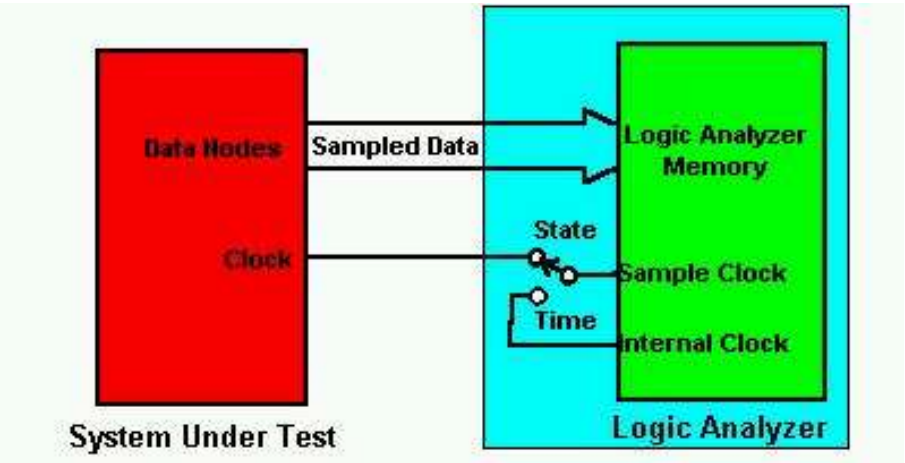
Logic Analyzers.

The logic analyzer is an instrument that provides visualization as to what is happening on the various microprocessor busses. There are two basic types of logic analyzers, the state and the timing. State analyzers are designed to monitor activity on the microprocessor address and data buses while the timing analyzer monitors activity on multiple lines such as microprocessor control buses or I/O handshake.

Hewlett Packard developed the State Analyzer and Biomation the Timing Analyzer for which their developers were co-recipients of the Electronics Magazine 1977 Achievement Award. The citation states the award was "for major instrumentation innovations that meet a vital need in the growing field of digital-systems design."

The basic diagram of a logic analyzer is shown. On a clock pulse, information on the many nodes probing the system under test (SUT) is latched into the logic analyzer memory as a "1" or "0" depending upon whether it was greater or less than the threshold voltage. The type of analyzer depends upon the clock source.



- For the state analyzer the clock is taken from the SUT, so it is also known as a synchronous analyzer. Another common name for the state analyzer is a software analyzer.
- With the timing analyzer the clock is generated internally within the analyzer. Since its clock is not synchronized to the SUT, the timing analyzer it is also known as an asynchronous analyzer.

The Logic Analyzer Trace or Display

The logic analyzer captures its data as a series of "0"s and "1"s. Current analyzers include post processing that translates these binary patterns into more meaningful displays. In the case of a state analyzer the information is disassembled back into microprocessor mnemonics and even the original high level source code, while for a timing analyzer the display or trace is a timing diagram.

Logic State Analyzer Trace or Display

The most simple state analyzer display is a binary list of 1's and 0's. This form of display

was used in the early analyzers. The first refinement was to display the trace in other formats such as octal, decimal or hexadecimal. Since the state analyzer is normally used to monitor program bus activity the displays are often mixed, with hexadecimal for the address and data bus and binary for the microprocessor control or status lines.

While hexadecimal displays may have been tolerated in the smaller projects they are not very user friendly and inverse assemblers that translated the 1's and 0's into processor specific mnemonics became standard with logic state analyzers. The example below is of portion of a trace using a 6502 microprocessor. It is the hexadecimal address and data information that is captured by the analyzer; the mnemonics etc. are generated by the software within the analyzer using the data and status information collected by the analyzer hardware.

