TechOnLine Publication Date: Oct. 13, 2004

How Much Bandwidth Does Your Logic Analyzer Need?

Brock J. LaMeres
Agilent Technologies

There is much confusion when it comes to discussing bandwidth and logic analyzers. Traditionally logic analyzers are thought of as a purely digital measurement tool. However, as datarates increase and rise times shrink, designers are being forced to understand the analog characteristics of this tool. One of the biggest problems that designers face is ensuring that their verification tools are able to function at these higher frequencies. Factors such as the equipment's bandwidth and loading can cause false negatives and break systems when not fully understood. It is imperative that digital system designers can trust their logic analyzer in order to achieve the fastest time-to-market. As frequencies continue to rise, the logic analyzer front-end needs to be treated with the same analog delicateness as an oscilloscope.

There are two main considerations that engineers must understand when analyzing the bandwidth of their system and validation tools. The first is the frequency components present in their digital signals on their PCB and how that relates to their logic analyzer's bandwidth. The second is how the probe loading will interact with these frequencies. Both considerations come down to the theory of how digital signals are translated into analog metrics and how to use these metrics to analyze whether a successful measurement can be made. The following sections will discuss the three techniques to translate a digital signal into analog metrics. The three translations are rise-time-to-bandwidth, toggle-rate-to-bandwidth, and pulse-width-to-bandwidth. Once a digital signal can be described in terms of bandwidth, then the loading and logic analyzer bandwidth can be easily analyzed.

Translating Rise-Time to Bandwidth

The most popular method to convert the rise-time of a digital signal into analog bandwidth is by using a 1-pole RC circuit to model the response of a standard load. The circuit is solved in both the time and frequency domains and rules-of thumb for rise-time and bandwidth are generated that are in terms of resistance and capacitance. The rules of thumb for rise-time and bandwidth are combined through substitution which result in the resistance and capacitance values dropping out leaving a linear relationship between rise-time and bandwidth. **Figure 1** shows the RC circuit that is used in this derivation to model a standard load.

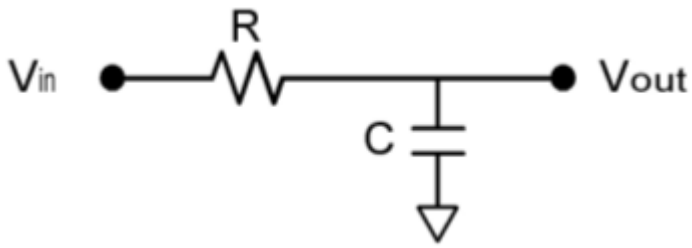


Figure 1: Solving an RC circuit in both the time and frequency domain results in a linear relationship between rise-time and bandwidth.

The first step in this derivation is to solve the circuit in the time domain assuming it is being driven with a unit step (u(t)). The general solution to this circuit is given by:

$$V_{OUT} = V_{IN}\left(1 - e^{-\frac{t}{RC}}\right) \quad \textbf{(1)}$$

Since rise-time is defined as the time it takes to transition from 10% to 90% of V_{OUT}, we can solve the equation to get two separate solutions. The first solution is the time it takes to transition from 0 volts to 10% of V_{OUT}. To accomplish this, V_{IN} is set to 1 volt and V_{OUT} is set to 0.1 volts. The second solution is obtained in the same way except that V_{OUT} is set to 0.9 volts. Since rise-time is defined as the time between these two solutions, the results are simply subtracted and yield a rule of thumb for the rise-time of an RC circuit.

$$t_{rise} = 2.2 \cdot RC \quad \textbf{(2)}$$

The second step of this derivation is to solve the same RC circuit in the frequency domain. The general solution to this circuit is given by:

$$\frac{V_{OUT}}{V_{IN}} = \frac{1}{1 + RCs} \quad \textbf{(3)}$$

Since bandwidth is defined as the frequency at which the magnitude of the response is attenuated by 30%, then this expression can be solved to generate a rule of the thumb.

$$BW = \frac{1}{2\pi RC} \quad \textbf{(4)}$$

Now that we have general expression for rise-time and bandwidth in terms of resistance and capacitance, we can combine the two expressions to yield a single linear relationship. This expression can now be used to quickly convert between the rise-time of a digital signal and the frequency components that the rise-times possesses.

