ASR
00246  0250  4BD0            SST       A3,ASR
00247  0252  5C04            RB(4)     *+2
00248  0254  F03A            RTN       A14
00249                 *
00250                 *
00251        0256  CKOK      EQU       *
00252  0256  81A0            LDK.L     A9,0                    SET LOADING ADDRESS
       0258  0000
00253                 *
00254                 *
00255                 *                                   GO LOAD SYSTEM
00256  025A  84A0            LDK.L     A12,MNLD-HCIPL
       025C  029C
00257  025E  948E            ADR       A12,A11
00258  0260  F693            CFR       A14,A12
00259                 *
00260                 *
00261  0262  810A            LDR       A1,A10                  TEST START ADDRESS
00262  0264  8C04            ABR(4)    A1                      EOS  OR FOF HAS BEEN READ
00263  0266  707F            HLT                               NO START ADDRESS
00264  0268  0000  BADDR     DATA      0
00265  026A  0300  RAFL      LDK       A3,0
00266  026C  0700            LDK       A7,0
00267  026E  8520            LDK.L     A5,BUFF-HCIPL
       0270  008A
00268  0272  950E            ADR       A5,A11
00269  0274  80CE            LD        A8,SAVA15-HCIPL,A11
       0276  0366
00270  0278  5610            RF(6)     INPA
00271  027A  0604            LDK       A6,4                    4*4 SO ASR
00272  027C  0111            LDK       A1,/11                  SEND X-ON
00273  027E  46D0            CIO       A6,S,ASR
00274  0280  4110            OTR       A1,0,ASR
00275  0282  5C04            RB(4)     *+2
00276  0284  4290            CIO       A2,H,ASR
00277  0286  4AD0            SST       A2,ASR
00278  0288  5C04            RB(4)     *+2
00279        028A  INPA      EQU       *
00280  028A  0650            LDK       A6,80                   LENGTH
00281  028C  7600  WER1      WER       A6,0                    INIT MLX WHATEVER
00282  028E  7501  WER2      WER       A5,1                    CHANNEL IS
00283  0290  8602            LDR       A6,A8
00284  0292  3E68            SRL       A6,8
00285        0294  CIOPTR    EQU       *
00286  0294  46C0            CIO       A6,S,PTR                AVAILABLE ON THE PAPER READER
00287  0296  5C04            RB(4)     *+2
00288  0298  824E            LD        A2,SAVA15-HCIPL,A11
       029A  0366
```

```
00289  029C  3AC3          SLC     A2,3            MLX ?
00290  029E  9234          RF(?)   INP2            NO
00291        02A0   SSTMLX  EQU     *
00292  02A0  49C0          SST     A1,PTR
00293  02A2  9C04          RB(4)   *+2
00294        02A4   TMTEST  EQU     *
00295  02A4  A120          ANKL    A1,/1007        TAPE MARK OR FATAL ERROR ?
       02A6  1007
00296  02A8  501A          RF(0)   MLX1            NO
00297  02AA  2107          ANK     A1,/7           FATAL FRROR ?
00298  02AC  5486          RF(4)   ECCLS           YES
00299                      *
00300  02AE  8120          LDK.L   A1,'1E'         PUT EOF IN BUFF
       02B0  3A45
00301  02B2  8135          STR     A1,A5
00302  02B4  8120          LDK.L   A1,'0F'
       02B6  4F46
00303  02B8  8159          ST      A1,2,A5
       02BA  0002
00304  02BC  0704          LDK     A7,4            UPDATA POINTRERS
00305  02BE  951C          ADR     A5,A7
00306  02C0  979E          RF      PRASCI
00307                      *
00308  02C2  9F5A   RAFL3  RB      RAFL            RELAY TOWARDS RAFL
00309                      *
00310        02C4   MLX1   EQU     *
00311  02C4  E134          LCR     A1,A5
00312  02C6  E920          CWK     A1,/18          IS IT ASCII
       02C8  0018
00313  02CA  95FA          RF(5)   RELAY           NO
00314  02CC  0750          LDK     A7,80           YES
00315  02CE  7E00   RER    RER     A6,0            READ REMAINING LENGTH
00316  02D0  A620          ANK.L   A6,/FFF
       02D2  0FFF
00317  02D4  9F18          SUR     A7,A6           COMPUTE TRANSMITTED LGT
00318  02D6  951C          ADR     A5,A7           UPDATE BUFF POINTER
00319  02D8  97A6          RF      PRASCI
00320  02DA  4A00   INP2   INR     A2,0,PTR
00321  02DC  500A          RF(0)   SWITCH          INR HAS BEEN ACCEPTED ;
00322                      *                       PROCESS THE CHARACTER
00323  02DE  4AC0   SSTPR2 SST     A2,PTR          TRY A SST
00324  02E0  9C08          RB(4)   INP2            TRY AGAIN INR WHEN SST REFUSED
00325  02E2  8108          LDR     A1,A2           SAVE STATUS IN A1
00326  02E4  1300          ADK     A3,0            CHECK WETHER ASCII OR OBJECT
00327  02E6  5278          RF(2)   PRASCI          RECORD WAS ASCII ; PRINT IT
00328  02E8  0600          LQK     A6,0            STORE A NON ASCII CHARACTER AT
00329  02EA  8635          STR     A6,A5           THE END OF BUFFER ;
00330                      *                       TO BE USED IN MX1
00331  02EC  9F4A          RB      TMTEST
00332        02EE   OBJINP EQU     *
00333  02EE  8082          LDR     A8,A8
00334  02F0  960C          RF(6)   EIGHT           8=8 DO NOT INPUT 2ND CHARACTER
00335  02F2  220F          ANK     A2,/F
00336  02F4  4E00   INR44  INR     A6,0,PTR
00337  02F6  9C04          RB(4)   *+2
00338  02F8  260F          ANK     A6,/F
00339  02FA  3A44          SLL     A2,4
00340  02FC  9218          ADR     A2,A6           JOIN THE TWO HALF CHARACTERS
```

```
00341          02FE    EIGHT   EQU     *
00342  02FE    EF04            CWR     A7,A1
00343  0300    501E            RF(0)   END2
00344  0302    E235            SCR     A2,A5
00345  0304    B408            XRR     A4,A2
00346  0306    1501            ADK     A5,1
00347  0308    EF20            CWK     A7,1
       030A    0001
00348  030C    5408            RF(4)   OBJIN1
00349  030E    0400            LDK     A4,0
00350  0310    8108            LDR     A1,A2
00351  0312    9108            ADR     A1,A2
00352  0314    1103            ADK     A1,3
00353  0316    1701    OBJIN1  ADK     A7,1
00354  0318    5F40    INP2R   RB      INP2
00355  031A    2207    FIRST   ANK     A2,/7
00356  031C    0150            LDK     A1,80
00357  031E    5F22            RB(7)   EIGHT
00358  0320    4280    END2    CIO     A2,H,PTR
00359          0322    SSTPTR  EQU     *
00360  0322    49C0            SST     A1,PTR
00361  0324    5C04            RB(4)   *-2
00362  0326    8082            LDR     A8,A8                   IF ASR WAIT
00363  0328    5606            RF(6)   *+8
00364  032A    904F            IM      COREND+2-HCIPL,A11      WAIT
       032C    036A
00365  032E    5C06            RB(4)   *-4
00366  0330    24FF            ANK     A4,/FF
00367  0332    50E6            RF(0)   PROL01+2
00368          0334    FCCLS   EQU     *
00369  0334    0180            LDK     A1,ECMSG-HCIPL
00370  0336    910F            ADR     A1,A11
00371  0338    0204            LDK     A2,4
00372  033A    84A0            LDK.L   A12,OUTMSG-HCIPL
       033C    00EA
00373  033E    948E            ADR     A12,A11
00374  0340    E693            CFR     A14,A12
00375  0342    207F    STOP    HLT
00376  0344    5FDC            RB(7)   RAFL
00377  0346    EA20    ASCINP  CWK     A2,/0D
       0348    000D
00378  034A    500F            RF(0)   END1
00379  034C    EF20            CWK     A7,68
       034E    0044
00380  0350    5878            RB(0)   INP2
00381  0352    E235            SCR     A2,A5
00382  0354    1501            ADK     A5,1
00383  0356    1701            ADK     A7,1
00384  0358    5F80            RB(7)   INP2
00385  035A    4280    END1    CIO     A2,H,PTR
00386          035C    SSTPR1  EQU     *
00387  035C    4AC0            SST     A2,PTR
00388  035E    5C04            RB(4)   *-2
00389          0360    PRASCI  EQU     *
00390  0360    018A            LDK     A1,BUFF-HCIPL
00391  0362    910F            ADR     A1,A11
00392  0364    821C            LDR     A2,A7
00393  0366    0320            LDK     A3,/20
```

```
00394  0368  F335            SCR     A3,A5
00395  036A  1501            ADK     A5,1
00396  036C  8320            LDK.L   A3,/0D0A
       036E  0D0A
00397  0370  8335            STR     A3,A5
00398  0372  1203            ADK     A2,3
00399  0374  3A61            SRL     A2,1
00400  0376  3A41            SLL     A2,1
00401  0378  8082            LDR     A8,A8                   IF ASR DO NOT OUTPUT ASCII
00402  037A  9608            RF(6)   PRASCA                  A SECOND TIME (ALREADY PRINTED WHEN READ
00403  037C  904F            IM      COREND+2-HCIPL,A11      BUT WAIT A BIT
       037E  036A
00404  0380  5C06            RB(4)   *-4
00405  0382  5702            RF      PRASCB
00406        0384      PRASCA  EQU     *
00407  0384  5700            RF      *+2                     ****************
00408        0386      PRASCB  EQU     *
00409  0386  028A            LDK     A2,BUFF-HCIPL
00410  0388  920E            ADR     A2,A11
00411  038A  8328            LDR*    A3,A2
00412  038C  EB20            CWK     A3,/3A45                * IE
       038E  3A45
00413  0390  5CD0            RB(4)   RAFL3
00414  0392  1202            ADK     A2,2
00415  0394  8328            LDR*    A3,A2
00416  0396  EB20            CWK     A3,/4F46                *OF
       0398  4F46
00417  039A  5CDA            RB(4)   RAFL3
00418                  *
00419                  *
00420                  *                       SET RETURN PARAMETERS
00421                  *
00422  039C  82CF            LD      A10,COREND-HCIPL,A11    START ADDRESS IN A10
       039F  0368
00423  03A0  84A0            LDK.L   A12,MNLD-HCIPL
       03A2  029C
00424  03A4  948E            ADR     A12,A11                 MNLD ADDRESS  IN A12
00425  03A6  81CF            LD      A9,SAVBAS-HCIPL,A11     LOADING BASE  IN A9
       03A8  00DC
00426  03AA  80CE            LD      A8,BADDR-HCIPL,A11      ENDING ADDRESS IN A8
       03AC  0116
00427  03AE  874F            LD      A7,MASTFG-HCIPL,A11     MASTER FLAG IN A7
       03B0  00DA
00428                  *
00429                  *
00430  03B2  F03A            RTN     A14                     RETURN TO CALLING
00431                  *
00432  03B4  5FF4      RAFL2   RB      RAFL3                   * RELAY TOWARDS RAFL
00433  03B6  5762      RELAY   RF      PROLOI+2
00434                  *
00435  03B8  1300      SWITCH  ADK     A3,0
00436  03BA  59CF              RB(1)   OBJINP
00437  03BC  227F              ANK     A2,/7F
00438  03BE  58E6              RB(0)   INP2
00439  03C0  EA20              CWK     A2,/7F
       03C2  007F
00440  03C4  58EC              RB(0)   INP2
00441  03C6  1300              ADK     A3,0
```

```
00442  03C8  5A84              RB(2)     ASCINP
00443  03CA  EA20              CWK       A2,/1F
       03CC  001F
00444  03CE  5116              RF(1)     ASCII
00445  03D0  EA20              CWK       A2,/14
       03D2  0014
00446  03D4  510C              RF(1)     OBJEC
00447  03D6  EA20              CWK       A2,8
       03D8  0008
00448  03DA  5206              RF(2)     OBJEC
00449  03DC  EA20              CWK       A2,/10
       03DE  0010
00450  03E0  5CCA              RB(4)     INP2R
00451  03E2  1301    OBJEC    ADK       A3,1
00452  03E4  5FCC              RB(7)     FIRST
00453  03E6  1801    ASCII    SUK       A3,1
00454  03E8  5FA4              RB(7)     ASCINP
00455                          EJECT
00456                 *
00457                 *
00458                 *  MAIN  LOADING  PART          ENTRY PARAMETERS :
00459                 *
00460                 *                                 A1 = MASTER FLAG
00461                 *                                 A9 = BASE ADDRESS
00462                 *
00463                 *
00464                 *
00465                 *                    1- OUTPUT MSG REQUESTING  TAPE ON READER
00466                 *                    2- PERFORMS A 'HALT' - THE USER HAS THEN THE
00467                 *                       POSSIBILITY TO ALTER THE BASE ADDRESS (REG,A9)
00468                 *                                  AND  THE MASTER FLAG  (REG,A1)
00469                 *                                                  (A1=0 MASTER)
00470                 *                                                  (A1=1 USER)
00471                 *                    3- LOADING PROCESS STARTS  WHEN USER DEPPRESS START
00472                 *                       BUTTON
00473                 *
00474                 *
00475                 *
00476                 *
00477         03EA    RAFL1    EQU       *
00478  03EA   5F38             RB        RAFL2
00479                 *
00480                 *
00481                 *                                 ERRONNOUS CLUSTER,PRINT ERR MESSAGE
00482         03EC    CLC01    EQU       *
00483  03EC   5F8A             RB        ECCLS
00484                 *
00485                 *
00486                 *
00487                 *
00488         03EE    MNLD     EQU       *
00489  03EE   8404             LDR       A4,A1                  SAVE  MASTER FLAG
00490  03F0   0200             LDK       A2,0
00491  03F2   824F             ST        A2,COREND-HCIPL,A11    RAZ  START  ADDRESS
       03F4   0368
00492                          IFT,      MODE=BOM
00493  03F6   8120             LDK.L     A1,SYSMSG-HCIPL
       03F8   0058
```

```
00494  03FA  910E            ADR       A1,A11
00495  03FC  0228            LDK       A2,/28
00496  03FE  5700            RF        **2                    ****************
00497                  *
00498                  *
00499  0400  8110            LDR       A1,A4                  RESTORE MASTER FLAG
00500                  *
00501  0402  207F            HLT                              HALT FOR EVENTUAL REGISTERS MODIF
00502                  *
00503  0404  814F            ST        A1,MASTFG-HCIPL,A11
       0406  00DA
00504                        XIF
00505  0408  81CF            ST        A9,SAVBAS-HCIPL,A11
       040A  00DC
00506                  *
00507                  *
00508                  *
00509                  * PROCESS LOADING : THIS  MODULE READ A  CLUSTER
00510                  *          AND  BRANCH ACCORDING TO THE CLUSTER TYPE
00511                  *          ON EXIT   A1 = BUFF  ADDRESS +1
00512                  *                    A2 = WORD  COUNT
00513                  *                    A3 = TYPE
00514                  *          THE TYPE MUST BE    3,4,7   IF NOT THIS :HALT
00515  040C  82A0      PROLO     LDK.L     A10,STAD-HCIPL         END ADDRESS
       040E  0080
00516  0410  928F                ADR       A10,A11
00517        0000      ABA       EQU       0
00518  0412  81CF                ST        A9,BADDR-HCIPL,A11     BADDR =BASE ADDRESS
       0414  0116
00519  0416  208F      PROGID    INH
00520        0418      PROL01    EQU       *
00521  0418  5F30                RB        RAFL1
00522  041A  8120                LDK.L     A1,BUFF-HCIPL
       041C  008A
00523  041E  910F                ADR       A1,A11
00524  0420  0401                LDK       A4,1
00525  0422  0300                LDK       A3,0
00526  0424  E324                LCR       A3,A1                  A3 = TYPE
00527  0426  1101                ADK       A1,1
00528  0428  0200                LDK       A2,0
00529  042A  E224                LCR       A2,A1                  A2 = WORD COUNT
00530  042C  1101                ADK       A1,1
00531  042E  EB20                CWK       A3,3
       0430  0003
00532  0432  500E                RF(0)     CLCODE                 BRANCH  ON CLUSTER CODE
00533  0434  EB20                CWK       A3,4
       0436  0004
00534  0438  503A                RF(0)     CLIMOD                 INTERNAL MODIFICATION
00535  043A  EB20                CWK       A3,7
       043C  0007
00536  043E  505F                RF(0)     CLEND                  END/START
00537  0440  5F2A                RB(7)     PROL01
00538                  ************************************
00539                  * CLUSTER CODE  TYPE 3
00540                  *          UPON ENTRY: A1=ADDRESS OF BUFF+1 (RBK
00541                  *                      A2=WORD COUNT
00542                  *                      A9=BADDRESS
00543                  *                      A10=ENDADDRESS
00544                  ********************************************
```

```
00545  0442  834E      CLCODE   LD       A3,BUFF+6-HCIPL,A11
       0444  0090
00546  0446  5C30               RB(4)    PROL01                    EMBK SET SKIP THE CLUSTER
00547  0448  834E      CLC01A   LD       A3,BUFF+4-HCIPL,A11
       044A  008E
00548  044C  A311               TM       A3,A4                     IS IT  RELOCATABLE SECTION
00549  044E  5004               RF(0)    CLC04
00550  0450  3301               XRK      A3,1
00551  0452  9306               ADR      A3,A9
00552  0454  8524      CLC04    LDR*     A5,A1                     A5=(RBK)
00553  0456  1106               ADK      A1,6                      A1= ADDRESS OF 8T CODE WORD IN BUFF
00554  0458  1A03               SUK      A2,3                      A2= NUMBER OF CODE WORD
00555                    *                                  A3= STORAGE ADDRESS
00556                    *                                  A4= MASK FOR RBK
00557                    *                                  A6= CODE WORD
00558  045A  3CE1      CLC05    SRC      A4,1
00559  045C  8624               LDR*     A6,A1
00560  045E  EB0A               CWR      A3,A10                    COMPARE LOAD ADDRESS WITH AD OF IPL
00561  0460  5876               RB(0)    CLC01
00562  0462  A511               TM       A5,A4
00563  0464  5002               RF(0)    CLC07
00564  0466  9606               ADR      A6,A9
00565  0468  862D      CLC07    STR      A6,A3                     STORE CODE WORDS
00566  046A  1102               ADK      A1,2
00567  046C  1302               ADK      A3,2
00568  046E  1A01               SUK      A2,1
00569  0470  5C18               RB(4)    CLC05
00570  0472  5F68               RB(7)    PROL0
00571                    *,
00572                    ********************************************************************
00573                    * INTERNAL MODIFICATION CLUSTER
00574                    ********************************************
00575  0474  0701      CLIMOD   LDK      A7,1                      A7= MASK FOR ADDRESS
00576  0476  8524               LDR*     A5,A1                     A5=(RBK)
00577  0478  1A01               SUK      A2,1
00578  047A  3CE1      CLIM1    SRC      A4,1
00579  047C  1102               ADK      A1,2
00580  047E  8324               LDR*     A3,A1                     A3=ADDRESS
00581  0480  A31D               TM       A3,A7                     IS IT RELOCATABLE
00582  0482  5008               RF(0)    CLIM2                     NO
00583  0484  3301               XRK      A3,1
00584  0486  9306               ADR      A3,A9                     YES AD BASE
00585  0488  EB0A               CWR      A3,A10                    ADDRESS OK
00586  048A  58A0               RB(0)    CLC01
00587  048C  1102      CLIM2    ADK      A1,2
00588  048E  8624               LDR*     A6,A1                     TAKE CODE WORD
00589  0490  A511               TM       A5,A4                     IS IT RELOCATABLE
00590  0492  5002               RF(0)    CLIM3
00591  0494  9606               ADR      A6,A9                     AD BASE
00592  0496  862D      CLIM3    STR      A6,A3                     STORE CODE WORD
00593  0498  1A02               SUK      A2,2
00594  049A  5C22               RB(4)    CLIM1
00595  049C  5F92               RB(7)    PROL0
00596                    ******************************************
00597                    *  CLUSTER  END/START
00598                    ******************************************
00599  049E  8324      CLEND    LDR*     A3,A1
```

