



Princess Sumaya University for Technology
Computer Engineering Department
22440: Microprocessor Lab

Experiment 4: Logic Analyzer and Parallel Port

1. Introduction

Logic analyzer is a very important instrument for microprocessor designers and maintainers. So is to a student of Microprocessor System Lab., like you. However, since an oscilloscope is more familiar to most users, why we choose a logic analyzer? Though a logic analyzer is 'tuned' to the digital world, it may yield more useful information. This property fits our goal to monitor many digital signals.

When should you use a logic analyzer?

- When you need to see lots of signals at once.
- When you need to look at signals in your system the same way your hardware does.
- When you need to trigger on a pattern of Highs and Lows on several lines and see the result.

In fact, most logic analyzers are really two analyzers in one. The first part is a *Timing Analyzer*, while the second part is the *State Analyzer*. A Timing Analyzer is analogous to an oscilloscope. It displays information in the same general form as a scope, with the horizontal axis representing time and the vertical axis has voltage amplitude. Because the waveforms in both instruments are time dependent, the displays said to be in "time domain".

2. HP 1650/51 Logic Analyzer

The HP 1650/51 Logic Analyzer front panel user interface consists of front panel keys, the Knob, and display. Using the front panel user interface, is a basis process of:

- Selecting the desired menu with menu keys.

- Placing the cursor on the desired field within the menu by rotating the Knob.
- Displaying the field options for current data by pressing the SELECT keys.
- Selecting the desired option by rotating the Knob or entering new data by using the Knob or the keypad.
- Starting and stopping data acquisition by using the RUN and STOP keys.

In order to apply the user interface quickly, you should know what the front panel control do, see **Fig. 1-1**.

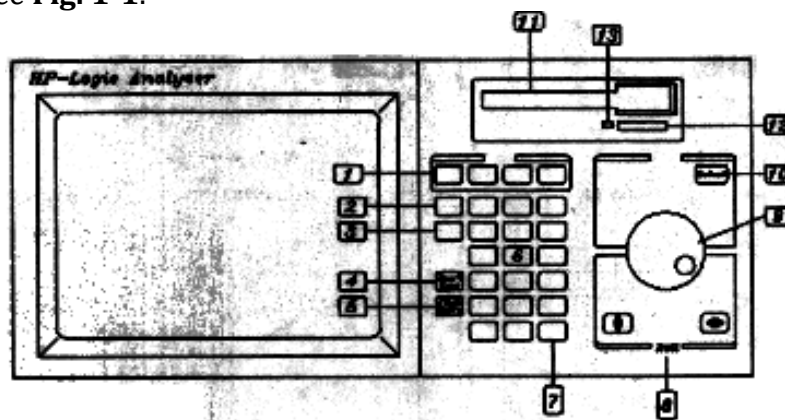


Fig. 1-1 HP 1650B/51B front panel

1. Menu Keys

The Menu Keys allow you to select the main menu in the Logic Analyzer. These keys are **FORMAT**, **TRACE**, **DISPLAY**, and **I/O**.

Format Menu Key

The **Format** Menu key allows you to access either the Timing format or the State format specification menus. You exit this menu by pressing another menu key or by returning to the System Configuration Menu.

Trace Menu Key

The **Trace** Menu key allows you to access either the Timing Trace or the State Trace specification menus. You exit it by the same methods shown above.

Display Menu Key

The **Display** Menu key allows you to access either the Timing waveforms display or the State listing display. You exit it by the same methods shown above.

I/O Menu Key

The **I/O** Menu key allows you to access the I/O menu. You can access it from any other menu and at any time. Pressing the **I/O** Menu key causes the I/O menu to pop up over the current menu on the display.

2. **RUN Key**

The **RUN** Key allows you to initialize a data acquisition and display cycle. An Analyzer is automatically forced into its Display Menu when a **RUN** is initiated.

3. **STOP Key**

The **STOP** Key allows you to stop data acquisition or printing. A single press always stops the data acquisition. The data displayed on screen depends on which acquisition mode 'Single or Repetitive' was used to acquire the data.

4. **Don't Care Key**

The **Don't Care** key allows you to enter **Don't Cares** in Binary, Octal, and Hexadecimal pattern specification fields. In alpha entry fields, this key enters a space and moves the underscore marker to the next space.

5. **Clear Entry Key**

The **Clear Entry** Key allows you to

- Return decimal values to the previous value in the decimal menu fields. Return values to don't care in menu fields with number bases other than decimal.
- Clear Alpha Entry menus.
- Move the underscore marker or cursor to its original position field.

6. **Hexadecimal Keypad**

The **HEX** keypad allows you to enter numeric values in numeric entry fields. The bases are: binary, octal, decimal, and hexadecimal.

