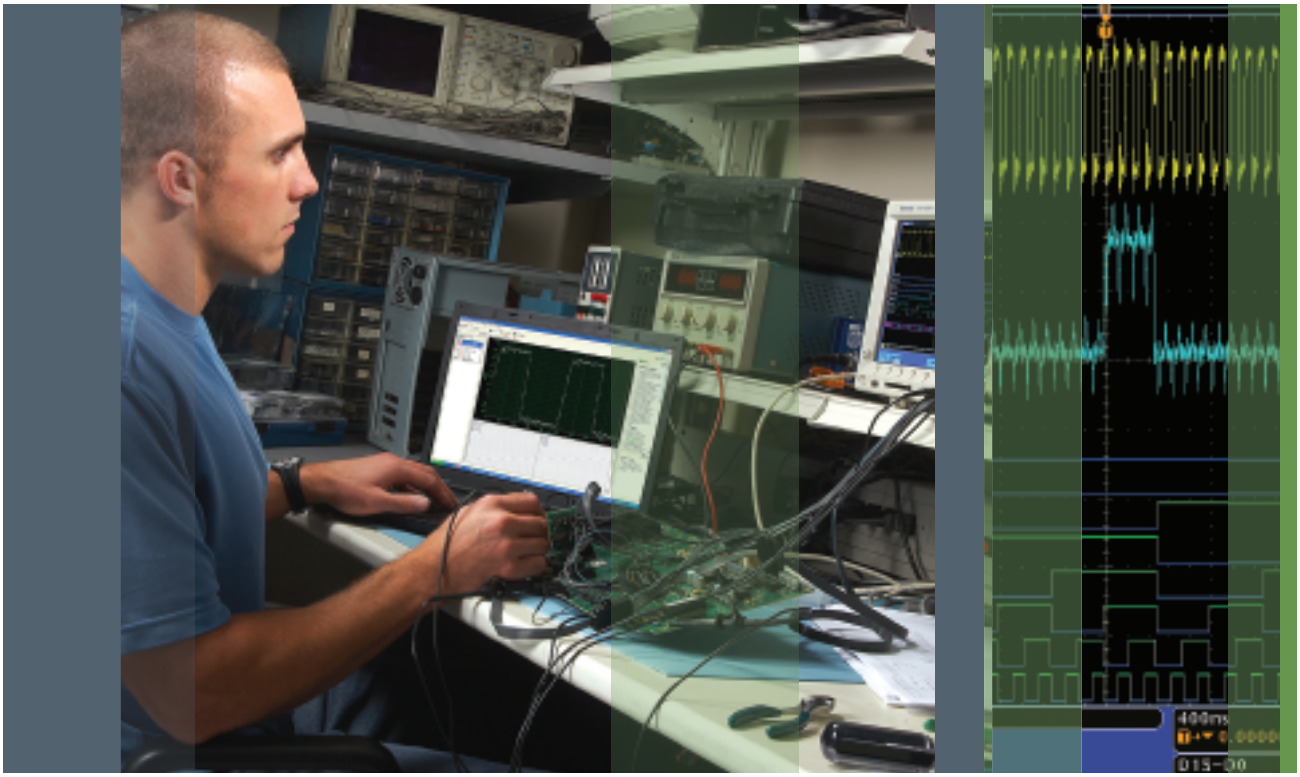
Primer



# Embedded Systems: Test & Measurement Challenges

Tektronix®

# Table of Contents

**Embedded Systems Overview . . . . . 3 - 6**

Industry Drivers . . . . . .3
Market Trends and Drivers . . . . . .4
Engineering Needs and Responses . . . . . .4
Elements of Embedded Designs . . . . . .5
Key Test Challenges of Embedded Designs . . . . . .6

**Microprocessors and Microcontrollers . . . . . 7 - 11**

Debugging Embedded Processors . . . . . .7
High-level-language Source Code Support . . . . . .8
Symbolic Support . . . . . .8
Performance Analysis Support . . . . . .9
Worst-Case Execution Time Support . . . . . .10

**Field Programmable Gate Arrays (FPGAs) . . . . 12 - 17**

FPGA Design Process Overview . . . . . .13
Design Phase . . . . . .13
Debug and Verification Phase . . . . . .13
FPGA Debug Methodologies . . . . . .13
Embedded Logic Analyzer Core . . . . . .13
External Oscilloscope . . . . . .14
External Mixed-Signal Oscilloscope . . . . . .14
External Logic Analyzer . . . . . .14
Choosing the Right FPGA Debug Methodology . . . . . .15
Summary . . . . . .17

**Memory . . . . . .18 - 25**

SDRAM . . . . . .20
DDR SDRAM . . . . . .20
DDR2 SDRAM . . . . . .21
DDR3 SDRAM . . . . . .22
Memory System Design . . . . . .23
Design Simulation . . . . . .23
Design Verification . . . . . .23
Verification Strategy . . . . . .23
SDRAM Verification . . . . . .24

**Low-speed Serial Buses . . . . . .26 - 30**

Measurement Challenges . . . . . 29
Triggering Versus Search . . . . . 30
Summary . . . . . 30

**High-speed Serial Buses . . . . . 31 - 36**

High-speed Serial Standards . . . . . .32
Measurement Challenges . . . . . .33
Summary . . . . . .36

**Power Supplies . . . . . 37 - 43**

Switch-mode Power-supply Basics . . . . . 37
Characterizing SMPS Performance . . . . . .38
Electrical Characteristics . . . . . .38
Magnetic Characteristics . . . . . .39
Input/output (I/O) Analysis . . . . . .39
Power-supply Design . . . . . .40
Measurement Basics . . . . . .40
Probing Considerations . . . . . .42
Automated Power Measurements . . . . . .42
Powerful Measurement Capabilities . . . . . .43

**Digital RF Technologies . . . . . 44 - 50**

Changes in Spectrum Usage . . . . . .44
Engineering Challenges . . . . . .45
Testing Digital RF Technologies . . . . . .46
Measurement Tools . . . . . .46
Digital RF Applications . . . . . .48
Signal Sources . . . . . .50

**Looking Ahead with Total Insight . . . . . 51**

# Embedded Systems Overview

Embedded systems are compact, specialized computing devices using dedicated microprocessors or microcontrollers, and typically perform very specific pre-defined tasks designed to service specific functions or integrated within much larger and more complex devices. Examples include automotive systems, avionics, network devices, industrial controls, medical prosthetics, consumer electronics and communication networks.

Thanks to the benefits of Moore's Law, the shrinking cost and growing performance of technology are leading to smaller and more powerful processing devices being integrated into a wide range of everyday items.

However, the truly significant developments in technology will be in the arena of decentralized, pervasive computing where previously separate devices will be both more intelligent and connected to networks.

Embedded devices, as the name implies, are sensors, processors and controllers that are incorporated into objects – which in turn interact with their environment, with people and with one another, and work in combination to support highly specialized applications and needs.

## Industry Drivers

Embedded technology is being used to deliver new applications and services in a variety of industries. Most significantly, several industries are leading the 'pervasive electronics' megatrend, including:

- **Consumer electronics:** Increasing design complexity is driving the consumer electronics industry to greater use of embedded systems. From advanced multi-function mobile telephones to digital televisions, devices such as computers, microprocessors, or other electronic components such as field programmable gate arrays (FPGAs) and transmission electronics are embedded within these applications. Increasingly, consumers want to integrate these devices with other products, and be able to upgrade these embedded devices as new advances become available.
- **Industrial electronics:** Factory automation, process controls, power generation and management, security and environmental monitoring all rely on embedded electronic systems. Business needs ranging from quality, cost reduction, efficient operations, data and information management are driving wider uses of electronic control and management systems.
- **Automotive:** It is estimated that about 25% of the value of a modern car lies in the electronics. This is further estimated to increase to about 40% by 2010. The average new car comes with more than a dozen microprocessors inside, controlling everything from the engine to the radio. Some high-end models include more than 100 microprocessors and microcontrollers. Many are subtle, like the auto-dimming chips inside a rear-view mirror. From 8-bit controllers to 32-bit processors, automotive embedded systems provide electronic controls for antilock brakes, traction and stability controls and engine management.
- **Avionics:** Embedded electronics are widely deployed in aircraft in areas such as cockpit instrumentation, air data, inertial systems, engine control, electrical power generation, hydraulics, fuel systems, autopilot, navigation, GPS, ILS, landing gear, flying surfaces, slats, flaps, etc. Innovations continue; for example, new fully automated adaptive rotor systems within helicopters are able to reduce noise on take-off and landing and also reduce vibration during flight.
- **Medical:** Embedded systems are widely used for patient monitoring and diagnostics, in operating rooms and within technical devices such as MRI and PET scanners. Surgical robots may soon take a place in operating rooms. 'System on a chip' (SOC) integrated technology has recently been used to create a chemical and biological laboratory capable of carrying out analysis at the point of care, without having to wait one or two days for laboratory results. In a 'non-technical' application, an artificial leg employing embedded systems can operate in unison with the biological leg. The embedded system is based on scientific gait analysis and biomechanical studies, performed under microprocessor and software control based on inputs from numerous sensors.

- **Communications:** Examples of embedded technologies abound in the communications industry, driven by the merging of traditional analog with newer digital technologies, shifts from conventional voice and time-division multiplexing (TDM) based networks towards multimedia networks integrating voice and data including video, into areas such as global positioning and tracking. Network processors are increasingly being used in switches, gateways, routers and other communications equipment such as security and control plane processors to manage performance, power consumption, traffic management, and other embedded functions. Increasingly, too, Voice over Internet Protocol (VoIP) chips are being embedded throughout networks.

## Market Trends and Drivers

In addition to the industries that are driving embedded-systems development, there are also a number of market forces at work:

- **Increasing functionality and intelligence:** Consumer demand is continuously growing for higher performance devices and richer and more intelligent user interfaces, with the ability to seamlessly communicate with other devices. New applications ranging from sensor networks to energy conservation continue to drive performance.
- **Decreasing size:** More and more capabilities are being designed into increasingly smaller designs and devices.
- **Decreasing cost:** Along with expectations of higher performance, prices are continuously driven down through design and manufacturing efficiencies with 'off the shelf' technologies and the resulting economies of scale.
- **Lower power consumption:** In most cases, particularly for portable devices, driving down the power requirement - leading to longer battery life or lower operating cost - is necessary for a product to succeed in the market.
- **Interoperability:** 'Plug & play' connectivity through standardized I/O interfaces that are both wired and wireless creates new opportunities for the application of embedded intelligence.

## Engineering Needs and Responses

In order to meet the challenges presented by the requirements of different industries and the marketplace, a number of areas require addressing by product engineering teams:

- **Industry standards adoption:** Embedded system elements and the buses connecting them are becoming increasingly reliant on, and compliant with, industry standards. This facilitates interoperability and the economies of scale associated with off-the-shelf technology. Tools with support for the many industry standards are needed to ensure performance characterization and standards compliance.
- **Advanced performance:** Reusable components level the playing field for the design and delivery of basic embedded systems. Real innovation in technology and applications requires the development of new technology and processes that utilize mixed signals in ever more complex designs. Highly capable test tools with extensive functionality and usability are needed to address these challenges.
- **Integration and connectivity:** While standards exist for many technologies commonly used within an embedded system, a major test requirement is to ensure that all elements are synchronized and perform as a seamless, integrated whole. End devices may contain multiple embedded systems, some of which need to communicate with one another and with the outside world. This is an extension of integration testing, to ensure integrated functions, timing operation and communication.This area requires test tools that enable evaluation of not only a single element but also an entire system.
- **Mixed signals:** In step with increasing functionality and performance, engineers often have to work with both analog and digital signals in their designs. This complicates the job of testing, requiring specialized tools to get inside the components and see what is going on with test points on the device under test.

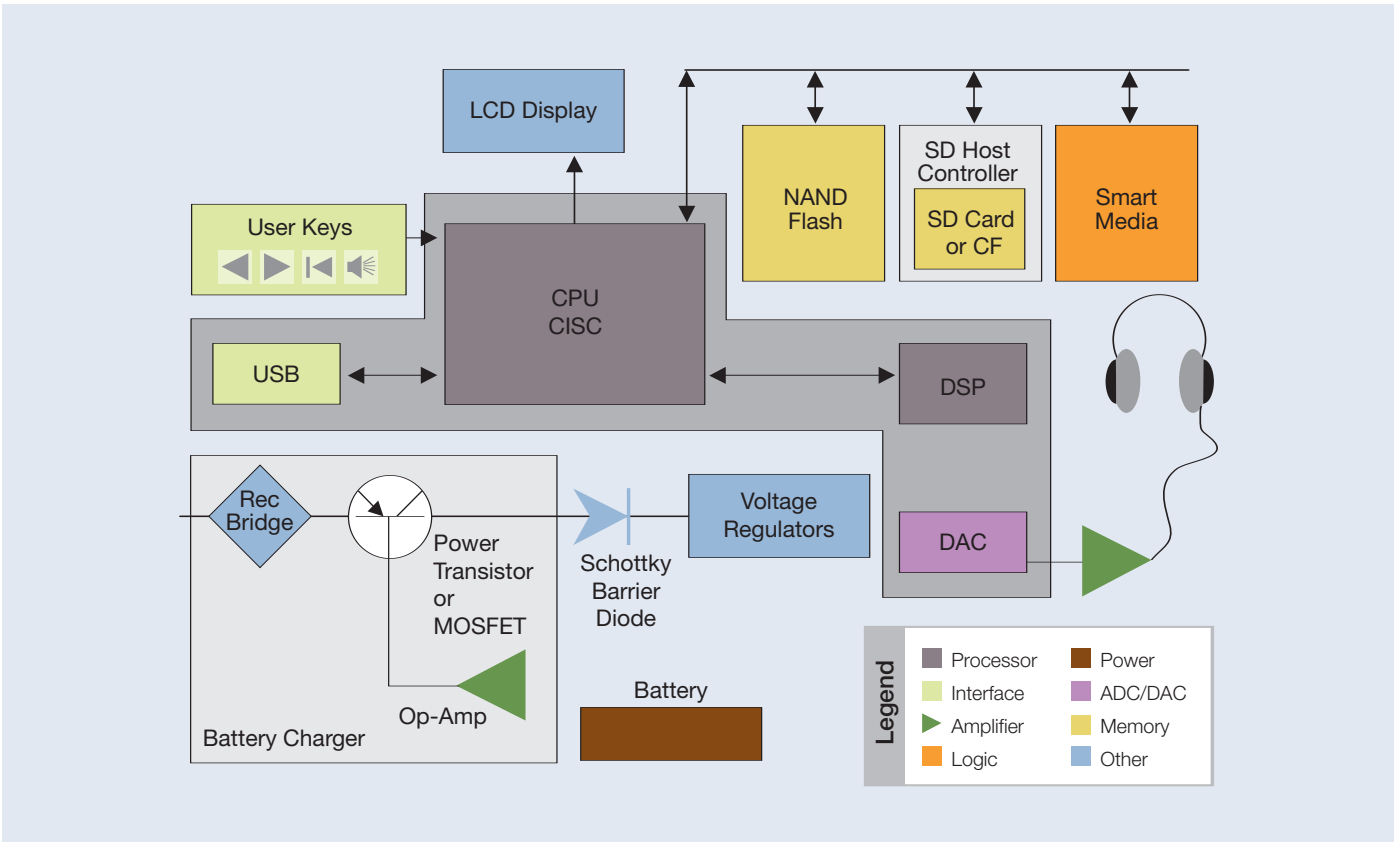



**Figure 1.1.** A block diagram of a typical MP3 music player. As with many embedded computing designs, the MP3 player uses mixed signals, containing digital serial buses for USB, digital signal processors, digital parallel buses to connect to memory, digital-analog converters, and an analog amplifier.

- **Challenging environments:** Modern electronic devices are expected to offer the latest performance and functionality along with reliability and the ability to work across changing environments. Aircraft avionics are clearly expected to work across extremes of temperature and altitude, and even consumer electronic devices need to work wherever people work or play. Powerful test tools are needed to stress products in the lab and ensure that they can also coexist in changing conditions and environments without electrical interference.

In this primer, we shall look at the various elements of embedded system in detail, and describe some of the challenges that arise in their implementation: challenges that are being addressed by a new generation of test & measurement tools.

## Elements of Embedded Designs

Figure 1.1 shows a typical modern device using embedded technology, and illustrates the diversity of devices typically used.

The brains of an embedded system are typically microprocessors, microcontrollers or digital signal processors (DSPs). Other types of processing devices are application-specific integrated circuits (ASICs) and field-programmable gate arrays (FPGAs). An FPGA is a programmable integrated circuit device that can be programmed for a wide variety of tasks. These devices are used for tasks ranging from consolidating functionality in a system to tasks requiring specialized computing power. The main benefit of FPGAs is their reprogrammable capability, which can significantly decrease design times while also adding functionality and lowering overall design costs.

Analog/digital converters (ADCs), digital/analog converters (DACs), sensors and transducers provide the interface to the physical world. These devices serve the most critical role of converting information from an analog world into digital data and accepting digital data and delivering it back into the analog environment. The number of these elements in a particular design depends on the type of device and application.

As the number of embedded hardware elements increase within a device, so do the number of communication paths between these elements. The exclusive use of parallel buses to connect all these elements would not be feasible because of other key drivers such the need to minimize cost (which demands optimum use of board space) and decrease size. This has led to a proliferation of designs using serial buses, which typically require only a few connections, compared to many for parallel buses. Although most embedded devices are not running at the speeds of high-performance computing systems, they utilize many industry standard serial buses, often with different buses in combination. Still others combine parallel and both low- and high-speed serial buses.

Embedded systems designs typically use low-speed serial buses such as $I^2C$, SPI, RS-232, FlexRay, CAN, LIN and USB. The complexities of these serial buses present significant validation and debug challenges. Engineers need tools with integrated serial triggering, protocol decoding, and comprehensive analysis capabilities. In addition, leading performance embedded designs use higher-speed serial buses such as Ethernet, PCI-Express, SATA or HDMI. Engineers need test instruments and software that provide high-speed serial data acquisition, debug, validation and compliance testing. In many instances, parallel buses provide the best interface between the processing system and memory devices relying on technologies like DDR. Consequently, engineers need test tools that are proficient in debugging parallel bus problems in addition to their serial bus capabilities.

## Key Test Challenges of Embedded Designs

One of the key test challenges faced by engineers is the ability to acquire and monitor different signals and protocols. Design engineers need to be able to generate a variety of signals to stress test the device under test (DUT) to determine how the design would behave in the real world. They need test solutions that can capture and visually reproduce these signals to verify the signal integrity. They need precise timing information between multiple digital signals on a bus for troubleshooting set-up and hold violations. In many cases, when hardware and software engineers are working together to troubleshoot the root cause of a specific problem, they require a view of information on a bus - both its electrical representation and also at a higher level of abstraction like the assembly code of a microprocessor or the decoded view of a serial bus protocol.

Many designs have a large number of hardware components for executing specific tasks that may be located on different parts of the circuit board. To ensure proper interaction between components, embedded engineers need to have a system-level view of the DUT. The challenge is to make sure that the component operations are synchronized, which means that the test equipment needs to be able to provide accurate information on timing performance in addition to creating higher levels of abstraction and analysis.

In many instances during development, not all components are available for test, making it necessary to ‘reproduce’ or simulate the signals of the missing component to test the overall operation of the device. If the signal has a complex wave shape, it can be acquired once with an oscilloscope and then replicated with an arbitrary waveform generator. In other cases, components need to be stress tested for their ability to handle impaired input signals by purposely including jitter, noise or other anomalies. To generate these signals, arbitrary/function and arbitrary waveform generators are the tools of choice.

Attaching a probe to the DUT poses another challenge. The small physical size of the devices, the large number of points on the board that need to be probed, and the fact that any probe adds capacitive loading altering the operational characteristics of the DUT are all factors that add to probing challenges. Probing solutions need to be designed to minimize the capacitive loading, make it easier for the engineer to connect to the DUT, and also be able to quickly ascertain which probe (or probe lead) is correlated to which trace on the screen of a test instrument.

The elements of embedded systems, and the test challenges that each of them imposes, are covered in the following sections.

## Microprocessors and Microcontrollers

The demand for more features and performance directly drives the need for more capable and faster microprocessors and microcontrollers at the heart of any electronic system. Embedded designs are becoming increasingly more distributed, requiring a corresponding increase in the number of microprocessors and microcontrollers in the design performing individual tasks. The increasing number of processing elements increases the number of communication paths in the design, adding to system complexity.