```
00600  04A0  500A              RF(0)     CLEN3A                FINISH WO START
00601  04A2  A311              TM        A3,A4
00602  04A4  5002              RF(0)     CLEN1
00603  04A6  9306              ADR       A3,A9
00604                    *
00605                    *
00606        04A8        CLEN1   EQU       *
00607  04A8  834F              ST        A3,COREND-HCIPL,A11
       04AA  0368
00608  04AC  814E        CLEN3A  LD        A1,BUFF+6-HCIPL,A11   UPDATE BASE ADDRESS
       04AE  0090
00609  04B0  914F              AD.S      A1,BADDR-HCIPL,A11
       04B2  0116
00610  04B4  9184              ADR       A9,A1
00611  04B6  5FAC              RB        PROLO
00612                    *
00613                    *
00614  04B8  0000        SAVA15  DATA      0
00615                    *
00616                    ********************************************************************************
00617                    *
00618  04BA  0000        COREND  DATA      0
00619                    *
00620                    *
00621                    ********************************************************************************
00622  04BC  0000              DATA      0
00623  04BE  0000              DATA      0
00624  04C0  0000              DATA      0
00625  04C2  0000              DATA      0
00626  04C4  0000              DATA      0
00627  04C6  0000              DATA      0
00628  04C8  0000              DATA      0
00629  04CA  0000              DATA      0
00630  04CC  0000              DATA      0
00631  04CE  0000              DATA      0
00632  04D0  0000              DATA      0
00633  04D2  0000              DATA      0
00634  04D4  0000              DATA      0
00635  04D6  0000              DATA      0
00636  04D8  0000              DATA      0
00637  04DA  0000              DATA      0
00638  04DC  0000              DATA      0
00639  04DE  0000              DATA      0
00640  04E0  0000              DATA      0
00641  04E2  0000              DATA      0
00642  04E4  0000              DATA      0
00643  04E6  0000              DATA      0
00644  04E8  0000              DATA      0
00645  04EA  0000              DATA      0
00646  04EC  0000              DATA      0
00647  04EE  0000              DATA      0
00648  04F0  0000              DATA      0
00649  04F2  0000              DATA      0
00650  04F4  0000              DATA      0
00651  04F6  0000              DATA      0
00652  04F8  0000              DATA      0
00653  04FA  0000              DATA      0
```

```
00654  04FC  0000                    DATA    0
00655  04FE  0000                    DATA    0
00656  0500  0000                    DATA    0
00657  0502  0000                    DATA    0
00658  0504  0000                    DATA    0
00659  0506  0000                    DATA    0
00660  0508  0000                    DATA    0
00661  050A  0000                    DATA    0
00662  050C  0000                    DATA    0
00663  050E  0000                    DATA    0
00664  0510  0000                    DATA    0
00665  0512  0000                    DATA    0
00666  0514  0000                    DATA    0
00667  0516  0000                    DATA    0
00668  0518  0000                    DATA    0
00669  051A  0000                    DATA    0
00670  051C  0000                    DATA    0
00671  051E  0000                    DATA    0
00672  0520  0000                    DATA    0
00673  0522  0000                    DATA    0
00674  0524  0000                    DATA    0
00675  0526  0000                    DATA    0
00676  0528  0000                    DATA    0
00677  052A  0000                    DATA    0
00678  052C  0000                    DATA    0
00679  052E  0000                    DATA    0
00680  0530  0000                    DATA    0
00681  0532  0000                    DATA    0
00682                                EJECT
00683         0534       IPL44       EQU     *
00684  0534   030C                    LDK     A3,NBREC
00685  0536   8420                    LDKL    A4,BIPL
       0538   0100  R
00686  053A   0785                    LDK     A7,/85
00687                          * BASIC WRITE
00688  053C   80A0                    LDKL    A8,DFCB44
       053E   0594  R
00689         0540       PUNCH4      EQU     *
00690  0540   0550                    LDK     A5,80
00691  0542   8620                    LDKL    A6,BUF44
       0544   05A0  R
00692         0546       PCH400      EQU     *
00693  0546   0100                    LDK     A1,0
00694  0548   0200                    LDK     A2,0
00695  054A   E130                    LCR     A1,A4
00696  054C   3964                    SRL     A1,4
00697  054E   F921                    CCK     A1,0
       0550   0000
00698  0552   5006                    RF(0)   PCH405
00699  0554   F921                    CCK     A1,/500
       0556   0500
00700  0558   9202                    RF(2)   PCH410          LESS THAN 5
00701         055A       PCH405      EQU     *
00702  055A   2910                    ORK     A1,/10
00703         055C       PCH410      EQU     *
00704  055C   3948                    SLL     A1,8
00705  055E   E130                    LCR     A1,A4
```

```
00706  0560  A120              ANKL    A1,/FF0F
       0562  FF0F
00707  0564  E921              CCK     A1,0
       0566  0000
00708  0568  5006              RF(0)   PCH415
00709  056A  E921              CCK     A1,/500
       056C  0500
00710  056E  5202              RF(2)   PCH420
00711        0570      PCH415  EQU     *
00712  0570  2910              ORK     A1,/10
00713        0572      PCH420  EQU     *
00714  0572  8139              STR     A1,A6
00715                * STORE WORD
00716  0574  1602              ADK     A6,2            NEXT WORD IN BUFFER
00717  0576  1401              ADK     A4,1            NEXT CHZR, IN TPL
00718  0578  1D01              SUK     A5,1
00719                *
00720  057A  5936              RB(1)   PCH400
00721  057C  EB20              CWK     A3,1            LAST RECORD ?
       057E  0001
00722  0580  5406              RF(4)   PCH430          NO
00723  0582  0113              LDK     A1,/13          XOFF
00724  0584  8141              ST      A1,BUF44+158
       0586  063E  R
00725        0588      PCH430  EQU     *
00726  0588  2804              LKM
00727  058A  0001              DATA    1
00728  058C  1B01              SUK     A3,1
00729  058E  5950              RB(1)   PUNCH4
00730                *
00731  0590  2804              LKM
00732  0592  0003              DATA    3
00733                *
00734        0594      DECB44  EQU     *
00735  0594  0003              DATA    3
00736  0596  05A0  R           DATA    BUF44
00737  0598  00A0              DATA    160
00738  059A  0000              DATA    0
00739  059C  0000              DATA    0
00740  059E  0000              DATA    0
00741  05A0              BUF44   RF8     80
00742                *
00743                *
00744                *
00745                           END     BIPLPR
```

SYMBOL TABLE

ABA	0000	A	ASCII	03F6	R	ASCINP	0346	R	ASR	0010	A
BADDR	0268	R	BIPL	0100	R	BIPLPR	0000	R	BOM	0001	A
BUF44	05A0	R	BUFF	01DC	R	CFZON	023A	R	CIOPTR	0294	R
CKOK	0256	R	CLC01	03FC	R	CLC01A	0448	R	CLC04	0454	R
CLC05	045A	R	CLC07	0468	R	CLCODE	0442	R	CLEN1	04A8	R
CLEN3A	04AC	R	CLEND	049E	R	CLIM1	047A	R	CLIM2	048C	R
CLIM3	0496	R	CLIMOD	0474	R	CNTFLG	014C	R	COREND	04BA	R
DECB	001E	R	DECB44	0594	R	DECBUF	0020	R	ECCLS	0334	R
ECMSG	01D2	R	EIGHT	02FE	R	END1	035A	R	END2	0320	R
FIRST	031A	R	H	0000	A	HCIPL	0152	R	HCIPL1	0150	R
HCIPL2	018A	R	INP2	02DA	R	INP2R	0318	R	INPA	028A	R
INR44	02F4	R	IPL	0104	R	IPL010	0116	R	IPL020	0122	R
IPL100	012C	R	IPL110	0130	R	IPL44	0534	R	IPL88	0000	R
LDFLG	0100	R	MASTFG	022C	R	MLX1	02C4	R	MNID	03FF	R
MODE	0001	A	NBREC	000C	A	OBJEC	03E2	R	OBJIN1	0316	R
OBJINP	02EE	R	OFLMSG	01D6	R	OTR	0242	R	OUTMSG	023C	R
PCH400	0546	R	PCH405	055A	R	PCH410	056C	R	PCH415	0570	R
PCH420	0572	R	PCH430	0588	R	PRASCA	0384	R	PRASCB	0386	R
PRASCI	0360	R	PROGLD	0416	R	PROLD	040C	R	PROL01	0418	R
PTR	0000	A	PUNCH	000C	R	PUNCH4	0540	R	RAFL	026A	R
RAFL1	03FA	R	RAFL2	03B4	R	RAFL3	02C2	R	RELAY	03B6	R
RER	02CE	R	S	0001	A	SA	0000	A	SAVA15	04B8	R
SAVBAS	022E	R	SSTMLX	02A0	R	SSTPR1	035C	R	SSTPR2	02DE	R
SSTPTR	0322	R	STAD	01D2	R	STOP	0342	R	SWITCH	03B8	R
SYSMSG	01AA	R	TMTEST	02A4	R	WER1	028C	R	WER2	028E	R

IPL52N

```
00000                          IDENT   IPL52N
00001                          ENTRY   IPL88
00002                          ENTRY   IPL44
00003                  *
00004                  *                       BELIER IPL GENERATION
00005          000C    NBREC   EQU     12
00006                  *
00007          0000    BIPLPR  EQU     *
00008          0000    IPL88   EQU     *
00009  0000    010C            LDK     A1,NBREC
00010  0002    8220            LDK.L   A2,BIPL-80
       0004    0080  R
00011  0006    0785            LDK     A7,/85                  BASIC WRITE
00012  0008    80A0            LDKL    A8,DECB
       000A    001E  R
00013          000C    PUNCH   EQU     *
00014  000C    1250            ADK     A2,80
00015  000E    8241            ST      A2,DECBUF               SET BUFF ADDR
       0010    0020  R
00016  0012    2804            LKM                             PUNCH ONE RECORD
00017  0014    0001            DATA    1
00018  0016    1901            SUK     A1,1                    COUNT DONE ?
00019  0018    590F            RB(1)   PUNCH                   NO
00020  001A    2804            LKM                             YES
00021  001C    0003            DATA    3                       EXIT
00022                  *
00023          001E    DECB    EQU     *
00024  001E    0003            DATA    3                       FILE CODE
00025  0020            DECBUF  RES     1
00026  0022    0050            DATA    80
00027  0024    0000            DATA    0
00028  0026    0000            DATA    0
00029  0028    0000            DATA    0
00030                          EJECT
00031                          RORG    BIPLPR+/100
00032                  *
00033          0100    BIPL    EQU     *
00034                  *                       1ST RECORD = LOW CORE IPL
00035                  *                       LOW CORE IPL,1 RECORD,LOADED AT /80 AND
00036                  *                       STARTED AT /84
00037                  *
00038  0100    FFFF    LDFLG   DATA    /FFFF                   LDFLG = -1 IF JUST LOADED
00039  0102    0000            DATA    0                       NOT USED
00040          0104    IPL     EQU     *
00041                  *                       ADDR OF IPL = BASE ADDR
00042  0104    8240            LDKL    A10,/84
       0106    0084
00043                  *                       CHECK IF JUST LOADED OR NOT
00044  0108    9048            IM      LDFLG-IPL,A10
       010A    FFFC
00045  010C    511E            RF(1)   IPL100                  NO,NOT THE 1ST TIME
```

```
00046                          *
00047                          *
00048                          *                                YES,COMPUTE HIGH CORE LIMIT
00049                          *
00050  010E  8720                     LDK.L    A7,/5555              PATTERN
       0110  5555
00051  0112  81A0                     LDKL     A9,/FC00              HIGH CORE -/400
       0114  FC00
00052        0116  IPL010    EQU      *
00053  0116  8727                     STR      A7,A9
00054  0118  EF26                     CWR*     A7,A9                 IF THE LOCATION (A9) EXISTS,(A7)=((A9))
00055  011A  5006                     RF(0)    IPL020                MATCH
00056  011C  99A0                     SUKL     A9,/2000              NO,NEXT LOWER BLOCK OF 4K
       011E  2000
00057  0120  5F0C                     RR       IPL010
00058        0122  IPL020    EQU      *
00059  0122  8386                     LDR      A11,A9                SAVE BASE ADDR OF HIGH CORE IPL
00060  0124  93A0                     ADKL     A11,2                 START ADD.
       0126  0002
00061  0128  8486                     LDR      A12,A9                SAVE LOAD ADD.
00062  012A  5704                     RF       IPL110
00063        012C  IPL100    EQU      *
00064  012C  94A0                     ADKL     A12,80
       012E  0050
00065                          * LOAD ADDRESS OF NEXT RECORD
00066        0130  IPL110    EQU      *
00067  0130  871E                     LDR      A7,A15                RESTORE CIO INST IN BOOT
00068  0132  273F                     ANK      A7,/3F
00069  0134  9720                     ADK.L    A7,/41C0              CIO INSTRUCTION
       0136  41C0
00070  0138  872D                     STR      A7,A3
00071                          *
00072  013A  0557                     LDK      A5,/57                CHANGE LOC. /5A OF BOOT IN ORDER TO
00073  013C  E541                     SC       A5,/5A                CANCEL LEADING CHARACTER FLAG
       013E  005A
00074                          *                                RE INITIALIZE A5,A6
00075  0140  0550                     LDK      A5,80                 # OF CHARACTERS
00076  0142  8612                     LDR      A6,A12                LOAD ADDR.
00077                          *                                CHECK IF COUNT DONE
00078  0144  9048                     IM       CNTFLG-IPL,A10
       0146  0048
00079  0148  890E                     ABR(1)   A11
00080  014A  0F42                     AB       /42                   START BOOT ABAIN
00081        014C  CNTFLG    EQU      *                              = NUMBER OF RECORDS OF HIGH CORE IPL
00082  014C  FFF5                     DATA     1-NBREC
00083  014E  0000                     DATA     0
00084  0150  0000                     DATA     0
00085  0152  0000                     DATA     0
00086  0154  0000                     DATA     0
00087  0156  0000                     DATA     0
00088  0158  0000                     DATA     0
00089  015A  0000                     DATA     0
00090  015C  0000                     DATA     0
00091  015E  0000                     DATA     0
00092  0160  0000                     DATA     0
00093  0162  0000                     DATA     0
00094  0164  0000                     DATA     0
00095  0166  0000                     DATA     0
```

```
00096  0168  0000          DATA    0
00097  016A  0000          DATA    0
00098  016C  0000          DATA    0
00099  016E  0000          DATA    0
00100  0170  0000          DATA    0
00101  0172  0000          DATA    0
00102  0174  0000          DATA    0
00103  0176  0000          DATA    0
00104  0178  0000          DATA    0
00105  017A  0000          DATA    0
00106  017C  0000          DATA    0
00107  017E  0000          DATA    0
00108  0180  0000          DATA    0
00109  0182  0000          DATA    0
00110  0184  0000          DATA    0
00111  0186  0000          DATA    0
00112  0188  0000          DATA    0
00113  018A  0000          DATA    0
00114  018C  0000          DATA    0
00115  018F  0000          DATA    0
00116  0190  0000          DATA    0
00117  0192  0000          DATA    0
00118  0194  0000          DATA    0
00119  0196  0000          DATA    0
00120  0198  0000          DATA    0
00121  019A  0000          DATA    0
00122  019C  0000          DATA    0
00123  019E  0000          DATA    0
00124  01A0  0000          DATA    0
00125  01A2  0000          DATA    0
00126  01A4  0000          DATA    0
00127  01A6  0000          DATA    0
00128  01A8  0000          DATA    0
00129  01AA  0000          DATA    0
00130  01AC  0000          DATA    0
00131  01AE  0000          DATA    0
00132  01B0  0000          DATA    0
00133  01B2  0000          DATA    0
00134  01B4  0000          DATA    0
00135  01B6  0000          DATA    0
00136  01B8  0000          DATA    0
00137  01BA  0000          DATA    0
00138  01BC  0000          DATA    0
00139  01BE  0000          DATA    0
00140  01C0  0000          DATA    0
00141  01C2  0000          DATA    0
00142  01C4  0000          DATA    0
00143  01C6  0000          DATA    0
00144  01C8  0000          DATA    0
00145  01CA  0000          DATA    0
00146  01CC  0000          DATA    0
00147  01CE  0000          DATA    0
00148  01D0  0000          DATA    0
00149  01D2  0000          DATA    0
00150  01D4  0000          DATA    0
00151  01D6  0000          DATA    0
00152  01D8  0000          DATA    0
00153  01DA  0000          DATA    0
```

```
00154  01DC  0000              DATA    0
00155  01DE  0000              DATA    0
00156  01E0  0000              DATA    0
00157  01E2  0000              DATA    0
00158  01E4  0000              DATA    0
00159  01E6  0000              DATA    0
00160  01E8  0000              DATA    0
00161  01EA  0000              DATA    0
00162  01EC  0000              DATA    0
00163  01EE  0000              DATA    0
00164  01F0  0000              DATA    0
00165  01F2  0000              DATA    0
00166  01F4  0000              DATA    0
00167  01F6  0000              DATA    0
00168  01F8  0000              DATA    0
00169  01FA  0000              DATA    0
00170  01FC  0000              DATA    0
00171  01FE  0000              DATA    0
00172  0200  0000              DATA    0
00173  0202  0000              DATA    0
00174  0204  0000              DATA    0
00175                          EJECT
00176                                                  HIGH CORE IPL
00177        0000      PTR     EQU     0               DEVICE ADDR WILL BE INITIALIZED BY HCIPL
00178        0010      ASR     EQU     /10
00179        0001      S       EQU     1
00180        0000      H       EQU     0
00181        0001      BOM     EQU     1
00182        0000      SA      EQU     0
00183                  *
00184        0000      MODE    EQU     SA              FOR MESSAGE  OUTPUT
00185                          RORG    BIPL+80         1ST RECORD OF HIGH CORE IPL
00186        0150      HCIPL1  EQU     *               1ST WORD OF THE CURRENT RECORD
00187  0150  FEFE              DATA    /FEFE
00188        0152      HCIPL   EQU     *
00189  0152  85A0              LDKI    A13,OUTMSG-HCIPL
       0154  00FA
00190  0156  958F              ADR     A13,A11         LOAD A13 WITH ADDR OF OUTMSG ROUTINE
00191  0158  868F              LDR     A14,A11
00192  015A  96A0              ADKL    A14,CFZON-HCIPL
       015C  00F8
00193                  *               ADDR OF STACK
00194  015E  871F              LDR     A7,A15          GET DEVICE ADDR
00195  0160  87CF              ST      A15,SAVA15-HCIPL,A11  SAVE  4*4  FLAG
       0162  0354
00196  0164  5606              BF(6)   *+8
00197  0166  904F              IM      COREND+2-HCIPL,A11
       0168  0358
00198  016A  5C06              BB(4)   *-4
00199  016C  273F              ANK     A7,/3F
00200                  *                       INIT IO INSTRUCTIONS
00201  016E  974F              ADS     A7,CIOPTR-HCIPL,A11
       0170  0142
00202  0172  974F              ADS     A7,INP2-HCIPL,A11
       0174  0188
00203  0176  974F              ADS     A7,END2-HCIPL,A11
       0178  01CE
```