<TBODY>

```
001   C185  20    JSR  C41E
002   C187  1E         operand
003   01FD  C1         internal
004   01FD  C1         write        <---- return address
005   01FC  87         write        <---- to stack
006   C187  C4         operand
007   C41E  ..                      Subroutine
...   ....  ..
047   C48F  60    RTS
048   C490  8D         prefetch
049   01FB  70         internal
050   01FC  87         read         <---- return address
051   01FD  C1         read         <---- from stack
052   C187  C4         internal
053   C188  ..                      Next Instruction
|<--Hardware-->|<--LSA software-->|<--- User Interpretation ---->|
```

</TBODY>

Logic State Analyzer Mnemonic Display (6502 mnemonics).

Note:

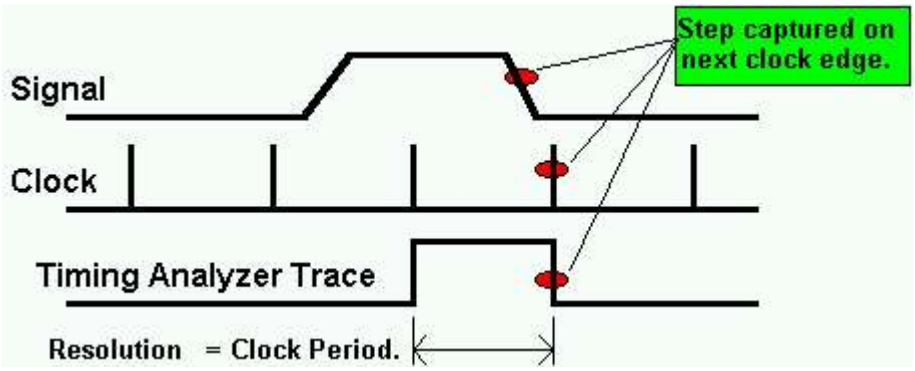
- The logic state analyzer captures **all** of the microprocessor bus activity so the trace includes the program fetch, the program execution, and any other states. In the example line 1 gives the op code fetch (the JSR) while in lines 2 and 6 the analyzer has captured the operand for the JSR. As part of the instruction execution the microprocessor places the return address onto the stack. This has been captured in lines 4 and 5 by the analyzer. Prior to placing the return address on the stack the microprocessor has performed some internal actions. The status of the external busses at that time is captured by the analyzer. (line 3)
- The analyzer trace is divided into 3.
    - The first group lists the hexadecimal information as captured by the logic analyzer hardware, this shows exactly what is executing on the microprocessor busses.
    - The second group is the logic analyzer's software interpretation of what is happening.
    - The final group is information added by the user, for example the interpretation of the stack write and read.

If the logic analyzer contains an integrated data base, such as with a microprocessor

development system, it is possible to translate the target addresses into labels or the assembly level mnemonics back into the high level language source code. In the example the JSR would correspond to a function call.

The Logic Timing Analyzer Display or Trace.

The logic timing analyzer normally displays its results, also know as a trace, as a timing diagram. This is illustrated for one channel below. The trace appears as a "text book" version of the actual signal(s). As shown the logic analyzer determines if signal from the system under test (SUT) is greater or less than a threshold voltage at each active clock edge.



For the logic analyzer the vertical resolution is only one bit, allowing only two states to be displayed, while the horizontal resolution matches that of the clock. The logic timing analyzer does not display finite rise times, voltage overshoots or ringing; it only displays if the sampled signal is greater or less than a preset threshold voltage. Provided the logic analyzer voltage threshold is set to be the same as that of the microprocessor in the SUT then the one bit vertical resolution is sufficient to explain the system functionality. That is, the user is able to make observations or statements such as "at this time the input signal is a logic high (or low)".

The time resolution of the logic timing analyzer is one clock period. If the state of the trace changes this change will be shown corresponding to the analyzer active clock edge. All the user can say is that the input change occurred somewhere in the previous clock period. This is illustrated in the example where the input signal has changed between the analyzer clock but the trace does not change until the sample clock. Often the user is only interested in the sequence in which the signals change state, so the relatively coarse horizontal (time) resolution is not a severe limitation. The functional description of the SUT will be of the form "in this time interval there was a signal transition from high to low (or low to high)".