Traditionally, higher performance was typically achieved by increasing the clock frequency. However, this 'quick fix' is unsuitable for embedded processors. It increases power consumption, leads to higher electromagnetic interference (EMI), requires very stable power supply, and increases the bottlenecks in accessing platform resources, such as flash memory.

The recent introduction of multi-core processors offers a solution. Instead of scaling performance by improving single-core performance, performance is now scaled by putting multiple cores on a single chip, effectively integrating a complete multiprocessor on one chip.

Moreover, multi-cores offer a better performance to power ratio than a single-core solution with similar performance.

As market drivers increase the hardware requirements and software complexity, so programming environments and languages have become more sophisticated and complex legacy embedded designs had relatively simple operating systems, but these have now been replaced with more capable high performance embedded systems, which in turn require more processing power.

## Debugging Embedded Processors

Embedded software developers are responsible for the flawless execution of their software in a real-time embedded system. They must ensure that both the input and output data-flow processing and system time constraints are validated to design specifications. Valid measurements however can only be performed on the actual target system; this requires very close collaboration with the hardware developers on the team.

When debugging embedded target systems incorporating microprocessors or microcontrollers, the digital design team often encounters four types of problems:

- **Logic problems:** These are simple mistakes in logic design or coding, or an incorrect design assumption. These problems are often caught by the hardware simulator or by the software debugger. This type of problem often comprises 80% of the debug problems encountered but may only consume 20% of the debug time. While tedious and time-consuming to find, these problems are relatively straightforward to fix. The remaining 20% of the problems fall into the remaining three categories and may take up to 80% of the debug time to find and fix.
- **Hardware/software interaction problems:** These problems are more difficult to debug and often require some form of physical trace tool. The logic analyzer has been the traditional physical trace tool of choice for such problems. Logic analyzers provide both the hardware and software analysis capability to examine the interaction of the hardware and the software in a target system. With the ability to correlate specific signals with the processor's instruction execution flow, the hardware and software developers can debug complex problems such as why certain instructions cause memory errors or why a specific data pattern causes crosstalk-induced glitches on the processor's data bus.

- **Real-time software problems:** These are the most difficult problems to debug since they occur only when the target system is running at full speed. Logic analyzers excel in solving these problems because they run at the full speed of the target system and provide powerful triggering and deep trace memory to capture the most elusive real-time faults. This capability is enhanced by a new generation of partner analysis tools that provide a solution to the problem of determining worst-case execution times (WCET) for software components running on advanced microprocessors. It enables engineers to use a systematic and scientific approach to ensuring that time constraints are met - allowing them, in effect, to engineer timing correctness into their systems rather than spending a great deal of time and effort trying to get timing bugs out. The ability to identify how to optimize the code and eliminate timing errors increases reliability and reduces the need for costly hardware upgrades.
- **Crash problems:** Embedded systems differ from non-embedded systems (e.g. PCs or workstations) in that they generally do not have protection from an errant program crashing the target system. Robust operating systems, such as those found on PCs or workstations, have a variety of mechanisms to isolate the system from a misbehaving application; embedded systems often do not. Thus, when the software on an embedded system crashes, it often takes the whole target system down thereby losing any information that might be useful in determining the root cause of the problem. Logic analyzers, especially those with deep acquisition memory, can provide the real-time history of the target system, up to and including the crash, to aid in quickly determining the cause of the crash.

Solving these problems requires a tool that both the embedded software and the hardware developer can use to resolve the difficult cross-domain problems that span the range from high-level language source code down to the underlying signal behavior.

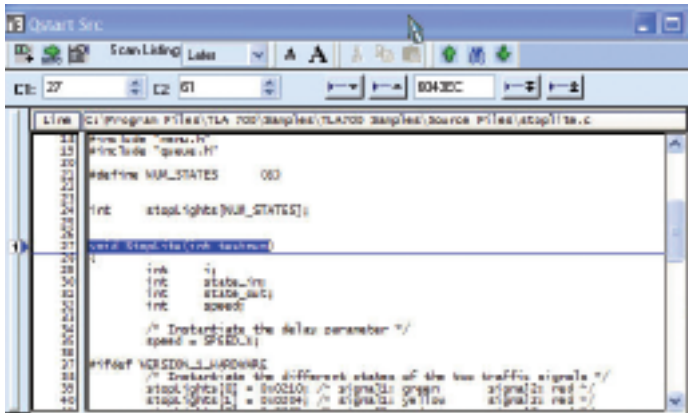


**Figure 2.1.** Source window where user's source code is displayed.

A modern high-performance logic analyzer (TLA series) with enhanced connection to software tools (WCET) provides engineers with data from a target system at the analogue, timing, state, and HLL source code levels, thereby providing a 'big picture' overview of target system behavior from signals to source code. Four such tools are particularly significant: **high-level language source code support; performance analysis; symbol support; and worst case software execution times analysis.**

## High-level-language Source Code Support

The logic analyzer's high-level language (HLL) source code consists of a data window where the user's source code is displayed (Figure 2.1).

The key to how the source window operates is the symbol file. In addition to the name and value of the symbol, it contains the line number and address of each source line. With this information, the source window can correlate the source code that the developer writes with the acquired data in the listing window.

There are a variety of mechanisms for navigating the data. Within the source window, it is possible to move to the next or previous executed source statement using the toolbar

buttons, scroll or page through the source file using the window sliders, move either of the two cursors, or switch to a different source file. From the listing window, move either of the cursors can be moved. When a cursor in either the source or listing window is moved, the corresponding cursor in the other window moves as well. It is possible to set a trigger condition based on the source line in the source window. The user can also control all the various window properties (e.g. variable font size, colors, show or hide line numbers, and tab spacing). Multiple source windows can be used for either different source files or for multi-processor support.

## Symbolic Support

The symbolic information contained within the object file that is loaded on the target system is the key to debugging at a symbolic and HLL source code level.

There are two types of symbol files:

**1. Pattern symbol files:** Pattern symbols have one value (the pattern) which can contain 'don't care' (X) values. Pattern symbols are often used for displaying variable names and in the control group of a disassembler display.

**2. Range symbol files:** Range symbols have two values - an upper and a lower bound - but cannot contain 'don't care' (X) values. Range symbols are often used in the address group of a disassembler display. Range symbols are found in the software executable file or object file that is downloaded to the target system.

Three types of symbols can be found within this file:

- **Function -** Range symbols describing the beginning and ending addresses of software functions.
- **Variable -** Range symbols describing the beginning and ending addresses of software variables.
- **Source code -** Range symbols describing the beginning and ending addresses of source statements. The logic analyzer's source window uses these range symbols to perform the linkage with the acquired data in the listing window. Any missing line numbers contain either comments or non-executable source lines. The 'beginning' and 'end' columns are generated by software tools to provide statement-level granularity within a line, which the source window can use to highlight individual statements in a single source line.

## Performance Analysis Support

Today's embedded software applications are getting increasingly larger, which makes it difficult to see the 'big picture' of the overall flow and execution time of the software. The embedded software developer will often get the code functioning correctly, but will leave performance tuning until near the end of the project. An often quoted rule of thumb is that '20% of the code executes 80% of the time'; however, the big question is which 20% of the code. What is needed is some form of overview tool to show which of the hundreds of software modules are consuming the majority of the processor's execution time. A logic analyzer with performance analysis capability can show where the software is spending its time. With this information, the embedded software developer can quickly zero in on those routines that, if optimized, have the greatest payback in improved performance.

Performance analysis support can provide two types of measurement:

- **Range overview (channel groups):** With range overview, the logic analyzer examines the data acquired for a user-selected logic analyzer module group and displays the hits in each range in a histogram format. It can use the same loaded symbol table (with absolute addresses) with no limit on the number of ranges. Ranges can be generated and displayed using either numbers (linear or logarithmic) or from a loaded symbol file. Data can easily be sorted (ascending or descending) by either range, count or percentage. The histogram window can be split to view both the beginning and end of the ranges. Various properties can be selected including colors, variable fonts and channel polarity. The user can also choose by percentage based on all samples or only matched samples.

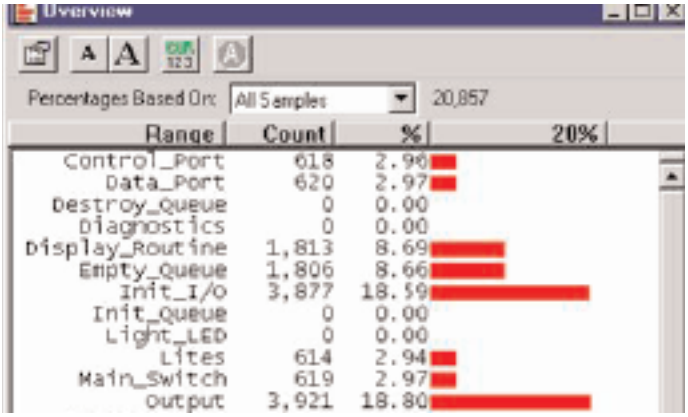



**Figure 2.2.** Range overview.

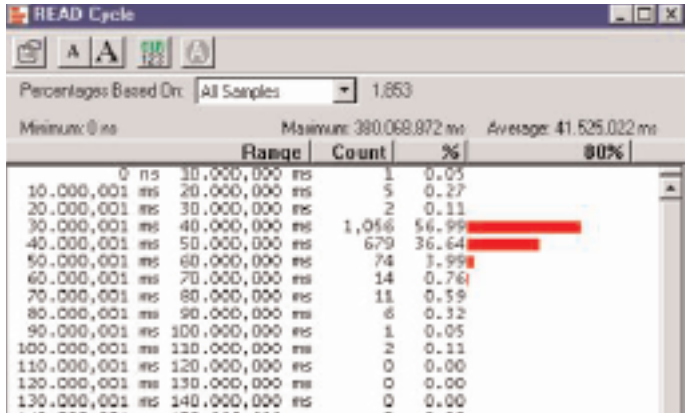



**Figure 2.3.** Single event (counter-timer).

- Range overview is especially useful for providing information on which software modules consume the most execution time. The developer must be careful, however, to ensure that recorded hits are actually measuring processor execution time, not 'pre-fetch' or other non-execution activity (Figure 2.2).
- **Single event (counter/timer):** Developers often have a routine that has a specified execution time for which they need to validate that it never exceeds the specification. 'Single event' is a measurement mode that uses the logic analyzer module's counter/timers to display the range of execution time/counts for a single routine. By using the analyzer's trigger machine to define an event, you can measure that event with high resolution. The resulting display shows the minimum, maximum and average time that a single event takes to execute over multiple runs (Figure 2.3). This capability is useful in monitoring time-critical routines such as interrupt/exception handlers to ensure that the system specification for maximum service time is never violated. Multiple histogram windows can be used for each logic analyzer module, which is a useful feature for multi-processor support.

## Worst-Case Execution Time Support

Using the tracing facilities of a high-performance logic analyzer (TLA series) combined with the RapiTime performance analysis tool from partner software tool vendor Rapita Systems Ltd. This provides powerful timing analysis capability for real-time software running on the latest embedded microprocessors.

RapiTime works by automatically adding lightweight instrumentation to the software. When the software is executed on the target microprocessor, these instrumentation points simply write a value, the instrumentation point ID, to an output port. The output port is monitored using a TLA which captures and accurately timestamps the instrumentation data, forming a detailed timing trace of program execution. The generated timing trace is then imported into RapiTime which performs code coverage analysis, performance profiling, and worst-case execution time analysis.

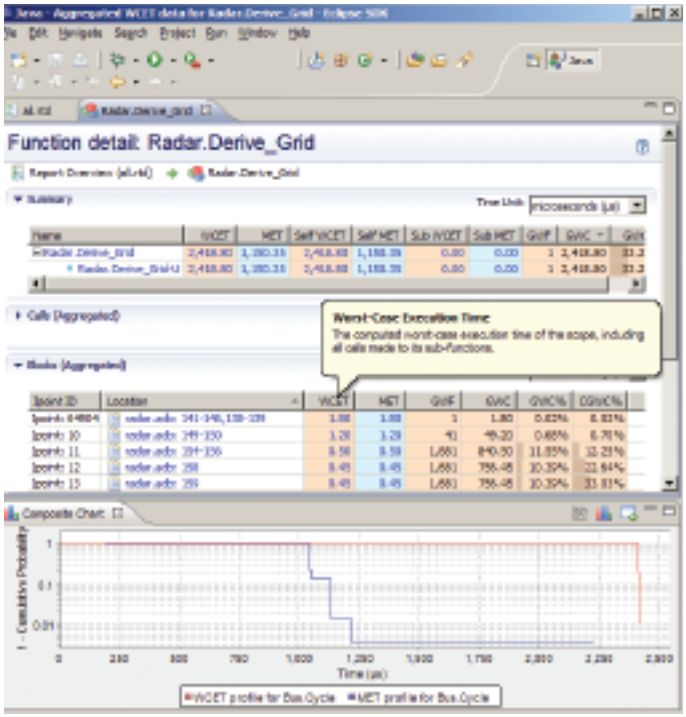



**Figure 2.4.** Results overview of worst-case execution times.

RapiTime enables users to determine the worst-case and longest observed execution times of software components; view the number of times each function or loop is called on the worst-case path; visualize the contribution of each software component, function, loop and basic block to the overall execution time and identify code on the worst-case path.

RapiTime users can locate worst-case hotspots, code that contributes the most to the worst-case execution time, although not necessarily much to the average case (Figure 2.4). Hence engineers can identify those functions where optimization will have the maximum benefit in ensuring that the software always meets its time constraints.

The RapiTime/TLA timing trace analysis is used to:

- **Calculate worst-case execution times** for each software component
- **Identify code on the worst-case path** via color coded source
- **Provide detailed analysis of worst-case hotspots** presenting the best opportunities for optimization
- **Display code coverage** to ensure confidence in the timing analysis results
- **Generate execution time profiles** to explore the variability in execution times due to hardware effects.

# Field Programmable Gate Arrays (FPGAs)

Field Programmable Gate Arrays (FPGAs) are a class of programmable logic devices (PLDs). The simplest PLDs are small blocks of programmable logic, typically containing a small number of combinatorial logic gates and sequential devices such as registers and flip-flops. By programming the block, the designer can realize many different logical functions. Examples are memory decoders and state machines. These devices are often factory-programmed.

Slightly more complex versions, with a simple interconnect matrix between multiple programmable logic arrays are called Complex PLDs (CPLDs). The logic blocks in these devices may also include SRAM cells to increase their versatility.

FPGAs are typically much larger arrays of logic, with a more sophisticated interconnect matrix. FPGAs may include microcontroller cores and other specialized circuits such as digital-locked loops (DLLs), boundary scan logic, high-speed serial interfaces, and low-speed serial programming interfaces (such as SPI and JTAG).

FPGAs are typically programmed by a serial memory device at power-up or by the microprocessor. Because of this programmability, easy field upgrades of product functionality are possible.

These highly complex FPGA devices contain various circuits and chip architectures which can be significantly different between the suppliers.

The design tools themselves usually produce an FPGA design which matches the functional definition. If the definition is wrong, either at the FPGA level or at the system level, the simulators will not provide a working design.

Even within the FPGA, there may be problems with the design caused by external issues, including:

- **Asynchronous events,** such as interactions with a microprocessor, low-speed serial interface, or user interface
- **System interactions,** such as unforeseen race conditions or delays
- **Noise and crosstalk** on signals and power supplies
- **Signal fidelity problems,** such as reflections, loading and electromagnetic interference (EMI)
- **Power supply effects,** like transients (high frequency, caused by large variations in power supply currents and inadequate bypassing) and load variations (lower-frequency power supply voltage variations due to power-system design problems)

It is also important to note that FPGA-level design problems may not be discovered in simulation – partly because it is too time-consuming to generate all the necessary test vectors to provide 100% code coverage, but also because the simulation is too slow to run for all possible conditions. It may be quicker to build a prototype and watch it operate in real time.

The phenomenal growth in design size and complexity continues to make the process of design verification a critical bottleneck for today's FPGA systems. Limited access to internal signals, advanced FPGA packages, and printed-circuit-board (PCB) electrical noise are all contributing factors in making design debugging and verification the most difficult process of the design cycle. An engineer can easily spend more than 50% of the design cycle time on debugging and verifying a design. To help with the process of design debugging and verification, new tools are required to help debug designs while they are running at full speed on the FPGA.

Of course, the simplest 'solution' is to do nothing special for FPGA debug. The designer can simply treat the FPGA just like any other IC and monitor the operation from the outside with an oscilloscope or logic analyzer. However, for designs of any complexity, this method is neither effective nor efficient.

## FPGA Design Process Overview

There are two distinct phases in bringing an FPGA system to market: the design phase and the debug and verification phase. The primary tasks in the design phase are entry, simulation and implementation. The primary tasks in the debug and verification phase are to validate the design and correct any bugs found.

## Design Phase

Not only is the design captured in this phase, but debugging begins with the use of simulation tools. The proper use of simulation has proven to be an effective way of finding and correcting design errors. However, simulation should not be relied upon as the only tool to debug a FPGA design. There are too many things that simulation just cannot catch. In the design phase it is also necessary to look ahead to the debug and verification phase and plan how the FPGA will be debugged in-circuit and at-speed. This should lead to the definition of the overall debug approach, help to identify any test and measurement tools required, and identify any impact that the chosen debug method has on the design of the board.

## Debug and Verification Phase

During the debug phase, it is necessary to find the hard problems that were not caught by simulation. Doing this in a time-effective way is a considerable challenge.

## FPGA Debug Methodologies

One of the key choices that needs to be made in the design phase is deciding which FPGA debug methodology to use. Ideally, a methodology is required that is portable to different FPGA designs, providing insight into both FPGA operation and system operation, and giving the designer the power to pinpoint and analyze difficult problems.

There are two basic in-circuit FPGA debug methodologies: the use of an embedded logic analyzer and the use of external test equipment, such as an oscilloscope, mixed signal oscilloscope, or logic analyzer. The choice of which methodology to use depends on the debug needs of the project.

## Embedded Logic Analyzer Core

The major FPGA vendors offer embedded logic analyzer cores. Examples include SignalTap® II from Altera and ChipScope® ILA from Xilinx. These intellectual property blocks are inserted into the FPGA design and provide both triggering capability and storage capability. FPGA logic resources are used to implement the trigger circuit and FPGA memory blocks are used to implement the storage capability. JTAG is used to configure the operation of the core and is used to pass the captured data to a PC for viewing. Because the embedded logic analyzer uses internal FPGA resources, this approach is most often used with larger FPGAs that can better absorb the overhead of the core. Typically the core should take up no more than 5% of the available FPGA logic resources. As with any debug methodology, there are some trade-offs that the designer should be aware of:

- **Number of pins versus internal resources:** Embedded logic analyzer cores use no additional pins since they are accessed via the existing JTAG pins. This means that this method can be used even if the design is pin-constrained. The trade-off is that internal FPGA logic resources and memory blocks are used that could otherwise be used to implement the design. In addition, since internal memory is used to capture the data, memory depths tend to be relatively shallow.
- **Probing versus operating mode:** The probing for an embedded logic analyzer core is simple. It uses the existing JTAG pins, so there is no need to worry about how to connect an external logic analyzer to the system. The trade-off is that, while the embedded logic analyzer provides visibility into the operation of the FPGA, there is no way of correlating that information to board-level or system-level information. The correlation of signals inside the FPGA with those outside the FPGA is often critical to solving the toughest debug challenges.
- **Cost versus flexibility:** Most FPGA vendors offer their embedded logic analyzer cores for less than the cost of a full-functioned external logic analyzer. The trade-off is that embedded logic-analyzer cores offer less functionality than a full-functioned logic analyzer - functionality that is often needed to capture and analyze tough debug challenges. For example, embedded logic analyzers can only operate in state mode - they capture data synchronous to a specified clock that is present in the FPGA design and therefore cannot provide accurate signal timing relationships.

## External Oscilloscope

For users who already have an oscilloscope, the lowest capital-cost FPGA debug technique is to insert some custom debug circuitry into the FPGA.

This will allow designers to:

- select internal signals and route them to external pins
- make special triggering signals (wide parallel triggering, serial triggering or complex multi-state triggering)

Used in conjunction with probing of external signals, this technique allows a designer to correlate the FPGA signals with other analog or digital signals in the system.