```
00204  017A  974F            ADS     A7,END1-HCIPL,A11
       017C  0208
00205  017E  974F            ADS     A7,INR44-HCIPL,A11
       0180  01A2
00206  0182  811F            LDR     A1,A15
00207  0184  39C8            SLC     A1,8                    TEST MULTI-SINGLE DEVICE CONT.
00208  0186  5602            RF(6)   HCIPL2
00209  0188  270F            ANK     A7,/F                   MULTI DEVICE CONTROLLER
00210         018A  HCIPL2    EQU     *
00211  018A  974F            AD.S    A7,SSTMLX-HCIPL,A11
       018C  014E
00212  018E  974F            ADS     A7,SSTPR2-HCIPL,A11
       0190  018C
00213  0192  974F            ADS     A7,SSTPTR-HCIPL,A11
       0194  01D0
00214  0196  974F            ADS     A7,SSTPR1-HCIPL,A11
       0198  020A
00215  019A  3F41            SLL     A7,1
00216  019C  974F            AD.S    A7,WPR1-HCIPL,A11       INIT WER/RER INST
       019E  013A
00217  01A0  974F            AD.S    A7,WER2-HCIPL,A11
       01A2  013C
00218  01A4  974F            AD.S    A7,RER-HCIPL,A11
       01A6  017C
00219  01A8  57AC            RF      CKOK
00220  01AA  4F42  SYSMSG    DATA    'OBJCT TAPE ON RF'
       01AC  4A43
       01AE  5420
       01B0  5441
       01B2  5045
       01B4  204F
       01B6  4E20
       01B8  5245
00221  01BA  4144            DATA    'ADER, THINK OF B'
       01BC  4552
       01BE  2E20
       01C0  5448
       01C2  494E
       01C4  4B20
       01C6  4F46
       01C8  2042
00222  01CA  4153            DATA    'ASE 1'
       01CC  4520
       01CE  3120
00223  01D0  0D0A            DATA    /0D0A
00224         01D2  STAD      EQU     *
00225  01D2  4543  ECMSG     DATA    'EC'
00226  01D4  0D0A            DATA    X'0D0A'
00227  01D6  4F56  OFLMSG    DATA    'OVFR'
       01D8  4652
00228  01DA  0D0A            DATA    /0D0A
00229  01DC        BUFF      RES     39
00230  022A  FFFF            DATA    /FFFF
00231  022C  0000  MASTFG    DATA    0
00232  022E  0000  SAVBAS    DATA    0
00233  0230                  RES     6                       * STACK  AREA
00234         023A  CFZON     EQU     *-2
00235         023C  OUTMSG    EQU     *
```

```
00236  023C  0304              LDK     A3,4
00237  023E  43D0              CIO     A3,S,ASR
00238  0240  E324              LCR     A3,A1
00239  0242  4310    OTR       OTR     A3,0,ASR
00240  0244  5C04              RB(4)   *-2
00241  0246  1101              ADK     A1,1
00242  0248  1A01              SUK     A2,1
00243  024A  5C0C              RB(4)   OTR-2
00244  024C  0300              LDK     A3,0
00245  024E  4390              CIO     A3,H,ASR
00246  0250  4BD0              SST     A3,ASR
00247  0252  5C04              RB(4)   *-2
00248  0254  F03A              RTN     A14
00249                  *
00250                  *
00251         0256    CKOK      EQU     *
00252  0256  81A0              LDK.L   A9,0                    SET LOADING ADDRESS
       0258  0000
00253                  *
00254                  *
00255                  *                                  GO LOAD SYSTEM
00256  025A  84A0              LDK.L   A12,MNLD-HCIPL
       025C  029C
00257  025E  948E              ADR     A12,A11
00258  0260  F693              CFR     A14,A12
00259                  *
00260                  *
00261  0262  810A              LDR     A1,A10                  TEST START ADDRESS
00262  0264  8C04              ABR(4)  A1                      EOS  OR EOF HAS BEEN READ
00263  0266  207F              HLT                             NO START ADDRESS
00264  0268  0000    BADDR     DATA    0
00265  026A  0300    RAFL      LDK     A3,0
00266  026C  0700              LDK     A7,0
00267  026E  8520              LDK.L   A5,BUFF-HCIPL
       0270  008A
00268  0272  950E              ADR     A5,A11
00269  0274  80CE              LD      A8,SAVA15-HCIPL,A11
       0276  0354
00270  0278  5610              RF(6)   INPA
00271  027A  0604              LDK     A6,4                    4*4 SO ASR
00272  027C  0111              LDK     A1,/11                  SEND X-ON
00273  027E  46D0              CIO     A6,S,ASR
00274  0280  4110              OTR     A1,0,ASR
00275  0282  5C04              RB(4)   *-2
00276  0284  4290              CIO     A2,H,ASR
00277  0286  4AD0              SST     A2,ASR
00278  0288  5C04              RB(4)   *-2
00279         028A    INPA      EQU     *
00280  028A  0650              LDK     A6,80                   LENGTH
00281  028C  7600    WER1      WER     A6,0                    INIT MLX WHATEVER
00282  028E  7501    WER2      WER     A5,1                    CHANNEL IS
00283  0290  8602              LDR     A6,A8
00284  0292  3E68              SRL     A6,8
00285         0294    CIOPTR    EQU     *
00286  0294  46C0              CIO     A6,S,PTR                AVAILABLE ON THE PAPER READER
00287  0296  5C04              RB(4)   *-2
00288  0298  824E              LD      A2,SAVA15-HCIPL,A11
       029A  0354
```

```
00289  029C  3AC3          SLC     A2,3            MLX ?
00290  029E  523A          RF(2)   INP2            NO
00291        02A0  SSTMLX   EQU     *
00292  02A0  49C0          SST     A1,PTR
00293  02A2  5C04          RB(4)   *+2
00294        02A4  TMTEST   EQU     *
00295  02A4  A120          ANKI    A1,/1007        TAPE MARK OR FATAL ERROR ?
       02A6  1007
00296  02A8  501A          RF(0)   MLX1            NO
00297  02AA  2107          ANK     A1,/7           FATAL ERROR ?
00298  02AC  5486          RF(4)   ECCLS           YES
00299                      *
00300  02AE  8120          LDK.L   A1,'EF'         PUT EOF IN BUFF
       02B0  3A45
00301  02B2  8135          STR     A1,A5
00302  02B4  8120          LDK.L   A1,'OF'
       02B6  4F46
00303  02B8  8155          ST      A1,2,A5
       02BA  0002
00304  02BC  0704          LDK     A7,4            UPDATA POINTRERS
00305  02BE  951C          ADR     A5,A7
00306  02C0  579E          RF      PRASCI
00307                      *
00308  02C2  5F5A  RAFL3   RB      RAFL            RELAY TOWARDS RAFL
00309                      *
00310        02C4  MLX1     EQU     *
00311  02C4  E134          LCR     A1,A5
00312  02C6  F920          CWK     A1,/18          IS IT ASCII
       02C8  0018
00313  02CA  55EA          RF(5)   RELAY           NO
00314  02CC  0750          LDK     A7,80           YES
00315  02CE  7E00  RFR     RER     A6,0            READ REMAINING LENGTH
00316  02D0  A620          ANK.L   A6,/FFF
       02D2  0FFF
00317  02D4  9F18          SUR     A7,A6           COMPUTE TRANSMITTED LGT
00318  02D6  951C          ADR     A5,A7           UPDATE BUFF POINTER
00319  02D8  5786          RF      PRASCI
00320  02DA  4A0C  INP2    INR     A2,0,PTR
00321  02DC  50DA          RF(0)   SWITCH          INR HAS BEEN ACCEPTED ;
00322                      *                       PROCESS THE CHARTER
00323  02DE  4AC0  SSTPR2  SST     A2,PTR          TRY A SST
00324  02E0  5C08          RB(4)   INP2            TRY AGAIN INR WHEN SST REFUSED
00325  02E2  8108          LDR     A1,A2
00326  02E4  1300          ADK     A3,0            CHECK WETHER ASCII OR OBJECT
00327  02E6  5278          RF(2)   PRASCI          RECORD WAS ASCII ; PRINT IT
00328  02E8  0600          LDK     A6,0            STORE A NON ASCII CHARACTER AT
00329  02EA  8695          STR     A6,A5           THE END OF BUFFER ,
00330                      *                       TO BE USED IN MX1
00331  02EC  5F4A          RB      TMTFST
00332        02EE  OBJINP   EQU     *
00333  02EE  8082          LDR     A8,A8
00334  02F0  560C          RF(6)   EIGHT           8=8 DO NOT INPUT 2ND CHARACTER
00335  02F2  220F          ANK     A2,/F
00336  02F4  4E00  INR44   INR     A6,0,PTR
00337  02F6  5C04          RB(4)   *+2
00338  02F8  260F          ANK     A6,/F
00339  02FA  3A44          SLL     A2,4
```

```
00340  02FC  9218            ADR     A2,A6                 JOIN THE TWO HALF CHARACTERS
00341        02FE  FF04  EIGHT EQU     *
00342  02FE  FF04            CWR     A7,A1
00343  0300  501E            RF(0)   END2
00344  0302  F235            SCR     A2,A5
00345  0304  8408            XRR     A4,A2
00346  0306  1501            ADK     A5,1
00347  0308  EF20            CWK     A7,1
       030A  0001
00348  030C  9408            RF(4)   OBJIN1
00349  030E  0400            LDK     A4,0
00350  0310  8108            LDR     A1,A2
00351  0312  9108            ADR     A1,A2
00352  0314  1103            ADK     A1,3
00353  0316  1701  OBJIN1    ADK     A7,1
00354  0318  5F40  INP2R     RB      INP2
00355  031A  2207  FIRST     ANK     A2,/7
00356  031C  0150            LDK     A1,80
00357  031E  5F22            RB(7)   EIGHT
00358  0320  4280  END2      CIO     A2,H,PTR
00359        0322  SSTPTR    EQU     *
00360  0322  49C0            SST     A1,PTR
00361  0324  5C04            RB(4)   *+2
00362  0326  8082            LDR     A8,A8                 IF A8R WAIT
00363  0328  5606            RF(6)   *+8
00364  032A  904F            IM      COREND+2-HCIPL,A11    WAIT
       032C  0358
00365  032E  5C06            RB(4)   *-4
00366  0330  24FF            ANK     A4,/FF
00367  0332  50D4            RF(0)   PROLO1+2
00368        0334  ECCLS     EQU     *
00369  0334  0180            LDK     A1,ECMSG-HCIPL
00370  0336  910F            ADR     A1,A11
00371  0338  0204            LDK     A2,4
00372  033A  84A0            LDK.L   A12,OUTMSG-HCIPL
       033C  00FA
00373  033E  948E            ADR     A12,A11
00374  0340  F693            CFR     A14,A12
00375  0342  207F  STOP      HLT
00376  0344  5FDC            RB(7)   RAFL
00377  0346  FA20  ASCINP    CWK     A2,/0D
       0348  000D
00378  034A  500E            RF(0)   END1
00379  034C  EF20            CWK     A7,68
       034E  0044
00380  0350  5878            RB(0)   INP2
00381  0352  F235            SCR     A2,A5
00382  0354  1501            ADK     A5,1
00383  0356  1701            ADK     A7,1
00384  0358  5F80            RB(7)   INP2
00385  035A  4280  END1      CIO     A2,H,PTR
00386        035C  SSTPR1    EQU     *
00387  035C  4AC0            SST     A2,PTR
00388  035E  5C04            RB(4)   *-2
00389        0360  PRASCI    EQU     *
00390  0360  018A            LDK     A1,BUFF-HCIPL
00391  0362  910F            ADR     A1,A11
00392  0364  821C            LDR     A2,A7
00393  0366  0320            LDK     A3,/20
```

```
00394  0368  E335              SCR     A3,A5
00395  036A  1501              ADK     A5,1
00396  036C  8320              LDK.L   A3,/0D0A
       036E  0D0A
00397  0370  8335              STR     A3,A5
00398  0372  1203              ADK     A2,3
00399  0374  3A61              SRL     A2,1
00400  0376  3A41              SLL     A2,1
00401  0378  8082              LDR     A8,A8                   IF ASR DO NOT OUTPUT ASCII
00402  037A  5608              RF(6)   PRASCA                  A SECOND TIME (ALREADY PRINTED WHEN READ
00403  037C  904F              IM      COREND+2-HCIPL,A11      BUT WAIT ABIT
       037E  0358
00404  0380  5C06              RB(4)   *+4
00405  0382  5702              RF      PRASCB
00406        0384      PRASCA  EQU     *
00407  0384  1100              ADK     A1,0                    PASS WAS CFR A14,A13 IN IPL52S
00408                  * TO AVOID PRINT MESSAGE
00409        0386      PRASCB  EQU     *
00410  0386  028A              LDK     A2,BUFF-HCIPL
00411  0388  920F              ADR     A2,A11
00412  038A  8328              LDR*    A3,A2
00413  038C  EB20              CWK     A3,/3A45                * IF
       038E  3A45
00414  0390  5CD0              RB(4)   RAFL3
00415  0392  1202              ADK     A2,2
00416  0394  8328              LDR*    A3,A2
00417  0396  EB20              CWK     A3,/4F46                *OF
       0398  4F46
00418  039A  5CDA              RB(4)   RAFL3
00419                  *
00420                  *
00421                  *                                       SET RETURN PARAMETERS
00422                  *
00423  039C  82CE              LD      A10,COREND-HCIPL,A11    START ADDRESS IN A10
       039E  0356
00424  03A0  84A0              LDK.L   A12,MNLD-HCIPL
       03A2  029C
00425  03A4  948E              ADR     A12,A11                 MNLD ADDRESS  IN A12
00426  03A6  81CE              LD      A9,SAVBAS-HCIPL,A11     LOADING BASE  IN A9
       03A8  00DC
00427  03AA  80CE              LD      A8,BADDR-HCIPL,A11      ENDING ADDRESS IN A8
       03AC  0116
00428  03AE  874E              LD      A7,MASTFG-HCIPL,A11     MASTER FLAG IN A7
       03B0  000A
00429                  *
00430                  *
00431  03B2  F03A              RTN     A14                     RETURN TO CALLING
00432                  *
00433  03B4  5FF4      RAFL2   RB      RAFL3                   * RELAY TOWARDS RAFL
00434  03B6  5750      RELAY   RF      PROLO1+2
00435                  *
00436  03B8  1300      SWITCH  ADK     A3,0
00437  03BA  99CF              RB(1)   OBJINP
00438  03BC  227F              ANK     A2,/7F
00439  03BE  58F6              RB(0)   INP2
00440  03C0  EA20              CWK     A2,/7F
       03C2  007F
00441  03C4  58FC              RB(0)   INP2
```

```
00442  03C6  1300              ADK       A3,0
00443  03C8  5A84              RB(2)     ASCINP
00444  03CA  EA20              CWK       A2,/1F
       03CC  001F
00445  03CE  5116              RF(1)     ASCII
00446  03D0  EA20              CWK       A2,/14
       03D2  0014
00447  03D4  510C              RF(1)     OBJEC
00448  03D6  EA20              CWK       A2,8
       03D8  0008
00449  03DA  5206              RF(2)     OBJEC
00450  03DC  EA20              CWK       A2,/10
       03DE  0010
00451  03F0  9CCA              RB(4)     INP2R
00452  03E2  1301    OBJEC    ADK       A3,1
00453  03F4  5FCC              RB(7)     FIRST
00454  03E6  1801    ASCII    SUK       A3,1
00455  03F8  5FA4              RB(7)     ASCINP
00456                          EJECT
00457                 *
00458                 *
00459                 *  MAIN  LOADING  PART          ENTRY PARAMETERS :
00460                 *
00461                 *                                 A1 = MASTER FLAG
00462                 *                                 A9 = BASE ADDRESS
00463                 *
00464                 *
00465                 *
00466                 *                    1- OUTPUT MSG REQUESTING  TAPE ON READER
00467                 *                    2- PERFORMS A 'HALT' - THE USER HAS THEN THE
00468                 *                       POSSIBILITY TO ALTER THE BASE ADDRESS (REG,A9)
00469                 *                                  AND  THE MASTER FLAG  (REG,A1)
00470                 *                                                  (A1=0 MASTER)
00471                 *                                                  (A1=1 USER)
00472                 *                    3- LOADING PROCESS STARTS  WHEN USER DEPPRESS START
00473                 *                       BUTTON
00474                 *
00475                 *
00476                 *
00477                 *
00478         03FA    RAFL1    EQU       *
00479  03FA   5F38             RB        RAFL2
00480                 *
00481                 *
00482                 *                                 ERRONNOUS CLUSTER,PRINT ERR MESSAGE
00483         03EC    CLC01    EQU       *
00484  03FC   5FBA             RB        ECCLS
00485                 *
00486                 *
00487                 *
00488                 *
00489         03FF    MNLD     EQU       *
00490  03EE   8404             LDR       A4,A1                SAVE  MASTER FLAG
00491  03F0   0200             LDK       A2,0
00492  03F2   824F             ST        A2,COREND-HCIPL,A11  RAZ  START  ADDRESS
       03F4   0356
00493                          IFT       MODE=BOM
00505                          XIF
```

```
00506  03F6  81CF              ST       A9,SAVBAS-HCIPL,A11
       03F8  000C
00507                  *
00508                  *
00509                  *
00510                  * PROCESS LOADING : THIS  MODULE READ A  CLUSTER
00511                  *          AND  BRANCH ACCORDING TO THE CLUSTER TYPE
00512                  *          ON EXIT   A1 = BUFF  ADDRESS +1
00513                  *                    A2 = WORD  COUNT
00514                  *                    A3 = TYPE
00515                  *          THE TYPE MUST BE    3,4,7   IF NOT THIS :HALT
00516  03FA  82A0      PROLO    LDK.L    A10,STAD-HCIPL          END ADDRESS
       03FC  0080
00517  03FE  928F              ADR      A10,A11
00518        0000      ABA      EQU      0
00519  0400  81CF              ST       A9,BADDR-HCIPL,A11      BADDR =BASE ADDRESS
       0402  0116
00520  0404  208F      PROGID   INH
00521        0406      PROL01   EQU      *
00522  0406  9F1F              RB       RAFL1
00523  0408  8120              LDK.L    A1,BUFF-HCIPL
       040A  008A
00524  040C  910F              ADR      A1,A11
00525  040E  0401              LDK      A4,1
00526  0410  0300              LDK      A3,0
00527  0412  E324              LCR      A3,A1                   A3 = TYPE
00528  0414  1101              ADK      A1,1
00529  0416  0200              LDK      A2,0
00530  0418  E224              LCR      A2,A1                   A2 = WORD COUNT
00531  041A  1101              ADK      A1,1
00532  041C  EB20              CWK      A3,3
       041E  0003
00533  0420  500F              RF(0)    CLCODE                  BRANCH  ON CLUSTER CODE
00534  0422  EB20              CWK      A3,4
       0424  0004
00535  0426  503A              RF(0)    CLIMOD                  INTERNAL MODIFICATION
00536  0428  EB20              CWK      A3,7
       042A  0007
00537  042C  505E              RF(0)    CLEND                   END/START
00538  042E  9F2A              RB(7)    PROL01
00539                  **********************************
00540                  * CLUSTER CODE  TYPE 3
00541                  *          UPON ENTRY: A1=ADDRESS OF BUFF+1 (RBK
00542                  *                      A2=WORD COUNT
00543                  *                      A9=BADDRESS
00544                  *                      A10=ENDADDRESS
00545                  ********************************************
00546  0430  834F      CLCODE   LD       A3,BUFF+6-HCIPL,A11
       0432  0090
00547  0434  9C30              RB(4)    PROL01                  EMBK SET SKIP THE CLUSTER
00548  0436  834F      CLC01A   LD       A3,BUFF+4-HCIPL,A11
       0438  008E
00549  043A  A311              TM       A3,A4                   IS IT  RELOCATABLE SECTION
00550  043C  5004              RF(0)    CLC04
00551  043E  3301              XRK      A3,1
00552  0440  9306              ADR      A3,A9
00553  0442  8524      CLC04    LDR*     A5,A1                   A5=(RBK)
00554  0444  1106              ADK      A1,6                    A1= ADDRESS OF ST CODE WORD IN BUFF
```