$$t_{rise} \cdot BW = 0.35 \quad \textbf{(5)}$$

Translating Toggle-Rate to Bandwidth

The digital toggle rate of a signal can be converted into a frequency representation using the Fourier Transform. In Fourier analysis, there are a set of basic transforms that can be scaled according to the application. In Fourier representation, a periodic signal is represented as a series of impulses occurring at the desired periodicity in the time domain. This type of representation is called the Shaw function (III(t)). Its transform into the frequency domain is another Shaw function (III(s)). The scaling between the two domains is accomplished by the Similarity Theorem which has an inverse relationship between the two domains (as in, closer spaced impulses in the time domain will result in wider spaced impulses in the frequency domain).

$$\Im\{III(t \cdot period)\} = \left(\frac{1}{period}\right) III\left(\frac{s}{period}\right) \quad \textbf{(6)}$$

A rule of thumb commonly used in digital systems is that the system must have enough bandwidth to capture the 3rd harmonic of the digital pulse train. When relating this to the Shaw function, the third harmonic refers to the third impulse in the frequency domain. **Figure 2** illustrates the transform and how the rule of thumb relates.

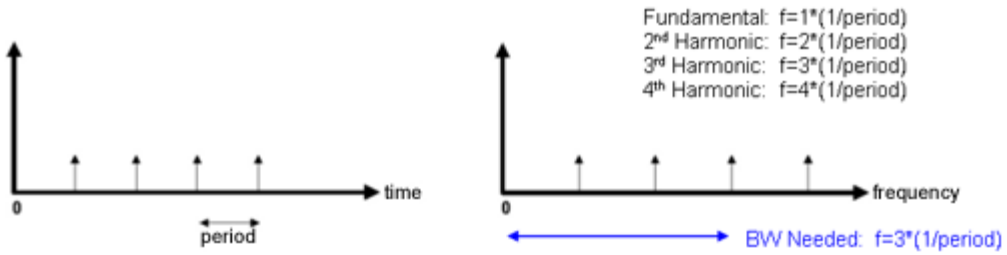


Figure 2: A periodic signal in the time domain is represented as a series of impulses spaced at the period of the signal. This translates into a series of impulses in the frequency domain that occur at the integer evaluation of (1/period).

The total bandwidth needed for a periodic signal can be expressed as:

$$BW_{periodic\_signal} = 3 \cdot \left(\frac{1}{period} \right) \quad \textbf{(7)}$$

Translating Pulse-Width to Bandwidth

Pulse width can also be converted into a frequency representation using the Fourier Transform. In Fourier representation, a pulse in the time domain is described as the rectangle function (Π(t)). When this function is translated into the frequency domain, it yields the sinc function (sinc(s)). Again, the scaling between the two domains is accomplished by the Similarity Theorem which has an inverse relationship between the two domains (as in, a narrower the pulse in the time domain yields a wider sinc envelope in the frequency domain).

$$\Im\{\Pi(t \cdot width)\} = \left(\frac{1}{width} \right) \mathrm{sinc} \left(\frac{s}{width} \right) \quad \textbf{(8)}$$

The sinc function produces a series of envelopes as the frequency increases. The zero crossings of the sinc function will occur at integer evaluations of the sinc function argument. In this case it will be evaluated at integer evaluations of (1/width). **Figure 3** shows how a time domain pulse is represented in the frequency domain.

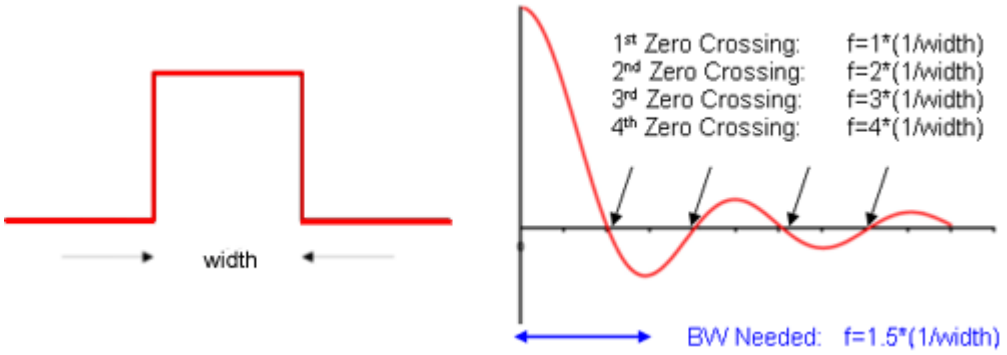


Figure 3: A pulse in the time domain transforms to a sinc function in the frequency domain. A rule of thumb is that the system needs enough bandwidth to capture half of the second sinc envelope.