If the design is carried out carefully, it is possible to minimize the usage of valuable FPGA logic resources, the usage of valuable FPGA pins, and the effect of the debug circuitry on the performance of the design. However, with this approach it is necessary to redesign and recompile the code each time the experiment is changed, which consumes valuable design time. In addition, the debug circuitry will consume some scarce FPGA gates and some FPGA pins. As a result, with the channel limitations of the oscilloscope and the pin limitation on the FPGA, this technique gives only limited visibility into complex designs.

## External Mixed-signal Oscilloscope

Mixed-signal oscilloscopes (MSOs) offer three key advantages over oscilloscopes in FPGA debug applications:

1. By providing 20 or more channels instead of only 2-4 channels, the MSO allows the user to see many more FPGA (and other) signals at once.
2. Because these channels can also be used for triggering, MSOs are able to identify specific pattern events much better than oscilloscopes.
3. Finally, MSOs allow parallel bus and event table displays of the digital signals, simplifying the interpretation of the digital signals in a complex design.

However, they do not allow the capture of hundreds of signals at once, complex 'if/then/else' triggering or synchronous acquisitions (sampling based on an FPGA clock).

## External Logic Analyzer

Because of some of the limitations of the embedded logic analyzer methodology, many FPGA designers have adopted a methodology that uses the flexibility of the FPGA and the power of an external logic analyzer. In this methodology, internal signals of interest are routed to unused pins of the FPGA, which are then connected to an external logic analyzer. This approach offers very deep memory, which is useful when debugging problems where the symptom and the actual cause are separated by a large amount of time. It also offers the ability to correlate the internal FPGA signals with other activity in the system. As with the embedded logic analyzer methodology, there are trade-offs to consider:

- **Pins versus internal resources:** The external logic analyzer approach uses very few (if any) logic resources and no FPGA memory. This frees these resources to implement the design's functionality. The trade-off is now that a number of incremental pins need to be dedicated to debugging rather than being used by the design.
- **Probing versus operating mode:** The external logic analyzer probing is a little more involved than the probing required for the embedded logic analyzer approach. Rather than being able to reuse the JTAG connector that is already on the board, the designer needs to determine how to probe the FPGA signals with the logic analyzer. The easiest way of doing this is to add a debug connector to the board. This will also enable the FPGA signals to be easily correlated with other signals in the system.
- **Cost versus flexibility:** While it is true that external logic analyzers have an initial higher cost than an embedded logic analyzer, it is possible to solve a wider range of problems with them. Not only is the logic analyzer useful for FPGA debug: it can also be used to solve other digital design challenges. There is also more flexibility in acquisition modes and trigger capability. With an external logic analyzer, it is possible to access up to 16 different trigger states, and data can be captured in timing mode with up to 125 ps resolution.

## Choosing the Right FPGA Debug Methodology

Both methodologies can be useful depending on your situation. The challenge is to determine which approach is appropriate for a particular design.

If the problems are likely to be isolated to functional problems within the FPGA, the use of an embedded logic analyzer may be all the debug capability that is required. If, however, larger system-level debug problems are anticipated that require the verification of timing margins, the correlation of internal FPGA activity with other activity on the board, or more powerful triggering capabilities, the use of external test equipment is more suited to meet debug needs.

For designers who need to look at at-speed timing information in addition to just state data, an external MSO or logic analyzer will show the detailed timing relationships of FPGA signals with up to 60 ps resolution. This helps to verify that events are really happening as they are designed to and allows the timing margins of a design to be verified. An embedded logic analyzer can only capture data synchronous to a specified clock that is present in the FPGA.

Depth of capture is another consideration, since a logic analyzer will offer access to a wider sample depth. In SignalTap II, for instance, the maximum sample depth is set to 128k, which is a device constraint. However, with an external MSO, it is possible to capture up to 10M samples, while with an external logic analyzer the figure goes up to 256M samples. This can help the designer to see more of the problem and its potential cause, thus shortening overall debug time.

A further choice depends on whether the design is more pin-constrained or resource-constrained. Using an embedded logic analyzer requires no additional output pins, but internal FPGA resources must be used to implement the logic analyzer functions. Using an external test instrument requires the use of additional output pins but minimizes (or eliminates) the need to use internal FPGA resources.

To overcome these limitations, a new method of FPGA debug has been created that delivers all of the advantages of the external test equipment approach while removing its primary limitations. First Silicon Solution's FPGAView, used with a Tektronix TLA Series logic analyzer or MSO4000 Series Mixed Signal Oscilloscope, provides a complete solution for debugging FPGAs and their surrounding hardware.

The combination of FPGAView and external test equipment, such as a full-featured mixed-signal oscilloscope or logic analyzer, allows an engineer to see inside the FPGA design and correlate internal signals with external signals. Productivity is increased because the time-consuming process of recompiling the design is eliminated and access is provided to multiple internal signals for each debug pin.

In addition, FPGAView can handle multiple test cores in a single device (useful for monitoring different clock domains) and multiple FPGA devices on a JTAG chain.

Because the FPGA architectures and vendor support packages are very different, the solutions are somewhat different for Xilinx and Altera FPGA support. However, at a high level, each solution consists of four basic parts:

- **Multiplexer:** This is an intellectual property (IP) block that forms a multiplexer. This core is compiled into the FPGA design. Its function is to connect to many internal nodes and route a few of them at a time to a few output pins on the FPGA, where they can be measured with external test equipment. In the case of Xilinx, this IP core is designed by FS2. In the case of Altera, it is a standard part of their development tools.
- **Control software:** This is the PC-based FPGAView control program, written by FS2, that controls both the multiplexer and the external test equipment. Running on either the TLA logic analyzer's embedded PC or an external PC, the program controls which of the multiplexer inputs is routed to the outputs, controls the external test equipment, and displays the signal names on the corresponding signals on the external test equipment.

- **Test equipment:** As indicated above, this can be either the TLA logic analyzer or the MSO4000 mixed-signal oscilloscope.
- **JTAG cable:** This is used by the control software to communicate from the PC to the FPGA to select the appropriate signals to route out.

Using FPGAView is a 4-step process. The first step is to configure the test core and insert it into the design. This is done using the logic analyzer interface editor in Altera's Quartus II software suite.

To customize the test core, the user can specify:

- Pin count: This signifies the number of pins from 1 to 256 that are dedicated to the logic-analyzer interface.
- Bank count: This signifies the number of internal signals (again from 1 to 256) to be mapped to each pin.
- Output/capture mode: This selects the type of acquisition to be performed. Either 'combination/timing' or 'registered/state' can be selected.
- Clock: If a 'state' capture mode is selected, this allows the user to select the sample clock for the test core.
- Power-up state: This parameter is used to specify the power-up state of the pins designated for use with the logic-analyzer interface.

After selecting the appropriate parameters for to meet the debug requirements, the engineer needs to select which pins will be used by the test core for output and which signals are to be probed and by which bank.

The second step is to configure FPGAView by establishing the connection to the JTAG interface and to the external test equipment. This allows the needed flexibility to match the tool to the debug environment. For example, FPGAView fully supports the remote-hosted mode of operation available in the TLA logic analyzers.

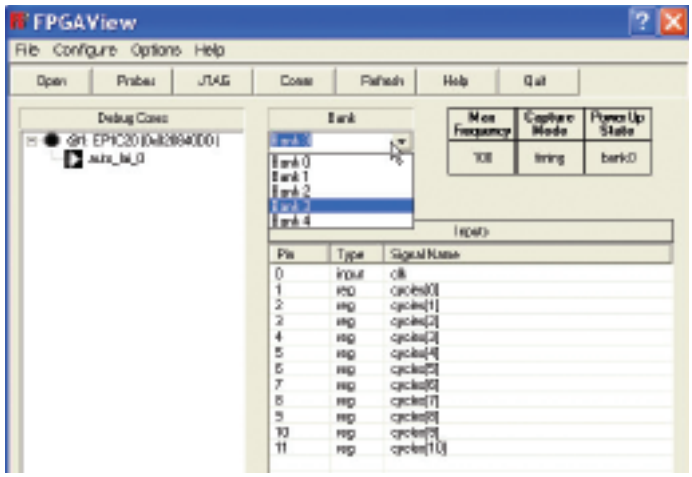



**Figure 3.1.** Using FPGAView to select the state bank.

The next step is to map the physical connection between the FPGA pins and the external test equipment. This will allow FPGAView to automatically update the signal names displayed on the external test equipment to match those of the signals currently being monitored by the test core.

The final step is to make the measurement. The bank list pull-down in FPGAView is used to select which bank is to be measured, and, once the bank is selected, FPGAView communicates to the FPGA via the JTAG interface and configures the test core so that the desired bank is selected. FPGAView also programs the external test equipment with these names into the assigned channels, making it easy to interpret the measurement results. To measure a different set of internal signals, the user simply chooses a different bank of signals. Correlating these FPGA signals with other signals in the system is done automatically by the external test equipment.

Having previously inserted an logic-analyzer interface block into the design, the engineer then uses FPGAView to select the desired bank of signals to monitor.

In Figure 3.1 FPGAView is being used to monitor the state bank, showing the state bits as well as the internal signals

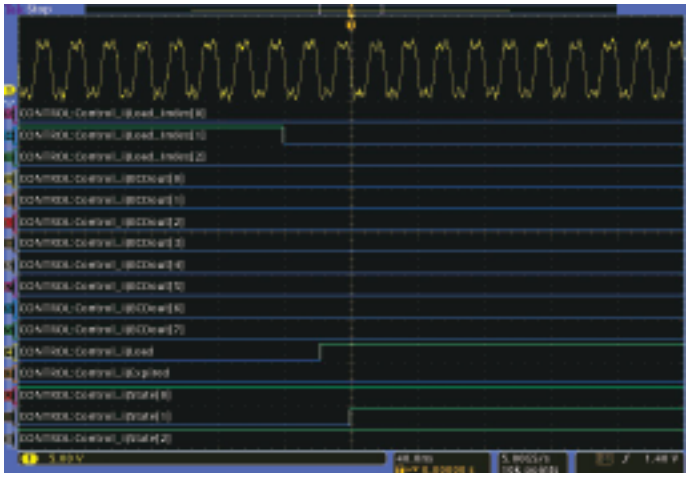


**Figure 3.2.** MSO Series mixed signal oscilloscope simplifies system debug.

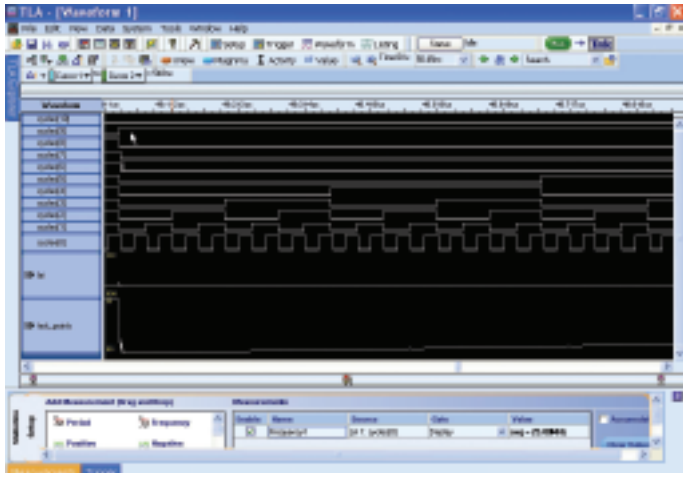


**Figure 3.3.** Examining the operation of the state machine when an input is applied.

associated with the state machine. Probing the 'load' and 'start' buttons of the system gives visibility of the external inputs to the FPGA as well the internal signals.

Setting up the external test equipment to trigger when the state machine exits the idle state makes it possible to examine the operation of the state machine when an input is applied. Correlating these FPGA signals with other signals in your system is then done automatically by the full-featured MSO or TLA as shown in Figure 3.2 and Figure 3.3.

## Summary

Choosing the right FPGA debug methodology can reduce debug and validation time. Embedded logic analyzers and external test equipment each have their own advantages and disadvantages. FPGAView eliminates most of the trade-offs involved in using external test equipment and enables real-time debugging of Xilinx and Altera FPGAs.

It allows design teams to view the internal operation of their FPGA designs and allows correlation of these signals with other board signals. Further benefits in increasing productivity and reducing debugging time result from the ability to change internal probe points in an instant without the need to recompile the design, while being able to monitor multiple internal signals for each debug pin makes the system easier to use and less intrusive than other debug methodologies.

## Memory

There is a continual demand for memories to be larger, faster, lower powered and physically smaller. These needs are the driving force in the advancement of DRAM (dynamic random access memory) technology. Mainstream DRAMs have evolved over the years through several technology enhancements, such as SDRAM (synchronous DRAM), DDR (double data rate) SDRAM, DDR2 (double data rate 2) SDRAM, and DDR3 (double data rate 3) SDRAM. This evolution has also been driven by how computer memories are used on DIMMs (dual-inline memory modules). DIMM implementations have expanded from unregistered DIMMs to include registered DIMMs and FB-DIMMs (fully buffered DIMMs).

Computer memories are not the only systems that continue to demand larger, faster, lower powered and physically smaller memories. Embedded systems applications have similar requirements and can also use DRAMs. However, memory systems are implemented differently in computers versus embedded systems.

Typically, computer memories are mounted on pluggable DIMMs that are easily installed in the computer during assembly. The computer user may upgrade the computer memory by adding or replacing the DIMMs after the computer has been purchased. As a result, memories used in computers require a high level of compatibility with current and future computers, as well as current and future memories used in conjunction with a DIMM.

On the other hand, embedded systems typically use a fixed memory configuration, meaning that the user does not modify the memory system after purchasing the product. The embedded systems manufacturer then has total control over which memories from specific manufacturers are used in the embedded systems product. It is common to optimize an embedded system's performance and cost by using one specific memory from one memory manufacturer.

As a result, it is less important in embedded systems, as compared to computer systems, to have a high level of multi vendor memory interoperability.

New DRAM designs are meeting computer and embedded systems memory requirements to be larger, faster, lower powered and physically smaller. As a result, the following

DRAMs changes are occurring:

- memory size is increasing
- the number of banks is increasing
- burst length is increasing
- supply voltage is decreasing
- logic voltage swings are decreasing
- clock rates are increasing
- data rates are increasing
- memory channels implementations are going from a large number of parallel signals to a reduced number of high-speed serial signals
- the number of memory channels is increasing
- circuit-board density is increasing

These trends are causing designers to use new techniques and tools to design, verify and debug their memory systems.

As memory clock rates increase and logic voltage swings decrease, signal integrity has become more of an issue for reliable memory operation. As result, there are trends for new DRAM features to focus on improving the signal integrity of the memory system. These features include dynamically controlled ODT (on-die termination), OCD (off-chip driver) calibration and fully buffered DIMMs with AMBs (advanced memory buffers).

An advantage of DRAM over other types of memory is its ability to be implemented with fewer circuits per memory cell on the integrated circuit. The DRAM's memory cell is based on storing charge on a capacitor. A typical DRAM cell is built with one capacitor and one or three FETs (field-effect transistors). A typical SRAM (static random access memory) cell takes six FET devices, resulting in fewer memory cells per same IC. SRAMs are simpler to use, easier to interface to and have faster data access times than DRAMs.

A DRAM's core architecture consists of memory cells organized into a two-dimensional array of rows and columns (Figure 4.1).

To access a memory cell requires two steps: first addressing a specific row and then addressing a specific column in the selected row. In other words, an entire row is read internally in the DRAM IC and then the column address selects which column of the row is to be read or to be written to the DRAM I/O (input/output) pins.

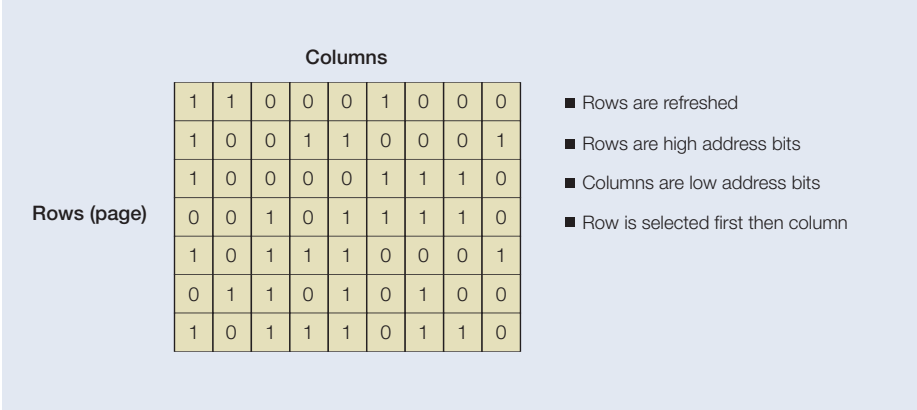



**Figure 4.1.** DRAM memory cells organized into a two-dimensional array of rows and columns.

DRAM reads are destructive, in that the data in the row of memory cells are destroyed in the 'read' operation. The row data therefore need to be written back into the same row after the completion of a 'read' or 'write' operation on that row. This operation is called ' pre-charge', and is the last operation on a row. It must be done before accessing a new row, and is referred to as closing an open row.

A DRAM row is called a memory page, and once the row is opened it is possible to access multiple sequential or different column addresses in that row. This increases memory access speed and reduces memory latency by not having to resend the row address to the DRAM when accessing memory cells in the same memory page. As a result, the row address consists of the computer's higher-order address bits and the column address of the lower-order address bits.

Since the row and column addresses are sent at different times, the row address and the column address are multi-plexed on the same DRAM pins in order to reduce package pin count, cost and size. Typically, the size of the row address is larger than the column address because the power usage is related to the number of columns.

Early DRAMs had control signals such as RAS# (row address select active low) and CAS# (column address select active low) to select the row and column addressing operations being performed. Additional DRAM control signals include WE# (write enable active low) for selecting write or read operation, CS# (chip select active low) for selecting the DRAM and OE# (output enable active low). The early DRAMs had control signals that were asynchronous and had various timing specifications covering their sequence and time relationships to determine the DRAM operating mode.

The read cycle of these early DRAMs had four steps:

1. RAS# goes low with a row address on the address bus.
2. CAS# goes low with a column address on the address bus.
3. OE# goes low and read data appears on DQ data pins. The time from the first step to the third step when data is available on DQ pins is called latency.
4. RAS#, CAS# and OE# go high (inactive) and wait for the internal pre-charge operation to complete restoration of the row data after the destructive read.

The time from the first step to completion of the last step is the memory cycle time. Signal timing of the above signals is related to the sequence of edges, and is asynchronous. There are no synchronous clock operations with these early DRAMs.

The DRAM memory cell needs to refresh to avoid losing its data contents. This requires refresh of the capacitor before it loses its charge. Refreshing memory is the responsibility of the memory controller, and the refresh time specification varies with different DRAM memories. The memory controller performs a refresh by doing a RAS# only cycle with the row address. At the end of the RAS# only cycle is the pre-charge

operation of restoring the row data that was the address in the RAS# only cycle. Typically, the memory controller would have a row counter that would sequentially generate all row addresses that were needed by the RAS# only refresh cycles.

There are two refresh strategies (Figure 4.2). The first strategy is for the memory controller to refresh all rows sequentially in a burst of refresh cycles and then return control of memory back to the processor for normal operation. The next burst of refresh operations occurs before reaching the maximum refresh time. The second refresh strategy is for the memory controller to interleave the refresh cycles with normal processor memory operations. This refresh method spreads out the refresh cycles over the maximum refresh time.

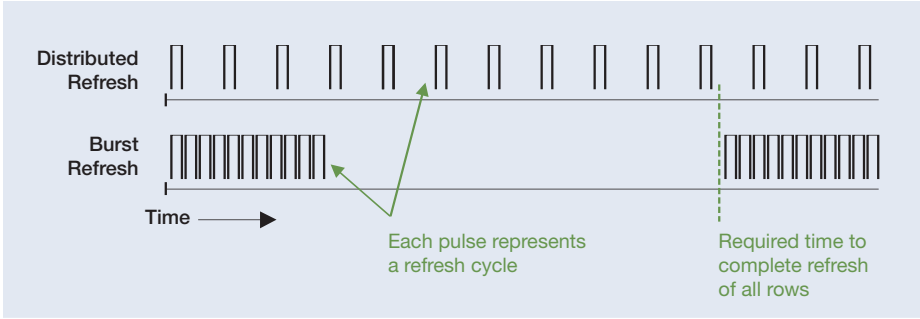



**Figure 4.2.** DRAM refresh implementations include distributed refresh and burst refresh.

Early DRAMs evolved and implemented the refresh counter on the DRAM IC to take care of sequentially generated row addresses. Internally to the DRAM IC, the refresh counter is an input to a multiplexer that controls the memory array row address. The other multiplexer input is from the row address from the external address input pins. This internal refresh counter eliminated the need for an external refresh counter circuit in the memory controller. Some of these DRAMs supported a CAS# before RAS# cycle to initiate a refresh cycle using the internally generated row address.