```
00555  0446  1A03              SUK     A2,3                    A2= NUMBER OF CODE WORD
00556                    *                                A3= STORAGE ADDRESS
00557                    *                                A4= MASK FOR RBK
00558                    *                                A6= CODE WORD
00559  0448  3CE1        CLC05 SRC     A4,1
00560  044A  8624              LDR*    A6,A1
00561  044C  EB0A              CWR     A3,A10                  COMPARE LOAD ADDRESS WITH AD OF IPL
00562  044E  9864              RB(0)   CLC01
00563  0450  A511              TM      A5,A4
00564  0452  5002              RF(0)   CLC07
00565  0454  9606              ADR     A6,A9
00566  0456  862D        CLC07 STR     A6,A3                   STORE CODE W=RD8
00567  0458  1102              ADK     A1,2
00568  045A  1302              ADK     A3,2
00569  045C  1A01              SUK     A2,1
00570  045E  9C18              RB(4)   CLC05
00571  0460  9F68              RB(7)   PROL0
00572                    *
00573                    ***************************************************************
00574                    * INTERNAL MODIFICATION CLUSTER
00575                    ***********************************************
00576  0462  0701        CLIMOD LDK    A7,1                    A7= MASK FOR ADDRESS
00577  0464  8524              LDR*    A5,A1                   A5=(RBK)
00578  0466  1A01              SUK     A2,1
00579  0468  3CE1        CLIM1 SRC     A4,1
00580  046A  1102              ADK     A1,2
00581  046C  8324              LDR*    A3,A1                   A3=ADDRESS
00582  046E  A31D              TM      A3,A7                   IS IT RELOCATABLE
00583  0470  5008              RF(0)   CLIM2                   NO
00584  0472  3301              XRK     A3,1
00585  0474  9306              ADR     A3,A9                   YES AD BASE
00586  0476  EB0A              CWR     A3,A10                  ADDRESS OK
00587  0478  988E              RB(0)   CLC01
00588  047A  1102        CLIM2 ADK     A1,2
00589  047C  8624              LDR*    A6,A1                   TAKE CODE WORD
00590  047E  A511              TM      A5,A4                   IS IT RELOCATABLE
00591  0480  5002              RF(0)   CLIM3
00592  0482  9606              ADR     A6,A9                   AD BASE
00593  0484  862D        CLIM3 STR     A6,A3                   STORE CODE WORD
00594  0486  1A02              SUK     A2,2
00595  0488  9C22              RB(4)   CLIM1
00596  048A  9F92              RB(7)   PROL0
00597                    ****************************************
00598                    *  CLUSTER  END/START
00599                    ****************************************
00600  048C  8324        CLEND LDR*    A3,A1
00601  048E  500A              RF(0)   CLEN3A                  FINISH NO START
00602  0490  A311              TM      A3,A0
00603  0492  5002              RF(0)   CLEN1
00604  0494  9306              ADR     A3,A9
00605                    *
00606                    *
00607        0496        CLEN1 EQU     *
00608  0496  834F              ST      A3,COREND-HCIPL,A11
       0498  0356
00609  049A  814F        CLEN3A LD     A1,BUFF+6-HCIPL,A11     UPDATE BASE ADDRESS
       049C  0090
```

```
00610  049E  914F              AD.S    A1,BADDR=HCIPL.A11
       04A0  0116
00611  04A2  9184              ADR     A9,A1
00612  04A4  5FAC              RB      PROLO
00613                    *
00614                    *
00615  04A6  0000        SAVA15  DATA    0
00616                    *
00617                    ********************************************************************************
00618                    *
00619  04A8  0000        COREND  DATA    0
00620                    *
00621                    *
00622                    ********************************************************************************
00623  04AA  0000              DATA    0
00624  04AC  0000              DATA    0
00625  04AE  0000              DATA    0
00626  04B0  0000              DATA    0
00627  04B2  0000              DATA    0
00628  04B4  0000              DATA    0
00629  04B6  0000              DATA    0
00630  04B8  0000              DATA    0
00631  04BA  0000              DATA    0
00632  04BC  0000              DATA    0
00633  04BE  0000              DATA    0
00634  04C0  0000              DATA    0
00635  04C2  0000              DATA    0
00636  04C4  0000              DATA    0
00637  04C6  0000              DATA    0
00638  04C8  0000              DATA    0
00639  04CA  0000              DATA    0
00640  04CC  0000              DATA    0
00641  04CE  0000              DATA    0
00642  04D0  0000              DATA    0
00643  04D2  0000              DATA    0
00644  04D4  0000              DATA    0
00645  04D6  0000              DATA    0
00646  04D8  0000              DATA    0
00647  04DA  0000              DATA    0
00648  04DC  0000              DATA    0
00649  04DE  0000              DATA    0
00650  04E0  0000              DATA    0
00651  04E2  0000              DATA    0
00652  04E4  0000              DATA    0
00653  04E6  0000              DATA    0
00654  04E8  0000              DATA    0
00655  04EA  0000              DATA    0
00656  04EC  0000              DATA    0
00657  04EE  0000              DATA    0
00658  04F0  0000              DATA    0
00659  04F2  0000              DATA    0
00660  04F4  0000              DATA    0
00661  04F6  0000              DATA    0
00662  04F8  0000              DATA    0
00663  04FA  0000              DATA    0
00664  04FC  0000              DATA    0
00665  04FE  0000              DATA    0
00666  0500  0000              DATA    0
```

```
00667  0502  0000                DATA    0
00668  0504  0000                DATA    0
00669  0506  0000                DATA    0
00670  0508  0000                DATA    0
00671  050A  0000                DATA    0
00672  050C  0000                DATA    0
00673  050E  0000                DATA    0
00674  0510  0000                DATA    0
00675  0512  0000                DATA    0
00676  0514  0000                DATA    0
00677  0516  0000                DATA    0
00678  0518  0000                DATA    0
00679  051A  0000                DATA    0
00680  051C  0000                DATA    0
00681  051E  0000                DATA    0
00682  0520  0000                DATA    0
00683                            EJECT
00684        0522      IPL44     EQU     *
00685  0522  030C                LDK     A3,NBREC
00686  0524  8420                LDKL    A4,BIPL
       0526  0100  R
00687  0528  0785                LDK     A7,/85
00688                  * BASIC WRITE
00689  052A  80A0                LDKL    A8,DFCB44
       052C  0582  R
00690        052E      PUNCH4    EQU     *
00691  052E  0550                LDK     A5,80
00692  0530  8620                LDKL    A6,BUF44
       0532  058E  R     ,
00693        0534      PCH400    EQU     *
00694  0534  0100                LDK     A1,0
00695  0536  0200                LDK     A2,0
00696  0538  E130                LCR     A1,A4
00697  053A  3964                SRL     A1,4
00698  053C  E921                CCK     A1,0
       053E  0000
00699  0540  5006                RF(0)   PCH405
00700  0542  E921                CCK     A1,/500
       0544  0500
00701  0546  5202                RF(2)   PCH410          LESS THAN 5
00702        0548      PCH405    EQU     *
00703  0548  2910                ORK     A1,/10
00704        054A      PCH410    EQU     *
00705  054A  3948                SLL     A1,8
00706  054C  E130                LCR     A1,A4
00707  054E  A120                ANKL    A1,/FF0F
       0550  FF0F
00708  0552  E921                CCK     A1,0
       0554  0000
00709  0556  5006                RF(0)   PCH415
00710  0558  E921                CCK     A1,/500
       055A  0500
00711  055C  5202                RF(2)   PCH420
00712        055E      PCH415    EQU     *
00713  055E  2910                ORK     A1,/10
00714        0560      PCH420    EQU     *
00715  0560  8139                STR     A1,A6
00716                  * STORE WORD
```

```
00717  0562  1602          ADK     A6,2                NEXT WORD IN BUFFER
00718  0564  1401          ADK     A4,1                NEXT CHZR. IN IPL
00719  0566  1D01          SUK     A5,1
00720                *
00721  0568  5936          RB(1)   PCH400
00722  056A  F820          CWK     A3,1                LAST RECORD ?
       056C  0001
00723  056E  5406          RF(4)   PCH430              NO
00724  0570  0113          LDK     A1,/13              XOFF
00725  0572  8141          ST      A1,BUF44+158
       0574  062C  R
00726        0576    PCH430  EQU     *
00727  0576  2804          LKM
00728  0578  0001          DATA    1
00729  057A  1B01          SUK     A3,1
00730  057C  5950          RB(1)   PUNCH4
00731                *
00732  057E  2804          LKM
00733  0580  0003          DATA    3
00734                *
00735        0582    DECB44  EQU     *
00736  0582  0003          DATA    3
00737  0584  058E  R        DATA    BUF44
00738  0586  00A0          DATA    160
00739  0588  0000          DATA    0
00740  058A  0000          DATA    0
00741  058C  0000          DATA    0
00742  058E          BUF44   RES     80
00743                *
00744                *
00745                *
00746                        END     BIPLPR
```

SYMBOL TABLE

```
ABA     0000  A   ASCII   03E6  R   ASCINP  0346  R   ASR     0010  A
BADDR   0268  R   BIPL    0100  R   BIPLPR  0000  R   BOM     0001  A
BUF44   058E  R   BUFF    01DC  R   CFZON   023A  R   CIOPTR  0294  R
CKOK    0256  R   CLC01   03EC  R   CLC01A  0436  R   CLC04   0442  R
CLC05   0448  R   CLC07   0456  R   CLCODE  0430  R   CLEN1   0496  R
CLEN3A  049A  R   CLEND   048C  R   CLIM1   0468  R   CLIM2   047A  R
CLIM3   0484  R   CLIMOD  0462  R   CNTFLG  014C  R   COREND  04A8  R
DECB    001E  R   DECB44  0582  R   DECBUF  0020  R   ECCLS   0334  R
ECMSG   01D2  R   FIGHT   02FE  R   END1    035A  R   END2    0320  R
FIRST   031A  R   H       0000  A   HCIPL   0152  R   HCIPL1  0150  R
HCIPL2  018A  R   INP2    02DA  R   INP2R   0318  R   INPA    028A  R
INR44   02F4  R   IPL     0104  R   IPL010  0116  R   IPL020  0122  R
IPL100  012C  R   IPL110  0130  R   IPL44   0522  R   IPL88   0000  R
LDFLG   0100  R   MASTFG  022C  R   MLX1    02C4  R   MNLD    03EE  R
MODE    0000  A   NBREC   000C  A   OBJEC   03E2  R   OBJIN1  0316  R
OBJINP  02FE  R   OFLMSG  01D6  R   OTR     0242  R   OUTMSG  023C  R
PCH400  0534  R   PCH405  0548  R   PCH410  054A  R   PCH415  055F  R
PCH420  0560  R   PCH430  0576  R   PRASCA  0384  R   PRASCB  0386  R
PRASCI  0360  R   PROGLD  0404  R   PROLO   03FA  R   PROL01  0406  R
PTR     0000  A   PUNCH   000C  R   PUNCH4  052E  R   RAFL    026A  R
RAFL1   03EA  R   RAFL2   03B4  R   RAFL3   02C2  R   RELAY   03B6  R
RER     02CE  R   S       0001  A   SA      0000  A   SAVA15  04A6  R
SAVBAS  022E  R   SSTMLX  02A0  R   SSTPR1  035C  R   SSTPR2  02DE  R
SSTPTR  0322  R   STAD    01D2  R   STOP    0342  R   SWITCH  03B8  R
SYSMSG  01AA  R   TMTEST  02A4  R   WER1    028C  R   WER2    028E  R
```

3.54 SIMPLE TEST PROGRAMS

The Simple Test programs that follow this paragraph are all Stand Alone programs that can be loaded into memory direct from the Control Panel switches. They should be tried if the Standard Test Programs either cannot be loaded or will not run. To avoid errors which may be introduced when loading the programs by hand it is suggested that once the program has been loaded and tested it should be reproduced on a suitable input media so that for future programs on an input media, together with suggestions for producing more sophisticated programs, will be found in paragraphs 3-64 to 3-74 of this section.

3.55 Memory Test Program MEMHAN

If the Standard Memory Test Program either cannot be used, or if you wish to test only a certain area of memory with a particular test pattern, try the following program which can be loaded either by hand from the Control Panel switches or by the IPL routine if it has been reproduced on a suitable input media. If the program is loaded from the Control Panel switches it can be loaded into any area of memory, but if it is loaded by the IPL routine the addresses of the instructions will be as shown in the program listing. The program loads the Test Pattern into A12. When all the required locations have been loaded, the program reads each location and compares each with the data in register A13. If an error is found during the read part of the program it will stop and the address where the error has occurred will be found in register A9, the pattern read will be found in register A8, and the expected pattern will be in register A13. If you wish the program to continue after an error has been detected press the RUN button. If there are no errors the program will run until the contents of A7 are zero.

3.56 Program CHECK

This program enables you to check the type of input from the Operator's device i.e. 8-bit data no parity, 7-bit data with parity etc. The program can be loaded by hand with the control panel switches or by using the IPL routine. Once loaded it is only necessary to load the start address into register A0 and then push the RUN button. You can now type in any ten characters from the keyboard which are stored in the Buffer (address 00A8). When the buffer is full the program branches back to the start of program and you can check the codes received for each character using the Read Memory Routine. If you wish to check more characters push the RUN button and repeat as necessary. This program is useful if you wish to check for certain characters in your programs.

3.57 ASR, PER3100, and Display

If the Standard Test Program either cannot be loaded or will not run, one of the following programs should be tried. They can all be loaded either by the Control Panel switches or by the IPL routine. Two of the programs (LINE and NOECHO) have been written for devices that have been wired for Even Parity so if your device is wired for any other mode of operation the parameters loaded into register A2 must be changed accordingly.

3.58 Program LINE

This program can be loaded either from the Control Panel switches or by using the IPL routine. If the program is loaded by hand it can be loaded into any area of memory, but if loaded by the IPL routine the locations of the instructions will be as shown in the program listing. The program enables a selected ASCII character to be sent to the device a selected number of times (the number of times selected MUST NOT exceed the maximum number of characters per line specified for the device being tested), then it sends the Carriage Return/Line Feed characters and branches back to the start of program and stops ready for a new character to be selected. If you wish the program can be modified to run continuously by changing two of the instructions as follows:

- Change the LDR instruction in line 27 to LDK A4,64 hexadecimal value 0440.
- Change the RB instruction value in line 41 so that it branches back to line 24 of the program by using hexadecimal value 5F24.

Once the RUN button has been pushed it will then run and continue repeating the same line until you stop the program by pushing the INST button.

MEMHAM

```
00000                          IDENT    MEMHAN
00001          0000    BEGIN   EQU      *
00002                          RORG     BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO WRITE AND CHECK READ
00005                  *       A SELECTED PATTERN INTO A SELECTED AREA OF MEMORY AND TO
00006                  *       REPEAT THE TEST A SELECTED NUMBER OF TIMES UP TO 255
00007                  *
00008                  *        LOAD THE NUMBER OF TIMES YOU WANT THE TEST REPEATED INTO
00009                  *       REGISTER A7 FROM THE CONTROL PANEL
00010                  *
00011                  *       LOAD THE START ADDRESS OF THE AREA TO BE TESTED INTO
00012                  *       REGISTER A11 FROM THE CONTROL PANEL
00013                  *
00014                  *       LOAD THE END ADDRESS OF THE AREA TO BE TESTED INTO
00015                  *       REGISTER A12 FROM THE CONTROL PANEL
00016                  *
00017                  *       LOAD THE TEST PATTERN INTO REGISTER A13 FROM THE CONTROL PANEL
00018                  *
00019                  *       LOAD THE START ADDRESS OF THE PROGRAM INTO REGISTER A0 FROM THE
00020                  *       CONTROL PANEL THEN PUSH THE RUN BUTTON
00021                  *
00022                  *
00023  0080  FFFF              DATA     /FFFF
00024  0082  0000              DATA     0
00025  0084  207F      START   HLT
00026  0086  B1BE      WRITE   LDR      A9,A11          LOAD START ADDRESS INTO A9
00027  0088  85A7      LOAD    STR      A13,A9          STORE TEST PATTEN INTO MEMORY
00028  008A  9140              ADKI     A9,2            UPDATE MEMORY ADDRESS COUNTER
       008C  0002
00029  008E  E992              CWR      A9,A12          CHECK FOR END ADDRESS IN A9
00030  0090  5D0A              RB(NG)   LOAD            IF NOT EQUAL LOAD NEXT ADDRESS
00031  0092  B1BE              LDR      A9,A11          LOAD START ADDRESS INTO A9
00032  0094  80A6      READ    LDR*     A8,A9           LOAD CONTENTS OF A9'S MEMORY ADDRESS
00033  0096  E89A              CWR      A8,A13          AND COMPARE WITH PATTERN IN A13
00034  0098  5002              RF(Z)    CONT            IF EQUAL CONTINUE WITH READ ROUTINE
00035  009A  207F              HLT                      IF NOT EQUAL STOP
00036  009C  9140      CONT    ADKI     A9,2            UPDATE ADDRESS COUNTER
       009E  0002
00037  00A0  E992              CWR      A9,A12          CHECK FOR END ADDRESS IN A9
00038  00A2  5D10              RB(NG)   READ            IF NOT EQUAL READ NEXT WORD
00039  00A4  1E01              SUK      A7,1            SUBTRACT 1 FROM REPEAT COUNTER
00040  00A6  5C22              RB(NZ)   WRITE           AND START NEXT W/R CYCLE IF NOT ZERO
00041  00A8  5E26              RB       START           WHEN FINISHED GO TO START OF PROGRAM FOR
00042                  *                               POSSIBLE NEW PARAMETERS
00043                          END      START
```

CHECK

```
00000                                IDENT    CHECK
00001          0000      BEGIN     EQU      *
00002                                RORG     BEGIN+/80
00003                      *
00004                      *         THIS PROGRAM ENABLES THE ENGINEER TO CHECK
00005                      *         THE CHARACTER CODE FROM THE OPERATORS DEVICE
00006                      *
00007                      *
00008                      *         LOAD THE PROGRAM START ADDRESS INTO A0 FROM THE
00009                      *         CONTROL PANEL - THEN PUSH THE RUN BUTTON
00010                      *
00011                      *
00012  0080    FFFF                  DATA     /FFFF
00013  0082    0000                  DATA     0
00014  0084    207F      START     HLT
00015  0086    20BF                  INH
00016                      *
00017  0088    0300                  LDK      A3,0                 /ZEROISE BUFF COUNTER
00018  008A    0201                  LDK      A2,1                 LOAD PARAMETERS FOR INPUT CHARACTER
00019                      *                                                                WITH NO
00020  008C    4200                  CIO      A2,1,/10             SEND START COMMAND TO DEVICE
00021  008E    5C04                  RB(NA)   *-2
00022  0090    4D10      INCHAR    INR      A5,0,/10             SEND INR INSTRUCTION TO DEVICE
00023  0092    5C04                  RB(NA)   *-2                  AND WAIT FOR CHAR TO BE INPUT
00024  0094    E54D                  SC       A5,BUFF,A3           STORE CHAR IN BUFF
       0096    00A8  R
00025  0098    1301                  ADK      A3,1                 UPDATE BUFF COUNTER
00026  009A    EB20                  CWK      A3,10                CHECK FOR BUFFER FULL
       009C    000A
00027  009E    5C10                  RB(NE)   INCHAR               ACCEPT NEW CHAR IF NOT FULL
00028  00A0    4290                  CIO      A2,0,/10             STOP INPUT FUNCTION
00029  00A2    4AD0                  SST      A2,/10               AND GET STATUS
00030  00A4    5C04                  RB(NA)   *-2
00031  00A6    5F2A                  RB       START
00032  00A8            BUFF      RES      5
00033                                END      START
```

LINE