7. **CHS Key**

The **CHS** (change sign) key allows you to change the sign (:t) of numeric variables.

8. **ROLL Keys**

When part of the data display is off screen, the Roll keys define which way the Knob will move the displayed data.

9. **KNOB**

The **Knob** has four major functions depending on what menu or pop-up menu you are in. The **Knob** allows you to:

- Move the cursor from field to field within the System Configuration and main menus.
- Roll display left or right and up or down.
- Position the cursor on options within pop-up menu. .Increment/decrement numeric values in numeric pop-up menu.

10. SELECT Key

The **Select** key allows you to open pop-up menus, choose options in them, cancel selections, and close pop-up menus.

11. Disk Drive

A 3.5 inch, double-sided, double-density drive operating system, it allows you to store and configuration as diverse assembler files besides loading the load logic analyzer.

12. Disk Eject Button

Press this button to eject a flexible disk from the disk drive.

13. Indicator Light

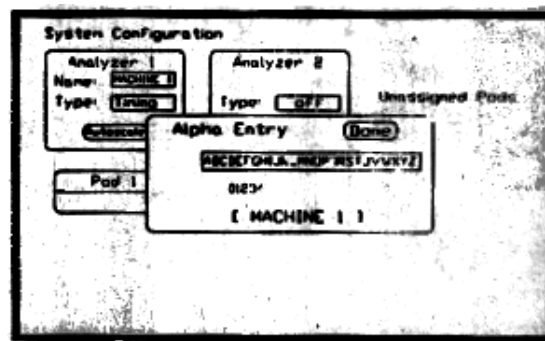
Illuminated when the Disk Drive is operating. Wait until this light is out before removing or inserting disks.

3. First Step

- Power ON your machine to load operating system. (The power switch is behind the machine).
- Use Knob to move cursor to certain item, and press **SELECT** key to do the function.

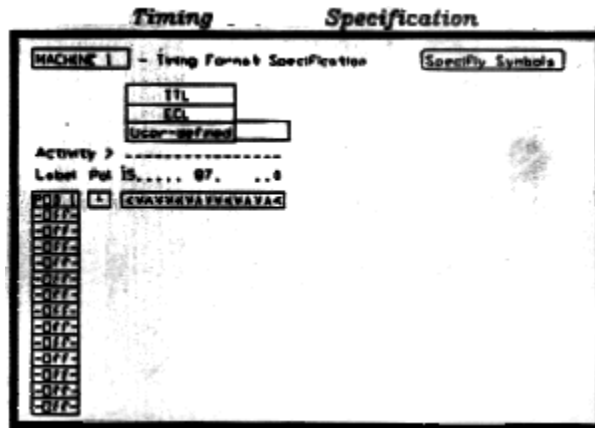
4. Enter Alpha Entry

You can customize your analyzer configuration by giving names to several items. The items that can be named are: the name of each analyzer, labels, symbols, filenames, and file descriptions. For example, you can rename an analyzer by placing the cursor on the name you wish to change in the System Configuration menu and press **SELECT**. You will see the Alpha Entry pop-up menu. You can use **Knob** to move cursor. See Fig. 1-3.



5. FORMAT Setting

The functions of Format are setting of pod thresholds and selection of channels. To set pod threshold to a specific voltage, place the cursor in the threshold portion of the pod field "TTL, ECL, or user defined" of any pod and press **SELECT**. See Fig. 1-4.



The operation of channel selection can be depicted by Fig. 1-4. 16 channels pod can be represented by 16-bit bit assignment fields. Symbol * (asterisk) indicates assigned bits. Symbol (period) indicates unassigned bits. You can use cursor and select key to assign bits in these menu. See Fig. 1-5.

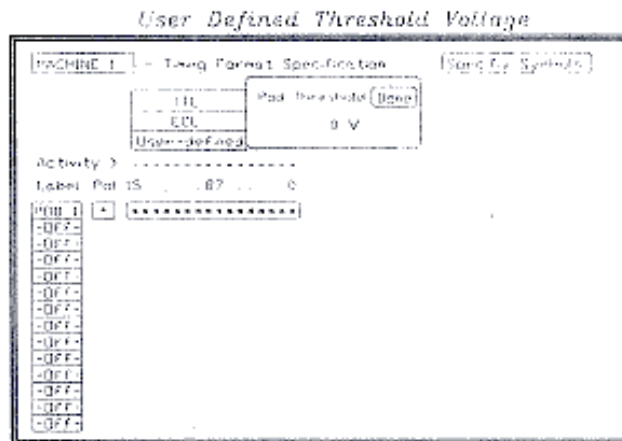


Fig. 1-5 User defined threshold voltage

TRACE Setting

The goal of Trace is to set method of data acquisition. The following is the depiction of Timing mode. See Fig. 1-6.

