*TI 99/4A SOFTWARE TEST SYSTEM *

This test system consists of two SSSD diskettes and this documentation. All were released by Texas Instruments to 99/4A User Groups on their records in Mid November, 1986. Each club was sent the disks and a printout of the documentation.

These are not to be sold except for a minimal charge to cover the disks and cost of copying. The Documentation was keying in to TI WRITER and transfered to disk by the Central Westchester 99'ers Club, Westchester County NY as a service to all 99/4A clubs.

One diskette requires the Mini Memory and makes entensive use of the EASYBUG feature of that module. Since EASYBUG is isolated from the console and peripheral memory, it can perform tests of the whole memory map even if there is a defect in VDP or expansion memory.
The other disk requires Extended Basic.

SYSTEM REQUIREMENTS:

TI 99/4A console, Mini Memory Command Module, Disk controller and Disk Drive, TI Extended Basic Command Module and Memory Expansion.

Table of Contents

* Loading Instructions for Mini Memory *

1. INSERT MIMI MEMORY COMMAND MODULE
2. TURN UNIT ON (TITLE SCREEN DISPLAYED)
3. PRESS ANY KEY (SELECTION LIST DISPLAYED)
4. SELECT #3 FOR MINI MEMORY (SELECTION LIST #2 DISPLAYED)
5. SELECT #3 FOR RE-INITIALIZE (PROMPT APPEARS AT TOP OF SCREEN)
6. PRESS PROCEED {FCTN 6} (SELECTION LIST #2 IS DISPLAYED)
7. SELECT #1 FOR LOAD AND RUN ("FILE NAME" IS DISPLAYED)
8. TYPE "DSK1.FILE NAME" PRESS ENTER
9. AFTER LOADING IS COMPLETE "FILE NAME" IS DISPLAYED AGAIN
1Ø. PRESS ENTER ("PROGRAM NAME" IS DISPLAYED)
11. TYPE "RUN" PRESS ENTER

THE PROGRAM SHOULD BE UP AND RUNNING, DISPLAYING A SELECTION LIST FROM ONE OF THE TEST PROGRAMS PROVIDED AND WAITING FOR A SELECTION TO BE MADE.
WHEN TESTING IS COMPLETE TURN UNIT OFF BEFORE REMOVING COMMAND MODULE.
THIS WILL ALLOW THE USE OF THE PROGRAM IN OTHER UNITES OR AT ANOTHER TIME BY FOLLOWING THE STEPS BELOW:

1. INSERT MINI MEMORY
2. TURN UNIT ON (TITLE SCREEN APPEARS)
3. PRESS ANY KEY (SELECTION LIST APPEARS)
4. SELECT #3 FOR MINI MEMORY (SELECTION LIST #2 APPEARS)
5. SELECT #2 FOR RUN ("PROGRAM NAME" APPEARS)
6. PRESS ENTER (PROGRAM BEGINS)

FILE NAME	TEST PROGRAMS
*DIAGNOSTIC	1. DIAGNOSTIC
	2. KEYBOARD TEST
	3. SPEECH TEST
	4. JOYSTICK TEST
*P/CARDS	1. PASCAL CARD TEST
	2. MEMORY EXPANSION TEST
	3. RS232 INTERFACE TEST
	4. CASSETTE TEST
	5. BIT-MAP MODE TEST

THE FOLLOWING TEST PROGRAMS REQUIRE EXTENDED BASIC COMMAND MODULE AND MEMORY EXPANSION FOR EXECUTION:

1. IMPACT SERIAL PRINTER TEST
2. SPEECH TEST
3. THERMAL PRINTER TEST
4. IMPACT PARALLEL PRINTER TEST

*BELOW IS EXECUTING INSTRUCTION FOR THE FOUR PRECEEDING PROGRAMS:

1. INSERT EXTENDED BASIC COMMAND MODULE
   (TITLE SCREEN DISPLAYED)
2. PRESS ANY KEY
   (SELECTION LIST DISPLAYED)
3. SELECT #2 FOR EXTENDED BASIC
   (*AUTO LOADING PROCESS BEGINS AND THE SELECTION LIST
   IS DISPLAYED)

* DIAGNOSTIC *

FILE NAME: DIAGNOSTIC PROGRAM NAME: RUN

THE DIAGNOSTIC TEST CONSISTS OF SEVEN INDIVIDUAL PROGRAMS LINKED TOGETHER TO FORM A COMPLETE CONSOLE TEST EXCLUDING KEYBOARD AND INPUT-OUTPUT PORT.

