Atari DS800/LNBUG User's guide

The document itself begins on the next page.

Document source:

Original backup tapes owned by Dutchman2000, obtained by Atarimania.
Documentary research and PDF layout by Laurent Delsarte.
Note that these backup tapes contain A LOT of information spread out in many folders, meaning it will take time to process the important bits.

Document identification:

Original file name:	T.RENEE.00004.DOC extracted from CEO.01JUN84
Title of document:	Atari DS800/LNBUG User's guide
Author(s):	(unknown)
Original file date:	(unknown)
Type of document:	Manual
Target audience:	Professional developers
Status:	Very advanced draft
Reference (Atari):	(unknown)
Reference (Laurent Delsarte):	For any discussion, this PDF has been given the reference **BKUP-19XX-XX-XX-MANU-0002B-8** which should be quoted in any communication.
Tags:	#Atari #8bit #6502 #400 #800 #DS800 #VCS #2600 #VAX #PDP11 #Develop #Debug #Cartridge #LNBUG

Comments:

The Atari DS800 is most likely the model shown by the late Curt Vendel in the "Atari 400 and 800 Computer Comprehensive Presentation 2019 VCFEAST (Vintage Computer Festival East)" video.

This document is a bit strange because whole sections seem redundant, including multiple copies of "Introduction", "Reset switch", "Single steps", etc. Are these leftovers from a previous version? Obviously, I'm leaving it "as is": I haven't deleted anything.
Also, the figures & schema mentioned in the document are all missing.

Obvious spelling mistakes and typos have been corrected.
Additions by Laurent Delsarte are indicated by [[and]].

Given the impressive number of documents still to be explored, it's not impossible that a more recent version of this same document will resurface in the future.

Revision B
This revision B (BKUP-19XX-XX-XX-MANU-0002B-8), supersedes the original revision A, (BKUP-19XX-XX-XX-MANU-0002A-9).
Changes: See footnote 6 page 19. Thanks to Avery Lee.

This page intentionally left blank

Atari DS800/LNBUG User's guide

Table of Contents

Notice

To all persons receiving this document

Introduction

The DS800 is an ATARI® 800™ Personal Computer System, modified internally to support cartridge program development.

These modifications include: A program development monitor, called LNBUG, its associated memory, control bits, and serial ports, and a processor bus port for a logic analyzer.

LNBUG in turn features: memory commands, program run and single stepping, breakpoints and patches, single pass assembly and disassembly, communications with a host computer for program editing, assembly/compilation, and downloading.

Workstation

(Figure 1 A)

A complete programmer's workstation would include:
1. DS800
2. Monitor terminal, such as ADM-3[1]
3. Host computer hookup, via direct line or modem
4. Television set
5. Any ATARI Personal Computer System peripherals and/or controllers
6. A logic analyzer, such as an HP 1611A[2]

A minimum workstation, which a programmer might take home, could include:
1. DS800
2. Terminal with a built-in modem, such as a TI 745[3]
3. Television set
4. ATARI Personal Computer System peripherals and/or controllers
   (See Figure 1A for a sample workstation)

Televisions, controllers, and peripherals connect as on an ordinary ATARI 800.

To run programs from RAM, 48K of RAM should be installed, and no cartridges, should be plugged in. However, using LNBUG you get only 24K user space.

The DS800 draws more power than a standard 800, so a higher current unit from Jerome Industries should be used. (If a standard unit is used, it might blow its fuse.)

The monitor terminal, the host computer, and the logic analyzer connect to a panel on the left side of the DS800. (See Figure 2)

1 [[2025 foot note]] https://en.wikipedia.org/wiki/ADM-3A
2 [[2025 foot note]] https://www.hpmemoryproject.org/wb_pages/wall_b_page_12.htm
3 [[2025 foot note]] https://en.wikipedia.org/wiki/Silent_700

Serial ports

(Figure 2)

There are two 9-pin "D" connectors on the left side panel. These are electrically and mechanically compatible with the RS-232-C ports found on the ATARI 850™ Interface Module. The left one (as viewed from the front side) is for communication with a host computer. The right one is for use with the local terminal. A 4-bit baud rate selection switch is mounted in the panel for each port.