```
00000                          IDENT    LINE
00001          0000    BEGIN   EQU      *
00002                          RORG     BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO SEND A SELECTED
00005                  *       ASCII CHAR TO THE OPERATOR'S DEVICE A SELECTED
00006                  *       NUMBER OF TIMES
00007                  *
00008                  *
00009                  *       LOAD THE ASCII CHARACTER INTO REGISTER A6 FROM THE
00010                  *       CONTROL PANEL
00011                  *
00012                  *       LOAD THE NUMBER OF TIMES YOU WANT IT REPEATED
00013                  *       INTO REGISTER A7 FROM THE CONTROL PANEL
00014                  *
00015                  *       LOAD THE PROGRAM START ADDRESS INTO REGISTER A0
00016                  *       FROM THE CONTROL PANEL AND PUSH THE RUN BUTTON
00017                  *
00018                  *
00019  0080  FFFF              DATA     /FFFF
00020  0082  0000              DATA     0
00021  0084  207F      START   HLT
00022  0086  208F              INH                          THIS PROGRAM RUNS IN INHIBIT MODE
00023                  *
00024  0088  0204              LDK      A2,4                LOAD PARAMETERS FOR OUTPUT CHARACTER
00025                  *                                    WITH EVEN PARITY
00026  008A  4200              CIO      A2,1,/10            SEND START COMMAND TO DEVICE
00027  008C  5C04              RB(NA)   *-2                 KEEP TRYING UNTIL ACCEPTED
00028  008E  841C              LDR      A4,A7               LOAD COUNT INTO A4
00029  0090  4610      OUCHAR  OTR      A6,0,/10            OUTPUT ASCII CHAR TO DEVICE
00030  0092  5C04              RB(NA)   *-2                 TRY AGAIN IF NOT ACCEPTED
00031  0094  1C01              SUK      A4,1                SUBTRACT ONE FROM COUNT
00032  0096  5C08              RB(NZ)   OUCHAR              SEND CHAR AGAIN IF NOT ZERO
00033                  *                                    IF ZERO
00034  0098  8520      CRLF    LDKL     A5,/0A0D            LOAD CR/LF CHARS
       009A  0A0D
00035  009C  4510      OUCRLF  OTR      A5,0,/10            SENDS CR/LF CHARS
00036  009E  5C04              RB(NA)   *-2
00037  00A0  3D68              SRL      A5,8                POSITIONS 0A CHAR FOR SENDING
00038  00A2  5C08              RB(NZ)   OUCRLF              BRANCHES FIRST TIME TO SEND SECOND CHAR
00039  00A4  4290              CIO      A2,0,/10            SENDS STOP COMMAND TO DEVICE
00040  00A6  4AD0              SST      A2,/10              AND THEN GETS STATUS WORD
00041  00A8  5C04              RB(NA)   *-2
00042  00AA  5F28              RB       START               RETURN TO START  COUNT DONE
00043                          END      START
```

3.59 Programs NOEC57 and ECHO57

These programs enable you to verify the input/output functions of the peripheral and the V24 Serial CU and they can be loaded either by hand from the Control Panel or by using the IPL routine. The only differences between the two programs are the parameters loaded into register A2, and with ECHO57 it is the CU that sends the character back to the device but for NOEC57 it is the program. In both programs 7 data bits with even parity are selected so if your CU has been programmed a different type of data (by means of the U-links) you will have to change the parameters in register A2 to aggree with the type of parity and number of data bits being used. Once loaded, put the start address in register A0 and then push the RUN button; thereafter the program will run until stopped by pushing the INST button. Any printable characters typed in via the keyboard will be reproduced on the console log, and any control characters (e.g. CR LF) are accepted and executed.

NOEC57

```
00000                          IDENT    NOEC57
00001          0000    BEGIN   EQU      *
00002                          BORG     BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO VERIFY THE INPUT/OUTPUT
00005                  *       FUNCTIONS OF THE CU AND ITS PERIPHERAL
00006                  *
00007                  *
00008                  *       LOAD THE PROGRAM START ADDRESS INTO A0 FROM THE
00009                  *       CONTROL PANEL - THEN PUSH THE RUN BUTTON
00010                  *
00011                  *
00012  0080  FFFF              DATA     /FFFF
00013  0082  0000              DATA     0
00014  0084  207F      START   HLT
00015  0086  208F              INH
00016                  *
00017  0088  0205      INCHAR  LDK      A2,5            LOAD PARAMETERS FOR INPUT CHARS NO ECHO
00018                  *                                MODE WITH 7 DATA BITS NO PARITY
00019  008A  4200              CIO      A2,1,/10        SEND START COMMAND TO DEVICE
00020  008C  5C04              RB(NA)   *-2
00021  008E  4D10              INR      A5,0,/10        SEND INR INSTRUCTION TO DEVICE
00022  0090  5C04              RB(NA)   *-2             AND WAIT FOR CHAR TO BE INPUT
00023  0092  4290              CIO      A2,0,/10        STOP INPUT FUNCTION
00024  0094  4AD0              SST      A2,/10          AND GET STATUS
00025  0096  5C04              RB(NA)   *-2
00026  0098  0204              LDK      A2,4            LOAD PARAMETERS FOR OUTPUT CHARACTER
00027                  *                                WITH EVEN PARITY
00028  009A  4200              CIO      A2,1,/10        SEND START COMMAND TO DEVICE
00029  009C  5C04              RB(NA)   *-2
00030  009E  4510              OTR      A5,0,/10        SEND CHAR BACK TO DEVICE
00031  00A0  5C04              RB(NA)   *-2
00032  00A2  4290              CIO      A2,0,/10        SEND STOP OUTPUT FUNCTION COMMAND
00033  00A4  4AD0              SST      A2,/10          AND GET STATUS
00034  00A6  5C04              RB(NA)   *-2
00035  00A8  5F22              RB       START+4         RETURN TO START OF INPUT ROUTINE
00036                          END      START
```

ECHO57

```
00000                          IDENT    ECHO57
00001          0000    BEGIN   EQU      *
00002                          RORG     BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO VERIFY THE INPUT/OUTPUT
00005                  *       FUNCTIONS OF THE OPERATOR'S PERIPHERAL IN ECHO MODE
00006                  *
00007                  *
00008                  *       LOAD THE PROGRAM START ADDRESS INTO A0 FROM THE
00009                  *       CONTROL PANEL - THEN PUSH THE RUN BUTTON
00010                  *
00011                  *
00012  0080    FFFF            DATA     /FFFF
00013  0082    0000            DATA     0
00014  0084    207F    START   HLT
00015  0086    20BF            INH
00016                  *
00017  0088    0225    INCHAR  LDK      A2,/25          LOAD PARAMETERS FOR ECHO MODE
00018                  *                                WITH 7 DATA BITS NO PARITY
00019  008A    42D0            CIO      A2,1,/10        SEND START COMMAND TO DEVICE
00020  008C    5C04            RB(NA)   *-2
00021  008E    4D10            INR      A5,0,/10        SEND INR INSTRUCTION TO DEVICE
00022  0090    5C04            RB(NA)   *-2             AND WAIT FOR CHAR TO BE INPUT
00023  0092    4290            CIO      A2,0,/10        STOP INPUT FUNCTION
00024  0094    4AD0            SST      A2,/10          AND GET STATUS
00025  0096    5C04            RB(NA)   *-2
00026  0098    5F12            RB       INCHAR
00027                          END      START
```

3.60 PROGRAMS FOR THE MCU3 CARD

The MCU3 card can control the operation of a Paper Tape Reader, a Paper Tape Punch, and a Serial Device. Parts of the logic are common to all three devices so if one of the suggested programs runs without error you can almost certainly eliminate the common logic and look for the fault in unique logic of the device whose program does not run. No programs are given for the Serial Device. You can check it using the examples given in paragraphs 3-57 to 3-59 modifying them as necessary for the device concerned. The programs to check the PTR and PTP are: PTR57, PUNCH, PCH57, and CHKTRD. The first two can be modified to run continuously by changing the displacement value of the RB to START instruction so that it branches back to the first instruction in the program instead of the HLT instruction at START. The listings of the programs give the instructions for operating them and the function of the programs is:

- PTR57 — will read any type of coded paper tape and the correct operation of the CU logic and the device can be verified by checking the contents of the buffer against the holes punched in the tape.
- PUNCH — will punch a selected area of memory onto tape and you can either load the data into memory yourself (using the Control Panel), or use an area of memory that you know contains data. The correct operation of the CU logic and the device can be verified by checking the tape produced against the data in the area of memory selected.
- PCH57 — will cause the punch mechanism to select every possible combination of punch pins from all holes to blank tape; you can also modify this program to punch a selected punch combination continuously by changing the value loaded into register A5 and replacing the SUK instruction by an RB to CONT instruction (hexadecimal 5F06). The correct operation of the device can be verified either by checking the tape by eye or by using the following program.
- CHKTRD — will check if the tape produced by PCH57 was correctly punched by adding the value of each character read into register A3 and when all the characters have been read checking the value in A3 against a check sum for correct operation. The only precaution necessary in using this program is that the tape must be accurately positioned so that the first character read is the all holes character and not blank tape. If the checksum is correct the program will halt on location /86, if the checksum is wrong the program will halt on location /AA. When a wrong checksum has been detected try the program two or three times more and check if the result in register A3 is the same every time, then check the tape to see if all the combinations have been punched. If not check the punch logic again. If all combinations have been punched check the reader logic.

In principle it is unlikely that either the PTR or PTP will be operated via the IOP channel so no programs have been given for this type of operation. However, if your system uses the IOP for either of these devices you can modify one of the programs suggested for the MCU2 card devices.

PTR57

```
00000                                  IDENT    PTR57
00001          0000        BEGIN       EQU      *
00002                                  BORG     BEGIN+/80
00003                      *
00004                      *           THIS PROGRAM ENABLES THE ENGINEER TO VERIFY THE
00005                      *           INPUT FUNCTIOON OF THE CU AND ITS DEVICE
00006                      *
00007                      *           LOAD NUMBER OF CHARS TO BE READ INTO A7
00008                      *
00009                      *           LOAD THE PROGRAM START ADDRESS INTO A0
00010                      *
00011                      *           LOAD TAPE ON READER THEN PUSH RUN BUTTON
00012                      *
00013                      *
00014   0080   FFFF                    DATA     /FFFF
00015   0082   0000                    DATA     0
00016   0084   207F        START       HLT
00017   0086   208F                    INH                                THIS PROGRAM RUNS IN INHIBIT MODE
00018   0088   861C                    LDR      A6,A7                     LOAD CHAR COUNT IN A6
00019   008A   0300                    LDK      A3,0                      ZEROISE BUFFER CHAR COUNT
00020   008C   0201                    LDK      A2,1                      LOAD PARAMETER FOR INPUT CHARACTER
00021   008E   42E0                    CIO      A2,1,/20                  SEND START COMMAND TO DEVICE
00022   0090   5C04                    BR(NA)   *+2
00023   0092   4D20        INPT        INR      A5,0,/20                  INPUT CHAR FROM READER
00024   0094   5C04                    BR(NA)   *+2
00025   0096   E54D                    SC       A5,BUFF,A3                STORE CHAR IN BUFFER
        0098   00AB   R
00026   009A   1301                    ADK      A3,1                      UPDATE BUFFER CHAR COUNT
00027   009C   1E01                    SUK      A6,1                      SUBTRACT ONE FROM CHAR COUNT
00028   009E   5C0E                    BR(NZ)   INPT                      RETURN TO READ ANOTHER CHAR IF NOT ZERO
00029   00A0   42A0                    CIO      A2,0,/20                  SEND STOP COMMAND TO DEVICE
00030   00A2   4AE0                    SST      A2,/20                    AND GET STATUS
00031   00A4   5C04                    BR(NA)   *+2
00032   00A6   5F24                    BR       START                     GO TO START OF PROGRAM
00033   00AB               BUFF        RES      40
00034                                  END      START
```

SYMBOL TABLE

```
BEGIN   0000  R    BUFF    00AB  R    INPT    0092  R    START   0084  R
```

PUNCH

```
00000                                   IDENT   PUNCH
00001                           *
00002                           *       THIS PROGRAM PUNCHES A SELECTED AREA OF MEMORY ONTO PAPER TAPE
00003                           *
00004           0000            BEGIN   EQU     *
00005                                   RORG    BEGIN+/80
00006                           *
00007                           *
00008                           *       LOAD THE FIRST ADDRESS OF THE AREA INTO REGISTER A12
00009                           *
00010                           *       LOAD THE LAST ADDRESS OF THE AREA INTO REGISTER A11
00011                           *
00012                           *       LOAD THE START ADDRESS OF THE PROGRAM INTO REGISTER A0
00013                           *
00014                           *       RUN THE RUN BUTTON
00015                           *
00016                           *
00017   0080    207F            START   HLT
00018   0082    8112                    LDR     A1,A12          SAVE THE START ADDRESS INTO A1
00019   0084    0200                    LDK     A2,0
00020   0086    42F0                    CIO     A2,1,/30
00021   0088    5C04                    RB(NA)  *-2
00022   008A    E524            CONT    LCR     A5,A1           LOAD THE CHAR INTO A5
00023   008C    4530                    OTR     A5,0,/30
00024   008E    5C04                    RB(NA)  *-2
00025   0090    1101                    ADK     A1,1            UPDATE AREA ADDRESS BY ONE
00026   0092    E90E                    CWR     A1,A11          CHECK IF LAST CHAR HAS BEEN PUNCHED
00027   0094    5C0C                    RB(NE)  CONT            NO? GO AND GET NEXT CHAR
00028   0096    4280                    CIO     A2,0,/30
00029   0098    4BF0                    SST     A3,/30
00030   009A    5C04                    RB(NA)  *-2
00031   009C    5F1E                    BR      START
00032                                   END     START
```

SYMBOL TABLE

```
BEGIN   0000  R   CONT    008A  R   START   0080  R
```

PCH57

```
00000                                IDENT    PCH57
00001                          *
00002                          *     THIS PROGRAM CHECKS THE FUNCTION OF THE PUNCH'S HOLE SELECTION LOGIC
00003                          *
00004          0000      BEGIN   EQU      *
00005                          RORG     BEGIN+/80
00006                          *
00007                          *
00008                          *     PUSH THE RUN BUTTON TO START THE PROGRAM
00009                          *
00010                          *
00011  0080    FFFF            DATA     /FFFF
00012  0082    0000            DATA     0
00013  0084    207F      START HLT
00014  0086    05FF            LDK      A5,/FF              LOAD DATA TO PUNCH ALL HOLES
00015  0088    0200            LDK      A2,0
00016  008A    42F0            CIO      A2,1,/30
00017  008C    5C04            RB(NA)   *-2
00018  008E    4530      CONT  OTR      A5,0,/30            SEND PUNCH COMMAND AND CHAR TO CU
00019  0090    5C04            RB(NA)   *-2
00020  0092    1D01            SUK      A5,1                MODIFY HOLE PATTERN
00021  0094    5E08            RB(NN)   CONT                NOT NEGATIVE? GO AND PUNCH NEXT PATTERN
00022  0096    42B0            CIO      A2,0,/30
00023  0098    4BF0            SST      A3,/30
00024  009A    5C04            RB(NA)   *-2
00025  009C    5F1A            RB       START
00026                          END      START

          SYMBOL TABLE

  BEGIN   0000  R   CONT    008F  R   START   0084  R
```

CHKTRD

```
00000                          IDENT   CHKTRD
00001          0000    BEGIN   EQU     *
00002                          BORG    BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM USES THE TAPE PRODUCED BY PCH51 TO VERIFY THE PTR READ CEL
00005                  *
00006                  *       LOAD THE PROGRAM START ADDRESS INTO A0
00007                  *
00008                  *       LOAD TAPE ON READER THEN PUSH RUN BUTTON
00009                  *
00010                  *
00011  0080  FFFF              DATA    /FFFF
00012  0082  0000              DATA    0
00013  0084  207F      START   HLT
00014  0086  208F              INH                             THIS PROGRAM RUNS IN INHIBIT MODE
00015  0088  06FF              LDK     A6,255                  LOAD CHAR COUNT IN A6
00016  008A  0300              LDK     A3,0                    ZEROISE BUFFER CHAR COUNT
00017  008C  0201              LDK     A2,1                    LOAD PARAMETER FOR INPUT CHARACTER
00018  008E  42E0              CIO     A2,1,/20                SEND START COMMAND TO DEVICE
00019  0090  5C04              BR(NA)  *+2
00020  0092  4D20      INPT    INR     A5,0,/20                INPUT CHAR FROM READER
00021  0094  5C04              BR(NA)  *+2
00022  0096  931A              ADR     A3,A6                   ADD THE CHAR VALUE TO A3
00023  0098  1E01              SUK     A6,1                    SUBTRACT ONE FROM CHAR COUNT
00024  009A  5C0A              BR(NZ)  INPT                    RETURN TO READ ANOTHER CHAR IF NOT ZERO
00025  009C  42A0              CIO     A2,0,/20                SEND STOP COMMAND TO DEVICE
00026  009E  4AF0              SST     A2,/20                  AND GET STATUS
00027  00A0  5C04              BR(NA)  *+2
00028  00A2  FB20              CWK     A3,32640                COMPARE VALUE IN A3 WITH CHECK SUM
       00A4  7F80
00029  00A6  5824              BR(E)   START                   IF EQUAL GO TO START
00030  00A8  207F              HLT
00031                          END     START
```

SYMBOL TABLE

```
BEGIN   0000  R    INPT    0092  R    START   0084  R
```

3.61 PROGRAMS FOR THE MCU2 CARD

The MCU2 card can control the operations of a Card Reader and a Line Printer. The remarks concerning the common logic of the card given in the previous paragraph (second sentence) also apply to this card. Both devices can be operated on either the IOP or Programmed channel so a program for each type is given that will enable the basic functions of the CU logic and the device to be checked. They can all be modified to run continuously by changing the displacement value of the RB to START instruction so that it branches back to the first instruction of the program instead of the HLT instruction at START. The listings of the programs give the instructions for operating them and the function of the programs is:

- CRPROG and CRIOP — will read one card each time the RUN button is pushed (unless you have modified the program to run continuously) and the two programs operate on the Programmed Channel and IOP Channel respectively. You can verify the correct transfer of data by checking the holes punched in the cards against the data stored in the buffer.
- LPPROG and LPIOP — will store the two selected characters into the buffer (total of 78 printable characters in the unmodified version) until it is full then stores the Line Feed/Carriage Return characters and prints the line of characters. After each line printed you may change the two characters that you want printed; you can also send the other control characters to the printer but in this case you must modify the number of times they are loaded into the buffer. The two programs operate on the Programmed Channel and IOP Channel respectively.

CRPROG