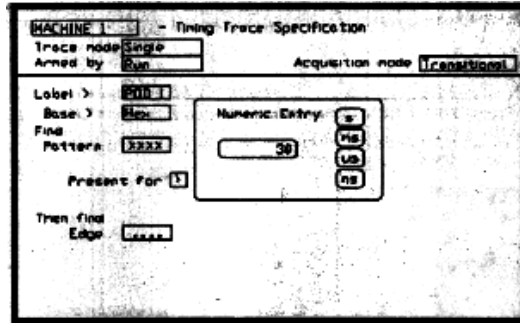


Fig. 1-6 User Trace function Specification

Trace mode

- Single: The Logic Analyzer will stop when trigger status occurs.
- Repetitive: The trigger status continuously display on the monitor.

Armed by

- RUN: Use RUN key to start system.
- BNC input: Use input signal of BNC connector to start system.

Acquisition mode

- Transitional: Sample the data at regular intervals.
- Glitch: Catch any transition between samples.

8. Steps of Timing Mode

- **Power ON** your Logic Analyzer, and let it be in system initialization.
- Press **Format** key, and you will see the display of Fig. 1-8.

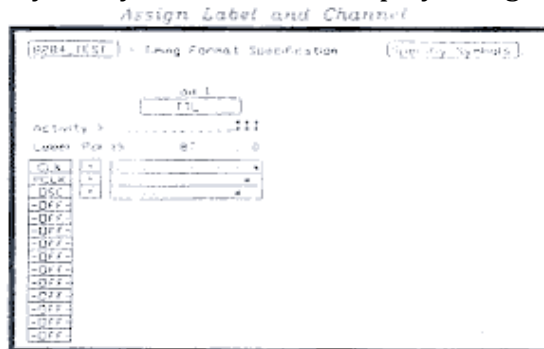


Fig. 1-8 Assign Label and channels

- Move **cursor** to items of **Label**, and press **SELECT** key.
- Move cursor to **Modify Label**, and press **SELECT** key.
- Enter an appropriate name to the observed by using **Alpha Entry pop-up Menu**, and repeat this step to all signals' names.
- Specify **channels** to the observed signals.
- Press **Trace** key, and you will see the display of Fig. 1-9.

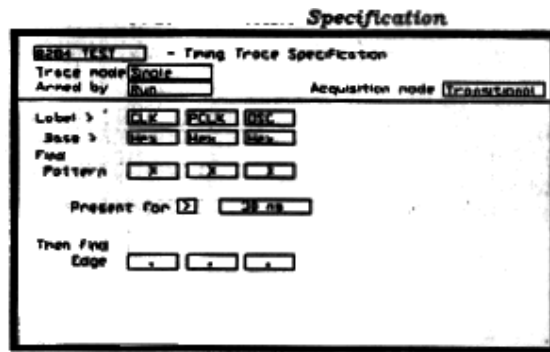


Fig. 9 Trace Functions Specification

- Put Trace mode in **Repetitive** mode.
- Put Acquisition *mode* in **Glitch** mode.
- Press **Run** key. Obtain the display, which is in Fig. 1-10.

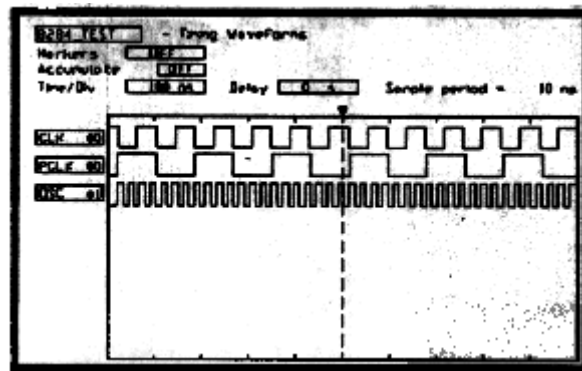


Fig. -10 Get Display

9. Timing Analyzer Measurement

Press **Display** key, move cursor to **Markers** item, and then press **SELECT** key. Choose **Time** option, and you will see the display of Fig. 1-11. The followings are the meanings of the display, and all of these can be controlled by using **Knob** and **SELECT** key.