Both ports transmit data on pin 4, receive data on pin 4, and use pin 5 as a common. (See the side board theory of operation, and the side board schematic diagram for details.) Use an ATARI 850™ "terminal" type cable to connect to a local terminal; use a "modem" type cable to connect to a modem, or to a host computer.

As noted above, each serial port has an associated baud rate switch, immediately to the right on the side panel. These switches are binary encoded, to select 1 of 16 baud rates.

To set the baud rate, first choose the desired rate from the table below, then read the switch code, and set the switches. The top switch is the "least significant bit"; the bottom switch is the "most significant bit". Closing each switch, to the right, is a "1".

Baud Rate switch selection chart[4]:

Baud rate	**50**	**75**	**110**	**134.5**	**150**	**300**	**600**	**1,200**
Top switch	RT	LT	RT	LT	RT	LT	RT	LT
2nd switch	RT	RT	LT	LT	RT	RT	LT	LT
3rd switch	RT	RT	RT	RT	LT	LT	LT	LT
Bottom switch	RT	RT	RT	RT	RT	RT	RT	RT

Baud rate	**1,800**	**2,000**	**2,400**	**3,600**	**4,800**	**7,200**	**9,600**	**19,200**
Top switch	RT	LT	RT	LT	RT	LT	RT	LT
2nd switch	RT	RT	LT	LT	RT	RT	LT	LT
3rd switch	RT	RT	RT	RT	LT	LT	LT	LT
Bottom switch	LT	LT	LT	LT	LT	LT	LT	LT

4 [[In this chart, I assume that RT (LT) stand for « Toggle switch in Right (Left) position », respectively.]]

Processor bus

(Figure 2)

The address, data and control buses of the ATARI 800 are brought out to a 40-pin flat cable connector on the left side panel. The signals are arranged so that a Hewlett-Packard 1611A logic analyzer can be plugged in for tracing processor activity.

A gated clock is provided so that ANTIC DMA signal and an ungated clock are also provided, so that the ANTIC itself can be traced.

An "External Select" input is provided, for attaching external memory mapped devices. However, no DC power is provided, since there is little space left inside.

Reset switch

(Figure 2)

At the right side of the panel is a double momentary reset switch. To cold start the LNBUG monitor, rock the switch down; To cold start the resident OS (Operating System), rock the switch up. (This needs to be changed to reflect board changes)

In the latter case, the LNBUG monitor PROMs, and all LNBUG's associated memory and I/O devices are disabled, and invisible to the processor. However, the top 8K or 16K of RAM, presumably containing a prototype cartridge program, may be write protected. (See next paragraph for details.)

Optional memory-protect switches

As an option the DS800 will write protect cartridge space RAM after an OS cold start, when LNBUG is out of the system. This is called "Internal" mode in the theory of operation. This prevents the OS from clobbering a prototype cartridge program when it scans memory sizing RAM. If memory protect is enabled, the OS will treat this as ROM.

WARNING!

Care should be taken to ensure that there is valid data at 9FFC and BFFC. If either location contains "00", the OS will try to jump through presumed vectors at 9FFE or BFFE.

Invalid jump vectors will crash the system. LNBUG does not initialize these vectors.

One switch enables or disables memory protection; the second switch selects between 8K (A000-BFFF) or 16K (8000-BFFF).

A memory map illustrating the programming environment, with or without LNBUG. (Figure 3)

Single step

LNBUG, using hardware in the DS800, can be used to single step 6502 object code. It is very useful for debugging a code. However, it should be noted that the single step hardware preempts the NMI (non-maskable interrupt) processor input. This means that vertical blank, display list, or warm start interrupts from the ANTIC are inhibited. Debugging of real time programs that depend on these interrupts cannot be done with single step; logic analyzer may be used.