## SDRAM

The DRAM's asynchronous operation caused many design challenges when interfacing it to a synchronous processor. SDRAM (synchronous DRAM) was designed to synchronize the DRAM operation to the rest of the computer system and to eliminate defining all the different modes of memory operations based on the sequence of CE# (chip enable active low), RAS#, CAS# and WE# edge transitions.

| DDR SDRAM | Data Rate | Memory Clock |
|---|---|---|
| DDR-266 | 266 Mb/s/pin | 133 MHz |
| DDR-333 | 333 Mb/s/pin | 166 MHz |
| DDR-400 | 400 Mb/s/pin | 200 MHz |

**Table 1.** DDR SDRAM data rates and clock speeds

SDRAM added a clock signal and the concept of memory commands. The type of memory command is determined by the state of CE#, RAS#, CAS# and WE# signals at the rising edge of the SDRAM clock. Data sheets describe the memory commands in table form based on the state of CE#, RAS#, CAS# and WE# signals. For example, an 'activate' command sends a row address to the SDRAM to open a row (page) of memory. Next is a sequence of 'de-select' commands to satisfy timing requirements before sending the read or write command with the column address. Once the row (page) of memory is opened with an 'activate' command, several read and write commands can operate on the data in that row (page) of memory. A pre-charge command is required to close the row before another row can open.

## DDR SDRAM

DDR (double data rate) SDRAMs increased the memory data rate performance by increasing clock rates, introducing the bursting of data and transferring two data bits per clock cycle (Table 1).

DDR SDRAMs burst multiple memory locations in a single read or single write command. A read memory operation entails sending an 'activate' command followed by a 'read' command. The memory responds after its latency with a burst of two, four, or eight memory locations at a data rate of two memory locations per clock cycle. Therefore, four memory locations are read from or written to in two consecutive clock cycles.

DDR SDRAMs have multiple banks to provide multiple interleaved memory access, which increases memory bandwidth. A bank is one array of memory; two banks are two arrays of memory; four banks are four arrays of memory and so on (Figure 4.3).

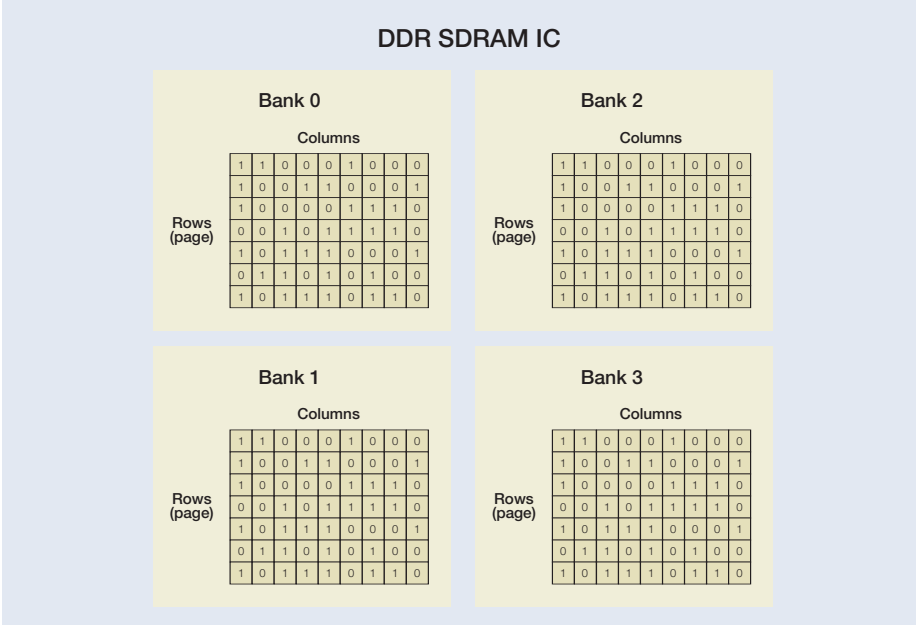



**Figure 4.3.** Multiple memory banks in a DDR SDRAM provide increased access flexibility and improved performance.

Four banks require two bits for bank addressing (BA0 & BA1). For example, a DDR SDRAM with four banks operates in the following manner. First, an 'activate' command opens a row in the first bank. A second 'activate' command opens a row in the second bank. Now any combinations of read or write commands can be sent to either the first bank or the second bank with their open rows. When read and write operations on the bank are completed, a pre-charge command closes the row, and the bank is ready for an 'activate' command to open a new row.

Note that the power required by the DDR SDRAM is related to the number of banks with open rows. More open rows require more power, and larger row sizes require more power. Therefore, for low-power applications one should open only one row at a time in each bank and not have multiple banks each with open rows.

Interleaving consecutive memory words in consecutive memory banks is supported when the bank address bits are connected to the lower-order address bits in the memory system. Consecutive memory words are in the same memory bank when the bank address bits are connected to the higher-order address bits in the memory system.

| DDR2 SDRAM | Data Rate | Memory Clock |
|---|---|---|
| DDR2-400 | 400 Mb/s/pin | 200 MHz |
| DDR2-533 | 533 Mb/s/pin | 266 MHz |
| DDR2-667 | 667 Mb/s/pin | 333 MHz |
| DDR2-800 | 800 Mb/s/pin | 400 MHz |

**Table 2.** DDR2 SDRAM data rates and clock speeds

## DDR2 SDRAM

DDR2 SDRAM has several improvements over DDR SDRAM. DDR2 SDRAM clock rates are higher, thus increasing the memory data rates (Table 2).

Signal integrity becomes more important for reliable memory operation as the clock rates increase. As clock rates increase, signal traces on the circuit boards start to act as transmission lines, and proper layout and termination at the end of the signal traces becomes more important.

Termination of the address, clock and command signals are somewhat straightforward because these signals are unidirectional and are terminated on the circuit boards. The data signals and data strobes are bidirectional. The memory controller hub drives them during a write operation, and the DDR2 SDRAM drives them during a read operation. To add to the complexity, multiple DDR2 SDRAMs are connected to the same data signals and data strobes. These multiple DDR2 SDRAMs can be on the same DIMM and on different DIMMs in the memory system. As a result, the data and data strobe drivers and receivers are constantly changing depending on the read/write operation and on which DDR2 SDRAM is being accessed.

DDR2 SDRAM improves the signal integrity of data signals and data strobes by providing ODT (on-die termination), an ODT signal to enable the on-die termination and the ability to program the on-die termination values (75 ohms, 150 ohms etc.) with the DDR2 SDRAM extended mode register. The on-die termination value and operation are controlled by the memory controller hub, and are a function of a DDR2 SDRAM DIMM's location and type of memory operation (reads or writes). ODT operation results in better signal integrity by creating a larger eye diagram for the 'data valid' window with increased voltage margins, increased slew rates, reduced overshoot and reduced ISI (inter-symbol interference).

DDR2 SDRAM reduces memory system power by operating at 1.8 V, which is 72% of DDR SDRAM's 2.5 V. In some implementations, the number of columns in a row has been reduced, resulting in lower power when a row is activated for read or writes.

Another benefit of the lower operating voltages is the lower logic voltage swings. For the same slew rate, the reduced voltage swings increase logic transition speeds to support faster clock rates. In addition, the data strobe can be programmed to be a differential signal. Using differential data strobe signals reduces noise, crosstalk, dynamic power consumption and EMI (electromagnetic interference) and increases noise margin. Differential or single-end data strobe operation is configured with the DDR2 SDRAM extended mode register.

| DDR3 SDRAM | Data Rate | Memory Clock |
|---|---|---|
| DDR3-800 | 800 Mb/s/pin | 400 MHz |
| DDR3-1066 | 1066Mb/s/pin | 533 MHz |
| DDR3-1333 | 1333Mb/s/pin | 667 MHz |
| DDR3-1600 | 1600 Mb/s/pin | 800 MHz |

**Table 3.** Expected DDR3 SDRAM data rates and clock speeds

A new feature introduced with DDR2 SDRAM is additive latency, which provides the memory controller hub with the flexibility to send the read and write commands sooner after the 'activate' command. This optimizes memory throughput and is configured by programming the additional latency using the DDR2 SDRAM extended mode register.

DDR2 SDRAM improves data bandwidth of 1 Gbit and 2 Gbit memories by using eight banks. The eight banks increase the flexibility of accessing large memory DDR2 SDRAMs by interleaving different memory bank operations. In addition, for large memories, DDR2 SDRAM supports a burst length of up to eight.

## DDR3 SDRAM

DDR3 SDRAM is a performance evolution beyond DDR2 SDRAM. DDR3 SDRAMs will support the next level of faster data rates and clock speeds (Table 3).

Other expected changes include reducing the DDR3 SDRAM operating voltage to 1.5 V, which is 83% of DDR2 SDRAM's 1.8 V. DDR3 SDRAM is the memory that will be used by FB-DIMM2 (fully buffered DIMM2) implementations. DDR3 SDRAM specifications are under development and are subject to change until they are approved by JEDEC.

## Memory System Design

The first few steps of product design are product requirements, product architecture design and subsystem design. One of the subsystem designs is the memory system. Memory system design is dependent on memory size, speed, power, existing standards, new developing standards, reuse of existing designs and other requirements.

The computer chipset manufacturers heavily influence memory system designs for computers. Some of these computer chipset manufacturers have their own testing procedures, validation processes and workshops to test products. Typically, these computer chipset manufacturers' web sites list memory products that have passed their compatibility testing.

## Design Simulation

A key part of memory system design is design simulation. The importance of comprehensive memory system design simulation cannot be understated. Experience has shown that a resistor value change of only a few ohms can have a significant impact on having a reliable operating memory system.

Memory system design simulation should include the effects of probing loading caused by any instrument when it is connected to the prototype memory system. The verification and debugging process will be very difficult if the prototype stops operating because of probe loading. Also, simulation should analyze the signals at probe test points with the loading of the instrument probe. The 'data valid' window will change along the signal trace from the memory controller hub driver to the SDRAM pins.

Probing test points should be as close as possible to the receiver pins so that the instrument shows the signal that the receiver is seeing. Sometimes this is not possible and interposers, test adapter boards and other special probing fixtures and aids are used to retrieve difficult to access signals. These probing aids should also be included in design simulations to understand their effect on the SDRAM signals and the measurement of the signals.

## Design Verification

Using new DRAM features in designs requires new design methods and techniques, which range from new techniques in design simulation to new BIOS operation. As a result, DRAM design implementations require complete verification and testing, ranging from circuit board construction to software operation to ensure reliable memory operation.

Product reliability will suffer if a memory system has infrequent random errors because of a design implementation that has not been fully verified. In addition, the customer may require a product to satisfy various compliance testing requirements that have been defined by JEDEC or by other manufacturers.

## Verification Strategy

It is important to have a strategy for effectively and quickly debugging design problems in any design implementation. Quick time-to-market product development requires verification/debugging planning early in the design. This plan should identify the following requirements:

- which design elements are new and which are reused design elements
- what to avoid and change based on past designs
- what level of validation and testing is needed, and whether the testing require special operating modes or signal patterns
- what special design-in features are needed (e.g. probing test points or test fixtures); has simulation analysis accounted for probing the prototype; are signal stimuli needed; is special software needed to exercise the hardware
- what environmental tests are needed (e.g. temperature, humidity etc.)
- what circuit operational visibility is necessary in order to debug it
- what regulatory compliance testing is required; will the validation/debug test points be used to test the product in manufacturing; will the validation/debug test points be used to repair the product in service; and how do engineers manage the risk of what they do not know today

| Verification | Tasks | Instruments |
|---|---|---|
| **Circuit Board Construction** | Single-ended trace impedances | Sampling Oscilloscope with TDR |
| | Differential trace impedances | Sampling Oscilloscope with TDR |
| | Trace lengths | Sampling Oscilloscope with TDR |
| | Crosstalk | Sampling Oscilloscope with TDT |
| **Electrical Power & Signals** | Power supply quality, noise, glitches & ground bounce | Oscilloscope |
| | Clock signal quality, rise & fall times/slew rates, spread spectrum clocking profile | Oscilloscope with jitter analysis software |
| | Command, address & data valid windows, clock, strobes & data signal skew | Oscilloscope with jitter analysis software |
| | FB-DIMM serial signals data valid windows | Oscilloscope with serial data compliance and analysis software, Signal Sources & FB-DIMM fixtures |
| **Protocols Sequences & Timing** | Memory system power up initialization protocols & timing | Logic analyzer with SDRAM support packages |
| | SDRAM mode register operation | Logic analyzer with SDRAM support packages |
| | SDRAM command protocols & timing | Logic analyzer with SDRAM support packages |
| | Read/Write data valid windows | Logic analyzer with SDRAM support packages |
| | Refresh operations | Logic analyzer with SDRAM support packages |
| | Memory channel traffic | Logic analyzer with FB-DIMM support packages |

**Table 4.** Verification tasks with associated test equipment.

For example, some verification strategies include building a validation prototype with numerous probing test points to verify the new system architecture with new ASICs/FPGAs. It is best that the validation prototype operates at full speed to verify at-speed operation and performance. Complex designs require more comprehensive visibility of their real-time operation in order to pinpoint problems quickly. Once the validation prototype is running correctly and has completed validation, the final prototype is implemented with reduced test points.

## SDRAM Verification

DRAM verification and testing techniques depend upon what is being designed. DRAM designs can be computer memory controller hub ICs, memory ICs, AMB ICs, DIMMs, computer motherboards and embedded systems. Each of these products requires different validation strategies, different validation tests and different test equipment. For example, a memory IC designer will not be verifying circuit board construction, whereas the DIMM designer will be verifying the DIMM circuit board construction. The memory controller is typically designed by the embedded systems designer because of its unique requirements to work with a specific processor and unique embedded system input/output configuration. As a result, a significant part of the design work is designing the memory controller and designing the circuit board layout between the memory controller and the memory ICs. Verifying this part of the design is critical for reliable operation.

DRAM verification and testing requires a range of test and measurement equipment such as sampling oscilloscopes, oscilloscopes, logic analyzers, probes, test fixtures, analysis software and compliance software (Table 4).

Test equipment needs to provide unobtrusive probing, precise acquisition and complete system visibility of electrical signals and protocol layers. To help designers quickly verify memory operation, powerful analysis capability is also necessary.

Monitoring a computer system or embedded system with a logic analyzer creates a powerful verification and debugging development environment. The logic analyzer is used to trace and correlate the processor bus activity, the memory activity and the input/output operations. Complete system visibility on the logic analyzer display provides critical design insight into real-time system operation. In addition, using integrated oscilloscope and logic analyzer probing, triggering and display provides complete design visibility from the software listing, the protocol listing, the digital waveforms and the analog waveforms on the same display. The result is a powerful, comprehensive and efficient analysis of the prototype.

Tektronix offers a comprehensive tool set including industry-leading oscilloscopes, true differential TDRs, and logic analyzers with Nexus Technology memory supports to enable embedded and computer designers to perform quick and accurate electrical testing and operational validation of their memory designs. Collectively, this tool set provides superior performance with unparalleled ease of use, making it an ideal solution for embedded systems and computer memory systems verification and debugging.

# Low-speed Serial Buses

In a serial bus, all the information is sent serially on the same few conductors (or sometimes just one). This means that a single signal may include address, control, data and clock information.

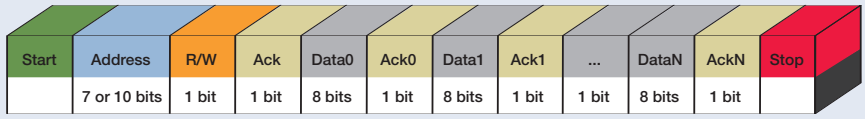



**Figure 5.1.** I²C message structure.

## Common low-speed serial buses include RS-232, USB, FlexRay, I²C, SPI, CAN, LIN, and many others.

In this section we will explore a few of these in more depth.

**I²C (Inter-Integrated Circuit):** The I²C bus was originally developed by Philips in the early 1980s to provide a low-cost way of connecting controllers to peripheral chips in TV sets, but has since evolved into a worldwide standard for communications between devices in embedded systems. The simple two-wire design has found its way into a wide variety of chips like I/O devices, A/D and D/A converters, temperature sensors, microcontrollers and microprocessors.

The I²C physical two-wire interface is comprised of bi-directional serial clock (SCL) and data (SDA) lines. I²C supports multiple masters and slaves on the bus, but only one master may be active at any one time. Any I²C device can be attached to the bus, allowing any master device to exchange information with a slave device.

Each device is recognized by a unique address and can operate as either a transmitter or receiver, depending on the function of the device. Initially, I²C only used 7-bit addresses, but evolved over time to allow 10-bit addressing as well. Three bit rates are supported: 100 kbit/s (standard mode), 400 kbit/s (fast mode), and 3.4 Mbit/s (high-speed mode). The maximum number of devices is determined by a maximum capacitance of 400 pF: equivalent to roughly 20-30 devices.

The I²C standard specifies the following format, as shown in Figure 5.1:

- **Start -** indicates that the device is taking control of the bus and that a message will follow
- **Address -** a 7- or 10-bit number representing the address of the device that will either be read from or written to
- **R/W -** one bit indicating if the data will be read from or written to the device
- **Ack -** one bit from the slave device acknowledging the master's actions. Usually each address and data byte has an acknowledge, but not always
- **Data -** an integer number of bytes read or written to the device
- **Stop -** indicates that the message is complete and the master has released the bus

**SPI (Serial Peripheral Interface):**
The SPI bus was originally developed by Motorola in the late 1980s for the 68000 Series of micro-controllers. Owing to the simplicity and popularity of the bus, many other manufacturers have adopted the standard over the years. It is now found in a broad array of components commonly used in embedded system design. SPI is primarily used between microcontrollers and their immediate peripheral devices. It is commonly found in cell phones, PDAs, and other mobile devices to communicate data between the CPU, keyboard, display, and memory chips.

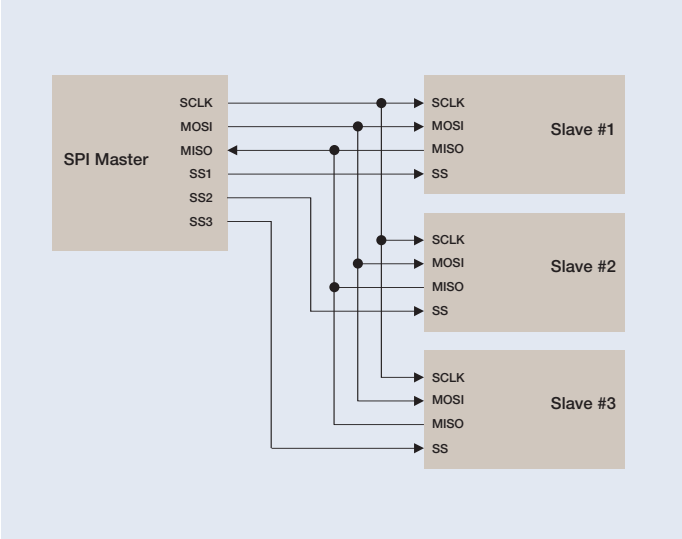



**Figure 5.2.** Common SPI configuration.

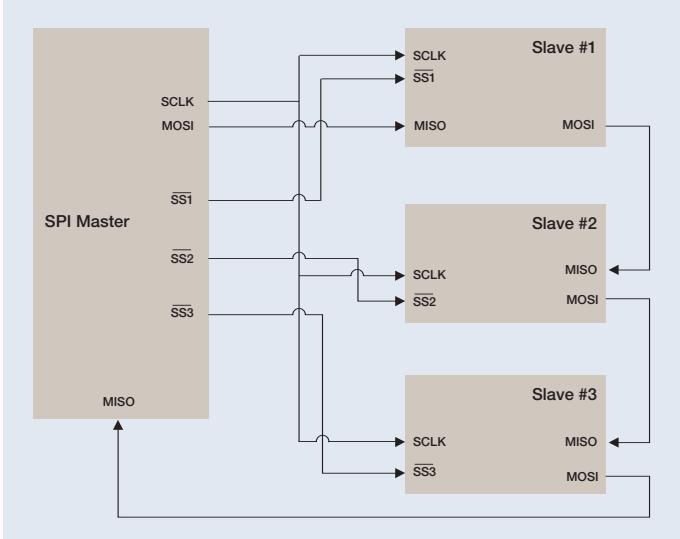



**Figure 5.3.** Daisy-chained SPI configuration.

The SPI bus is a master/slave, 4-wire serial communications bus. The four signals are clock (SCLK), master output/slave input (MOSI), master input/slave output (MISO), and slave select (SS). Whenever two devices communicate, one is referred to as the 'master' and the other as the 'slave'. The master drives the serial clock. Data is simultaneously transmitted and received, making it a full-duplex protocol. Rather than having unique addresses for each device on the bus, SPI uses the SS line to specify which device data is being transferred to or from. As such, each unique device on the bus needs its own SS signal from the master. If there are three slave devices, there are three SS leads from the master: one to each slave, as shown in Figure 5.2.