```
00000                                  IDENT    CRPROG
00001                          *
00002                          *
00003                          *       THIS PROGRAM READS A CARD VIA THE PROGRAMMED CHANNEL AND STOPS SO THAT
00004                          *       CONTENTS OF THE BUFFER CAN BE CHECKED AGAINST THE PUNCHED HOLES IN THE
00005                          *
00006                          *
00007          0000            BEGIN    EQU      *
00008                                   BORG     BEGIN+/80
00009                          *
00010                          *
00011                          *       LOAD THE CARD(S) TO BE READ INTO THE CARD READER
00012                          *       AND START THE CARD READER
00013                          *
00014                          *       PUSH THE RUN BUTTON
00015                          *
00016
00017  0080  FFFF                       DATA     /FFFF
00018  0082  0000                       DATA     0
00019  0084  207F              START    HLT
00020  0086  20BF                       INH
00021                          *
00022  0088  0200                       LDK      A2,0                  LOAD ZERO INTO A2
00023  008A  0100                       LDK      A1,0                  ZEROISE WORD COUNTER
00024  008C  8245              STORE    ST       A2,BUFF,A1            STORE ZERO INTOBUFF ADDRESS
       008E  0088  R
00025  0090  1102                       ADK      A1,2                  UPDATE WORD COUNT OF BUFF ADDRESS
00026  0092  E920                       CWK      A1,42                 CHECK IF LAST WORD REACHED
       0094  002A
00027  0096  5C0C                       BB(NE)   STORE                 NO? GO AND STORE NEXT WORD
00028  0098  0100                       LDK      A1,0                  ZEROISE CHAR COUNTER
00029  009A  0601                       LDK      A6,1                  LOAD PARAMS FOR INPUT FUNCTION
00030  009C  46C6                       CIO      A6,1,6                SEND CIO START COMMAND TO CARD READER
00031  009E  5C04                       BB(NA)   *+2
00032  00A0  4D06              READ     INB      A5,0,6                GET CHARS FROM CU
00033  00A2  5C04                       BB(NA)   *+2
00034  00A4  8545                       ST       A5,BUFF,A1            ANS STORE IN BUFFER
       00A6  0088  R
00035  00A8  1102                       ADK      A1,2                  UPDATE BUFFER ADDRESS
00036  00AA  E920                       CWK      A1,/50                CHECK IF LAST COLUMN READ
       00AC  0050
00037  00AE  5C10                       BB(NE)   READ                  NO? GO AND READ NEXT COLUMN
00038  00B0  4686                       CIO      A6,0,6                SEND STOP COMMAND TO CU
00039  00B2  4CC6                       SST      A4,6                  GET STATUS FROM CU
00040  00B4  5C04                       BB(NA)   *+2
00041  00B6  5F34                       BB       START                 GO AND WAIT FOR NEW DATA
00042  00B8                    BUFF     BFS      42
00043                                   END      START

          SYMBOL TABLE

BEGIN   0000  R   BUFF    00B8  R   READ    00A0  R   START   0084  R
STORE   008C  R
```

CRIOP

```
00000                                  IDENT    CRIOP
00001                          *
00002                          *
00003                          *       THIS PROGRAM READS A CARD AND STOPS  SO THAT THE CONTENTS
00004                          *       OF THE BUFFER CAN BE CHECKED AGAINST THE PUNCHED HOLES IN THE CARD
00005                          *
00006                          *
00007          0000            BEGIN   EQU      *
00008                                  BORG     BEGIN+/80
00009                          *
00010                          *
00011                          *       LOAD THE CARD(S) TO BE READ INTO THE CARD READER
00012                          *       AND START THE CARD READER
00013                          *
00014                          *       PUSH THE RUN BUTTON
00015                          *
00016
00017  0080  FFFF                      DATA     /FFFF
00018  0082  0000                      DATA     0
00019  0084  207F              START   W.T
00020  0086  20BF                      INH
00021                          *
00022  0088  0200                      LDK      A2,0                  LOAD ZERO INTO A2
00023  008A  0100                      LDK      A1,0                  ZEROISE WORD COUNTER
00024  008C  8245              STORE   ST       A2,BUFF,A1            STORE ZERO IN BUFF ADDRESS
       008E  0080  R
00025  0090  1102                      ADK      A1,2                  UPDATE WORD COUNT OF BUFF ADDRESS
00026  0092  E920                      CWK      A1,42                 CHEK IF LAST WORD REACHED
       0094  002A
00027  0096  5C0C                      RB(NE)   STORE                 NO? GO AND STORE NEXT WORD
00028  0098  8120                      LDKL     A1,/8050              LOAD PARAMS FOR FIRST WER
       009A  8050
00029  009C  710C                      WER      A1,/C                 AND SENT TO IOP
00030  009E  8120                      LDKI     A1,BUFF               LOAD BUFF ADDRESS FOR SECOND WER
       00A0  0080  R
00031  00A2  710D                      WER      A1,/D                 AND SEND TO IOP
00032  00A4  0601                      LDK      A6,1                  LOAD PARAMS FOR INPUT FUNCTION
00033  00A6  46C6                      CIO      A6,1,6                SEND CIO START COMMAND TO CARD READER
00034  00A8  5C04                      RB(NA)   *-2
00035  00AA  4CC6                      SST      A4,6                  GET STATUS FROM CIU
00036  00AC  5C04                      RB(NA)   *-2
00037  00AE  5F2C                      RB       START                 GO AND WAIT FOR NEW DATA
00038  0080              BUFF    RES      42
00039                                  END      START

          SYMBOL TABLE

 BEGIN   0000  R   BUFF    0080  R   START   0084  R   STORE   008C  R
```

LPPROG

```
00000                                   IDENT   LPPROG
00001                           *
00002                           *
00003                           *       THIS PROGRAM STORES TWO SELECTED CHARACTERS INTO A BUFFER
00004                           *       THEN PRINTS THEM AND STOPS WAITING NEW CHARACTERS TO BE SELECTED
00005                           *
00006                           *
00007           0000    BEGIN   EQU     *
00008                           RORG    BEGIN+/80
00009                           *
00010                           *
00011                           *       LOAD THE TWO CHARACTERS TO BE PRINTED INTO REGISTER A3
00012                           *
00013                           *       PUSH THE RUN BUTTON
00014                           *
00015
00016   0080    FFFF                    DATA    /FFFF
00017   0082    0000                    DATA    0
00018   0084    207F    START   HLT
00019   0086    20BF            INH
00020                           *
00021   0088    850C                    LDR     A5,A3                   LOAD DATA INTO A5
00022   008A    0100                    LDK     A1,0                    ZEROISE CHAR COUNTER
00023   008C    8545    REPT    ST      A5,LPBUF,A1             STORE THE CONTENTS OF A5 INTO LPBUF
        008E    00C0  R
00024   0090    1102                    ADK     A1,2                    UPDATE LPBUF ADDRESS
00025   0092    E920                    CWK     A1,78                   AND CHECK IF LINE FULL
        0094    004E
00026   0096    5C0C                    BR(NE)  REPT
00027   0098    8520                    LDKL    A5,/0D0A                LOAD LINE FEED CARRAIGE RETURN CHARS
        009A    0D0A
00028   009C    8545                    ST      A5,LPBUF,A1             AND STORE IN LPBUF
        009E    00C0  R
00029   00A0    0100                    LDK     A1,0                    ZEROISE CHAR COUNTER
00030   00A2    0600                    LDK     A6,0                    LOAD PARAM FOR OUTPUT FUNCTION
00031   00A4    4607                    CIO     A6,1,/07                SENT START COMMAND TO PRINTER
00032   00A6    5C04                    BR(NA)  *-2                     TRY AGAIN IF NOT ACCEPTED
00033   00A8    E544    OUCHAR  LC      A5,LPBUF,A1             GET CHAR FROM LPBUF
        00AA    00C0  R
00034   00AC    4507                    OTR     A5,0,7                  AND SEND TO LINE PRINTER
00035   00AE    5C04                    BR(NA)  *-2
00036   00B0    1101                    ADK     A1,1                    UPDATE LPBUF ADDRESS
00037   00B2    E920                    CWK     A1,80                   CHECK IF LAST CHAR SENT
        00B4    0050
00038   00B6    5C10                    BR(NE)  OUCHAR                  NOT EQUAL? GOAND SEND NEXT CHAR
00039   00B8    4687                    CIO     A6,0,7                  SEND STOP COMMAND TO LINE PRINTER
00040   00BA    4CC7                    SST     A4,7                    GET STATUS
00041   00BC    5C04                    BR(NA)  *-2
00042   00BE    5F3C                    BR      START                   GO AND WAIT FOR NEW DATA
00043   00C0            LPBUF   BFS     80
00044                                   END     START

        SYMBOL TABLE            BEGIN   0000  R   LPBUF   00C0  R   OUCHAR  00A8  R   REPT    008C  R
                                START   0084  R
```

LPIOP

```
00000                                  IDENT    LPIOP
00001                          *
00002                          *
00003                          *       THIS PROGRAM STORES TWO SELECTED CHARACTERS INTO A BUFFER
00004                          *       THEN PRINTS THEM AND STOPS WAITING NEW CHARACTERS TO BE SELECTED
00005                          *
00006                          *
00007          0000            BEGIN   EQU      *
00008                                  BORG     BEGIN+/80
00009                          *
00010                          *
00011                          *       LOAD THE TWO CHARACTERS TO BE PRINTED INTO REGISTER A3
00012                          *
00013                          *       PUSH THE RUN BUTTON
00014                          *
00015
00016  0080  FFFF                      DATA     /FFFF
00017  0082  0000                      DATA     0
00018  0084  207F              START   HLT
00019  0086  20BF                      INH
00020                          *
00021  0088  B50C                      LDR      A5,A3                 LOAD DATA INTO A5
00022  008A  0100                      LDK      A1,0                  ZEROISE CHAR COUNTER
00023  008C  8545              REPT    ST       A5,LPBUF,A1           STORE THE CONTENTS OF A5 INTO LPBUF
       008E  0088  R
00024  0090  1102                      ADK      A1,2                  UPDATE LPBUF ADDRESS
00025  0092  F920                      CWK      A1,78                 AND CHECK IF LINE FULL
       0094  004F
00026  0096  5C0C                      BR(NE)   REPT
00027  0098  8520                      LDKL     A5,/0D0A              LOAD LINE FEED CARRAIGE RETURN CHARS
       009A  0D0A
00028  009C  8545                      ST       A5,LPBUF,A1           AND STORE IN LPBUF
       009E  0088  R
00029  00A0  8120                      LDKL     A1,/4050              LOAD PARAMS FOR FIRST WER
       00A2  4050
00030  00A4  710E                      WER      A1,/E                 AND SEND TO IOP
00031  00A6  8120                      LDKL     A1,LPBUF              LOAD FIRST ADDRESS OF LPBUF
       00A8  0088  R
00032  00AA  710F                      WER      A1,/F                 AND SEND TO IOP
00033  00AC  0600                      LDK      A6,0                  LOAD PARAM FOR OUTPUT FUNCTION
00034  00AE  46C7                      CIO      A6,1,/07              SENT START COMMAND TO PRINTER
00035  00B0  5C04                      BR(NA)   *-2                   TRY AGAIN IF NOT ACCEPTED
00036  00B2  4CC7                      SST      A4,7                  GET STATUS
00037  00B4  5C04                      BR(NA)   *-2
00038  00B6  5F34                      BR       START                 GO AND WAIT FOR NEW DATA
00039  00B8                    LPBUF   RES      80
00040                                  END      START
```

SYMBOL TABLE

```
BEGIN   0000  R   LPBUF   00B8  R   REPT    008C  R   START   0084  R
```

3.62 Program COPY

This program enables you to punch a copy of any paper tape. It can be loaded by hand from the Control Panel or by using the IPL routine. Once loaded push the FEED HOLES button on the Paper Tape Punch to provide a leader of about 10cm, then load the paper tape you want punched on the Paper Tape Reader and push the RUN button. The program will read the tape and punch a copy simultaneously. When the Reader and Punch stop push the FEED HOLES button to provide a trailing edge for the tape. NOTE: The program has been written to reproduce any length of paper tape so no character counter has been included so the program must be loaded for each tape you wish to reproduce.

3.63 Program DUMP

This program enables the engineer to dump a selected area of memory on either the ASR or PER3100. It can be loaded either by hand from the Control Panel switches or using a program tape that contains the MINI-IPL (paragraph 3-72) and program produced by one of the methods described in paragraph 3-64 to 3-70. If the MINI-IPL method is used care must be taken that the program does not overwrite part of the area you wish dumped. The instructions for operating this program are given in the program listing and when the last address has been dumped the program branches back to the start ready for possible new parameters.

COPY

```
00000                          IDENT    COPY
00001                  *
00002                  *       THIS PROGAM  READS A PAPER TAPE AND C PUNCHES A COPY
00003                  *
00004          0000    BEGIN    EQU      *
00005                          RORG     BEGIN+/80
00006  0080    207F    START    HLT
00007  0082    0101            LDK      A1,1
00008  0084    0200            LDK      A2,0
00009  0086    01F0    READ     CIO      A1,1,/20
00010  0088    5C04            RB(NA)   *-2
00011  008A    4D20            INR      A5,0,/20
00012  008C    5C04            RB(NA)   *-2
00013  008E    01A0            CIO      A1,0,/20
00014  0090    0CF0            SST      A4,/20
00015  0092    5C04            RB(NA)   *-2
00016  0094    2420            ANK      A4,/20
00017  0096    EC20            CWK      A4,/20
       0098    0020
00018  009A    581C            RB(F)    START
00019  009C    02F0            CIO      A2,1,/30
00020  009E    5C04            RB(NA)   *-2
00021  00A0    4530            OTR      A5,0,/30
00022  00A2    5C04            RB(NA)   *-2
00023  00A4    02B0            CIO      A2,0,/30
00024  00A6    0BF0            SST      A3,/30
00025  00A8    5C04            RB(NA)   *-2
00026  00AA    5F26            RB       READ
00027                          END      START
```

DUMP

```
00000                                IDENT   DUMP
00001          0000          BEGIN   EQU     *
00002                                RORG    BEGIN+/80
00003                        *
00004                        *       THIS PROGRAM ENABLES THE ENGINEER TO DUMP A SELECTED
00005                        *       AREA OF MEMORY ON EITHER THE ASR OR PER 3100
00006                        *
00007                        *
00008                        *       LOAD THE STARTING ADDRESS OF THE AREA TO BE
00009                        *       DUMPED INTO REGISTER A7 FROM THE CONTROL PANEL
00010                        *
00011                        *       LOAD THE ENDING ADDRESS OF THE AREA INTO
00012                        *       REGISTER A8 FROM THE CONTROL PANEL
00013                        *
00014                        *       LOAD THE PROGRAM STARTING ADDRESS INTO
00015                        *       REGISTER A0 FROM THE CONTROL PANEL THEN
00016                        *       PUSH THE RUN BUTTON
00017                        *
00018                        *
00019  0080  FEFF                    DATA    /FEFF
00020  0082  0000                    DATA    0
00021  0084  207F            START   HIT
00022  0086  20BF                    INH                             THE PROGRAM RUNS IN INHIBIT MODE
00023                        *
00024  0088  0404                    LDK     A4,4                    LOAD PARAMETERS FOR OUTPUT CHARACTER
00025                        *                                       WITH EVEN PARITY
00026  008A  4400                    CIO     A4,1,/10                SEND START COMMAND TO DEVICE
00027  008C  813C            WORD    LDR*    A1,A7                   LOAD THE CONTENTS OF THE MEMORY ADDRESS
00028                        *                                       CONTAINED IN A7 INTO A1
00029  008E  0204                    LDK     A2,4                    LOAD CHARACTER COUNT FOR WORD
00030  0090  060F            CONT    LDK     A6,/F                   LOAD MASK PATTERN
00031  0092  A604                    ANR     A6,A1                   FIND THE PRINTABLE CHAR ADDRESS IN TABLE
00032  0094  E558                    LC      A5,TABLE,A6             AND LOAD INTO REGISTER A5
       0096  00C0  R
00033  0098  E549                    SC      A5,BUFF+1,A2            THEN STORE IN BUFF
       009A  0001  R
00034  009C  39E4                    SRC     A1,4                    SHIFT TO ISOLATE NEXT CHAR
00035  009E  1A01                    SUK     A2,1                    SUBTRACT ONE FROM CHAR COUNT
00036  00A0  5C12                    RB(NZ)  CONT                    AND BRANCH BACK IF NOT LAST CHAR IN WORD
00037  00A2  E348            OUCHAR  LC      A3,BUFF,A2              LOAD CHAR FROM BUFF INTO A3
       00A4  0000  R
00038  00A6  4310                    OTR     A3,0,/10                AND SEND TO THE DEVICE
00039  00A8  5C04                    RB(NA)  *-2
00040  00AA  1201                    ADK     A2,1                    UPDATE BUFF CHAR COUNT
00041  00AC  FA20                    CWK     A2,6                    AND CHECK FOR LAST CHAR SENT
       00AE  0006
00042  00B0  5C10                    RB(NF)  OUCHAR                  AND BRANCH BACK IF NOT LAST CHAR
00043  00B2  1702                    ADK     A7,2                    UPDATE MEMORY ADDRESS COUNTER
00044  00B4  FF02                    CWR     A7,A8                   AND CHECK FOR LAST ADDRESS DUMPED
00045  00B6  5D2C                    RB(NG)  WORD                    IF NOT LAST OUTPUT NEW WORD
```

```
DUMP
00046                          *                                    IF LAST ADDRESS THEN
00047  00B8  4490                      CIO       A4,0,/10          SEND STOP COMMAND TO THE DEVICE
00048  00BA  4CD0                      SST       A4,/10            AND GET STATUS
00049  00BC  5C04                      RB(NA)    *+2
00050  00BE  5F3C                      RB        START             DUMP FINISHED GO TO START OF PROGRAM
00051  00C0  3031              TABLE   DATA      '0123456789ABCDEF'
       00C2  3233
       00C4  3435
       00C6  3637
       00C8  3839
       00CA  4142
       00CC  4344
       00CE  4546
00052  00D0  0A0D              BUFF    DATA      /0A0D
00053                                  END       START

          SYMBOL TABLE

BEGIN   0000  R   BUFF    00D0  R   CONT    0090  R   DUCHAR  00A2  R
START   0084  R   TABLE   00C0  R   WORD    008C  R

          ASS.ERR.    00000
:EOF
PROG ELAPSED TIME:  00H-00M-00S-000MS-
```

3.64 PRODUCING PROGRAMS ON PAPER TAPE

For short programs (less than 80 characters), the tape produced can be input using the normal IPL routine; for programs with more than 80 characters MINI-IPL, suggested in this section, will have to be included before your own programs which can then be loaded using the IPL routine.

3.65 Using the ASR Punch in LOCAL Mode

If the ASR is fitted with a punch the hexadecimal characters of the program can be input from the keyboard provided the following rules are observed:

- The LOCAL/REMOTE switch must be set to LOCAL and the punch must be switched ON.
- The punch will treat any input from the keyboard as data and will reproduce the appropriate code on tape so if you make an error while typing in the instructions there is no way it can be corrected and you will have to begin again.
- To produce the tape leader and the trailing end of tape press the HERE IS key three times.
- The number keys 0 to 9 can be used normally but the letter keys A B C D E and F MUST NOT be used, instead use the J K L M N and O keys respectively which will produce the correct hexadecimal code on the tape.
- If the number of characters exceeds the line length the punch must be switched OFF before you press the CR and LF keys, and it must be switched ON again before you continue typing in the program instructions.
- The resulting program tape is in 8-bit ASCII but the Bootstrap can only decode Object code (either 8+8 or 4x4) so to load the program tape using the IPL routine the Control Panel switches must first be set for 4x4 mode (i.e. switches 0, 3 and 10 set to value 1). The program will then be loaded from the PTR.

3.66 If the program produced is less than 80 8-bit characters the Bootstrap will keep on reading (looking for the missing characters) until all the tape has passed through the read head. This does not affect the instructions that have been loaded but you must stop the CPU by pushing the INST button, then push the MC button to reset the device and CU logic. Any parameters can now be loaded into registers and the program will run normally.

3.67 Program ASC4x4

This program allows you to write programs using the operator's peripheral. The hexadecimal characters of the program can be entered in a similar way to that described above using the number keys 0 to 9 and letter keys J to O. However, the program is more flexible in that erroneous characters can be ignored and certain other characters are recognised which make the production of a program tape much easier. Once the program has been loaded put the start address of the program into register A0 and push the RUN button. The program will run and in addition to the keys mentioned above the following keys are significant:

- ↑ or ^ — The symbol and position of this key will depend on the device being used but its function and 7-bit code (/5E) are the same. The use of this key is to delete a character typed in error. When the program recognises this character it down dates the buffer counter and branches back to the input routine to accept a new character.
- Space Bar — The Space Bar is used to tell the program that the four hexadecimal characters of the program instruction are valid and that any corrections have been made. When the program recognises this character code (/20) it branches to the translate routine and stores them in the output buffer; it then branches back to the input routine for four new characters.
- # — This key is used to indicate that the last program character has been typed. When the program recognises this character code (/23) it branches to the CR/LF routine.

3.68 The program translates the valid characters received into 8-bit ASCII format and store them in a buffer until 40 valid characters have been received or the character has been recognised. It then sends the CR and IF characters to the Operator's device and starts the punch routine. If the last character code has not been recognised the program branches back to the input routine to accept new characters; if it has been recognised it branches back to the start of program and stops.

ASC4X4