As mentioned before, a common rule of thumb is that the system needs enough bandwidth to capture the third harmonic of the digital pulse train. In Fourier representation, a digital pulse train is described as a rectangle function convolved with the Shaw function. In the time domain, this produces a series of pulses repeating at the maximum toggle rate of the data stream. The pulse is represented using

the rectangle function and the periodicity of the data stream is represented with the Shaw function. The convolution operator combines the two functions in the time domain. In the frequency domain, the rectangle function transforms into the sinc function, the Shaw function transforms into another Shaw function¸ and the convolution operator transforms into a multiplication operation. The multiplication of the Shaw and sinc functions in the frequency domain has the effect of producing Shaw impulses that are bound by the envelope of the sinc function.

$$\Im\{\Pi(t \cdot width) * III(t \cdot period)\} = \left[\left(\frac{1}{width}\right) \cdot \sin c\left(\frac{s}{width}\right)\right] \cdot \left[\left(\frac{1}{period}\right) \cdot III\left(\frac{s}{period}\right)\right] \quad \textbf{(9)}$$

For a 50% duty cycle pulse train, the Shaw impulses will occur at every integer evaluation of (1/period). The envelope of the sinc function will have zero crossings at every integer evaluation of (1/width), which will cancel out the Shaw impulses at these frequencies. What is left are the Shaw impulses evaluated at every odd integer evaluation of (1/period). The first odd integer impulse evaluation is called the fundamental frequency. The remaining odd integer impulse evaluations are called harmonics. As stated before, the system needs enough bandwidth to capture the third harmonic of the pulse train. In the special case just described, this occurs in the middle of the second sinc envelope. The assumption that this is sufficient bandwidth can be extended as the pulse width decreases. It can now be said that a system needs enough bandwidth to capture 1/2 of the second sinc envelop to reliably deliver the associated pulse. This relationship can be written as:

$$BW_{pulse} = 1.5 \cdot \left(\frac{1}{width}\right) \quad \textbf{(10)}$$

Evaluating the Logic Analyzer Bandwidth

Logic analyzers specify the bandwidth of their front-end circuitry similar to oscilloscope front-ends. When deciding whether a logic analyzer has adequate performance for debugging a particular digital signal, the previously mentioned transforms can be used. The best way to illustrate this is through an example. A system has the following specifications for its digital signals. Each of these specifications can be converted to their analog bandwidths. The logic analyzer needs to have enough bandwidth to accommodate the highest of these analog frequencies.

	System Specifications	Corresponding Bandwidth
Maximum Toggle Rate	600 MHz (1.2 Gb/s)	1.800 GHz
Minimum Pulse Width	800ps (48% duty cycle)	1.875 GHz
Rise-Time	250ps	1.4 GHz
Logic Analyzer Bandwidth Needed		1.875 GHz

Evaluating the Logic Analyzer Probe Loading

Logic Analyzers also specify the loads of their various probing options. This is typically in the form of a lumped capacitance and/or an impedance vs. frequency profile. For a quick analysis of whether the probe load will break the system, the lumped capacitance can be used. When considering the probe load, the frequency at which its capacitance begins to shunt out the target signal must be high enough as to not effect the three previously mentioned metrics. The probe capacitance forms a RC filter with the transmission lines in the system (typically 50Ω). This will have a response given by:

$$f_{3dB-probe} = \frac{1}{2\pi(Z_0 // Z_0)C_{probe}} \quad \textbf{(11)}$$

If we use the example in the previous section, we can determine the maximum capacitance that the

probe can present on the system without sever degradation. In the above example, the system had 1.875 GHz of bandwidth present in its digital signals. Plugging this into the above expression returns the maximum probe capacitance that can be tolerated by the system. In this case it is 3.4pF.

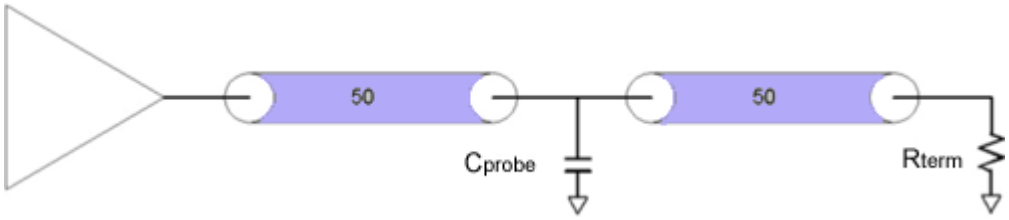


Figure 4: The capacitive load of the logic analyzer probe forms an RC filter with the impedance of the transmission line. In order to not disturb the signal being probed, the 3dB frequency of the probe's RC load must be larger than the highest analog frequency present in the digital pulse train.