Atari DS800 Systems architecture

The DS800 is an ATARI® 800™ Personal Computer System internally modified to support cartridge program development. Changes made are primarily to support a particular 6502 program development monitor called "LNBUG".

LNBUG works with a separate terminal for monitor communications, and a host computer for program editing, assembly, and downloading object code.

The DS800 also provides a processor bus port for either a logic analyzer, or an external memory mapped peripheral device (such as a floppy disc drive controller).

The standard 9V AC power supply is inadequate to power the additional circuitry, so a higher current unit (Jerome Industries) is required.

(See Figure 1A for a programmer's workstation.)

LNBUG needs the following hardware:

1. 8K Bytes of PROM memory for the LNBUG code itself.
2. 1K Bytes of RAM for its internal variables, etc.
3. An RS-232-C Port for the programmer/user's terminal.
4. An RS-232-C Port for the host computer.
5. Single stepping logic.
6. Power up reset logic.

Additional DS800 Features Include:

7. A 40-pin flat cable connector for use as a bus extender or logic analyzer port. (Compatible with the HP 1611A-65)
8. Switch controlled write protection for cartridge RAM.
9. Independent 5V power regulation.

All of the hardware added is partitioned between two new circuit boards.

The production CPU board is replaced by a new DS800 CPU board, which includes data buffers, miscellaneous logic, and a 40-pin flat cable connector. The flat cable in turn folds around inside the RF shield, comes out through a slot machined in the casting, and mates to the side board.

The DS800 side board, mounted in the open space within the left side of the ATARI 800 plastic shell, contains the memory, logic, and RS-232-C ports for LNBUG. A rectangular hole is cut in the plastic shell, matching the hole on the opposite side of the ATARI 800, for RS-232-C port connectors, bus extender connector, and switches.

(See Figure 4 for a System Block Diagram.)

Future extensions

Two additional components could be designed which would add useful capabilities to the DS800.

The current design does not provide for OS changes. To do so, new "personality module" should be designed, consisting of: 16K dynamic RAM, logic to deselect that RAM for D000-D7FF for the I/O devices, and the "overhead" logic from the standard ROM personality module.

This would leave RAM from D800 to FFFF, and RAM in the C000-CFFF area reserved for future OS extensions.

NOTE

+12V and -5V power is not brought on to the personality module.
Intel 2118 type +5V only 16K dynamic RAMs must be used.

Also need 1K or 2K EPROM for a Booter for disc-based software.

The current system is dependent on a host computer for mass storage, program assemble, and such. To make a "stand alone" system, an external disc controller with 8-inch floppies should be designed as a peripheral for the bus extender port.

The device should also contain a disc bootstrap program PROM mapped at 7FFC, and a suitable 6502 DOS, including a DEC compatible file handler and a suitable editor and macro-assembler.

Introduction

The DS800 side board contains most of the hardware necessary to convert a production ATARI® 800™ into a program development system. This includes: memory, I/O devices, logic, switches, connectors and associated DC power regulation.

(Refer to the "DS800 Side Board Schematic Diagram".)

This side board is mounted between the RF shield casting and the plastic shell, in the left side of the shell, opposite the power supply board. Switched 9V AC power is tapped from the power supply board.

A processor bus extender 40-pin flat cable is brought out through a slot machined in the casting.

A rectangular panel is attached to the side board, symmetric with the right-side panel, and brought to the outside in a similar way.

The side panel includes a reset switch, a bus extender connector, two 9-pin RS-232-C connectors, and their associated baud rate switches.

Reset switch

The reset switch is a double momentary rocker switch, mounted on the left side panel. It controls the two major states of the DS800, "Internal" mode, and "External" mode.