```
00000                          IDENT    ASC4X4
00001                    *
00002                    *     THIS PROGRAM ALLOWS ENGINEERS TO WRITE PROGRAMS ON THE OPERATOR'S
00003                    *     PERIPHERAL AND THEN PUNCHES THE PROGRAM IN ASCII FOR 4X4 LOADING
00004                    *
00005          0000      BEGIN    EQU      *
00006                            BORG     BEGIN+/AA
00007                    *
00008                    *     THIS ROUTINE ACCEPTS CHARACTERS FROM THE
00009                    *     OPERATOR'S PERIPHERAL IN THE NON ECHO MODE
00010                    *
00011                    *
00012  00AA    207F      START    HLT
00013  00AC    20BF               INH
00014  00AE    0100      OUTCT    LDK      A1,0                 OUTPUT BUFFER COUNTER ZEROISED
00015  00B0    0300      INCT     LDK      A3,0                 ZEROISE INPUT BUFFER COUNTER
00016  00B2    0601      INPT     LDK      A6,1
00017  00B4    46D0               CIO      A6,1,/10
00018  00B6    5C04               BB(4)    *-2
00019  00B8    4D10               INR      A5,0,/10
00020  00BA    5C04               BB(4)    *-2
00021  00BC    4690               CIO      A6,0,/10
00022  00BE    4CD0               SST      A4,/10
00023  00C0    5C04               BB(4)    *-2
00024  00C2    0600               LDK      A6,0
00025  00C4    46D0               CIO      A6,1,/10
00026  00C6    5C04               BB(4)    *-2
00027  00C8    4510               OTR      A5,0,/10
00028  00CA    5C04               BB(4)    *-2
00029  00CC    4690               CIO      A6,0,/10
00030  00CE    4CD0               SST      A4,/10
00031  00D0    5C04               BB(4)    *-2
00032  00D2    257F               ANK      A5,/7F               MODIFIES CHARACTER TO 7 DATA BITS
00033                    *
00034                    *     THIS ROUTINE CHECKS FOR  ERRONEOUS CHARS AND END OF PROGRAM
00035                    *
00036  00D4    FD20               CWK      A5,/5E               CHECK IF LAST CHAR WAS TYPED IN ERROR
       00D6    005E
00037  00D8    5404               BF(NE)   SPCE
00038  00DA    1801               SUK      A3,1                 THE LAST CHAR WAS TYPED IN ERROR SO
00039  00DC    5F2C               BB       INPT                 DOWNDATE CHAR COUNTER AND READ NEXT CHAR
00040  00DE    FD20      SPCE     CWK      A5,/20               CHECK FOR SPACE BAR CHARACTER
       00E0    0020
00041  00E2    500E               BF(E)    TRAN                 GO TO TRAN IF 4 VALID CHARS INPUT
00042  00E4    FD20               CWK      A5,/23               CHECK FOR END OF PROGRAM   #
       00E6    0023
00043  00E8    5038               BF(E)    CRLF
00044                    *
00045                    *     THIS ROUTINE STORES INPUT CHARS IN TEMPORARY BUFFER
00046                    *
```

```
00047  00FA  E540          SC      A5,TEMBUF,A3        STORE CHAR IN TEMBUF
       00FC  0158  R
00048  00FE  1301          ADK     A3,1                UPDATE INPUT BUFFER COUNT
00049  00F0  5F40          BR      INPT                GO TO INPUT ROUTINE
00050                *
00051                *      THIS ROUTINE TRANSLATES CHARS INTO ASCII FORMAT
00052                *
00053  00F2  0300    TRAN  LDK     A3,0                ZEROISE TEMBUF COUNTER
00054  00F4  B208    TRAN1 XRR     A2,A2               CLEAR REGISTER
00055  00F6  E54C          LC      A5,TEMBUF,A3        GET CHAR FROM TEMBUF
       00F8  0158  R
00056  00FA  B214          LDR     A2,A5               SAVE CHAR
00057  00FC  250F          ANK     A5,/0F              SAVE RIGHT MOST BITS
00058  00FE  2240          ANK     A2,/40
00059  0100  EA20          CWK     A2,/40              CHECK IF CHAR IS DECIMAL
       0102  0040
00060  0104  5404          BF(NE)  DECIM
00061  0106  15C0          ADK     A5,/C0              ALPHA CHAR
00062  0108  9702          BF      STORE
00063  010A  1580    DECIM ADK     A5,/80              DECIMAL CHAR
00064                *
00065                *      THIS ROUTINE STORES THE PROGRAM CHARS IN BUFFER
00066                *      AND SENDS CR LF AFTER 40 VALID CHARS HAVE BEEN RECIEVED
00067                *
00068  010C  E545    STORE SC      A5,BUFF,A1          STORE VALID CHAR
       010E  015E  R
00069  0110  1101          ADK     A1,1                UPDATE CHAR COUNTER
00070  0112  1301          ADK     A3,1                UPDATE TEMBUF COUNT
00071  0114  EB20          CWK     A3,4                CHECK IF ALL 4 CHARS HAVE BEEN TRANSLATE
       0116  0004
00072  0118  9C26          BR(NE)  TRAN1               TRANSLATE NEW CHARACTER IF NOT ZERO
00073  011A  E920          CWK     A1,40               COMPARE FOR BUFFER FULL
       011C  0028
00074  011E  5002          BF(F)   CRLF                IF YES SEND CR/LF CHARS THEN PUNCH BUFFE
00075  0120  5F72          BR      INCT                IF ZERO GO TO INPUT ROUTINE FOR NEW CHAR
00076  0122  070A    CRLF  LDK     A7,/0A              LOAD LF CHAR
00077  0124  46D0          CIO     A6,1,/10
00078  0126  4710          OTR     A7,0,/10            AND SEND TO TYPEWRITER
00079  0128  5C04          BR(4)   *-2
00080  012A  070D          LDK     A7,/0D              LOAD CR CHAR
00081  012C  4710          OTR     A7,0,/10            AND SEND TO TYPEWRITER
00082  012E  5C04          BR(4)   *-2
00083  0130  4690          CIO     A6,0,/10
00084  0132  4CD0          SST     A4,/10
00085  0134  5C04          BR(4)   *-2
00086                *
00087                *      THIS ROUTINE PUNCHES THE ASCII PROGRAM ON TAPE
00088                *
00089  0136  0200          LDK     A2,0                ZEROISE  BUFF COUNTER
00090  0138  46F0          CIO     A6,1,/30            START PUNCH
00091  013A  5C04          BR(4)   *-2
00092  013C  E348    CONT  LC      A3,BUFF,A2          LOAD CHAR FROM BUFF
       013E  015E  R
00093  0140  4330          OTR     A3,0,/30            AND PUNCH
00094  0142  5C04          BR(4)   *-2
00095  0144  1201          ADK     A2,1                UPDATE BUFF COUNTER
00096  0146  1901          SUK     A1,1                SUBTRACT 1 FROM CHAR COUNT
```

```
00097  0148  5C0E              RB(4)   CONT            AND CHECK FOR LAST CHAR
00098  014A  4680              CTO     A6,0,/30        THEN STOP PUNCH
00099  014C  4CF0              SST     A4,/30
00100  014E  5C04              RB(NA)  *-2
00101  0150  FD20              CWK     A5,/23          CHECK FOR LAST PROGRAM CHAR
       0152  0023
00102  0154  5CAA              RB(4)   OUTCT           NO?  ACCEPT MORE CHARACTERS
00103  0156  5EAE              RB      START
00104  0158            TEMBUF  RES     3               TEMPORARY BUFFER
00105  015E            BUFF    RES     25
00106                          END     START
```

SYMBOL TABLE

```
BEGIN   0000  R   BUFF    015E  R   CONT    013C  R   CRLP    0122  R
DECIM   010A  R   INCT    0080  R   INPT    0082  R   OUTCT   00AE  R
SPCE    00DE  R   START   00AA  R   STORE   010C  R   TEMBUF  0158  R
TRAN    00E2  R   TRAN1   00E4  R
```

3.69 The program has 74 instructions and in the program listing the start address is shown as 00AA. This is so that if your only means of producing program tapes is by one of the examples given in this section you can include the MINI-IPL in front of the program and load it using the 4x4 IPL routine. If you have access to an Assembler and Linkage Editor you will have no trouble in producing a master tape of this program, but if you have to load it by hand from the Control Panel switches before reproducing it by one of the methods suggested in this section great care must be taken that you do not introduce errors into the program whilst it is being loaded.

3.70 Program HEXTAP

This program operates exactly the same way as the program ASC4x4 and all instructions regarding the use of that program apply to this program. The only differences are that this program operates in Echo mode and the output tape is punched in 8+8 hexadecimal format. If you include the MINI-IPL in front of the program it can then be loaded from the PTR using the normal IPL routine.

HEXTAP

```
00000                          IDENT    HEXTAP
00001                  *
00002                  *       THIS PROGRAM ALLOWS ENGINEERS TO WRITE PROGRAMS ON THE OPERATOR'S
00003                  *       PERIPHERAL AND PUNCHES THE PROGRAM TAPE IN HEXADECIMAL 8+8 FORMAT
00004                  *
00005          0000    BEGIN   EQU      *
00006                          RORG     BEGIN+/AA
00007                  *
00008                  *       THHIS ROUTINE ACCEPTS CHARACTERS FROM THE OPERATOR'S PERIPHERAL
00009                  *       IN ECHO MODE
00010                  *
00011  00AA  207F      START   HLT
00012  00AC  20BF              INH
00013  00AE  0100      OUTCT   LDK      A1,0                  OUTPUT BUFFER COUNTER ZEROISED
00014  00B0  0300      INCT    LDK      A3,0                  ZEROISE INPUT BUFFER COUNTER
00015  00B2  0621      INPT    LDK      A6,/21
00016  00B4  4600              CIO      A6,1,/10
00017  00B6  5C04              RB(4)    *-2
00018  00B8  4D10              INR      A5,0,/10
00019  00BA  5C04              RB(4)    *-2
00020  00BC  4690              CIO      A6,0,/10
00021  00BE  4CD0              SST      A4,/10
00022  00C0  5C04              RB(4)    *-2
00023  00C2  257F              ANK      A5,/7F                MODIFIES CHARACTER TO 7 DATA BITS
00024                  *
00025                  *       THIS ROUTINE CHECKS FOR  ERRONEOUS CHARS AND END OF PROGRAM
00026                  *
00027  00C4  ED20              CWK      A5,/5E                CHECK IF LAST CHAR WAS TYPED IN ERROR
       00C6  005E
00028  00C8  5404              RF(NE)   SPCE
00029  00CA  1B01              SUK      A3,1                  THE LAST CHAR WAS TYPED IN ERROR SO
00030  00CC  5F1C              RB       INPT                  DOWNDATE CHAR COUNTER AND READ NEXT CHAR
00031  00CE  ED20      SPCE    CWK      A5,/20                CHECK FOR SPACE BAR CHARACTER
       00D0  0020
00032  00D2  500E              RF(E)    TRAN                  GO TO TRAN IF 4 VALID CHARS INPUT
00033  00D4  ED20              CWK      A5,/23                CHECK FOR END OF PROGRAM   #
       00D6  0023
00034  00D8  502C              RF(E)    CRLP
00035                  *
00036                  *       THIS ROUTINE STORES INPUT CHARS IN TEMPORARY BUFFER
00037                  *
00038  00DA  E54D              SC       A5,TEMBUF,A3          STORE CHAR IN TEMBUF
       00DC  013E  R
00039  00DE  1301              ADK      A3,1                  UPDATE INPUT BUFFER COUNT
00040  00E0  5F30              RB       INPT                  GO TO INPUT ROUTINE
00041                  *
00042                  *       THIS ROUTINE TRANSLATES CHARS INTO ASCII FORMAT
00043                  *
00044  00E2  0300      TRAN    LDK      A3,0                  ZEROISE TEMBUF COUNTER
00045  00E4  0404              LDK      A4,4                  LOAD SHIFT COUNTER
```

```
00046  00F6  B208        TRAN1   XRR     A2,A2                   CLEAR REGISTER
00047  00F8  3A44                SLL     A2,4                    SHIFT CHARS 4 PLACES LEFT
00048  00FA  E54C                LC      A5,TEMBUF,A3            GET CHAR FROM TEMBUF
       00FC  013E  R
00049  00FE  250F                ANK     A5,/0F                  SAVE RIGHT MOST BITS
00050  0100  9214                ADR     A2,A5                   ADD 4 BIT CHAR FROM A5 INTO A2
00051  0102  1301                ADK     A3,1                    UPDATE TEMBUF COUNTER
00052  0104  1C01                SUK     A4,1                    DOWNDATE SHIFT COUNTER AND IF NOT ZERO
00053  0106  5C10                RB(NZ)  TRAN1+2                 BRANCH AND TRANSLATE NEW CHAR
00054                     *
00055                     *       THIS ROUTINE STORES THE PROGRAM CHARS IN BUFFER
00056                     *       AND SENDS CR LF AFTER 40 VALID CHARS HAVE BEEN RECIEVED
00057                     *
00058  0108  8245        STORE   ST      A2,BUFF,A1              STORE TWO 8 BIT HEXADECIMAL CHARS IN BUF
       010A  0144  R
00059  010C  1102                ADK     A1,2                    UPDATE CHAR COUNTER
00060  010E  F920                CWK     A1,20                   COMPARE FOR BUFFER FULL
       0110  0014
00061  0112  5002                RF(E)   CRLP                    IF YES SEND CR/LF CHARS THEN PUNCH BUFFE
00062  0114  5F56                RB      INCT                    IF ZERO GO TO INPUT ROUTINE FOR NEW CHAR
00063  0116  070A        CRLP    LDK     A7,/0A                  LOAD LF CHAR
00064  0118  0600                LDK     A6,0                    LOAD START OUTPUT FOR DEVICE
00065  011A  4609                CTO     A6,1,/10
00066  011C  4710                OTR     A7,0,/10                AND SEND TO TYPEWRITER
00067  011E  5C04                RB(4)   *-2
00068  0120  070D                LDK     A7,/0D                  LOAD CR CHAR
00069  0122  4710                OTR     A7,0,/10                AND SEND TO TYPEWRITER
00070  0124  5C04                RB(4)   *-2
00071  0126  4690                CTO     A6,0,/10
00072  0128  4C00                SST     A4,/10
00073  012A  5C04                RB(4)   *-2
00074                     *
00075                     *       THIS ROUTINE PUNCHES THE ASCII PROGRAM ON TAPE
00076                     *
00077  012C  0200                LDK     A2,0                    ZEROISE  BUFF COUNTER
00078  012E  46F0                CTO     A6,1,/30                START PUNCH
00079  0130  5C04                RB(4)   *-2
00080  0132  E348        CONT    LC      A3,BUFF,A2              LOAD CHAR FROM BUFF
       0134  0144  R
00081  0136  4330                OTR     A3,0,/30                AND PUNCH
00082  0138  5C04                RB(4)   *-2
00083  013A  1201                ADK     A2,1                    UPDATE BUFF COUNTER
00084  013C  1901                SUK     A1,1                    SUBTRACT 1 FROM CHAR COUNT
00085  013E  5C0E                RB(4)   CONT                    AND CHECK FOR LAST CHAR
00086  0140  4680                CTO     A6,0,/30                THEN STOP PUNCH
00087  0142  4CF0                SST     A4,/30
00088  0144  5C04                RB(NA)  *-2
00089  0146  ED20                CWK     A5,/23                  CHECK FOR LAST PROGRAM CHAR
       0148  0023
00090  014A  5C8E                RB(4)   OUTCT                   NO?  ACCEPT MORE CHARACTERS
00091  014C  5F94                RB      START
00092  014E              TEMBUF  RES     3                       TEMPORARY BUFFER
00093  0154              BUFF    RES     25
00094                             END     START
```

SYMBOL TABLE

```
BEGIN   0000  R   BUFF    0144  R   CONT    0122  R   CRLP    0106  R
INCT    00B0  R   INPT    00B2  R   OUTCT   00AE  R   SPCE    00CE  R
START   00AA  R   STORE   00F8  R   TEMBUF  013E  R   TRAN    00F2  R
TRAN1   00F6  R
```

3.71 SUGGESTIONS FOR SOPHISTICATED PROGRAMS

As a general rule sophisticated programs perform more than one function and are longer than 20 instructions (so cannot easily be loaded by hand). An easy way to produce these programs is to take routines from simple programs (or routines you have written yourself) and link them together adding specific routines to perform the tasks you want. The two previous programs are examples of this technique. They comprise routines for the operator's device, the Paper Tape Punch, and a linking routine that translates the character codes input into either ASCII or hexadecimal character code. Make sure that when the same register is used for more than one function you do not corrupt data needed in another part of the program, that the buffer address is the same, that the displacement value of branch instructions is correct for the new program, and that all superfluous instructions from either routine have been omitted. The program examples that follow have all been produced in this way and should enable you to produce your own programs capable of performing specific and more complicated tasks even with a basic system.

3.72 MINI-IPL Routine

For programs longer than 80 characters some form of software IPL is needed to load the final part of the program into memory. You will remember that the Bootstrap is programmed to load 80 8-bit characters from the device (whose address and input mode are set up on the DATA switches) before it branches to address 84, so if we start the MINI-IPL at this address it will then load the rest of the program. The first four characters of the IPL (/FFFF and 0000) are not used by the routine but are needed to make sure that the first instruction of the routine is in location /84. This instruction loads the number of characters the IPL has to read into register A5. To calculate the character count loaded into A5 count the number of locations needed by the MINI-IPL (starting with data /FFFF) plus your program and multiply the result by two to give the number of characters. Now subtract 80 (which is the number of characters loaded by the Bootstrap) and the remainder is the number you need to load into register A5. If this number is greater than 255 you will have to change the LDK instruction into an LDKL putting the count into the second word. Make sure that the count is accurate otherwise the IPL will either not load all the instructions (count too short) or will load the blank tape of the trailing end as all zeros. In the latter case if only one or two such characters are read it should not affect the program, but if the End of tape is detected then it will almost certainly cause the program to abort. The rest of the instructions form a simple routine to read characters from the PTR and store them in the address pointed to by register A6, which is the register used by the Bootstrap to load the first 80 characters. When this routine has read all the characters (register A5's value equals zero) the PTR is stopped and, unless the first instruction of your program is Halt, your program is entered and starts to run. You can modify this routine to suit your own requirements, for example the start address of the program can be changed to any location in memory greater than /9E (which is the normal start address) by inserting a Load Constant instruction, as the first instruction of the routine, that contains the new start address of the program. You will also have to insert an RF or ABL instruction (with the new start address as the displacement value) after the RB instruction in location 9C. The remainder of the first 80 characters must all be zero otherwise the modified IPL will overwrite part of your program.

MINI-IPL