In this illustration, each slave only talks to the master. However, SPI can be wired with the slave devices daisy-chained, each performing an operation in turn, and then sending the results back to the master, as shown in Figure 5.3.

As this scenario shows, there is no 'standard' for SPI implementation. In some cases, where communication from the slave back to the master is not required, the MISO signal may be left out altogether.

When an SPI data transfer occurs, an 8-bit data word is shifted out MOSI, while a different 8-bit data word is being shifted in on MISO. This can be viewed as a 16-bit circular shift register. When a transfer occurs, this 16-bit shift register is shifted by eight positions, thus exchanging the 8-bit data between the master and slave devices. A pair of registers, clock polarity (CPOL) and clock phase (CPHA), determine the edges of the clock on which the data is driven. Each register has two possible states, which allows for four possible combinations, all of which are incompatible with one another.

This means that a master/slave pair must use the same parameter values to communicate. If multiple slaves are used that are fixed in different configurations, the master will have to reconfigure itself each time it needs to communicate with a different slave.

**CAN (Controller Area Network):**
The CAN bus was originally developed in the 1980s by Robert Bosch GmbH as a low-cost communications bus between devices in electrically noisy environments. Mercedes-Benz became the first automobile manufacturer in 1992 to employ CAN in their automotive systems. Today almost every automobile manufacturer uses CAN controllers and networks to control devices such as: windshield wiper motor controllers, rain sensors, airbags, door locks, engine timing controls, anti-lock braking systems, power train controls and electric windows. Owing to its electrical noise tolerance, minimal wiring, excellent error-detection capabilities and high-speed data transfer, CAN is rapidly expanding into other applications such as industrial control, marine, medical, aerospace and more.

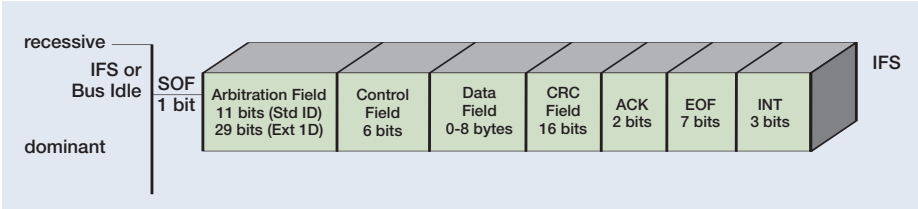



**Figure 5.4.** CAN data/remote frame.

The CAN bus is a balanced (differential) 2-wire interface running over a shielded twisted pair (STP), unshielded twisted pair (UTP) or ribbon cable. Each node uses a male 9-pin D connector. Non-return-to-zero (NRZ) bit encoding is used, with bit stuffing to ensure compact messages with a minimum number of transitions and high noise immunity. The CAN bus interface uses an asynchronous transmission scheme in which any node may begin transmitting any time the bus is free. Messages are broadcast to all nodes on the network.

In cases where multiple nodes initiate messages at the same time, bitwise arbitration is used to determine which message is higher priority. Messages can be one of four types: data frame, remote transmission request (RTR) frame, error frame, or overload frame.

Any node on the bus that detects an error transmits an error frame which causes all nodes on the bus to view the current message as incomplete and tells the transmitting node to resend the message. Overload frames are initiated by receiving devices to indicate they are not ready to receive data yet. Data frames are used to transmit data, while remote frames request data. Data and remote frames are controlled by start and stop bits at the beginning and end of each frame, and include the following fields: arbitration field, control field; data field, CRC field, and an ACK field, as shown in Figure 5.4.

The frame begins with a start of frame (SOF) bit. The arbitration field includes an identifier (address) and the remote transmission request (RTR) bit used to distinguish between a data frame and a data request frame, also called a remote frame. The identifier can either be standard format (11 bits - version 2.0A) or extended format (29 bits - version 2.0B).

The control field consists of six bits including the identifier extension (IDE) bit, which distinguishes between a CAN 2.0A (11-bit identifier) standard frame and a CAN 2.0B (29-bit identifier) extended frame. The control field also includes the data length code (DLC), which is a 4-bit indication of the number of bytes in the data field of a data frame or the number of bytes being requested by a remote frame.The data field consists of zero to eight bytes of data, followed by a 15-bit cyclic redundancy check (CRC) code and a recessive delimiter bit.

The acknowledge (ACK) field is two bits long. The first is the slot bit, transmitted as recessive, but then overwritten by dominant bits transmitted from any node that successfully receives the transmitted message. The second bit is a recessive delimiter bit. Seven recessive bits indicate the end of frame (EOF). The intermission (INT) field of three recessive bits indicates that the bus is free. Bus idle time may be any arbitrary length, including zero.

A number of different data rates are defined, with 1 Mbit/s being the fastest, and 5 kbit/s the minimum rate. All modules must support at least 20 kbit/s. Cable length depends on the data rate used. Normally, all devices in a system transfer information at uniform and fixed bit rates. The maximum line length can be thousands of meters at low speeds; 40 meters at 1 Mbit/s is typical. Termination resistors are used at each end of the cable.

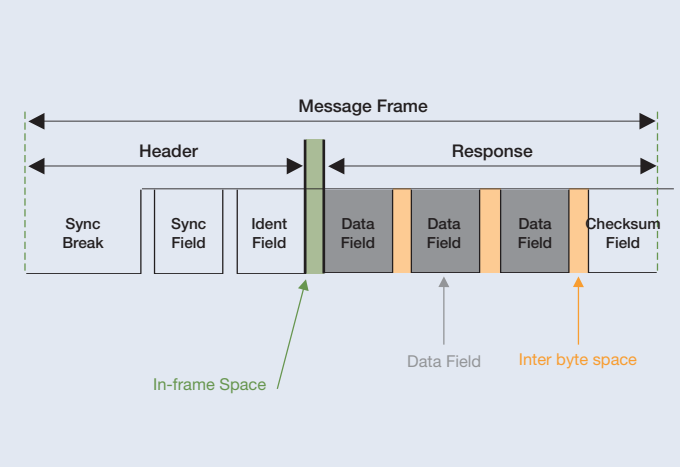



**Figure 5.5.** LIN message format.

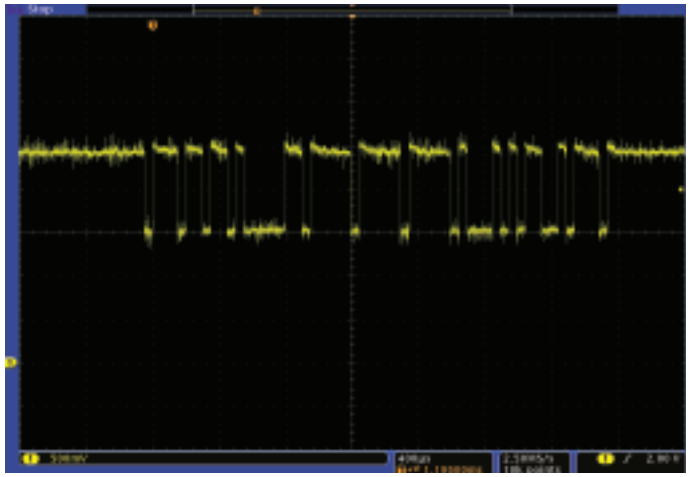


**Figure 5.6.** One message acquired from a CANbus.

**LIN (Local Interconnect Network):** The LIN bus is a single-wire serial communications protocol based on the common UART interface. LIN uses single-master/multiple-slave networking architecture, and was developed for automotive sensor and actuator networking applications for door control, lights and window operation. As with CAN, the LIN master node connects the LIN network with higher-level networks. LIN supports a maximum of 20 kbit/s data rate, resulting from the requirements for electromagnetic interference and clock synchronization.

A slave task is activated upon reception and filtering of the identifier, and starts the transmission of the message response. The response is comprised of two, four or eight data bytes and one check-sum byte. The header and response part form one message format. A LIN message format is shown in Figure 5.5.

## Measurement Challenges

An illustration of the measurement challenges posed by low-speed serial buses is shown by the CAN bus waveform in Figure 5.6.

This message contains a start of frame, an identifier (address), a data length code, data, CRC, and end of frame as well as a few other control bits. To further complicate matters, the clock is embedded in the data, and 'bit stuffing' is used to ensure an adequate number of edges for the receiving device to lock to the clock. Even to the very trained eye, it would be extremely difficult to quickly interpret the content of this message.

Even with a simpler serial standard such as $I^2C$, it is still significantly harder to observe what is being transmitted over the bus than it is with a parallel protocol. $I^2C$ uses separate clock and data lines, so at least in this case the clock can be used as a reference point. However, there is still a need to find the start of the message (data going low while the clock is high), which would traditionally involve manually inspecting and writing down the data value on every rising edge of the clock, and then organizing the bits into the message structure. It can easily take a couple of minutes of work just to decode a single message in a long acquisition, and the user will still have no idea if that is the message being looked for. If it is not, then this tedious and error-prone process has to be started over again on the next message.

It would be nice to just trigger on the message content that the user is looking for. However, the state and pattern triggers traditionally used on oscilloscopes and logic analyzers will not do any good here. They are designed to look at a pattern occurring at the same time across multiple channels. To work on a serial bus, their trigger engines would need to be tens to hundreds of states deep (one state per bit). Even if this trigger capability existed, it would not be an easy task programming it state-by-state for all these bits.

Fortunately, help is at hand in the form of next-generation serial and triggering analysis tools that are designed to work with the latest Tektronix oscilloscopes.

**Figure 5.7.** Analysis of low-speed serial buses.


**Figure 5.8.** I²C bus setup menu.

The system shown in Figure 5.7 allows the user to define inputs to the oscilloscope as a bus using the setup menu shown in Figure 5.8. Although this example refers to I²C, similar capabilities are available for many other low-speed serial buses.

By simply defining which channels clock and data are on, along with the thresholds used to determine logic 'ones' and 'zeroes', the oscilloscope is able to understand the protocol being transmitted across the bus. With this knowledge, the oscilloscope can trigger on any specified message-level information and then decode the resulting acquisition into meaningful, easily interpreted results. Gone are the days of edge triggering, hoping that the event of interest has been acquired, and then manually decoding message after message looking for the problem.

These triggers allow the user to isolate the particular bus traffic of interest, while the decoding capability offers the ability to instantly see the content of every message transmitted over the bus in the acquisition.

## Triggering Versus Search

As emphasized earlier, a capable triggering system is required to isolate the event of interest on the serial bus. However, once the data are acquired (i.e. the oscilloscope is stopped), and analysis is required, triggering does not apply any more. What users need is an oscilloscope with trigger-like resources for analyzing acquired waveform data.

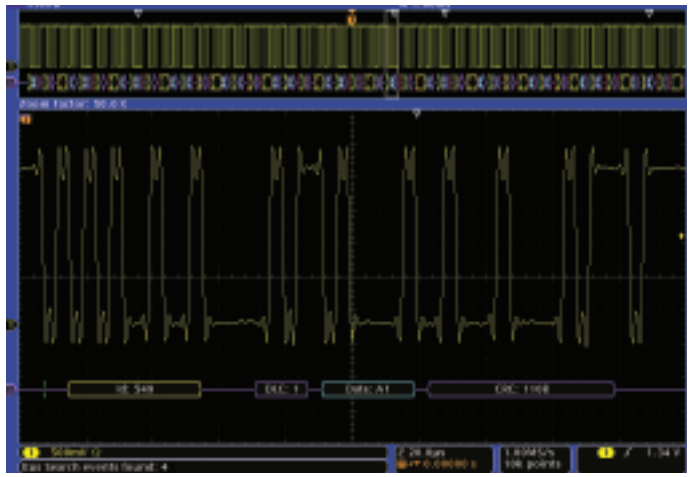


**Figure 5.9.** Searching on specified Identifier and Data in a CAN bus acquisition.

This capability is provided by a feature known as Wave Inspector®, which incorporates powerful search features. All the bus trigger capabilities discussed above are also available as search criteria on already acquired data. For example, in Figure 5.9 the oscilloscope has searched through a long acquisition for every CAN message that has specific address and data content, and marked each one with a hollow white triangle at the top of the display. Navigating between occurrences is as simple as pressing the front panel 'previous' and 'next' buttons.

Searches are also available for the more traditional trigger types, including edges, pulse widths, runt, setup & hold times, logic and rise/fall times.

## Summary

While there are many benefits in transitioning from parallel to low-speed serial buses in embedded systems design, there are also a number of challenges that the design engineer faces. With traditional test and measurement tools it is much more difficult to trigger on the event that the user is looking for. It can be nearly impossible to tell what information is present by just looking at the analog signal, and it is an extremely time-consuming and error-prone process to have to manually decode a long period of bus activity to diagnose problems. Tektronix oscilloscopes equipped with the appropriate triggering and analysis tools change everything: with their powerful trigger, decode and search capabilities, today's design engineers can solve embedded system design issues with exceptional efficiency.

## High-speed Serial Buses

At the other end of the spectrum from low-speed serial buses, the emergence of high-speed buses has largely been driven by increases in digital information and new entertainment options, which are changing consumers' expectations for a more satisfying and consistent real-time experience. This real-time experience has been enabled by exponential growth in computing performance, but is incomplete without providing equally fast access to content and data, no matter what the source may be. Faster delivery and response along with easier connectivity have created new technical requirements in capacity and speed. Servicing this need has established greater demands for increasing data exchanges at ever increasing bandwidths.

Several new serial data bus architectures, including PCI-Express, XAUI, RapidIO, HDMI and SATA, provide order-of-magnitude greater data throughput than possible only a few years ago.

To ensure interoperability through all phases of product development, standardization is required. Leading technology companies have already developed products using 2.5 Gbit/s and 3 Gbit/s speeds; 5 Gbit/s technologies are on the way, and 10 Gbit/s is already in use for network communications. The following section outlines the high-speed serial bus standards that are most relevant in the embedded systems environment:

- **SATA (Serial ATA):** SATA is the serial bus technology that replaces parallel ATA for peripheral storage devices. It offers scalable performance starting at 1.5 Gbit/s and moving to 3 Gbit/s.
- **HDMI (High Definition Multimedia Interface):** The latest consumer electronics technology is enabling multi-media applications. Using the earlier DVI (Digital Visual Interface) as a starting point, HDMI provides high-speed transport of digital video and audio, and operates from 250 Mbit/s to 1.65 Gbit/s. The latest HDMI version 1.3 extends data transfers rates to 3.4 Gbit/s.
- **PCI-Express:** This serial technology is replacing legacy parallel PCI/PCI-X buses used across PC and server applications. PCI-Express 1.0 is a multilane 2.5 Gbit/s serial interface addressing performance applications including high-speed graphics and imaging. Gen2 (PCI-Express 2.0) will double that performance to 5.0 Gbit/s.
- **Ethernet:** The most pervasive networking technology has evolved from operating at moderate speeds of 10 Mbit/s (10 BaseT) and 100 Mbit/s (100 BaseTX) to Gigabit Ethernet (GbE) with four lanes at 250 Mbit/s (1000 BaseT) and 10 Gbit/s. At 10 Gbit/s, four XAUI electrical lanes are operating at 3.125 Gbit/s.

What all these latest standards have in common are faster edge rates and narrower data pulses, which combine to create unique, exacting demands on designers. As multi-gigabit data rates become common in digital systems, signal integrity - the quality of the signal necessary for proper operation of an integrated circuit - is becoming a paramount concern for designers. One bad bit in the data stream can have a dramatic impact on the outcome of an instruction or transaction. Factors that can cause impairments in the transmitted signal quality include:

- **Gigabit signal speeds:** Ultra fast transfer rates, low-voltage differential signals, and multi-level signaling are more susceptible to signal integrity issues, differential skew, noise and analog interference. Serial buses can be implemented as single lanes and also as multiple-lane architectures for increased data throughput, adding to overall design complexity and potential for lane skew timing violations.
- **Jitter:** With high data rates and embedded clocks, modern serial devices can be susceptible to jitter, which creates transmission errors and degrades bit-error-rate performance. Jitter is the deviation from ideal timing of an event, and typically comes from crosstalk, system noise, simultaneous switching outputs and other regularly occurring interference signals.

- **Transmission-line effects:** A transmission line is a medium through which energy and signals are transmitted. It can consist of simple passive circuit elements like wires, cables, and chip printed-circuit-board (PCB) interconnections. With serial data technologies, the signal transmitter, transmission line, and receiver constitute a serial data network. Transmission effects such as reflections and impedance discontinuities can significantly impact signal quality and lead to transmission errors.
- **Noise:** Noise is any unwanted signal that appears in the sampled data. Noise comes from both external sources, such as the AC power line, and internal sources, such as digital clocks, microprocessors, and switched-mode power supplies. Noise can be transient in nature or be broadband random noise and can lead to jitter and signal integrity problems.

Higher-speed digital signals with embedded clocks have characteristics that appear more and more analog-like, making design validation and system integration ever more challenging. Demand for precise validation, characterization and stress testing under a wide variety of conditions adds to the challenge with signals that can become unreliable under even a small amount of distortion or jitter.

## High-speed Serial Standards

Driven by consumer demand, design engineers constantly need to integrate new functionality into their designs by incorporating leading-edge technology. To aid these engineers, the design and operation of each of the serial data technologies is defined in a unique standards document, usually the product of an industry committee. Each standard also requires specialized measurements and compliance testing procedures, which add engineering work beyond the already complex design task. These standards address areas such as data signaling and encoding, packetization, clock embedding, transmission properties, and compliance test procedures. With clear standards and test procedures for high-speed serial technologies, designers are able to create products that provide off-the-shelf interoperability.

Typical high-speed serial data test phases include:

- **Design verification:** Design and test engineers need to ensure that the real-world operation of their designs meets the design simulation and technical performance specifications, and that they are fully functional. This requires complete characterization, debug and analysis under realistic and limited stress conditions.
- **Characterization:** During the initial testing of a component, designers will characterize performance to determine if the signal behavior is within specification. A host of precise measurements needs to be made, often in real time, to determine if the high-speed buses are working properly. Characterization typically includes measuring rise and fall times, edge-to-edge timing, jitter and jitter tolerances, signal path skew, bus settling times, and data-path variations.
- **Troubleshooting:** Any discovered characterization or operational issues require engineers to debug and isolate failures or anomaly conditions such as hardware timing, crosstalk, signal quality and/or software design problems.
- **Compliance test:** Ensuring ‘plug and play’ compatibility between multiple vendors and products requires confirmation of final designs to industry-specific serial data standards. Compliance testing can be a complex and potentially time-consuming step to ensure interoperability, and usually requires advanced testing and analysis capabilities. Serial standards normally include extensive amplitude, timing, jitter, impedance and eye-diagram measurements within their compliance testing specifications.

Engineers will often need to fully evaluate performance at both the chip and system level. At gigahertz speeds, there is greater susceptibility to timing problems in the form of jitter, impedance discontinuities between a transmitter and receiver, or system-level interaction between hardware and software. Comprehensive analysis tools are required at all design phases to provide deeper insight and to fully characterize and isolate marginal design or system effects.