In "External mode, a reset is vectored through location 7FFC, and all memory mapped devices on the side board are enabled;
in "Internal" mode, a reset is vectored through location FFFC, and all side board memory devices are disabled.

Please note only one mode is available for a DS800 system, either Internal or External, but not both in one system.

For example, in a system running the LNBUG monitor, the switch has the following effects: An "External" reset causes a cold start entry into LNBUG. The LNBUG PROM at 6000-7FFF will be super-imposed into the address space, the ACIA's will be mapped at D7XX; a 1K RAM will be mapped at C000-C3FF for LNBUG's internal uses, and 2-bit control latch will appear as a write only device at 7FXX.

An "Internal" reset, on the other hand, will cause a cold start into the resident OS through location FFFC, and all the above-mentioned memory mapped devices will vanish. If the optional memory protect logic is enabled, the RAM in the cartridge address space will be write protected (See below for details).

Note that the IRQ line from the ACIAS is also inhibited by the "Internal" state. This prevents unrecognizable serial port interrupts from crashing a cartridge under test.

Power up memory mapping

One of the two bits in the control latch controls address decoding immediately following an "External" reset. When set it does two things:

1. Asserts the "EXB – External Bus" line to the DS800 CPU board, which in turn shuts out memory devices within the casting.

2. Gates the A15 input to U7 low. The latter will cause any reset vector to FFFC to be steered into 7FFC instead.

Any cold start routine called through 7FFC should first write a "02" to location 7FFC, to clear that bit. It is also cleared by an "Internal" reset.

Single step logic

The second control bit controls single stepping. When set, by writing "0A" to 7FFF, it forces a non-maskable interrupt on any processor opcode fetch from memory outside an exempted region, the address decode PROM selects the exempted addresses; in a LNBUG system, this includes the LNBUG code itself plus the NMI handler within the ROM OS at E7XX.

This single step logic also inhibits NMI's from the ANTIC. Therefore, vertical blank, display list, and warm start interrupts will be ignored. "Real Time" debugging should be done with a logic analyzer, using the processor bus port.

Memory protection

Logic is included to write protect cartridge (8000-BFFF) space RAM while running in "Internal" state. Two option switches, S4A and S4B, enable this feature and select 8K or 16K of write protection. Ordinarily, these switches should be configured when the DS800 is assembled.

The purpose of this feature is to allow a prototype cartridge program to be tested in its target environment, i.e. with all side board memory devices disabled. A problem arises when the resident ROM OS sizes RAM (on cold start).

In "External" mode the OS sweeps memory until it runs into the side board PROM (LNBUG code, for example), before it finds or clobbers the RAM containing the prototype cartridge code.

In "Internal" mode, however, the OS will find the cartridge space RAM and clobber any code. The memory protect logic, if enabled, prevents this.

Note that the memory protect logic is disabled in "External" mode.

WARNING!

If cartridge memory is protected, the OS will treat it as cartridge ROM. If the OS finds a 00 in either 9FFC or in BFFC it will attempt to jump indirect through 9FFE or BFFE.

If there are not valid jump addresses in these locations, the machine will probably crash.

A table of memory protection switch configurations is shown below:

S4A	S4B	Function
1	1	No Write Protect
1	A13	No Write Protect
SBEN	1	Write Protect 8000-BFFF
SBEN	A13	Write Protect A000-BFFF

Address decoding

Address decoding for the memory mapped devices on the side board is done by a bipolar PROM (U7, and 82S115). It is inhibited by the bus extender signal EX "External Select" and enabled by the state flip flop, from the reset switch.

It is a 512X8 device: The 9 inputs are: R/W, A15 gated with "Power On Memory", and A14-A8. The outputs can then be decoded on 1-page (256-byte) boundaries. The outputs are all active low, with pull-ups. D0-D6 are used; D7 is a spare.