**** 1. COLOR TEST

THIS TEST PLACES ALL 16 COLORS ON THE SCREEN FOR VISUAL INSPECTION AND WAITS FOR KEY ENTRY FROM THE OPERATOR BEFORE PROCEEDING TO THE NEXT TEST.

**** 2. SOUND TEST

THIS TEST REQUIRES A MID RANGE VOLUME SETTING. IT CONSISTS OF 5 DISTINCT SOUNDS WHICH EXERCISE ALL 3 SOUND GENERATORS AND NOISE GENERATOR.

**** 3. SPRITE-COINCIDENCE TEST:

THIS TEST PLACES 11 SPRITES ON THE SCREEN AND MOVES 2 OF THEM DOWN AND ACROSS THE SCREEN FOR VISUAL INSPECTION WHICH CHECKING FOR BOTH FIFTH SPRITE AND COINCIDENCE.

**** 4. CHARACTER & TEXT MODE

THIS TEST DISPLAYS IN THE TEXT MODE, 4 ROWS OF 1ØØ CHARACTER FOR APPROX-IMATELY THREE SECONDS FOR VISUAL INSPECTION.

**** 5. VDP TEST (DYNAMIC RAMS)

**** 6. ROM TEST

**** 7. GROM TEST

**** 8. 681Øs (STATIC RAMS)

THE PREVIOUS LISTED MEMORY DEVICES ALL FOLLOW THE SAME TESTING FORMAT. IF THE DEVICE TESTS GOOD, "GOOD __?__" IS DISPLAYED AND THE PROGRAM PRO-CEEDS TO THE NEXT TEST. IF A DEVICE IS DEFECTIVE, "BAD __?__" IS DISPLAYED AND THE PROGRAM STOPS AT THAT POINT. AFTER THE DIAGNOSTIC TEST IS COMPLETE, PRESS ENTER TO RETURN TO THE SELECTION LIST.

THE FOLLOWING INFORMATION CONCERNING THE KEYBOARD, SPEECH AND JOYSTICK TEST COMPLETES THE SELECTION LIST ACCOMPANYING THE DIAGNOSTIC TEST.

* FILE NAME: DIAGNOSTIC * PROGRAM NAME: RUN

2. KEYBOARD TEST:

A. FOLLOW PROMPTS (PRESS SPACE BAR OR !, 1, @, A, x, ETC.) DISPLAYED. WHEN SMALL CHARACTER "x" IS INDICATED, THE ALPHA LOCK MUST BE RELEASED. ENTER "x" AND RELOCK ALPHA LOCK. THE TEST IN THEN CONTINUED.

B. IF A MULTIPLE ENGRY IS DETECTED, "MULTIPLE ENTRY TEST "_" EKY" IS DISPLAYED. THE ASCII CHARACTER IS PLACED IN THE QUOTATIONS AND A ROW IS RESERVED BELOW PROMPT FOR TESTING. AFTER THE KEY IS THOROUGHTLY TESTED, PRESS ENTER AND THE TEST RESUMES AT THE POINT OF INTERUPTION.

C. IF A KEY DOESN'T ENTER OR IF A WRONG ENTRY IS DETECTED, THE FOLLOWING PROMPT IS DISPLAYED "WRONG ENTRY TEST"_" KEY" WHERE THE CORRECT ENTRY IS DISPLAYED IN QUOTATIONS AND A ROW IS RESERVED BELOW FOR TESTING. PRESS ENTER TO RESUME THE TEST.

3. SPEECH TEST:

THIS TEST RQUIRES A MID-RANGE VOLUME SETTING. IT SAYS "READ TO START. DID YOU UNDERSTAND ME?" AS THE WORDS ARE DISPLAYED. THIS TEST WAS WRITTEN TO TEST INPUT-OUTPUT CIRCUITRY OF THE CONSOLE.

4. JOYSTICK TEST:

THIS TEST DISPLAYS THE NUMBERS 1 & 2 INDICATING THE JOYSTICK CONTROL NUMBER. AS THE JOYSTICK IS MOVED, THE EIGHT CONTACT POINTS ARE INDIDICATED BY AN ASTERISK DISPLAYED IN THE RELATIVE POSITION AROUND THE NUMBER. THE FIRE BUTTON OF JOYSTICK #1 CLEARS THE ASTERISKS AND THE FIRE BUTTON OF JOYSTICK #2 RETURNS TO THE SELECTION LIST.