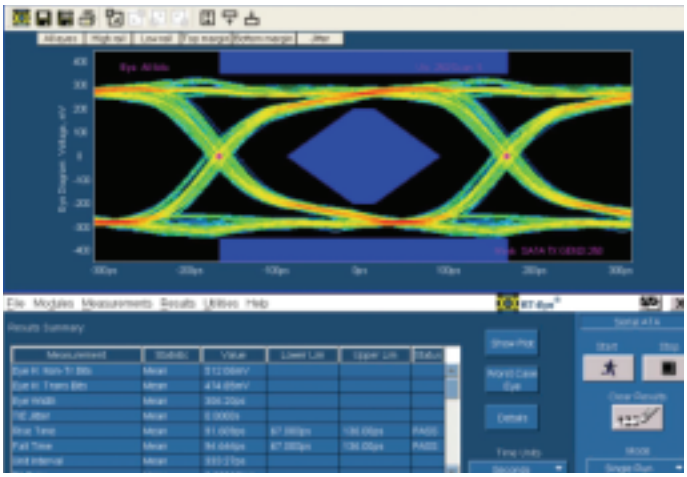


**Figure 6.1.** Eye measurements on an oscilloscope.

## Measurement Challenges

Some of today's most difficult serial measurement challenges range from producing eye diagrams quickly and effectively to capturing impedance characteristics to inform SPICE models. Consider these examples:

- **Eye diagram analysis software:** Software tools running integrally on proven measurement platforms can add efficiency to the serial measurement process and adapt as standards change. Consequently, software is a key battleground in the competition to solve compliance measurement problems of all kinds. Current applications, such as the on shown in Figure 6.1, use plug-ins to optimize the broader toolset to the needs of a particular standard. This approach enables users to stay in step with changes in the standards. For example, emerging PCI Express Gen II measurement specifications require the removal of de-emphasis effects from the data stream before making jitter measurements. This requirement is not presently incorporated in all standards, although it may get ported to other serial bus standards eventually. This situation is tailor-made for a plug-in solution that can be adapted to meet the new specification without affecting the measurement application as a whole.

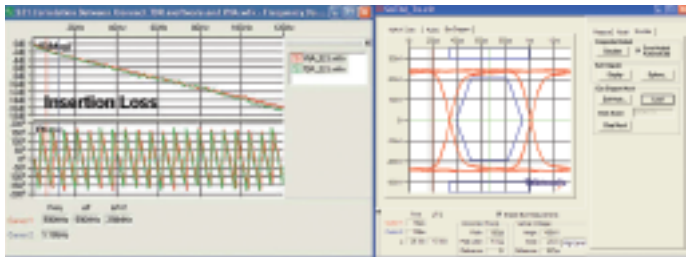



**Figure 6.2.** TDR-based transmission pathway measurements.

- **Transmission models in the development of compliant serial devices:** In a perfect world, aberration-free differential serial data signals would travel through noiseless transmission channels and arrive intact at the receiver. In the real world, however, things are very different. Signals can get badly degraded by high-frequency losses, crosstalk, and other effects. Increasingly, serial measurement procedures are taking this into account. One innovative approach for serial data network analysis (SDNA) applications uses TDR-based S-parameter measurement tools which are emerging as an efficient, cost-effective solution. The tools offer the performance to support uncompromised SDNA measurements with ample dynamic range for serial applications. Moreover, these platforms have gained a host of software tools to speed and simplify SDNA work. SDNA requires time and frequency-domain characterization of the interconnect link using TDR measurements and uses the data to support SPICE modeling, eye-diagram analysis, and a wide range of impedance parameters. Current tools, such as those shown in Figure 6.2, can run on a sampling oscilloscope, alongside eye diagram and jitter analysis applications, to provide a comprehensive tool set for multi-gigabit differential serial data link characterization and compliance verification.

- **Solving a jitter measurement problem:** Serial FB-DIMM (fully buffered double inline memory module) signals are notorious for their tendency to accumulate noise and crosstalk, with jitter as the consequence. Until now it has been very difficult to isolate the data jitter components in the midst of these degraded signals. A new technique, best described as 'common-mode jitter de-embedding', offers an effective solution for FB-DIMM jitter testing using a real-time oscilloscope. FB-DIMM architecture maintains a reference clock channel that is separate from the data channel. Both these paths are influenced by the same board losses; both exhibit essentially the same amount of noise and crosstalk. But the data channel also has some jitter contributed by the transmitter. This is obscured by the noise and signal degradation but nevertheless has the effect of increasing the channel's bit error rate. Figure 6.3 shows one example of multiple views for jitter analysis. The common-mode jitter de-embedding technique acquires both channels and finds the differences. Since both channels carry the same noise and crosstalk, the difference between the clock channel and the data channel is the jitter value. Common-mode jitter de-embedding has been proposed to the appropriate working groups, and is currently being evaluated.

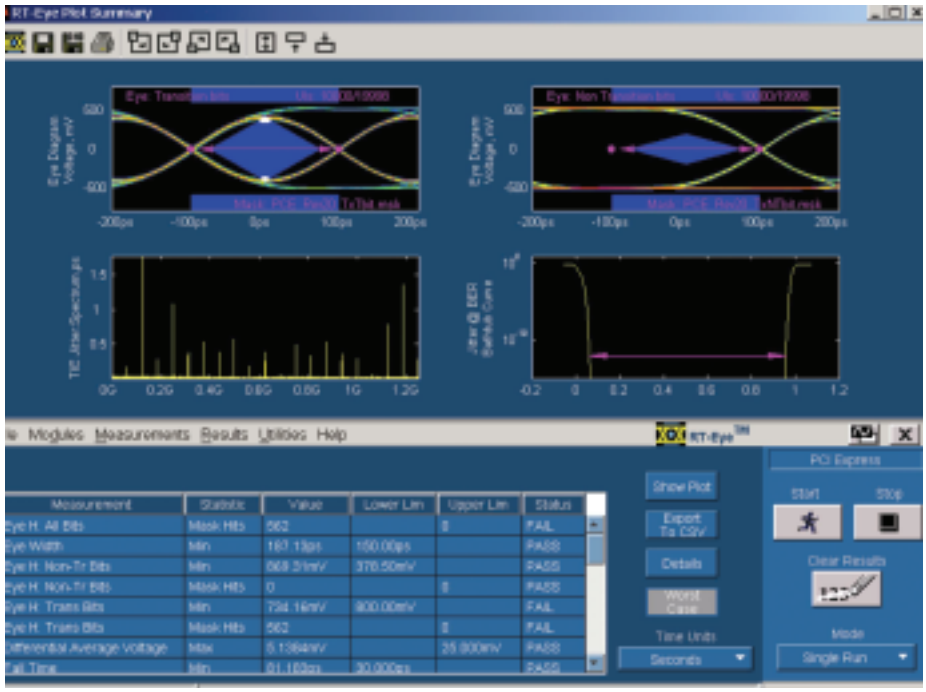


**Figure 6.3.** Multiple views of jitter analysis.

- **Virtual test points to reveal hidden signals:** Observability is a challenge when measuring serial receiver performance. The receiver input in a serial device is an almost meaningless access point for viewing signals. This is because the serial receiver itself processes the input signal through a built-in decision feedback equalization (DFE) filter designed to offset the degradation that occurs during transmission through cables, PCB traces, and connectors. The signal that goes into the active portion of the receiver - where eye diagrams and other characteristics must be evaluated - is encapsulated inside the device and is thus inaccessible. Some oscilloscopes now include built-in finite impulse response (FIR) filters to mimic the effect of the receiver's DFE filter. The user can load the same coefficients into the oscilloscope as were used to design the filter in the device under test. With the filter applied, the oscilloscope user can probe the input pin and view the signal as it would be seen if the device could be probed internally. This virtual test point reveals the receiver's post-filter signal, even though the physical test point is a pin on the device package.

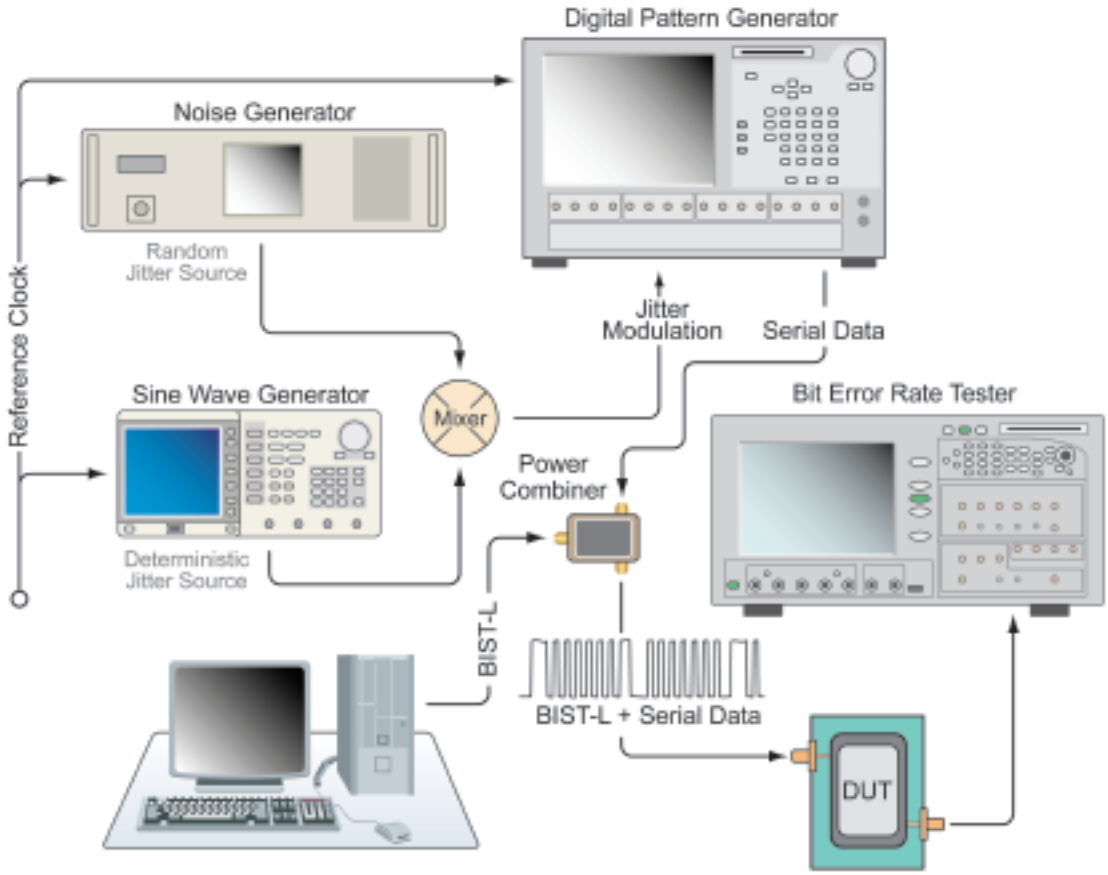



**Figure 6.4.** Modulated digital waveforms are the basis of this complex jitter tolerance measurement set-up.

- **Simplifying receiver jitter measurements:** Figure 6.4 illustrates a traditional jitter tolerance measurement setup using a pattern generator with external modulators for receiver testing. The DUT must be driven first with a setup sequence (BISTFIS), followed without interruption by a jitter-laden data signal. Clearly this is a complex arrangement, and there are some compromises in signal quality as well. The solution is to use an arbitrary waveform generator (AWG) using direct digital synthesis to provide sufficient bandwidth to minimize complexity. With this approach, jitter in any form can be merged into the test signal itself, and the effects of both random and deterministic jitter can be modeled. Moreover, the AWG can incorporate the BIST-FIS instructions as part of the data, eliminating the power combiner and its effects on signal fidelity.
- **Systemwide serial bus behavior:** Almost without exception, today's digital system environments include a mix of serial and parallel communication buses. Even a small, basic serial system is likely to have a debug port that delivers data in a parallel format, or parallel data buses handling internal transactions. Consequently, there is a need for tools that can capture and correlate multiple bus data streams, serial and parallel, all at once. The logic analyzer has long been a cornerstone of parallel bus acquisition,

while the protocol analyzer has performed the same role with serial data. The ideal solution would merge these capabilities and reduce complexity while offering inherent synchronization of serial and parallel data. The logic analyzer platform is a candidate but until now, serial acquisition with a logic analyzer has been possible only with the help of complex external bus support packages.

Recently—and uniquely in its class—the Tektronix TLA7000 Series has gained integrated serial acquisition features for PCI Express Gen I and II. Thus the TLA7000 Series becomes the sole solution that can, with one instrument, reveal serial and parallel bus interactions throughout a system. The probes used with the TLA7000 serial modules, unlike those of a protocol analyzer, are repeater-less: the data from the System Under Test does not go through a repeater that regenerates the signal, which potentially could mask some types of errors. Using repeater-less probing, the logic analyzer can view the physical layer directly.

Figure 6.5 depicts a page of serial data processed for display. Integrated serial/parallel acquisition provides a broad view of system operation and bus interactions. The logic analyzer incorporates a host of efficient analysis capabilities in a proven platform that is well-supported and familiar to most engineers. Thanks to its modularity, the instrument can be configured to acquire hundreds of parallel channels along with the serial buses. Much of what goes on inside a logic analyzer is software-based. Consequently, the analysis tools can respond readily to changing standards.

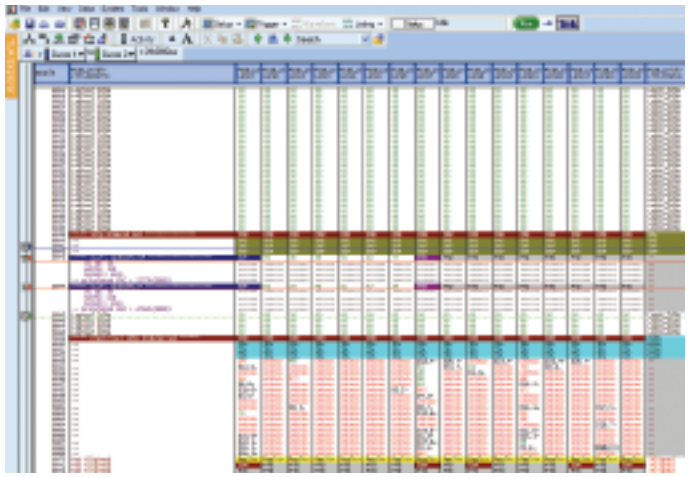


**Figure 6.5.** PCI Express trace captured by the TLA7000 Series logic analyzer.

## Summary

With so much complexity and change in the field of high-speed serial buses, engineers need verification and test solutions that help them find and correct design problems quickly and easily. In particular, they require complete serial data test solutions that enable them to develop products and ensure compliance to the latest serial data test requirements.

A new generation of measurement tools - including real-time oscilloscopes, equivalent time or sampling oscilloscopes, differential probing, logic analyzers and signal sources - has emerged to help engineers deal with serial measurement challenges. Taken together, these next generation instruments and application software solutions comprise an entirely new test bench for high-speed serial data. These solutions deliver the performance to capture, display and analyze the most complex and challenging serial data signals.

# Power Supplies

Every embedded system needs a power supply – usually to provide a stable low DC voltage to power its individual components. This will typically be a high-efficiency switch-mode power supply (SMPS) using high-frequency conversion techniques, and will itself need careful design and testing to ensure that it provides the right output and meets various standards on both its own performance (stability, ripple etc.) and how it affects other equipment (electromagnetic interference, EMC, etc.).

New switch-mode power-supply (SMPS) architectures with much higher data speeds and gigahertz-class processors that need higher current and lower voltages are creating new pressures for power-supply designers in the areas of efficiency, power density, reliability and cost.

To address these demands, designers are adopting new architectures like synchronous rectifiers, active power-factor correction and higher switching frequencies. These techniques bring unique challenges including high power dissipation at the switching device, thermal runaway and excessive EMI/EMC.

Because the power dissipated in a switch-mode power supply determines the overall efficiency of, and the thermal effect on, the power supply, the measurement of power loss at the switching device and the inductor/transformer assumes great importance. Engineers therefore need measurement and analysis equipment that can rapidly and accurately measure and analyze instantaneous power loss under changing load conditions.

## Switch-mode Power-supply Basics

The SMPS is known for its ability to handle changing loads efficiently. The power 'signal path' of a typical SMPS includes passive, active and magnetic components.

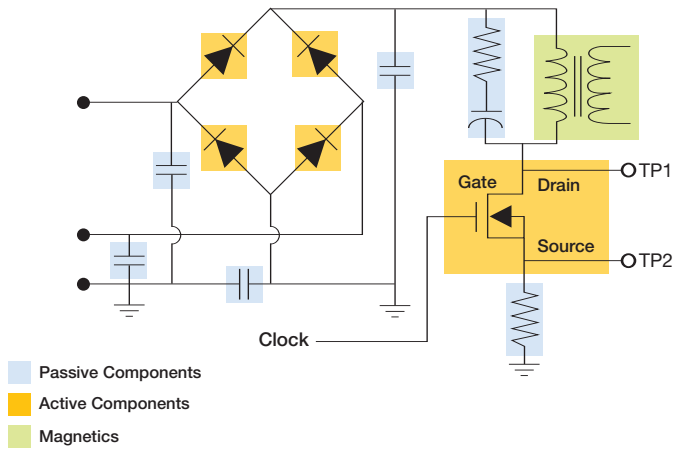



**Figure 7.1.** Switch-mode power supply simplified schematic.

The SMPS minimizes the use of lossy components such as resistors and linear-mode transistors, and emphasizes components that are (ideally) lossless: switch-mode transistors, capacitors and magnetics.

SMPS devices also include a control section containing elements such as pulse-width-modulated regulators, pulse-rate-modulated regulators, and feedback loops.

Control sections may have their own power supplies. Figure 7.1 illustrates a simplified SMPS schematic showing the power conversion section with active, passive, and magnetic elements.

SMPS technology relies on power semiconductor switching devices such as metal oxide semiconductor field-effect transistors (MOSFETs) and insulated-gate bipolar transistors (IGBTs). These devices offer fast switching times and are able to withstand erratic voltage spikes. Equally important, they dissipate very little power in either the 'on' or 'off' states, achieving high efficiency with low heat. For the most part, the switching device determines the overall performance of an SMPS. Key parameters of switching devices include: switching loss, average power loss, safe operating area, and more.

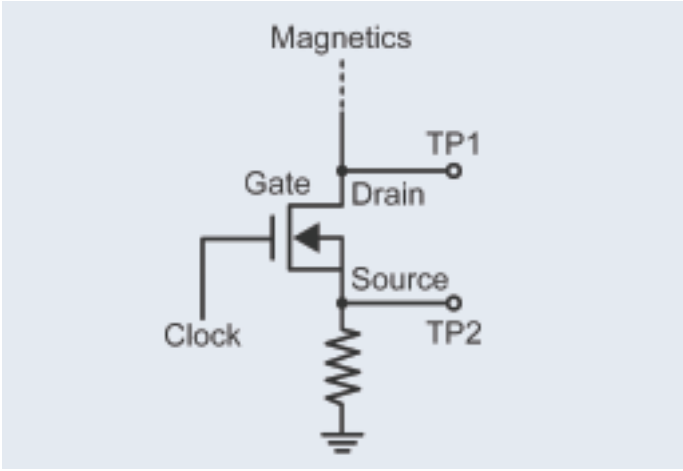



**Figure 7.2.** MOSFET switching device, showing measurement points.

In an AC/DC converter, the voltage across the switching device has a very high dynamic range. The voltage across the switching device during the 'on' state depends upon the type of switching device. In the MOSFET illustrated in Figure 7.2, the 'on' voltage is the product of channel resistance and current. In bipolar junction transistors (BJTs) and IGBT devices, the voltage is primarily based on the saturation voltage drop (VCEsat). The 'off' state voltage depends on the operating input voltage and the topology of the switch-mode converters. A typical DC power supply designed for computing equipment operates on universal utility voltage ranging from 80 to 264 VRMS.

## Characterizing SMPS Performance

Before any measurement can be made, it is important to understand the definition of terms used to characterize the performance and quality of an SMPS. These essential parameters may be divided into three categories: electrical characteristics, magnetic characteristics and input/output analysis parameters.

## Electrical Characteristics

Electrical measurements usually involve the characterization and measurement of the voltages across and currents through the switching device in the supply, which may be a MOSFET or an IGBT. The parameters of interest are turn-on loss, turn-off loss, power loss, dynamic 'on' resistance and safe operating area.