A table of outputs, selected devices, and example address ranges (for a system running the LNBUG 6.0 program) is shown below:

Device	Output	Address range	Function
2532	D3	6000-6FFF * Read	LNBUG code, Lower 4K
2532	D1	7000-7FFF * Read	LNBUG code, Upper 4K
4118	D0	C000-C3FF	LNBUG RAM, 1K
68B50's	D6	D700-D7FF	Serial ports
7474	D2	7F00-7FFF * Write	Power on/single step control bits
(EXB) External Bus	D5	D0+D1+D2+D3+D6	Board select
(SSEN) Single Step Enable	D4	6000-7FFF + E700-E7FF	Single step inhibit

The memory devices themselves are straight forward. The AND gate on A15*POM[5] steers the reset vectors into the side board, to 7FFC[6] instead of into the resident OS ROM at FFFC.

The 68B50 ACIAS are additionally decoded with address lines A3 and A4: The local terminal port is at D708 and D709; The host computer port is D710 and D711.

In the LNBUG version, the single step control signal (D4) inhibits single step NMI's being generated either from LNBUG itself, or from the interrupt handling code within the ATARI 800 OS.

The "External Bus" (EXB, J4-15) signal is used to superimpose memory mapped devices into the processor's address space. It is asserted by any of three conditions:

1. Any side board devices selected by the decode PROM.
2. The power up flip flop is set by an "External" reset.
3. The "External select" (EXS) signal from J3-9 is asserted.

5 [[POM = Power On Memory.]]

6 [[The original text mentions 4FFC but expert suggests it must be 7FFC instead]]

Serial communications

Two independent RS-232-C type serial ports are included on the side board.

In a LNBUG system, one is for use with a local terminal (J2), and one is for use with a host computer (J1). Both ports are implemented with standard MC68B50 "asynchronous communication interface adapters" with U2 and U3 using standard 1488 and 1489 RS-232-C interface parts (U8 and U9).

Baud rates are generated by a SMC COM8116 dual baud rate generator (U1), selected in turn by two quad dip switches (S1 and S2) mounted in the side panel.

These switches are binary encoded to select a baud rate. The top switch, marked "1", is the "least significant bit". The next three switches are in increasing order going down.
A logical "1" on any switch is selected by opening the switch, to the left;
a logical "0" is selected by closing the switch, to the right.
A table of switch settings vs. baud rates is shown below[7]:

S-4	S-3	S-2	S-1	Baud rate	S-4	S-3	S-2	S-1	Baud rate
0	0	0	0	50	1	0	0	0	1,800
0	0	0	1	75	1	0	0	1	2,000
0	0	1	0	110	1	0	1	0	2,400
0	0	1	1	134.5	1	0	1	1	3,600
0	1	0	0	150	1	1	0	0	4,800
0	1	0	1	300	1	1	0	1	7,200
0	1	1	0	600	1	1	1	0	9,600
0	1	1	1	1,200	1	1	1	1	19,200

To set the switches for 300 baud, for example, set the top switch to the right, the next to the left, the next to the right, and the bottom switch to the left. To select 9,600 baud, set all switches to the left except the top one.

The "terminal" port on the side panel, J2, is a 3-wire minimal RS-232-C connection. Input data is expected on pin 4, output is driven on pin 3, and common is on pin 5. The mark level is about -9V, and the space level is about +9V. RTS and CTS signals are strapped together; DTR, and DCD are also strapped.
Use an ATARI 850™ type "Terminal" cable to connect to the local terminal.

The "host computer" post may be connected to a simple 3-wire link, as the terminal port is, or it may be connected to a Bell 103 type modem (Or a Novation Cat, for example). In support of a modem, the RTS, CTS and DCD signals are connected to the 68B50 ACIA. DTR and DSR are strapped. The CTS and DCD inputs have 10K ohms pull-ups, as default if these are not driven externally.
Use an ATARI 850™ type "Modem" cable to connect to a modem or host computer.

7 [[The original table indicates that 4,800 baud is 1111, but logic dictates that it should be 1100.]]