*FILE NAME: P/CARD *PROGRAM NAME: RUN

*P-CODE CARD TEST:

THIS PROGRAM WAS WRITTEN TO INDIVIDUALLY TEST THE TWO ROMS, EIGHT GROMS AND ASSOCIATED CIRCUITRY. THE PROGRAM ALSO INDICATES WHICH DEVICE IS BEING TESTED AND STOPS IF AN ERROR IS DETECTED AND INDICATES THE AREA IN WHICH THE ERROR WAS DETECTED.

*MEMORY EXPANSION TEST:

THIS PROGRAM TEST THE 32K BYTES OF MEMORY EXPANSION. IT INDICATES THE AREA OF MEMORY BEING TESTED AND STOPE IF AN ERROR IS DETECTED AND INDICATES THE AREA IN WHICH THE ERROR WAS DETECTED.

*RS232 INTERFACE TEST:

THIS PROGRAM TESTS THE ROM AND BOTH THE SERIAL AND PARALLEL PORTS. THIS TEST REQUIRES A SPECIAL SERIAL TO PARALLEL CONNECTOR. IF AN ERROR IS DECTECTED, THE PROGRAM STOPS AND A DIAGNOSTIC ERROR MESSAGE IS DISPLAYED.

{THE THREE PRECEDING PROGRAMS WERE WRITTEN IN A SONSTANT LOOP. THIS MEANS {THAT THE TESTING CONTINUES UNTIL AN ERROR IS DETECTED, THE UNIT IS TURNED {OFF, OR A QUIT IS PERFORMED.

*CASSETTE TEST:

THIS PROGRAM TESTS THE CONSOLE CASSETTE CIRCUITRY BY WRITING 2K BYTES FROM THE CONSOLE TO CASSETTTE AND THEN BACK. A BYTE FOR BYTE COMPAR-ISON IS PERFORMED AND THE RESULT IS INDICATED. TO EXECUTE, SELECT #4 AND FOLLOW PROMPTS.

*BIT-MAP MODE TEST:

THIS PROGRAM SWITCHES FROM GRAPHIC TO BITMAP MODE AND BACK AT APPROXIMATELY 2 SECOND INTERVALS. "DEFECTIVE BIT MAP MODE" WILL BE DISPLAYED AT THE TOP AND BOTTOM OF THE SCREEN IF DEFECTIVE. THIS MODE IS USED FOR MORE DESCRIP-TIVE GRAPHICS SUCH AS PARSEC.

THE FOLLOWING TEST PROGRAMS REQUIRE EXTENDED BASIC COMMAND MODULE FOR EXECUTION. (SEE PAGE 2 FOR MORE INFORMATION.)

*SERIAL IMPACT PRINTER TEST:

THIS PROGRAM PRINTS (USING THE SERIAL PORT) THE CHARACTERS, SIZING, AND TYPE DENSITIES AVAILABLE THROUGH SOFTWARE IN GRAPHIC AND TEXT MODES. THE BUZZER AND CARRIAGE RETURN ARE ALSO TESTED. THE PRINTER MUST BE SET AT 3ØØ BPS FOR TESTING.

*SPEECH TEST:

THIS PROGRAM RECEITES THE ENTIRE RESIDENT VOCABULARY OF THE SPEECH SYNTHESIZER, AND IS USEFUL IN DETECTING INTERMITTENT OR THERMAL RELATED SPEECH PROBLEMS.

*THERMAL PRINTER TEST:

THIS PROGRAM IS WRITTEN TO ACTIVATE ALL THE HEATING ELEMENTS USED BY THE THERMAL PRINTER. THIS IS DONE BY PRINTING ROWS OF CHARACTERS AND DOTS FOR VISUAL INSPECTION.

*IMPACT PARALLEL PRINTER TEST:

THIS PROGRAM PRINTS (USING THE PARALLEL PORT) THE CHARACTERS, SIZING, AND TYPE DENSITIES AVAILABLE THROUGH SOFTWARE IN BOTH GRAPHIC AND TEXT MODES. THE BUZZER AND CARRIAGE RETURN ARE ALSO TESTED.