- **Turn-on/turn-off losses:** The energy lost in a switching device when it transitions from an 'off' to an 'on' state is defined as turn-on loss. Similarly, turn-off loss is described as the energy lost when the switching device transitions from an 'on' to an 'off' state. Transistor circuits lose energy during switching time due to the dissipative elements in the parasitic capacitance and inductance and charge stored in the diode. A proper analysis of these losses is essential to characterize the supply and to gauge its efficiency. The challenge in measuring turn-on and turn-off losses is to filter out the ringing present on the switching signal. This ringing is often mistaken for an 'on' or 'off' transition. The source of this ringing is parasitic capacitance and inductance in the circuit.
- **Power loss:** Power loss is defined as the difference between power input and output as a result of an energy transfer between two points. Switching and conduction losses account for a majority of the power loss in a supply.
- **Dynamic 'on' resistance:** The dynamic 'on' resistance of a switching device is its resistance when it is in the 'on' state. It is dynamic because the resistance is not constant and may vary with change in voltage or current.
- **Safe operating area (SOA):** The safe operating area of a transistor defines the conditions over which the device can operate without self-damage – specifically, how much current can run through the transistor at a given voltage. Exceeding these limits may cause the transistor to fail.

The SOA is inclusive of other limitations of the device such as maximum voltage, maximum current, power, junction temperature, secondary breakdown etc. SMPS designers use the information from the SOA to test the switching device in the power supply over diverse operating conditions. The challenge with measuring the SOA and the dynamic 'on' resistance is to successfully and accurately capture voltage and current data for the device under various load scenarios, operating temperature changes and variations in line input voltages. The more accurately the measurement system captures the voltage and current values, the more accurate this analysis will be.

## Magnetic Characteristics

Inductors and transformers make up the magnetic components of a switching power supply. It is important to understand the loss mechanisms associated with these components in order to accurately characterize the performance of the supply. The measurements of interest are inductance, magnetic power loss and magnetic properties such as B-H characteristics.

- **Inductance:** Inductors are commonly used as filters at the input and output of the supply. As transformers, they help to sustain oscillation in switching supplies and are used to couple voltage and current from the primary to the secondary while providing isolation between circuit elements. The inductance value depends on the current and voltage source, frequency of operation, excitation signal and wave shape. Hence it is important for the designer to monitor the behavior of these devices under operating conditions.
- **Magnetic power loss:** In order to accurately characterize the efficiency, reliability and thermal performance of a switching supply, it is important to have an understanding and awareness of the magnetic power losses that may occur in the supply. Core losses and copper losses are the two kinds of losses associated with magnetic elements:

  - Core losses consist of eddy-current losses and hysteresis losses, and may be calculated using the data sheet from the core element's manufacturer. These losses are dependent on the frequency of operation and the AC flux swing, and are independent of DC flux.

  - Copper losses occur because of to the resistance of the copper winding wire.
- **Magnetic properties and B-H characteristics:** The B-H curve is important to the design of the inductor or transformer used in an SMPS. This curve provides the relationship between the flux density (B) and the magnetic field intensity (H) of the core material. The slope of this characteristic is the permeability (m) of the core material, which influences the inductance. Inductance and permeability of the core material are greatly reduced at high flux densities, when the core saturates. The measurement of the B/H curve provides verification of the core loss and saturation (or lack thereof) of the magnetic elements in a switching supply. It also provides a measure of the energy lost per cycle in a unit volume of core material. Other magnetic properties of interest include the magnetic field strength (H), saturation flux density (BS), remanence (Br), coercive force (Hc) and initial permeability (μi).

## Input/output (I/O) Analysis

The analysis and comparison of the voltage and current at the input and output of the supply are essential in understanding the overall quality of performance of the supply as well as its efficiency. Ripple, noise, harmonics and power quality are all important parameters for accurate I/O analysis.

- **Ripple:** AC voltage that is superimposed onto the DC output of a power supply is expressed as a percentage of the normal output voltage or as peak to peak volts. Linear power supplies usually see a ripple that is close to twice the line frequency (100-120 Hz), whereas switching power supplies may see a switching ripple in the hundreds of kilohertz.

- **Noise:** High dv/dt and di/dt slew rates, parasitic inductance and capacitance, layout geometry and the switching frequency all contribute to the noise at the output of the switching supply. In general, noise in the SMPS may be correlated with the switching frequency, or it may be broadband.
- **Power-system harmonics and distortion:** Harmonics may be defined as sinusoidal components that have a frequency that is an integral multiple of the fundamental frequency. Distortion is defined as any deviation of a system's output signal waveform from that which should result from the input signal's waveform. Distortion may be caused by nonlinearities in an active device such as an operational amplifier, or by passive components such as coaxial cables, or by reflections in the propagation path. The term 'total harmonic distortion' (THD) is used to quantify distortion. The term expresses the distortion as a percentage of the fundamental of voltage and current waveforms. In simple terms, a pure sine wave will have no harmonics and no distortion, and a non-sinusoidal wave will have both distortion and harmonics. These harmonics are a big concern in power systems. Switching power supplies have the tendency to generate odd-order harmonics, which may find their way back to the power grid. The total percentage of harmonic distortion returned to the grid can rise as the number of switching supplies connected to the grid is increased. The analysis and reduction of harmonics becomes more critical as distortion causes heat build-up in the cabling and transformers of the power grid.
- **Power quality:** Ideally, the output of a power supply should not have any switching harmonics or other non-ideal noise components: something that, realistically, is not possible. Power-quality measurements such as true power, apparent (reactive) power, power factor, crest factor, current harmonics and THD help to gauge the quality of the output of the supply. Power-quality measurements are essential to determine the effects of distortions caused by nonlinear loads.

## Power-supply Design

Ideally, every power supply would behave like the mathematical models used to design it. But, in the real world, components are imperfect, loads vary, line power may be distorted, and environmental changes alter performance. Moreover, changing performance and cost demands complicate power-supply design. Consider these questions:

1. How many watts beyond rated capacity can the power supply sustain, and for how long?
2. How much heat does the supply dissipate, what happens when it overheats, and how much cooling airflow does it require?
3. What happens when the load current increases substantially? Can the device maintain its rated output voltage? How does the supply react to a dead short on its output?
4. What happens when the supply's input voltage changes?

The designer is asked to create a power supply that takes up less space, reduces heat, cuts manufacturing costs, and meets ever-tougher EMI/EMC standards. Only a rigorous regime of measurements can guide the engineer towards these goals.

## Measurement Basics

Historically, characterizing the behavior of a power supply has meant taking static current and voltage measurements with a digital multimeter and performing painstaking calculations on a calculator or PC. Today most engineers turn to the oscilloscope as their preferred power measurement platform. Modern oscilloscopes can be equipped with integrated power measurement and analysis software which simplifies set-up and makes it easier to conduct measurements over time. Users can customize critical parameters, automate calculations, and see results – in engineering units rather than just raw numbers - in seconds.

To those accustomed to making high-bandwidth measurements with an oscilloscope, power measurements, with their relatively low frequencies, might appear simple. In fact, power measurements present a host of challenges that the high-speed circuit designer never has to confront.

When preparing for switch-mode power supply measurements, it is important to choose tools that can do the job, and to set up those tools so that they will perform accurately and repeatably.

The oscilloscope must, of course, have the basic bandwidth and sample rate to handle the switching frequencies within an SMPS. Power measurements require at least two channels: one for voltage and one for current. Equally important are the facilities that make power measurements easier and more reliable. Following are some of the considerations:

- Does the instrument offer a solution to handle switching device 'on' and 'off' voltages in the same acquisition? These signals may have a ratio of 100,000:1
- Are dependable, accurate voltage and current probes available, and is there an expedient way to align their differing delays
- Is there an efficient process to minimize quiescent noise in the probes
- Can the instrument be equipped with sufficient record length to capture long line-frequency wavelengths at high sample rates

These characteristics form the underpinnings of meaningful and efficient power-supply design measurements.

The voltage across a switching device can be very large, and is 'floating', that is, not referenced to ground. There are variations in the pulse width, period, frequency and duty cycle of the signal. Waveforms must be faithfully captured and analyzed for imperfections.

The demands on the oscilloscope are exacting. Multiple probe types – single-ended, differential, and current – are required simultaneously. The instrument must have a deep memory to provide the record length for long, low-frequency acquisitions. And it may be called upon to capture signals of vastly differing scales in one acquisition.

Digital oscilloscopes running power measurement and analysis software packages become a powerful alternative to the power meters and harmonic analyzers traditionally used for power quality measurements.

One important consideration is that the oscilloscope must be able to capture harmonic components up to the 50th harmonic of the fundamental. Power line frequency is usually 50 Hz or 60 Hz, according to applicable local standards, although in some military and avionics applications the line frequency may be 400 Hz. In addition, signal aberrations may contain frequencies that are even higher. In today's high-speed oscilloscopes, oversampling of multiple GS/s ensures that fast-changing events are captured with high resolution. In contrast, conventional power meters can overlook signal details because their relatively slow response time.

The oscilloscope's record length has to be sufficient to acquire an integral number of cycles, even at very high sampling resolution. An oscilloscope's ability to capture events over a period of time depends on the sample rate used and the depth (record length) of the memory that stores the acquired signal samples. The memory fills up in direct proportion to the sample rate. When the sample rate is set high enough to provide a detailed high-resolution view of the signal, the memory fills up quickly.

For many SMPS power measurements, it is necessary to capture a quarter-cycle or half-cycle (90° or 180°) of the line frequency signal; some even require a full cycle. The object is to accumulate enough signal data to support calculations that counteract the effect of variations in the line voltage.

An oscilloscope such as the Tektronix DPO Series can be configured with several million points of memory depth, which is sufficient to store the needed amount of the line frequency signal at an appropriate sample rate.

## Probing Considerations

To make power measurements with a digital oscilloscope, it is necessary to measure voltage across and current from the drain to the source of the MOSFET switching device or the collector-to-emitter voltage across an IGBT. This task requires two separate probes: a high-voltage differential probe and a current probe.

High-voltage differential probes designed for power applications are the preferred tools for observing floating voltage signals.

Current probing is a special consideration. There are several implementations of current probing architecture:

- The AC current probe is based on current transformer technology. The current-transformer probe is non-intrusive but cannot sense the DC component in the signal, which can result in inaccurate measurements.
- The current shunt requires an interruption in the circuit, and can cause a voltage drop within the probe itself, potentially compromising power measurement accuracy.
- The AC/DC current probe is typically based on Hall-effect sensor technology. This device senses AC/DC current non-intrusively, and is able to read both the AC and the DC components with one connection. The AC/DC current probe has become the tool of choice for challenging power-quality measurements in switch-mode power supplies.

Each of these probes has its own characteristic propagation delay. The difference in these two delays, known as skew, causes inaccurate amplitude and time-correlated measurements. It is important to understand the impact of the probes' propagation delay on maximum peak power and area measurements. Power is the product of voltage and current, and if the two multiplied variables are not perfectly aligned, the result will be incorrect. The accuracy of measurements such as switching loss suffers when the probes are not properly 'de-skewed'.

With the oscilloscope-based power measurement software, the chosen probe combination is automatically de-skewed.

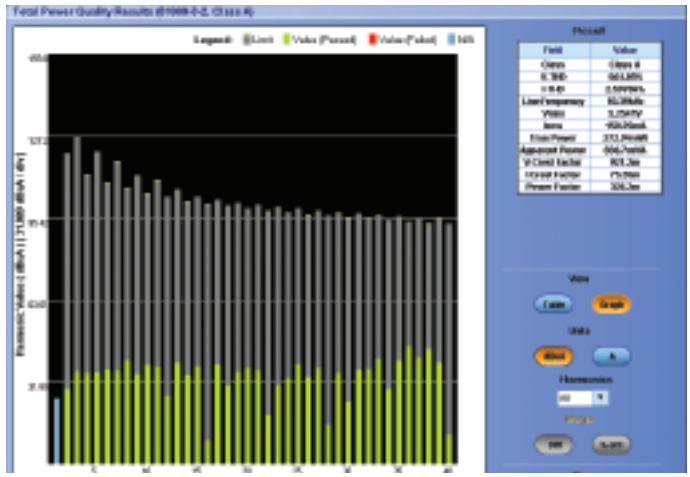


**Figure 7.3.** Power-quality results using a DPO7000 Series oscilloscope and DPOPWR measurement and analysis software.

The software takes control of the oscilloscope and adjusts the delay between the voltage and current channels using live current and voltage signals to remove the difference in propagation delay between the voltage and current probes.

A static de-skew function is also available that relies on the fact that certain voltage and current probes have constant and repeatable propagation delays. The static de-skew function automatically adjusts the delay between selected voltage and current channels based on an embedded table of propagation times for selected probes.

## Automated Power Measurements

A digital oscilloscope equipped with advanced power measurement and analysis software is transformed into a no-compromise automated power measurement platform. The software automatically sets up the oscilloscope and its initial measurement parameters. These can be fine-tuned manually if desired.

Figure 7.3 shows the power quality and current harmonic readings acquired with an oscilloscope running this software. The display brings out a full range of measurements including true power, apparent power, crest factor, total harmonic distortion and power factor, as well as a bar-graph display of current harmonics.

## Powerful Measurement Capabilities

The power supply is integral to virtually every type of line-powered electronic product, and the switch-mode power supply (SMPS) has become the dominant architecture in digital computing, networking, and communications systems. A single switch-mode power supply's performance - or its failure - can affect the fate of an embedded system.

Measurements are the only way to ensure the reliability, stability, compliance, and safety of an SMPS design. SMPS measurements fall into three principal categories: active device measurements; passive device measurements (mostly magnetics); and power-quality tests. Some measurements may deal with floating voltages and high currents; others require mathematics-intensive analysis to deliver meaningful results.The modern digital oscilloscope has become the tool of choice for characterization and troubleshooting measurements. When equipped with the appropriate probing tools and automated measurement software, they dramatically simplify challenging SMPS measurements while providing fast, accurate answers.

Software tools speed measurement procedures and minimize setup time. Most power-quality measurements can be automated by full-featured power measurement software running on the oscilloscope itself, performing lengthy procedures in seconds. By reducing the number of manual calculations the oscilloscope acts as a very versatile and efficient power meter.

Techniques such as non-intrusive current probing preserve circuit integrity, while software-based tools such as smoothing filters ensure repeatable results. Complex mathematical chores are also handled automatically by the integrated software.

Thanks to their versatility and ease of use, oscilloscopes have supplanted many traditional single-function tools in the power supply design laboratory.

## Digital RF Technologies

The full power of digital computing technologies has finally come to the radio world, with the corollary that the world of embedded systems must now encompass ways of dealing with wireless systems.

These developments have had a profound impact on the RF world, from the pace of innovation to the increasing availability of more powerful, specialized and inexpensive integrated circuits to perform traditional analog radio functions.

And, in turn, the availability of these advanced digital RF technologies suggests ever more ways to expand wireless communications.

The key technical elements driving the 'wireless everywhere' trend are:

- **Radio/wireless links from the same IC processes as PCs:** Advanced components such as digital signal processors (DSPs) and Field programmable gate arrays (FPGAs) have replaced many traditional analog radio components.
- **Moore's Law arrives in radio:** The same rate of performance advance and economy of scale that applies to digital technology in the computing industry now applies equally for many aspects of the radio world. This is a dramatic change for radio communications. New technologies are rapidly becoming available and are quickly being deployed, often in advance of codified standards.
- **Inexpensive, intelligent digital RF becomes ubiquitous:** In addition to basic cost benefits, digital intelligence is inherently included in the cost of a digital RF chip. Thus, while wireless devices are getting much cheaper, they are also getting 'smarter'.
- **Consumer-driven technology:** Consumers want truly ubiquitous applications and services - wireless at any time and anywhere on any global network - for convenience and enhanced user experience.
- **Device to device connectivity and communication:** The digital camera automatically talks to the photo printer. Home entertainment systems will soon no longer require a tangle of wires and connections. Smart buildings can be reconfigured to meet changing floor plans and new environmental control requirements.
- **Internet everywhere:** A computer, PDA or phone automatically finds the fastest connection no matter where it is, and these fast connections are becoming widely available.

Whether commercial or military, customers want more information, faster, anywhere, anytime. Digital RF technologies make wireless access available in more places, to more people, and for less money.

The demand for greater capabilities continues to drive innovation, ranging from power amplifiers to advanced radar systems and home networking. The combination of advanced technical capabilities brought about by digital RF and the increasing customer demand for more functionality and mobility has generated an explosion of innovation in wireless communications.

### Changes in Spectrum Usage

Radio spectrum is a scarce resource. The spectrum allocations of today are widely recognized as inefficient. Allocated spectrum is under-utilized much of the time, and interference problems abound. The advent of digital RF is powering rapid innovation in software-defined and cognitive radio technologies. Software-defined radio (SDR) technologies promise much lower-cost, flexible user devices that support multiple network interoperability and multi-media services. Cognitive-radio (CR) techniques greatly improve spectrum utilization, interference avoidance, and support new security and personalization features. Many view the inflated cellular spectrum

auctions of the recent past as signifying a fundamental flaw in how we deal with this scarce radio spectrum resource. Digital RF powered SDR and CR technologies provide the new path forward and will fundamentally change spectrum allocation methodologies.

This radical change in spectrum allocation and utilization methodology is occurring over a broad range of applications, from commercial and military communications to advanced radar and other remote sensing applications. While much of the innovation related to military communications is sensitive information, one can look to the commercial world for examples of this rapid innovation and dramatic change. Technologies such as WiFi, Bluetooth, cordless telephones, ZigBee and many others share a very limited amount of spectrum known as the ISM (industrial, scientific, medical) spectrum bands. The ISM bands are essentially unregulated 'free fire' zones, and, as stated by the International Telecommunications Union, 'Radio communication services operating within these bands must accept harmful interference, which may be caused by these [ISM] applications.'

Although the potential for interference is high in the ISM band, and interoperability must be maintained in these applications, innovation nevertheless is exploding through the use of Digital RF technologies.

## Engineering Challenges

Digital RF technologies enable time-varying techniques to more efficiently use available spectrum, avoid interference and ensure seamless operation.

For example, (Wireless LAN) WLAN signals seek out clear frequencies and adapt their modulation type to best use the available channel. Techniques used today include:

- **Frequency hopping:** In order to reduce the effects of fading and interference, and in some cases to improve security by lowering the likelihood of intercept, some digital RF systems employ frequency hopping, where the signal is present on one frequency at one instant and appears next on a different frequency. The challenge for engineers when designing is to ensure that the hop occurs to the correct frequency, and that the signal settles at the new frequency within a specified time. Determining hop characteristics requires that the test and measurement solutions have sufficient bandwidth to simultaneously view both the beginning and ending frequency, and to be able to trigger on the transition between frequencies.
- **Signal bursting:** Some RF systems employ bursted signals in a time-division duplexed (TDD) fashion. This is done to make the most efficient use of a spectrum, allowing both the uplink and downlink to occupy the same frequency. It also reduces the cost of user equipment by greatly simplifying the radio. The challenge in the design of these systems is to ensure that they turn on and transmit without distortion, quickly, and that they turn off at the right time. Measuring bursted signals requires the ability to trigger on the signal. Time-domain triggering is essential, while frequency-domain triggering is more flexible and allows the user to ignore adjacent signals. Once captured, the signal needs to be examined for its signal quality during the turn-on and turn-off phases. This involves measuring the power, frequency and modulation quality versus time, an application greatly aided by easily correlated multi-domain analysis.
- **Adaptive modulation:** Adaptive modulation is used to optimize system throughput and make the best use of tight spectrum allocations. Modulation techniques may change from very robust binary phase-shift keying (BPSK) to a high data rate 64-state quadrature amplitude modulation (64 QAM), depending on channel conditions such as fading and interference. These modulation changes can be programmed to occur on a packet-to-packet basis. The challenge for the designer is to make certain that the modulation changes occur seamlessly so that no data is corrupted. The measurement challenge is to measure these changes seamlessly. The ability to auto-detect the modulation type and perform easy multi-domain analysis greatly aids fast evaluation of these characteristics.

These techniques all exhibit frequency and modulation changes that occur over time, resulting in RF signals that are becoming increasingly complex and transient in nature. This, in turn, creates problems that are harder than ever to find and identify and troubleshoot. These transient and time-varying transmission techniques help RF devices to avoid interference and maximize peak power - and often evade detection.