Timing Marker

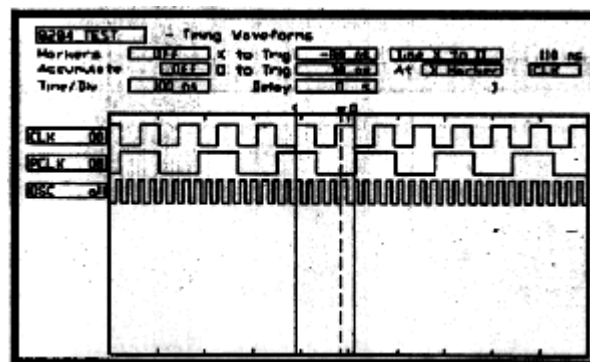


Fig. 1-11 Timing Marker

- **X to Trig.:** It is the time from marker X to Trigger (dashed line).
- **Time X to 0:** It is the time from marker X to marker 0.
- **0 to Trig.:** It is the time from marker 0 to Trigger (dashed line).
- **At:** It is a signal name indicating base time position.
- **Sec/div.:** It is a time unit.
- **Delay:** It is the time from Trigger to center line.

Parallel Port

What is a Port?

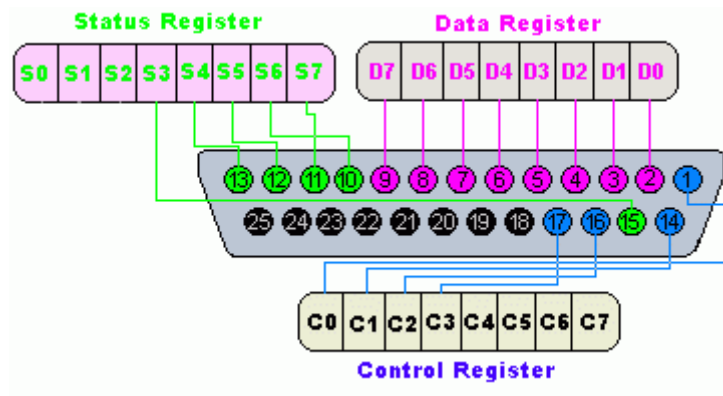
In the computer world, a *port* is a set of signal lines that the microprocessor, or CPU, uses to exchange data with other components. Typical uses for ports are communicating with printers, modems, keyboards, and displays, or just about any component or device except system memory. Most computer ports are digital, where each signal, or bit, is 0 or 1.

What is the “parallel port”?

A parallel port is a type of interface found on computers, for connecting various peripherals. In computing, a parallel port is a parallel communication physical interface. It is also known as a printer port or Centronics port. This port transfers multiple bits at once.

Hardware

The pin outs of DB25 connector is shown in the picture below



The lines in DB25 connector are divided in to three groups, they are

- 1) Data lines (data bus)
- 2) Control lines
- 3) Status lines

As the name refers, data is transferred over data lines , Control lines are used to control the peripheral and of course , the peripheral returns status signals back computer through Status lines. These lines are connected to Data, Control And Status registers internally . The details of parallel port signal lines are given below:

Pin No (DB25)	Signal name	Direction	Register - bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

Parallel port registers

The Data, Control and status lines are connected to their corresponding registers inside the computer. So by manipulating these registers in program, one can easily read or write to parallel port with programming languages like 'C' and BASIC.

The registers found in standard parallel port are,

- 1) Data register
- 2) Status register
- 3) Control register

As their names specifies, Data register is connected to Data lines, Control register is connected to control lines and Status register is connected to Status lines. (Here the word connection does not mean that there is some physical connection between data/control/status lines. The registers are virtually connected to the corresponding lines.). So whatever you write to these registers, will appear in corresponding lines as voltages, Of course, you can measure it with a multimeter or logic analyzer or any other measurement device. And whatever you give to Parallel port as voltages can be read from these registers (with some restrictions). For example, if we write '1' to Data register, the line Data0 will be

driven to +5v. Just like this, we can programmatically turn on and off any of the data lines and Control lines.

Where these registers are?

In an IBM PC, these registers are IO mapped and will have unique address. We have to find these addresses to work with parallel port. For a typical PC, the base address of LPT1 is 0x378 and of LPT2 is 0x278. The data register resides at this base address, status register at baseaddress+1 and the control register is at baseaddress+2. So once we have the base address, we can calculate the address of each registers in this manner. The table below shows the register addresses of LPT1 and LPT2.

Register	LPT1	LPT2
data register(baseaddress + 0)	0x378	0x278
status register (baseaddress + 1)	0x379	0x279
control register (baseaddress + 2)	0x37a	0x27a

Programming Concepts

As an I/O device you can simply access the parallel port using its address and read or write any data to it using the IN, OUT instructions.

For Example:

If you are to write 0xFF to the data register of the parallel port you have to write

```
Mov dx,378h  
Mov Al,0ffh  
Out dx,al
```

In the other hand, if you are to read a data from the same register, do the following:

```
Mov dx,378h  
IN al,dx
```

Assignment:

Write a program to read a number from keyboard then displays it to the parallel port data lines.

Use the logic analyzer to accomplish the estimated output.