TROUBLE-SHOOTING TIPS

TEST PROGRAM	COMPLAINT	DEFECTIVE DEVICE
*DIAGNOSTIC	DISTORTED VIDEO	CONSOLE/MONITOR/MODULATOR
"	NO COLOR	" " "
"	NO SOUND	" " "
"	DISTORTED SPRITE MOTION	"
"	ERRATIC PRINT	"
"	WON'T RUN KNOWN GOOD PROG	"
*KEYBOARD TEST	WON'T ENTER	KEYBOARD
"	INTERMITTANT ENTRY	"
"	MULTIPLE ENTRY	"
"	WON'T PRINT LARGE CHARACTERS	"
*SPEECHTEST	NO SPEECH	SYNTHESIZER/CONSOLE
"	GARBLED SPEECH	" "
"	STATIC IN SPEECH	" "
*JOYSTICK TEST	NO RESPONSE	JOYSTICKS/CONSOLE
"	WON'T MOVE UP	" " (release
"	BOTH JOYSTICKS MOVE	Alpha lock)
"	SIMULTANEOUSLY	CONSOLE
"	WON'T MOVE DIAGONALLY	JOYSTICKS

TROUBLE-SHOOTING TIPS

TEST PROGRAMS	COMPLAINT	DEFECTIVE DEVICE
*PACSCAL CARD TEST	LIGHT COMES ON AND COMPUTER LOCKS UP	PASCAL CARD
"	WON'T RUN KNOWN GOOD PASCAL PROGRAM	"
"	NO LIGHT COMPUTER POWERS UP AND DISPLAYS TITLE SCREEN.	"
*MEMORY EXP. TEST	COMPUTER DISPLAYS "NO MEMORY EXPANSION"	MEMORY EXPANSION CARD
"	WON'T EXECUTE PROGRAM FROM MEMORY EXPANSION.	"
"	MEMORY EXP. LIGHT COMES ON & COMPUTER LOCKS UP.	"
*RS232 INTERFACE	LIGHT COMES ON AND COMPUTER LOCKS UP.	RS232 INTERFACE
"	WON'T PRINT	"
*CASSETTE TEST	WON'T RECORD	CASSETTE/CONSOLE/CABLE
"	WON'T READ CASSETTE	" " "
"	READS O.K. BUT WON'T RUN PROG	" "

*PASCAL CARD REQUIRES MEMORY EXPANSION FOR NORMAL OPERATION

*THE LIGHT MENTIONED IN THE COMPLAINTS ABOVE IS REFERRING TO THE CARD INDICATOR LIGHT

TI 99/4A RS232 CARD TEST INTERPRETATION

The assembly language test for the RS232 card is designed to exercise all of the functions available on the device. If during the course of the test cycle an error is detected, the test will halt and one of several error messages will be printed to the monitor screen. This document is intended to aid the repair technician in determining the cause of the failure. The error messages are presented here in the order in which they are encountered in the test.

1. ILLEGALLY DRIVEN BUS - The software turns off all peripheral CRU enable bits and reads at 4ØØØ; if the result is not zero, this error is displayed. The fault may be located by using EASYBUG debugger to read the bus.

2. BAD ROM - A CRC check is performed on the data in the ROM. Failure to obtain the correct checksum results in this error. Do not replace the rom until it is verified that it is being accessed properly and there are do databus faults.

3. CRU BIT BAD - Performs a walking data read/write on the CRU registers at 13Ø2 through 13ØE. The failed bit can be determined from the last digit of the error address (1-7). These bits can be manipulated with EASYBUG for troublshooting.

4. PARALLEL PORT EXPEDTED/ACTUAL - Test first turns on the ourput control of te parallel port ourput buffer, ther performs a walking data read/write test. After the data test, the output control is turned off and the paralle port is read. Since the test fixture has a pull down resister for each data line, any bits read as a one are reported as OPEN BUS LINES.

5. SERIAL PORT DTR ERRORS - Since the Data Terminal Ready inputs must work in order to transmit data, the clear to send outputs (flag 1 & 2) have been connected to DTRØ and DRT1 in the test fixture. The flags are set high and low while the CTS/DTR inputs are sampled at both UARTS. An error is reported as which UART was read, along with the input error condition. It should be remembered that flags 1 & 2 were tested during the CRU test. However, the I/O circuitry may be verified by setting the flags in EASYBUG (flag 1 @ 13ØA, 2 @ 13ØC). This is the first test to require a response from the 99Ø2s. The inputs are read as CRU bits at 1376 & 1378 for DTRØ and 13B6 & 13B8 for DTR1.