## Testing Digital RF Technologies

The explosion of digital RF has created a highly complex technology environment. With non-assigned channels, adaptive modulation, peer-to-peer communications, and countless devices transmitting simultaneously within a limited radio-frequency spectrum, frequent collision and interference problems occur. These collisions cause intermittent or jammed communication. In the commercial world, this is frustrating to customers and businesses. In the military and government world, it can literally mean the difference between life and death.

In order to avoid the 'digital cliff' where a system or network simply stops working from overload or interference, it becomes critical to ensure that these devices do not transmit RF energy at unwanted times or unwanted frequencies and are able to function correctly in the presence of interference.

This situation leads to two test and measurement challenges, which can be summarized as 'find it' and 'characterize it'. The first testing challenge is to discover the interfering or spurious signal, whether internally generated by a device or originating externally. Then, once the interference is found, it must be thoroughly characterized. The signal of interest may have lower amplitude than other signals in the same frequency band and may happen infrequently, making it very difficult to capture.

## Measurement Tools

As can be seen from the above discussion, time cannot be ignored in any discussion on digital RF. As a result, digital RF creates the need for test tools whose capabilities mirror the time-varying nature of today's signals.

The traditional RF measurement tools - swept spectrum analyzers and vector signal analyzers (VSAs) - are generally not up to the task of dealing with digital RF technologies and devices. Because they essentially tune a narrow filter across a range of frequencies to generate a single frequency-domain display, otherwise known as a 'sweep', traditional swept analyzers can only compile a collection of uncorrelated RF spectral activity. Even the fastest swept analyzer may miss many intermittent or rapidly changing signals.

VSAs rely on post-capture analysis techniques, and do not perform real-time tasks, such as frequency-domain triggering, which are required in the modern digital RF world of ever-changing, brief and bursted signals. The lack of appropriate tools requires engineers to employ off-line and often home-grown solutions that can be inefficient, time consuming, complicated and often very expensive.

Many complex problems are illuminated for the first time by time-correlating RF signal behavior in the frequency, time and modulation domains. Tools that are able to trigger on and capture transient events and easily provide time-correlated, multi-domain displays of the signal dramatically reduce the time that engineers spend on troubleshooting problems. Such a tool is the real-time spectrum analyzer (RTSA): a new class of instrument that is designed to specifically solve digital RF problems.

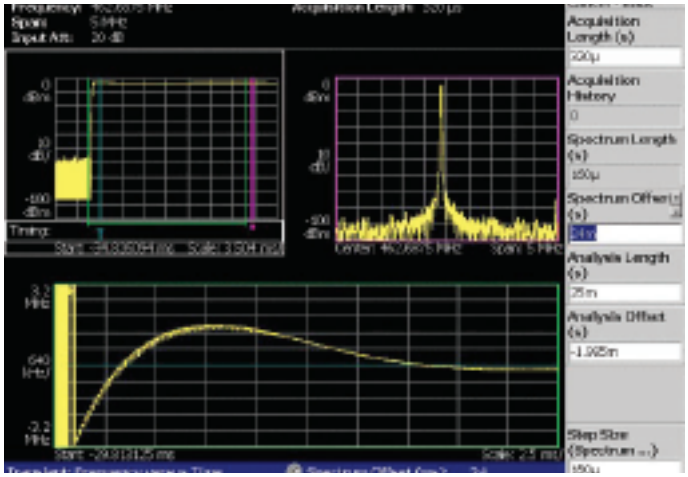



**Figure 8.1.** RTSAs deliver time-correlated, multi-domain analysis capability.

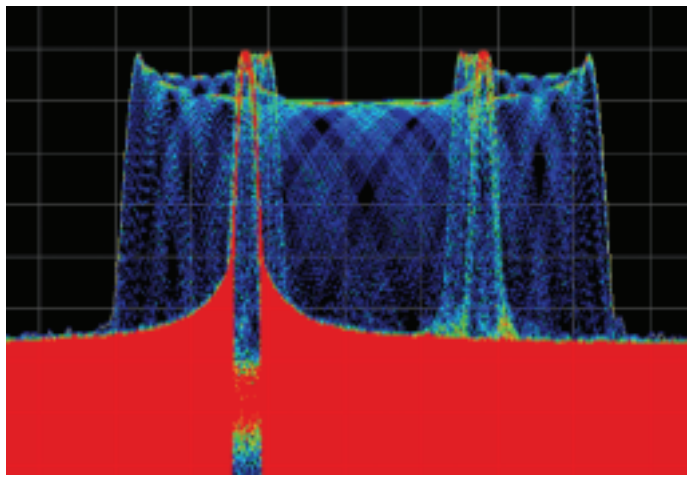


**Figure 8.2.** Revolutionary DPX spectrum display reveals transient signal behavior that helps you discover instability, glitches and interference.

RTSAs allow engineers to discover the unexpected problems that are commonplace in digital RF by selectively triggering on time- and frequency-domain anomalies and acquiring a seamless time record of a span of RF frequencies into memory. As shown in Figure 8.1, this ability to detect and capture the relevant spectrum event enables more useful time-correlated, multi-domain (time, frequency and modulation) analysis, all without the need to recapture the signal.

The key to these RTSA capabilities is a waveform processing technology known as DPX®, which uses high-speed parallel processing to produce a real-time 'live' RF spectrum display.

The DPX technology can produce nearly three orders of magnitude improvement in the spectrum processing rate compared to swept spectrum and vector signal analyzers, and the resulting display provides engineers with a view of RF signal instabilities and transients that would previously have escaped detection. Figure 8.2 shows how DPX greatly improves the spectrum update rate and display of information, transforming volumes of RF data into a single-glance view that clearly shows where signal anomalies exist. This capability makes it ideal for RF engineers who need to look for hidden, intermittent or infrequent signals.

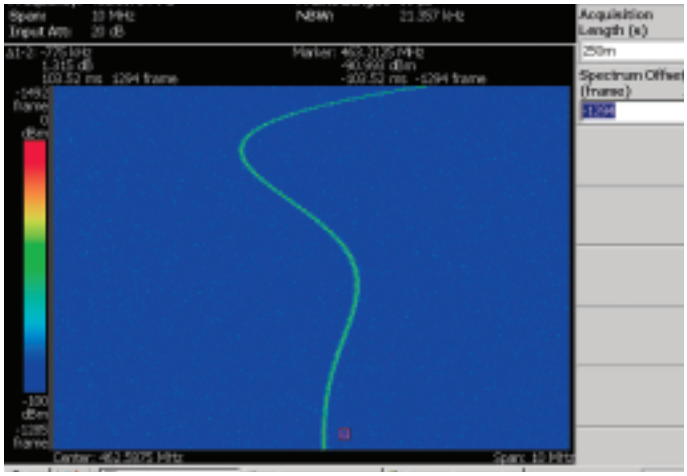


**Figure 8.3.** Spectrogram showing oscillator turn-on settling characteristic enables measurement frequency variance over time.

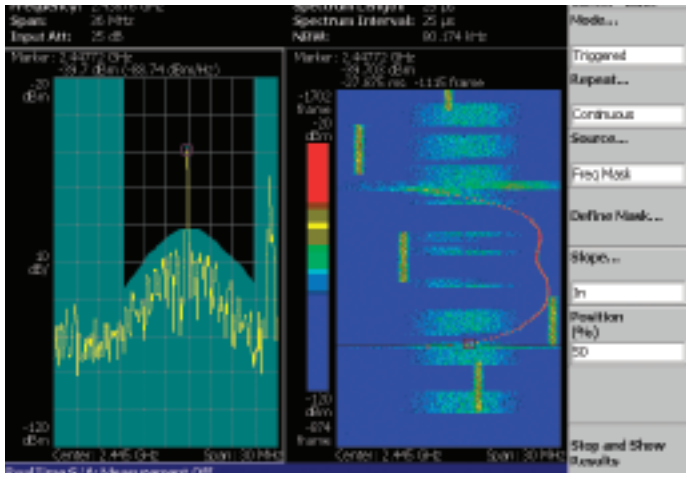


**Figure 8.4.** Using the Frequency Mask Trigger and Spectrogram to determine if interference with specific WLAN packets is coming from a Bluetooth transimitter or a microwave oven.

Another important feature is the spectrogram display, which plots frequency and amplitude changes over time - many minutes of time in some cases. The frequency, time and modulation domains are assembled into visible time-correlated images, while the spectrogram itself summarizes the long-term view. As shown in Figure 8.3, this enables an intuitive, three-dimensional look at the time-varying signal behavior, otherwise unseen in the traditional frequency-domain displays available on swept spectrum and many vector signal analyzers.

Also novel is the frequency mask trigger feature, shown in Figure 8.4, which allows users to define both the frequency and amplitude (power) conditions under which the instrument captures the signal information. This enables engineers to quickly home in on suspect frequencies or monitor signals continuously but acquire them only when the signal changes. Long memory enables engineers to capture all signal information just once and immediately perform a complete analysis. The event (for example, an interfering or transient signal) being analyzed may only happen once or very infrequently, so it is critical to capture all the information the first time.

By displaying a seamless record of frequency and power changes over time, an RTSA can solve many transient problems ranging from modulation switching on software-defined radio systems to identification of rogue pulses in radar transmission to dynamic modulation changes during a WLAN transmission.

## Digital RF Applications

The RTSA can help engineers to test digital RF technologies across a myriad of applications such as:

- **Cellular and WLAN:** Mobile phone and other wireless infrastructure manufacturers face significant challenges in spectrum control, power efficiency and production cost. DSP technology has enabled advances in power-amplifier linearization to address performance and efficiency. Ineffective implementation of these advanced solutions can lead to bursted and transient behavior that is difficult, if not impossible, to detect using a traditional spectrum analyzers. Real-time spectrum analyzers allow engineers to discover, detect and analyze transient amplifier behavior that traditional tools miss. Mobile devices now incorporate

a large number of transmit and receive chains (such as in phone/WLAN/Bluetooth/RFID combination devices). The real-time spectrum analyzer has enabled engineers to sort out these potential sources of self-interference in the time and frequency domains, ensuring transparent operation of these complex systems.

- **RFID:** The RFID reader and tag have a complex response. Aside from interference from adjacent communication, a reader must be able to isolate the response of individual tags in a multi-tag environment. In some systems, the RFID readers must be agile and hop over multiple frequencies to mitigate interference and multi path environments. A real-time spectrum analyzer can capture the interaction of the reader and tag completely and analyze multiple different formats of RFID systems.
- **Military and aerospace communications:** The software-defined radio and cognitive radio devices have an extensive amount of software replacing traditional analog hardware functionality. When unexpected performance occurs in the form of spectral emissions, it is necessary to isolate software and hardware problems. Real-time spectrum analyzers enable time correlation of frequency-domain events and can trigger a time-correlated oscilloscope and logic analyzer. Events can be isolated from RF to analog and digital domains.
- **Radar:** Spectral emissions from radar systems can interfere with wireless communications. Certain radar spectral emissions can also leave a signature that allows the easy detection and classification of the radar. It is important that a radar system only emits a desired spectral emission to satisfy the intended function. Using pulse analysis software, the real-time spectrum analyzer simplifies complex measurements on radar signals, including individual and pulse train analysis and phase-to-phase measurements. For the first time, engineers can detect in band interference due to intended or unintended emissions which would previously have gone undetected due to the limitations of other spectrum analyzer architectures.
- **Spectrum monitoring and surveillance:** With the explosion of wireless devices, the detection and classification of interference in both the licensed and the unlicensed spectrum is an ever-growing issue. The architecture and frequency-mask-trigger capabilities of the real-time spectrum analyzer make it possible to detect interference signals with 100% probability of intercept, making it an ideal platform for mission-critical spectrum monitoring and surveillance. The instrument's real-time display capabilities also open a new level of signal classification based on the 'live' RF display and persistency control. RF signals will be discovered and seen live for the first time.
- **UWB:** UWB communications are gaining acceptance as a low-cost alternative to tethered communications and are soon expected to be ubiquitous. Next-generation wired and wireless interface standards are looking to UWB communications for low-cost short-range communications alternatives. UWB intentionally uses many licensed and unlicensed frequency bands. Through the use of low power and frequency hopping, the desired effect is to mitigate the potential interference to other systems using the same bands. High-bandwidth oscilloscopes provide the ability to fully characterize UWB and UWB WiMedia signals during all operation modes, and assist in other non-standard UWB applications. The capture depth and frequency range can cover the intentional and unintentional artifacts of UWB communication. Next-generation high-speed arbitrary waveform generators will soon provide the instantaneous bandwidth and dynamic range resolution needed to directly generate the required RF and microwave signals for UWB, thus eliminating the need for complex systems with multiple instruments and filters.

The RTSA also provides a measurement-centric user interface, enabling engineers to more easily perform needed measurements. An engineer is able to select a variety of measurements, and the analyzer then automatically determines the appropriate settings needed to satisfy the measurements. These intelligent controls provide optimized acquisition and analysis parameters to speed set-up and measurement time, but the engineer still has the ability to manually override the settings for complete control if needed.

## Signal Sources

An important part of testing digital RF systems is the requirement to generate complex, fast changing signals that owe as much to the digital world as to the RF world. This process requires digitally configured signal sources such as arbitrary waveform generators (AWGs). These products can be used in two ways:

- Testing receivers by using signals that emulate real-world signals in applications such as evaluating an RFID reader's ability to cope with multiple tags or range sensitivity or validating the performance of UWB and radar receivers.
- Using the AWG as part of the prototype at the development stage and even as part of operational equipment. By using AWGs, users are able to experiment with multiple types of waveforms - broadband chirp signals in radar systems, for example.

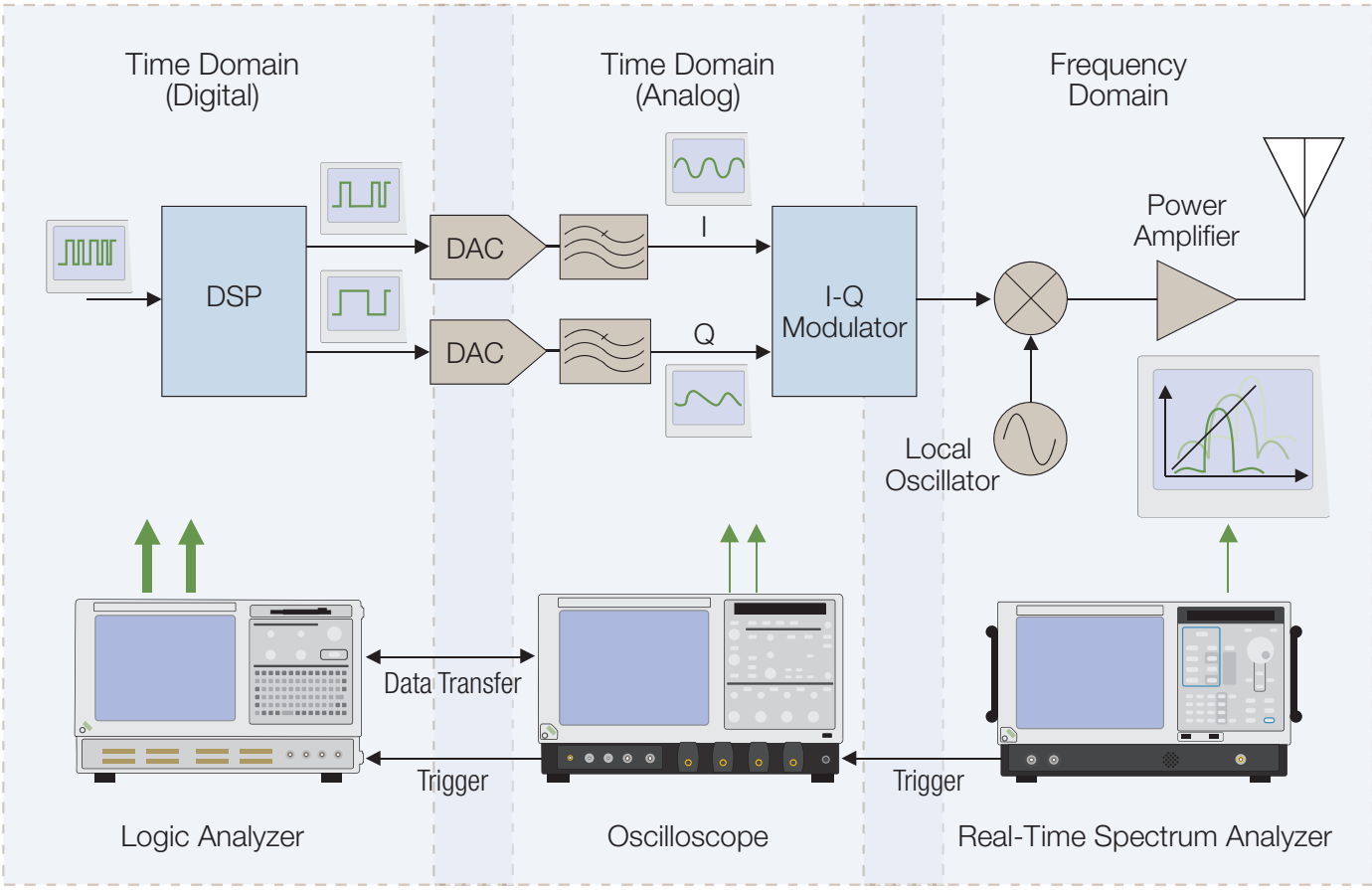



**Figure 9.1.** To isolate the source of problems in a modern embedded design such as a software-defined radio, it is important to trace the RF signal back through the digital path in order to identify the point of interference. Since the signal information changes form in a SDR design from digital words to continuously variable analog voltages, several pieces of test equipment may be needed to diagnose the exact source of problems.

## Looking Ahead with Total Insight

It is appropriate that digital RF technology formed the last section of this primer, for in many ways this sector embodies all the elements that make the embedded market such a challenging one. For example, several digital RF technologies are currently being developed which require an even wider bandwidth than is available with today's highest performing spectrum analyzers.

These ultra-wideband applications include the emerging WiMedia radios and standards being built on these for applications such as Certified Wireless USB. Fortunately, test solutions do exist for this category of technology. The evaluation and design of a UWB radio requires measurement instruments that have wide bandwidth, good dynamic range, fast signal capture, and the ability to perform spectral analysis on 'live' signals.

We are now in a situation where the biggest challenge is looking at the whole system: not just the individual components. Analog and digital, hardware and software, high and low frequencies, power and micro power signals all coexist alongside one another in embedded systems, and engineers need tools that deliver total insight by allowing them to examine all these elements and the interactions between them.

Figure 9.1 provides a summary of the way in which different measurement tools will address the challenges of embedded systems development.

## Contact Tektronix:


**ASEAN / Australasia** (65) 6356 3900
**Austria** +41 52 675 3777
**Balkan, Israel, South Africa and other ISE Countries** +41 52 675 3777
**Belgium** 07 81 60166
**Brazil & South America** (11) 40669400
**Canada** 1 (800) 661-5625
**Central East Europe, Ukraine and the Baltics** +41 52 675 3777
**Central Europe & Greece** +41 52 675 3777
**Denmark** +45 80 88 1401
**Finland** +41 52 675 3777
**France** +33 (0) 1 69 86 81 81
**Germany** +49 (221) 94 77 400
**Hong Kong** (852) 2585-6688
**India** (91) 80-22275577
**Italy** +39 (02) 25086 1
**Japan** 81 (3) 6714-3010
**Luxembourg** +44 (0) 1344 392400
**Mexico, Central America & Caribbean** 52 (55) 5424700
**Middle East, Asia and North Africa** +41 52 675 3777
**The Netherlands** 090 02 021797
**Norway** 800 16098
**People's Republic of China** 86 (10) 6235 1230
**Poland** +41 52 675 3777
**Portugal** 80 08 12370
**Republic of Korea** 82 (2) 6917-5000
**Russia & CIS** +7 (495) 7484900
**South Africa** +27 11 206 8360
**Spain** (+34) 901 988 054
**Sweden** 020 08 80371
**Switzerland** +41 52 675 3777
**Taiwan** 886 (2) 2722-9622
**United Kingdom & Eire** +44 (0) 1344 392400
**USA** 1 (800) 426-2200
For other areas contact Tektronix, Inc. at: 1 (503) 627-7111

Updated 17 October 2007

**For Further Information**

Tektronix maintains a comprehensive, constantly expanding collection of application notes, technical briefs and other resources to help engineers working on the cutting edge of technology. Please visit **www.tektronix.com**



10/07 DM 54W-21287-0


Tektronix®
Enabling Innovation