Processor bus connector

The "Processor Bus" port on the side panel, J3, is intended for use as a logic analyzer port (such as an HP 1611A-A65) or as a bus extender for external memory or I/O devices (like a floppy disc drive controller). It is pinout-compatible with the HP 1611 analyzer, which in turn maps a 40-pin 6502 IC into a 40-pin flat cable connector. The modifications to a 6502 pinout are described here:

J3-4 (6502-39) is the PH2 clock gated by the DMA signal, so that a logic analyzer can ignore ANTIC DMA cycles. It is derived on the DS800 CPU board from PHO and DMA, and it is buffered by a 74LS32 on the DS800 side board. It leads the PH2 MOS output by typically 20NSEC.

J3-8 (6502-37) is that same PH2 output buffered by a 74LS244, but heavily loaded within the shield. It is not gated by DMA. ANTIC DMA cycles are identified on J3-12 (6502-35), a no connect. This is an active low signal, sampled by PHI.

The R/W signal on J3-14 (6502-36) is gated by the write protect signal, described above. It is driven by a 74LS02, but it is heavily loaded.

An "External Select" input "EXS" is provided on PIN 9 (6502-5, NC). This allows external memory mapped devices to be inserted anywhere in the processor's address space. A low level on this pin will inhibit the address decode PROM on the side board, and inhibit a 74LS245 data bus buffer on the DS800 CPU board, which disconnects the processor from any memory device within the shield. However, the address bus and the R/W signal are still driven within the system, so unintended memory writes can occur.

The address bus and data bus are aligned correctly. However, they should be buffered. On the CPU board, the processor is buffered from the internal data bus by a 74LS245; the processor/ANTIC address bus is buffered by a pair of 74LS244's (except A15). Each data bus pin has 1 LS load and 7 MOS loads on it, plus an 8-inch flat cable. The address lines have from 1 to 4 loads, 0 to 4 MOS loads plus the flat cable. R/W is driven by a 74LS02, but it is loaded by the entire system. Ground is provided on J3-1, 6, and 40 (6502-1, 38, 21). The S>O> input (6502-38) is ignored, since it is tied to the ground on the CPU board. The pin J3-15, which would correspond to the VDD/+5 pin 8 on the 6502, is a no connect.

There is very little margin on the power available within the DS800, limited both by the size of the external AC power supply and by the heat dissipation possible within the plastic: therefore, there is no power to spare for external devices.

CPU board theory of operation

The DS800 CPU board was designed with three goals in mind:
1. Standalone compatibility with production CPU board.
2. NTSC/PAL compatibility.
3. Add as little logic as possible for compatibility with external LNBUG hardware.

The logic on the CPU board is based on the PAL version, which is inherently NTSC/PAL compatible by swapping the ANTIC and GTIA. (Refer to "DS800 CPU board schematic diagram").

Four ICs were added to crowded board:

1. A **second 74LS244** address bus buffer was added, to compensate for the cable and side board capacitive loads.

2. A **74LS245** data bus buffer was added, both for loading considerations, and to allow external memory devices to be superimposed in the ATARI 800 address space.

3. **74LS00** and **74LS02** gates were added for:
   a. Control Data Bus Buffer
   b. Gate R/W for write protection
   c. Gate NMI for single stepping
   d. Derive a PH2 gated by DMA, so that a Logic Analyzer can distinguish processor cycles from ANTIC cycles.
   e. Buffer the clock into the PAL oscillator.

The LNBUG board also drives the RES and IRQ lines, but since these are open collector, no extra logic is necessary.

All 4 control signals from the side board:
1. "Write Protect" (WP),
2. "External Bus" (EXB),
3. "Single Step Enable" (SSEN),
4. "Single Step NMI" (SNMI),

are pulled inactive with resistors, so that the board will run stand alone. The 40-pin flat cable is pinned out so that an HP 1611A logic analyzer is directly compatible.