6. TOO MUCH TIME TRANSMIT BUFFER X - between the DTR input test and this error, several things have occured: the UARTs have been initialized to 8 data bits, 1 satop bit, 3ØØ baud, no parity. Once this is doen, the selected UART (Ø first) is turned on by writing a one to the RTS bit(UART Ø @ 136Ø, UART1 @ 13AØ). This should cause the carrier detect led for the selected UART to scitch. The program now reads the Transmit Buffer Register Empty status bit (CRU @ 136C & 13AC). If this bit is not a one bythe specified time, "TOO MUCH TIME TRANSMIT BUFFER X" is printed. Once the stransmit buffer is empty, the transmit buffer is loaded with a data word. As soon as it is loaded, transmission should begin.

7. TOO MUCH TIME RECEIVE BUFFER X - Now the program begins testing the Receive Buffer Loaded bit in the receiving UART. This bit is set when a valid input word and stop bit is detected. If this does not occur wihtin a specified time limit, the error "TOO MUCH TIME RECEIVE BUFFER X" is printed.

8. SERIAL DATA ERROR - Once the 99Ø2 detects a valid input word made up ofa start bit, 8 data bits, and a stop bit, and the RBL bit is set, the received data word is compared to the word that was sent to the transmittin 99Ø2. If the words are not equal, the error "SERIAL DATA ERROR X TO X" is printed, depending on the direction of transmission. After good data is detected, RTS is turned off, causing the carrier detect LED to switch to its reset condition.

Serial prt testing begins with UART Ø transmitting and UART 1 receiving. When a successful T/R cycle is complete, the same data is transmitted in th opposite direction. The data word is shifted left one bit, and the entire process is repeated eight times.

Occasionally and RS232 card will not allow the system to complete power up initialization, resulting in a "blue screen lockup". This condition is usualy traceable to a problem on the ILA output to External Interrupt.

The next page is a memory map to aid in using EASYBUG debugger to troubleshoot the card. Happy hunting.

To obtain a photo copy of the diagrams neede to make up the special Serial and Parallel Port connectors to be used in testing the RS232, send a Self Addresses, stamped legal size envelope to Art Byers, 1261 Williams Drive, Shrub Oak, NY 1Ø588 and request: "Test System Diagrams". In addition to the above, you will be sent a photocopy of the RS232 block diagram and the RS232 CARD UART MEMORY MAP. It just was not feasable to reproduce the diagrams on TI WRITER.

RS232 CARD MEMORY MAP

4ØØØ - 4FFE	DSR ROM
5ØØØ - 5FFE	PARALLEL I/O

RS232 CARD CRU OUTPUT BIT DEFINITION

ADDR	BIT	DEFINITION
13ØØ	Ø	DSR ROM page enable, 1= enabled
13Ø2	1	Parallel Port mode set, 1 = input mode
13Ø4	2	Parallel Port Strobe bit
13Ø6	3	Spare Parallel Port bit
13Ø8	4	Flag Ø
13ØA	5	Clear to Send, RS232 Port Ø, Ø = active
13ØC	6	Clear to Send, RS232 Poprt 1, Ø= active
13ØE	7	Indicator LED control, 1= LED on

RS232 CARD CRU INPUT BIT DEFINITION

ADDR	BIT	DEFINITION
13ØØ	Ø	Spare
13Ø2	1	Parallel Port configuration sense
13Ø4	2	Parallel Port Acknowledge sense bit
13Ø6	3	Spare Parallel Port Sense bit
13Ø8	4	Flag Ø
13ØA	5	Clear to Send, RS# Port Ø sense # # #
13ØC	6	Clear to Send, RS232 Port 1 sense
13ØE	7	LED state sense

99Ø2 UART BASE ADDRESSES

UART Ø	= 134Ø
UART 1	= 138Ø

SOFTWARE CONTROLLED TROUBLESHOOTING TECHNIQUES

One of the difficulties of troubleshooting microporcessor based systems like the 99/4A, in which system control is handled by dedicated ROM, lies in the fact that the flow of the program (and therefore address, data, and control lines) cannot be modified by the technician. Once power has been applied, the system is under the control of firmware masked in ROM. Practicaly speaking, this means that if a circuit node for an address line is probed with a scope, the result will be a very long complex pulse train as the microprocessor responds to the instructions of firmware. The technician can see the high/low signals from the scope but cannot determine if the entire address is correct or if the timing is proper. While the processor is under control of this firmeware, it is not possible to stop the system to probe multiple points in order to determine proper address, data, or control.