```
00012                    *
00013                    *        THIS IS A MINI IPL FOR LOADING PAPER TAPE IN 8+8 HEXADECIMAL FORMAT
00014                    *
00015  0080  FFFF               DATA     /FFFF
00016  0082  0000               DATA     0000
00017  0084  0550               LDK      A5,80              LOAD CHAR COUNT FOR REST OF PROGRAM
00018  0086  0101               LDK      A1,1               LOAD PARAMETERS FOR CIO START
00019  0088  41E0               CIO      A1,1,/20           AND SEND TO CU
00020  008A  5C04               RB(NA)   *-2
00021  008C  4F20          INR  INR      A7,0,/20           SEND INR TO CU
00022  008E  5C04               RB(NA)   *-2
00023  0090  E739               SCR      A7,A6              STORE CHARACTER IN LOCATION
00024  0092  1601               ADK      A6,1               UPDATE MEMORY ADDRESS
00025  0094  1D01               SUK      A5,1               DECREMENT CHAR COUNTER
00026  0096  5C0C               RB(NZ)   INR                AND READ ANOTHER CHAR IF NOT ZERO
00027  0098  41A0               CIO      A1,0,/20           SEND STOP COMMNAD TO CU
00028  009A  4FF0               SST      A7,/20             AND THEN SEND SST COMMAND
00029  009C  5C04               RB(NA)   *-2
00030                    *
```

3.73 Program MINDUM

This program is an example of how to link the MINI-IPL to a program longer than 80 characters (in this case 82 characters), so that it can be loaded using the IPL routine. The program chosen for the exercise was DUMP and the first step was to count the number of characters, including the two dummy instructions Data /FFFF and Data 0000, which was 82 and therefore two characters too long to be loaded with the IPL routine. The number of characters needed for the MINI-IPL (excluding the two dummy instructions shown in the listing) is 26 so the total number of characters to be loaded by the MINI-IPL is 28. The program tape was produced using the program HEXTAP starting with the two dummy instructions, then the MINI-IPL and finally by the instructions for the program DUMP (excluding the two dummy instructions shown in the listing). The completed tape can now be loaded using the IPL routine and the instructions for running the programs are the same as those given for DUMP.

MINDUM

```
00000                          IDENT    MINDUM
00001          0000    BEGIN   EQU      *
00002                          RORG     BEGIN+/80
00003                  *
00004                  *       THIS PROGRAM ENABLES THE ENGINEER TO DUMP A SELECTED
00005                  *       AREA OF MEMORY ON EITHER THE ASR OR PER 3100
00006                  *
00007                  *       IT CONTAINS THE MINI-IPL AND CAN BE LOADED
00008                  *       USING THE IPL ROUTINE
00009                  *
00010                  *       LOAD THE STARTING ADDRESS OF THE AREA TO BE
00011                  *       DUMPED INTO REGISTER A7 FROM THE CONTROL PANEL
00012                  *
00013                  *       LOAD THE ENDING ADDRESS OF THE AREA INTO
00014                  *       REGISTER A8 FROM THE CONTROL PANEL
00015                  *
00016                  *       PUSH THE RUN BUTTON
00017                  *
00018                  *
00019                  *       THIS IS A MINI IPL FOR LOADING PAPER TAPE IN 8+8 HEXADECIMAL FORMAT
00020                  *
00021  0080  FFFF              DATA     /FFFF
00022  0082  0000              DATA     0000
00023  0084  051C              LDK      A5,28                 LOAD CHAR COUNT FOR REST OF PROGRAM
00024  0086  0101              LDK      A1,1                  LOAD PARAMETERS FOR CIO START
00025  0088  41F0              CIO      A1,1,/20              AND SEND TO CU
00026  008A  5C04              RB(NA)   *-2
00027  008C  4F20      INR     INR      A7,0,/20              SEND INR TO CU
00028  008E  5C04              RB(NA)   *-2
00029  0090  F739              SCR      A7,A6                 STORE CHARACTER IN LOCATION
00030  0092  1601              ADK      A6,1                  UPDATE MEMORY ADDRESS
00031  0094  1D01              SUK      A5,1                  DECREMENT CHAR COUNTER
00032  0096  5C0C              RB(NZ)   INR                   AND READ ANOTHER CHAR IF NOT ZERO
00033  0098  41A0              CIO      A1,0,/20              SEND STOP COMMNAD TO CU
00034  009A  4FE0              SST      A7,/20                AND THEN SEND SST COMMAND
00035  009C  5C04              RB(NA)   *-2
00036                  *
00037                  *       THE PROGRAM STARTS HERE
00038                  *
00039  009E  207F      START   HLT
00040  00A0  20BF              INH                            THE PROGRAM RUNS IN INHIBIT MODE
00041                  *
00042  00A2  0404              LDK      A4,4                  LOAD PARAMETERS FOR OUTPUT CHARACTER
00043                  *                                      WITH EVEN PARITY
00044  00A4  4400              CIO      A4,1,/10              SEND START COMMAND TO DEVICE
00045  00A6  813C      WORD    LDR*     A1,A7                 LOAD THE CONTENTS OF THE MEMORY ADDRESS
00046                  *                                      CONTAINED IN A7 INTO A1
00047  00A8  0204              LDK      A2,4                  LOAD CHARACTER COUNT FOR WORD
00048  00AA  060F      CONT    LDK      A6,/F                 LOAD MASK PATTERN
00049  00AC  A604              ANR      A6,A1                 FIND THE PRINTABLE CHAR ADDRESS IN TABLE
```

```
00050  00AE  F558            LC      A5,TABLE,A6         AND LOAD INTO REGISTER A5
       00B0  00DA  R
00051  00B2  F549            SC      A5,BUFF+1,A2        THEN STORE IN BUFF
       00B4  00FB  R
00052  00B6  39E4            SRC     A1,4                SHIFT TO ISOLATE NEXT CHAR
00053  00B8  1A01            SUK     A2,1                SUBTRACT ONE FROM CHAR COUNT
00054  00BA  5C12            RB(NZ)  CONT                AND BRANCH BACK IF NOT LAST CHAR IN WORD
00055  00BC  F348    OUCHAR  LC      A3,BUFF,A2          LOAD CHAR FROM BUFF INTO A3
       00BE  00FA  R
00056  00C0  4310            OTR     A3,0,/10            AND SEND TO THE DEVICE
00057  00C2  5C04            RB(NA)  *+2
00058  00C4  1201            ADK     A2,1                UPDATE BUFF CHAR COUNT
00059  00C6  FA20            CWK     A2,6                AND CHECK FOR LAST CHAR SENT
       00C8  0006
00060  00CA  5C10            RB(NE)  OUCHAR              AND BRANCH BACK IF NOT LAST CHAR
00061  00CC  1702            ADK     A7,2                UPDATE MEMORY ADDRESS COUNTER
00062  00CE  FF02            CWR     A7,A8               AND CHECK FOR LAST ADDRESS DUMPED
00063  00D0  5D2C            RB(NG)  WORD                IF NOT LAST OUTPUT NEW WORD
00064                  *                                  IF LAST ADDRESS THEN
00065  00D2  4490            CIO     A4,0,/10            SEND STOP COMMAND TO THE DEVICE
00066  00D4  4CD0            SST     A4,/10              AND GET STATUS
00067  00D6  5C04            RB(NA)  *+2
00068  00D8  5F3C            RB      START               DUMP FINISHED GO TO START OF PROGRAM
00069  00DA  3031    TABLE   DATA    '0123456789ABCDEF'
       00DC  3233
       00DE  3435
       00E0  3637
       00E2  3839
       00E4  4142
       00E6  4344
       00E8  4546
00070  00EA  0A0D    BUFF    DATA    /0A0D
00071                  END     START
```

SYMBOL TABLE

```
BEGIN   0000  R   BUFF    00FA  R   CONT    00AA  R   INR     008C  R
OUCHAR  00BC  R   START   009F  R   TABLE   00DA  R   WORD    00A6  R
```

3.74 Program MEM57

This is another example of how to link several routines to the simple program MEMHAN to produce a sophisticated program that will carry out a check of the entire memory without intervention from the engineer unless there is a detected error. It is more of a confidence program (after a memory fault has been repaired) than a specific test of a selected area whilst trying to locate a fault. The routines used are:

- The MINI-IPL routine so that the program can be loaded using the IPL routine. You will see that the only change necessary (from the other two listings in this section) is to load the character count register A5 with a count of 176.
- A routine to check the size of memory in an unknown system without doing a physical check on the cards. The routine assumes a maximum memory size of 32K and tries to write in the last location. If unsuccessful it subtracts 4K from the address in register A9 and tries to store the pattern in the new address. In the original routine when the size of memory had been found it branched to a Halt instruction so that the memory size could be read from register A9; so for this program the Halt instruction became the first instruction of the next routine.
- A routine to store the Start and End addresses of the area tested into a buffer called MEMADD. The previous routine has already computed the End address and this is store in the first word of the buffer; the Start address is the first location after the end of this program and it is stored in the second word of the buffer.
- Four Test Patterns and a Software Timer so that the series of test patterns will be repeated a number of times (in this case 25).
- The program MEMHAN slightly modified so that the Start and End addresses are taken from the buffer MEMADD instead of from registers, and the Test Patterns would be selected automatically.
- Routines to compute the New Start and New End addresses, to compute the New Load Area for the program, and to compute the New Addresses for the buffers for MEMADD, Test Patterns, and Timer.
- A routine to transfer the program to the New Load Area (which will be in the top end of memory) and to re-start the program MEMHAN so that it checks the bottom end of memory.

Once the program has been loaded it starts automatically and when it has checked the complete memory it stops on a Halt instruction that has been loaded into the last address of memory. If an error is detected during the running of the program it will stop on the HLT instruction in the Read routine and you can use the same technique, to check the location in error and the Write and Read patterns, as described for MEMHAN.

MEM57

```
00000                                  IDENT    MEM57
00001                          *
00002                          *       THIS PROGRAM TESTS THE ENTIRE MEMORY
00003                          *
00004          0000            BEGIN   EQU      *
00005                                  BORG     BEGIN+/80
00006                          *
00007                          *       THIS IS A MINI IPL FOR LOADING PAPER TAPE IN 8+8 HEXADECIMAL FORMAT
00008                          *
00009  0080    FFFF                    DATA     /FFFF
00010  0082    0000                    DATA     0000
00011  0084    0580                    LDK      A5,176                LOAD CHAR COUNT FOR REST OF PROGRAM
00012  0086    0101                    LDK      A1,1                  LOAD PARAMETERS FOR CIO START
00013  0088    41E0                    CIO      A1,1,/20              AND SEND TO CU
00014  008A    5C04                    BR(NA)   *+2
00015  008C    4F20            INR     INR      A7,0,/20              SEND INR TO CU
00016  008E    5C04                    BR(NA)   *+2
00017  0090    E739                    SCR      A7,A6                 STORE CHARACTER IN LOCATION
00018  0092    1601                    ADK      A6,1                  UPDATE MEMORY ADDRESS
00019  0094    1D01                    SUK      A5,1                  DECREMENT CHAR COUNTER
00020  0096    5C0C                    BR(NZ)   INR                   AND READ ANOTHER CHAR IF NOT ZERO
00021  0098    41A0                    CIO      A1,0,/20              SEND STOP COMMNAD TO CU
00022  009A    4FE0                    SST      A7,/20                AND THEN SEND SST COMMAND
00023  009C    5C04                    BR(NA)   *+2
00024                          *
00025                          *       THIS ROUTINE COMPUTES THE MEMORY SIZE
00026                          *
00027  009E    8720            START   LDKL     A7,/5555              TEST PATTERN
       00A0    5555
00028  00A2    81A0                    LDKL     A9,/FC00              LOAD MAXIMUM MEMORY SIZE
       00A4    FC00
00029  00A6    8727            COMP    STR      A7,A9                 TRY TO STORE PATTERN
00030  00A8    EF26                    CWR*     A7,A9                 CHECK IF LOCATION EXISTS
00031  00AA    5006                    BE(0)    PARAMS                YES? GO TO PROGRAM
00032  00AC    99A0                    SUKL     A9,/2000              NO? SUBTRACT 4K
       00AE    2000
00033  00B0    5F0C                    BR       COMP                  AND TRY AGAIN
00034                          *
00035                          *       THIS ROUTINE LOADS THE START AND END ADDRESSES
00036                          *
00037  00B2    81C1            PARAMS  ST       A9,MEMADD             STORE END ADDRESS
       00B4    00C0  R
00038  00B6    81A0                    LDKL     A9,/180               LOAD START ADDRESS FOR WRITE/READ ROUTIN
       00B8    0180
00039  00BA    81C1                    ST       A9,MEMADD+2           AND STORE IN MEMADD BUFFER
       00BC    00C2  R
00040  00BE    570F                    BE(7)    LOAD                  GO TO STRAT OF WRITE/READ ROUTINE
00041                          *
00042                          *
00043  00C0    0000            MEMADD  DATA     0                     RESERVE TO LOCATIONS
```

```
00044  00C2  0000          DATA    0                       FOR MEMORY ADDRESSES
00045                *
00046  00C4  5555    PATEN DATA    /5555                   TEST
00047  00C6  AAAA          DATA    /AAAA                   PATTERNS
00048  00C8  FFFF          DATA    /FFFF                   FOR WRITE
00049  00CA  0000          DATA    /0000                   READ
00050                *
00051  00CC  0019    TIMER DATA    25                      SOFTWARE TIMER COUNT
00052                *
00053                *       THESE INSTRUCTIONS PREPARE THE PROGRAM PARAMETERS
00054                *
00055  00CE  0700    LOAD  LDK     A7,0                    ZEROISE REPEAT COUNTER
00056  00D0  0200    LOAD1 LDK     A2,0                    ZEROISE PATTERN COUNTER
00057  00D2  85CA    LOPAT LD      A13,PATEN,A2            LOAD TEST PATTERN
       00D4  00C4  R
00058                *
00059                *       THIS IS THE WRITE MEMORY ROUTINE
00060                *
00061  00D6  81C0    TEST  LD      A9,MEMADD+2             LOAD START ADDRESS
       00D8  00C2  R
00062  00DA  85A7    WRITE STR     A13,A9                  WRITE TEST PATTERN IN MEMORY
00063  00DC  91A0          ADKL    A9,2                    UPDATE MEMORY ADDRESS
       00DE  0002
00064  00E0  E9C0          CW      A9,MEMADD               CHECK FOR LAST ADDRESS LOADED
       00E2  00C0  R
00065  00E4  5C0C          BB(NE)  WRITE                   REPEAT ROUTINE IF NOT LAST ADDRESS
00066                *
00067                *       THIS IS THE READ ROUTINE
00068                *
00069  00E6  81C0          LD      A9,MEMADD+2             LOAD START ADDRESS
       00E8  00C2  R
00070  00EA  80A6    READ  LDR*    A8,A9                   READ PATTERN FROM MEMORY
00071  00EC  E8CA          CW      A8,PATEN,A2             COMPARE MEM CONTENTS WITH PATTERN
       00EE  00C4  R
00072  00F0  5002          BF(F)   CONT                    IF EQUAL CONTINUE READ ROUTINE
00073  00F2  207F          HLT                             IF NOT EQUAL STOP ROUTINE
00074  00F4  91A0    CONT  ADKL    A9,2                    UPDATE MEMORY ADDRESS
       00F6  0002
00075  00F8  E9C0          CW      A9,MEMADD               CHECK FOR LAST ADDRESS READ
       00FA  00C0  R
00076  00FC  5C14          BB(NE)  READ                    CONTINUE IF NOT LAST WORD
00077  00FE  1202          ADK     A2,2                    UPDATE PATTERN COUNTER
00078  0100  EA20          CWK     A2,8                    CHECK FOR LAST PATTERN
       0102  0008
00079  0104  5C34          BB(NE)  LOPAT                   IF NOT LOAD NEW PATTERN
00080  0106  1701          ADK     A7,1                    UPDATE SOFTWARE TIMER COUNT
00081  0108  EF40          CW      A7,TIMER                CHECK FOR LAST TEST
       010A  00CC  R
00082  010C  5C3E          BB(NE)  LOAD1                   IF NO CONTINUE WRITE/READ ROUTINES
00083                *
00084                *       THIS ROUTINE COMPUTES THE NEW START AND END ADDRESSES
00085                *
00086  010E  84C0          LD      A12,MEMADD              SAVE END ADDRESS
       0110  00C0  R
00087  0112  8140          LD      A1,MEMADD               LOAD END ADDRESS FOR COMPUTATION
       0114  00C0  R
00088  0116  024E          LDK     A2,/4E                  LOAD PROGRAM SIZE
```

```
00089  0118  9908            SUR     A1,A2           GIVES NEW START ADDRESS
00090  011A  8580            LDR     A13,A1          SAVE NEW START ADDRESS
00091  011C  8141            ST      A1,NEWADD       STORE NEW LOAD ADDRESS
       011E  0122  R
00092  0120  5702            RF(7)   *+4             BYPASS NEWADD
00093  0122  0000    NEWADD  DATA    0               RESERVE ONE LOCATION FOR NEW ADDRESS
00094  0124  190A            SUK     A1,10           GET NEW END ADDRESS
00095  0126  8141            ST      A1,MEMADD       AND STORE IN MEMADD
       0128  00C0  R
00096                        *
00097                        *       THIS LOADS THE NEW START ADDRESS IN MEMADD+2
00098                        *
00099  012A  0200            LDK     A2,0
00100  012C  8241            ST      A2,MEMADD+2
       012E  00C2  R
00101                        *
00102                        *       THIS LOADS THE NEW END ADDRESS IN THE WRITE/READ ROUTINES
00103                        *
00104  0130  110A            ADK     A1,10
00105  0132  8141            ST      A1,WRITE+8
       0134  00E2  R
00106  0136  8141            ST      A1,CONT+6
       0138  00FA  R
00107                        *
00108                        *       THIS LOADS THE NEW ADDRESS OF MEMADD+2
00109                        *
00110  013A  1102            ADK     A1,2
00111  013C  8141            ST      A1,TEST+2
       013E  0008  R
00112  0140  8141            ST      A1,READ-2
       0142  00F8  R
00113                        *
00114                        *THIS LOADS THE NEW ADDRESS OF PATEN
00115                        *
00116  0144  1102            ADK     A1,2
00117  0146  8141            ST      A1,LDPAT+2
       0148  0004  R
00118  014A  8141            ST      A1,READ+4
       014C  00FE  R
00119                        *
00120                        *       THIS LOADS THE NEW ADDRESS OF TIMER
00121                        *
00122  014E  110A            ADK     A1,8
00123  0150  8141            ST      A1,CONT+22
       0152  010A  R
00124                        *
00125                        *       THIS ROUTINE DOES THE TRANSFER
00126                        *
00127  0154  83A0            LDKI    A11,40
       0156  0028
00128  0158  8D40    REP     MI      10,MEMADD
       015A  00C0  R
00129  015C  86A0            LDKL    A14,20
       015E  0014
00130  0160  96C1            ADS     A14,REP+2
       0162  015A  R
00131  0164  8D37            MSR     10,A13
```

```
00132  0166  95A0              ADKI    A13,20
       0168  0014
00133  016A  9BA0              SUKI    A11,10
       016C  000A
00134  016E  5C1A              BR(NZ)  REP
00135  0170  83A0              LDKL    A11,/207F
       0172  207F
00136  0174  9DA0              SUKI    A13,2
       0176  0002
00137  0178  A3B7              STR     A11,A13
00138  017A  9DA0              SUKL    A13,64
       017C  0040
00139  017E  BF16              ABR     A13
00140                          END     START
```

SYMBOL TABLE

BEGIN	0000	R	COMP	00A6	R	CONT	00F4	R	TNR	008C	R
LDPAT	00D2	R	LOAD	00CE	R	LOAD1	00D0	R	MEMADD	00C0	R
NEWADD	0122	R	PARAMS	00B2	R	PATEN	00C4	R	READ	00FA	R
REP	0158	R	START	009E	R	TEST	00D6	R	TIMER	00CC	R
WRITE	00DA	R									