There are three signal ground lines in this connector, but no 5V power. 5V DC power for the side board is derived separately, because the current margins on the main supply are insufficient.

New features of LNBUG 6.x (June 3, 1980)

LNBUG 6 is an updated version of all previous LNBUG monitors, obviously expected to be as complete as ever needed but undoubtedly due to change:

1. COOL RESET - Resets everything except patches, breakpoints, and macro commands. Type "C" upon reset.

2. MACRO COMMAND FILES - Save up to eight command lines for later use without retyping addresses, data, etc. (J command).

3. LOADING AND WRITING - To tape now has a switch (") to allow selection of either terminal port or auxiliary port.

4. UPLOADING AND DOWNLOADING AND OTHER PDP11 COMMUNICATIONS - LNBUG 6 contains a simple method to communicate with a time share system including uploading from LNBUG and downloading to LNBUG. For previous LNBUG 3 users, D replaces S for return to LNBUG.

5. READ-MODIFY SEQUENCE - When listing a line at a time using the "/" immediate command, the beginning of the last line is opened for modification rather than the address at which the whole listing began.

6. ZERO PAGE RAM - To allow development of Stella (Video Computer System) programs, the required LNBUG zero page RAM has been moved from 00-03 to 80-96. (Stella has no RAM at the bottom of zero page). This should make little difference in user program operation as LNBUG retains a pseudo zero page for the user. See "CAUTIONS" for further details.

7. CARRIAGE RETURN DELAY - Instead of delaying a clock-dependent time for hard copy, the delay program simply outputs 10 null characters instead - (Baud rate dependent).

8. INTERRUPTS AND SINGLE STEPPING - A correction has been made to allow single stepping through an interrupt program without fouling up the interrupt status. However, due to inherent conflict between hardware and the variable complexity of interrupts structures versus a debug monitor (real time versus stop time), problems may still arise. It is recommended that a trace analyzer be used for real time operations.

9. COLLEEN RAM OS - See section at end of manual for update on interrupt vector handling. Replace Address Map "ROM" with Colleen Address Map "PROM".

10. RAM RELOCATION - RAM has been moved to C000-C3FF.

11. SINGLE STEP - With Colleen ROM OS you must use the new address PROM to prevent single stepping at E7XX.

12. RELOCATION REGISTERS - Seven more relocation registers have been added to allow flexible debugging of several relocatable program sections.

    Example: Setup - 1000, 2Y Run - 1000Y2/G Y=0-7

13. BREAK KEY now sends a break condition to the auxiliary port when in the "$" mode.

14. PROBLEMS with proceeding from a breakpoint and using multiple passes through a breakpoint or single step have been fixed.

15. RUN SUBROUTINE COMMAND - To run a subroutine and return to LNBUG upon the RTS, specify as ADR G@ or G@.

16. ENABLE INTERRUPTS during LNBUG - Set the Z command. To reset specify ZX.

17. USER CAN INITIALIZE STACK at desired location by typing "S" and then MSD of stack address immediately following reset button. LNBUG then does a cold start. A cool or warm start reinitializes stack at previous value. This feature is useful for development of programs where no memory resides at 01FF.

18. PROM programming has been moved to RAM. BYTSVN must be downloaded first before command can be used.

Other development system changes and additions (June 3, 1980)

1. New baud rate switches to be mounted on front panels.
2. Colleen processor card with Colleen chips.
3. Colleen address map and RAM requirements have been changed.
4. Changes to gradually be implemented on the TMI cards add higher quality connectors, AUX. port polarity switch, separate connector to allow the use of a terminal for LNBUG only and one for the trace memory.
5. Trace memory - See trace memory manual.
6. Updated trace memory manual and LNBUG 5 manual.

Support

For answers to questions, please contact:

Larry Nicholson,
Cyan Engineering,
Grass Valley
(916) 273-6194.