The Mini Memory module has provided a way of escape from this delemma. The EASYBUG DEBUGGER contained in the Mini Memory asslows the user to:

1) Address any device in the 99ØØ address field.
2) inspect and, optionally, modify the contents of RAM.
3) Display the contents of GROM and ROM.
4) Execute assembly language programs from EASYBUG.
5) Directly access devices which are controlled by the TMS 99ØØ Microprocessor's Serial I/O Port, the communications register unit (CRU).

Although these functions are designed to allow debugging of assembly level software, EASYBUG, as a side effect, allows the user to manipulate the hardware in ways not previously possible without a great deal of trouble.

Quite a lot of manipulation is possible by using EASYBUG subroutines as they are available by simple menu selection. More detailed checking of data, address, and control is possible with the use of very short assembly language programs designed to give predictable and regularly repeating results at certain circuit nodes.

This technique requires that the user understand the operation of the 99/4A computer. It will always be necessary for the technician to understand what signal should be at a circuit node, when it should be there, and why. Obviously these techniques are useless on units that are dead, locked up or otherwise incapable of executing a program.

In summation, the EASYBUG can do several things to aid the technician:

1) Provide a means to control the CRU bus, and therefore CRU devices such as the TMS 99Ø1.
2) Allow activation of signals that are normally inactive to verify operation.
3) Provide a means to escape control of system firmware, allowing the technician to test functions in detail.

The approach to describing use of this technique will be in the form of a series of "EASYBUG NOTES". These will be labeled as to use and procedure.

EASYBUG NOTE #1: CONTROLLING THE 99Ø1

Required Reference: CRU map for 99Ø1

Use EASYBUG command for CRU singlebit I/O. Referring to the 99Ø1 CRU map, enter C and the address for the function to be exercised. A one or zero may then be entered to enable or disable the function.

EXAMPLE: Audio Gate

DISPLAY	ENTRIES	RESULT
?	CØØ3Ø (ENTER)	
CØØ3Ø=ØØ	1 (ENTER)	99Ø1 PIN 27=LL1
CØØ31=Ø1	(MINUS)	
CØØ3Ø=Ø1	Ø (ENTER)	99Ø1 PIN 27-LLØ
CØØ31=ØØ		

99Ø1 INPUT/OUTPUT MAP

ADDRESS	CRU BIT	PORT DESIGNATION	PIN	FUNCTION
ØØØØ	Ø	CONTROL		CONTROL
ØØØ2	1	INTERRUPT 1	17	EXTERNAL
ØØØ4	2	INTERRUPT 2	18	VDP VERTICAL SYNC.
ØØØ6	3	INTERRUPT 3	9	KEYBOARD: :/.,MN=
				JOYSTICK: FIRE
ØØØ8	4	INTERRUPT 4	8	KEYBOARD: LKJH SPACE
				JOYSTICK: LEFT
ØØØA	5	INTERRUPT 5	7	KEYBOARD: POIUY ENTER
				JOYSTICK RIGHT
ØØØC	6	INTERRUPT 6	6	KEYBOARD: Ø9876
				JOYSTICK: DOWN
ØØØE	7	INTERRUPT 7 (P15)	34	KEYBOARD: 12345
				JOYSTICK UP
ØØ1Ø	8	INTERRUPT 8 (P14)	33	KEYBOARD: ASDFG SHIFT
ØØ12	9	INTERRUPT 9 (P13)	32	KEYBOARD: QWERT
ØØ14	1Ø	INTERRUPT 1Ø (P12)	31	KEYBOARD: ZXCVB
ØØ16	11	INTERRUPT 11 (P11)	3Ø	NOT USED
ØØ18	12	INTERRUPT 12 (P12)	29	RESERVED
ØØ1A - 1E	13-15	INTERRUPT 13-15,23,27,28		NOT USED
ØØ2Ø	16	PROGRAMMABLE Ø	38	NOT USED
ØØ22	17	PROGRAMMABLE 1	37	NOT USED
ØØ24	18	PROGRAMMABLE 2	26	BIT 2 OF KEYBD SELECT(LSB)
ØØ26	19	PROGRAMMABLE 3	22	BIT 1 OF KEYBD SELECT
ØØ28	2Ø	PROGRAMMABLE 4	21	BIT Ø OF KEYBD SELECT(MSB)
ØØ2A	21	PROGRAMMABLE 5	2Ø	ALPHA LOCK KEY
ØØ2C	22	PROGRAMMABLE 6	19	CASSETTE MOTOR CONTROL 1
ØØ2E	23	PROG. 7 / INT. 15	23	CASSETTE MOTOR CONTROL 2
ØØ3Ø	24	PROG. 8 / INT. 14	27	AUDIO GATE
ØØ32	25	PROG.1Ø / INT. 12	28	MAG TAPE DATA OUT
ØØ36	27	PROG.11 / INT. 11	3Ø	MAG TAPE DATA INPUT
ØØ38 - 3E	28-32	PROG.12 -PROG 15	31-34	NOT USED