Conclusion

As data rates increase in digital systems, engineers are being forced to understand the limitations of their validation tools. By understanding the frequency components present in their digital signals, engineers can quickly evaluate the performance ability of their logic analyzer and the logic analyzer probe load. By taking the time upfront to evaluate the performance of their tools, engineers can increase the probability of making a successful measurement and dramatically reduce their turn-on and validation time.

About the Author

Brock J. LaMeres received his BSEE from Montana State University in 1998 and his MSEE from the University of Colorado in 2001. He is currently working on his Ph.D. at the University of Colorado where his research focus is High-Speed IO for next generation IC's. LaMeres is a hardware design engineer for Agilent Technologies in Colorado Springs where he designs logic analyzer probes and high-speed transport systems.

Tech Groups:
Analognet
Communicationsnet
DSPnet
EDAnet
Embeddednet
SOCnet
T&Mnet

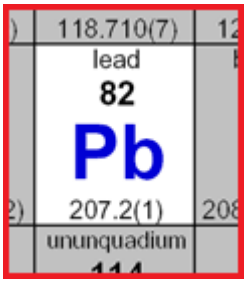
Educational Resources:
Design Challenge
Live Webcasts
On-Demand Webcasts
Courses
VirtuaLabs
Feature Articles
Technical Papers
Forums

Tech Topics:
Bluetooth
CSoC
Design Services
Internet/Network Security
Java
Linux
MEMS
Power Supplies
RTOS
Timing Closure
UML
USB / Firewire
VoIP
Web-Enabled Design

Company Directory:

Additional Resources:
Calendar
Jobs
News
Products
TechOnLine Newsletters

SPOTLIGHT



Pb Free

By the end of this year, chip packaging, solder, solder paste, and other aspects of the electronics industry will start getting more environmentally friendly. New legislation passed by the European Parliament in 2003 will reduce the amounts of heavy metals such as lead, mercury, cadmium, and chromium in the environment. In Getting the Lead Out: European Initiative Will Have Global Impact, Jack Shandle and Lee Goldberg examine how the electronics industry will be lead-free by 2006.

- **TechOnLine Site Survey:** Complete for a Chance to Win Cash

WHAT'S NEW

On-Demand Webcasts

- Mobile Entertainment: Delivering Handheld Multimedia and Gaming Solutions
  Intel • 60 min.

Courses

- ZigBee Overview and Products
  Technology Course • 72 min.

VirtuaLabs

- M16C/62P with FoUSB
  Renesas Technology
  OS: Renesas HEW3
  Processor: M16C

Feature Articles

- How Much Bandwidth Does Your Logic Analyzer Need?
  Brock LaMeres
  Agilent Technologies

Technical Papers

- DSP Co-

LIVE WEBCASTS

Live Webcasts

- **Oct 19, 2004:** ModelSim Designer
  Mentor Graphics • 60 min.
- **Oct 20, 2004:** Controlling Auxiliary Output Voltages with the UCC2540 PWM
  Texas Instruments • 60 min.
- **Oct 20, 2004:** Enabling Real-Time JPEG2000 with FPGA Architectures
  Altera • 60 min.
- **Oct 21, 2004:** Building Flexible Data Delivery Into Your Connected Industrial Devices
  Wind River and Tilcon Software • 60 min.
- **Oct 21, 2004:** Providing the Fastest Turnaround Times: Enhancing and Tuning The Calibre Design-to-Silicon Platform Engine
  Mentor Graphics • 60 min.

More Live Webcasts >>

View On-Demand Webcasts >>

MEMBER COMPANY SPOTLIGHT

Texas Instruments

Controlling Auxiliary Output Voltages with the UCC2540 PWM: This 60 min. webcast will present a unique technique to generate multiple output voltages using the new UCC2540 Pulse Width Modulator. Register today for this Oct. 20th webcast.

MEMBER COMPANIES

Select a Member

When do you plan to use an SA/PA in an SoC design?

- Within 3 months
- 3-6 months
- 6-12 months
- Over 1 year
- Never

Submit

TECHTOPIC SPOTLIGHT

TechOnLine's TOP 10 Ten Most Popular

Ever wonder what courses and articles your peers view the most on TechOnLine? Find out which 10 courses, webcasts, feature articles, and technical papers were most popular with the TechOnLine community for each month.

- Course Top 10
- Feature Article Top 10
- Tech Paper Top 10
- Webcast Top 10