EASYBUG NOTE #2: CONTROLLING MEMORY SELECTION LOGIC

Required Reference: System memory Map

Use EASYBUG command to modify CPU memory. Referring to the system memory map, enter M and an address that is in the block decoded by memory enable in question. when ENTER is pressed, the enable line should go active(LOW) for approximately 2us. (A storage scope or a sharp eye is required here.)

EXAMPLE: ROM gate

DISPLAY	ENTRIES	RESULT
?	M7ØØØ (ENTER)	U5Ø4 PIN 12= LLØ for 2us
M7ØØ1	(ENTER)	SAME

SYSTEM MEMORY MAP

HEX ADDRESS

ØØØØ - 1FFF Console ROM space
2ØØØ - 3FFF Memory Expansion
4ØØØ - 5FFF Peripheral Expansion (precoded to I/O connector)
6ØØØ - 7FFF Game cartridge ROM/RAM (precoded to GROM connector)
8ØØØ - 9FFF Microprocessor RAM, VDP, GROM, SOUND and SPEECH select.
AØØØ - BFFF Memory Expansion
CØØØ - DFFF Memory Expansion
EØØØ - FFFF Memory Expansion

EASYBOG NOTE #3 PROGRAM CONTROL OF DATA, ADDRESS AND CONTROL

Required reference: System Memory Map

Use EASYBUG command for modify CPU memory. Starting at an unused location in Mini memory RAM, enter the following program:

```
7ØØØ Ø2
7ØØ1 ØØ
7ØØ2 XX
7ØØ3 XX
7ØØ4 Ø2
7ØØ5 Ø1
7ØØ6 XX
7ØØ7 XX
7ØØ8 C4
7ØØ9 Ø1
7ØØA 1Ø
7ØØB FE
```

Program Explanation:

```
Ø2ØØ Load Immediate register Ø.
XXXX Data to be loaded in register Ø  (Address)
Ø2Ø1 Load immediate register 1
XXXX Data to be loaded in register 1. (Data)
C4Ø1 MOV R1, *RØ (Move contents of register 1 to the address specified
     by the contents of register Ø)
1ØFE JMP -2 (do previous instruction again)
```

Start program by using EASYBUG command: EXXXX (where is the address of the first word of the program in RAM.

Application: This program will cause the computer to execute a move instruction in a two-instruction loop.

This creates a situation which has several advantages:

1) Address lines (when Valid) should have a known state determined by the data that was loaded into register Ø.
2) Data lines (when valid) should have a kjnown state determined by the data that was loaded into register 1.
3) Control line signals should become regular & predictable due to the repetitious nature of the program.

Some applications for this program follow:

1) Memory Selection Logic Example

   Test Memroy Block Enable.
      Enter address word in program as some value from 4ØØØ-5FFF inclusive. Data word does not matter. Upon program execution, MBE* should go active. This is applicable to any signal in memory selection logic.

2) Data and Address Example

   Program running as before, address @ 4ØØØ. (Do not attempt to write to read only memory, damage to components may result).
   Use both channels on scope.
   Channel 1 - WE* (99ØØ_pin 61)
   Channel 2 - DATA or address lin in question.

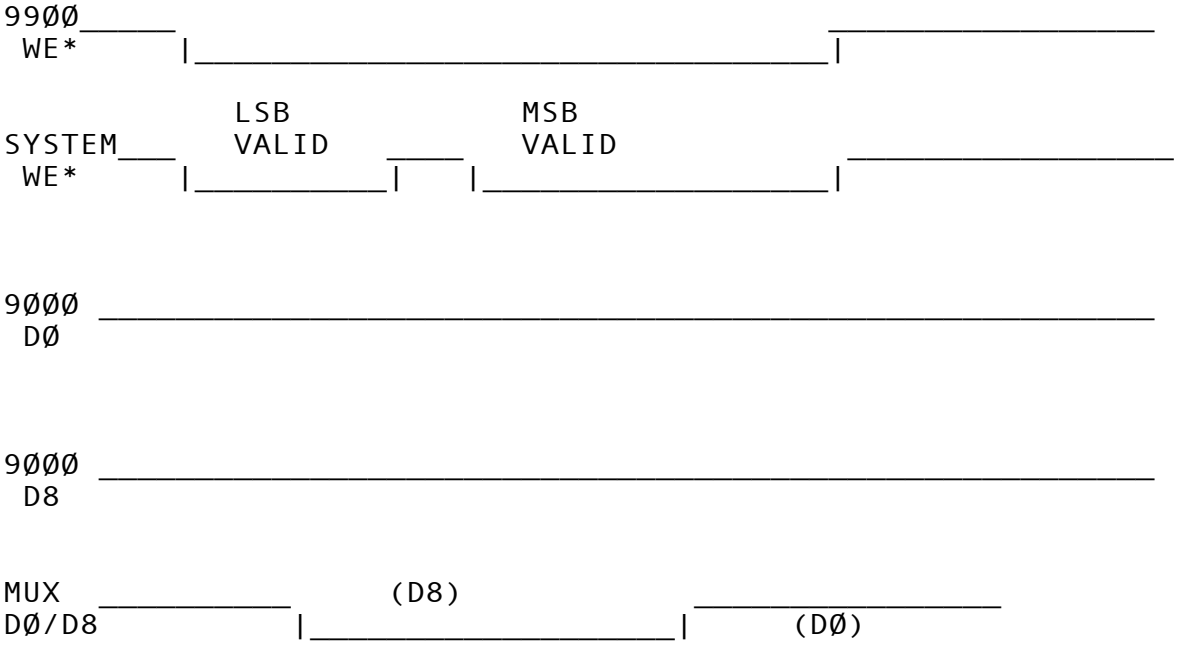
Data and address are valid when WE* is active (LOW)

Address lines and data lines should match data words for data and address in the program when WE* is true.

3) Multiplex Data Write

   Run program with address @4ØØØ. Data may be varied. Connect scope channel 1 to U6Ø6-3 (System WE*).
   Data word loaded in program can be traced through the multiplexers using channel two. System write enable* (chan 1) should display two negative going pulses. During the first of these LSB data is valid, and during the second, MSB data is valid.

EXAMPLE DATA = FFØØ



```
VERSION 1.2
ØØØ1              **TOGGLE CRU OUTPUT BIT **
ØØØ2
ØØØ3 7118              AORG >7118
ØØØ4                   DEF RUN
ØØØ5
ØØØ6 7118 Ø2ØC RUN     LI   R12,>11ØØ       *LOAD CRU BIT ADDRESS
     711A 11ØØ
ØØØ7 711C 1DØØ LOOP    SBO  Ø                 TOGGLE HIGH
ØØØ8 711E 1EØØ         SBZ  Ø                 TOGGLE LOW
ØØØ9 712Ø 1ØFD         JMP  LOOP
ØØ1Ø              ** POKE DATA FROM LINES 5 THROUGH I IN ANY FREE **
ØØ11              ** MINI MEMORY ADDRESS SPCAE IN EASYBUG MODE    **
ØØ12              ** RUN FROM EASYBUG EXECUTE MODE
ØØ13                   END
ØØØØ ERRORS
```

```
99/4 ASSEMBERL                                                PAGE ØØØ1
VERSION 1.2
ØØØ1              ** TEST CRU INPUT BIT
ØØØ2
ØØØ3 7118              AORG >7118
ØØØ4                   DEF RUN
ØØØ5
ØØØ6 7118 Ø2ØC RUN     LI  R12,>11ØØ         LOAD CRI BIT ADDRESS
     711A 11ØØ
ØØØ7 711C 1FØØ LOOP    TB  Ø                 INPUT BIT
ØØØ8 711E 1ØFE         JMP LOOP
ØØØ9              ** POKE DATA FORM LINES 5 THROUGH 7 IN ANY FREE **
ØØ1Ø              ** MINI MEMORY ADDRESS SPACE IN EASYBUG MODE.   **
ØØ11              ** RUN FROM EASYBUG EXECUTE MODE                **
ØØ12                   END
ØØØØ ERRORS
```