

High Performance Digital Test in VXI: Designing Instruments As Systems

Eric Truebenbach
Staff Engineer - Teradyne, Inc

Abstract

VXI is the standard of choice for designing custom test systems. Emerging standards such as *VXIplug&play* now also allow general purpose test systems to be constructed in the VXI framework. But until recently, the high performance digital functions needed for testing rather than just controlling a digital interface have been absent from VXI. As VXI matures, high digital performance needs to be included to complete VXI's capabilities.

This article examines the key hardware characteristics of mature digital instruments and systems, the suitability of the VXI standard for accommodating these characteristics, and implementations of digital functions in a VXI instrument. Special attention is given to applying system design methods to the design of an instrument, in order to achieve highest performance in actual use. A specific hardware implementation is presented along with an analysis of its success at meeting the key requirements.

Background

The VXI standard has long promised, and delivered, instrumentation that is denser and has a higher performance than benchtop instruments. In addition, the simple act of taking instruments from the clutter of the laboratory workbench and plugging them in to a single chassis, controlled by a standard interface, has created a whole greater than the sum of its parts: the collection of instruments has become a system. And the trends in VXI hardware and software show that the creation of systems from instruments continues. Starting with the basic interconnections permitted by the backplane, and adding the greater ease of control made possible by *VXIplug&play*, new products are binding the system together ever more tightly. VXI is not just a packaging convenience, but a means to build a custom test system without compromise.

The march of instruments from the benchtop into the VXI chassis has, however, left one function behind. While even complex analog instruments such as oscilloscopes and spectrum analyzers are available as VXI instruments, very few digital instruments have been offered for the VXI backplane. Perhaps this is because digital instruments never really were on benchtop in the first place. And those digital instruments that are available tend to be used for *control* of the digital interface of a Unit Under Test (UUT), rather than to perform *testing* of the digital performance. As the digital content of every electronic assembly grows, VXI needs to include high performance digital testing to fulfill its promise of offering the highest modular performance. Digital instruments in VXI need to offer the performance that analog VXI instruments already offer, to allow digital testing without compromises in speed, accuracy, or flexibility.

Digital Instruments

To explore how high performance digital can be implemented in VXI, one first needs to understand what basic types of digital instruments are available. These are, in order of performance: I/O ports, Bus Emulators, Digital Word Generators (DWGs), and Digital Performance Testers.

I/O ports

An I/O port is the simplest type of digital instrument. Sometimes it isn't even an instrument at all, since an I/O port may be implemented as the parallel port on a PC. I/O ports are used for the simplest type of control, where a slow and uncontrolled data application rate, limited number of I/O channels, and a fixed logic family are sufficient to control a UUT. An I/O port offers simplicity and low cost, but at the expense of performance. A PC parallel port on a very fast computer can run at about 80 kHz bidirectionally. Besides its low speed, an I/O port gives absolutely no control over logic levels or timing. Without these features, I/O ports cannot be used to test a UUT digital port anywhere near its specifications.

Bus Emulators

Bus emulators are designed to control a digital bus port on a UUT. Some types of bus emulators mimic only one type of bus, such as the ISA bus, but more general-purpose bus emulators may be programmed to emulate different types of buses. Bus emulators improve upon the basic I/O port by caching the digital data in local memory, called pattern memory, behind the port. This gives some control over timing, since the memory can be loaded at a low speed, and then the digital data can be sourced from memory at a fixed clock rate. A simple state machine steps through the pattern memory at a fixed rate to produce the signal transitions at the memory outputs. At the same time, responses from the UUT are written in to a response memory controlled by the same state machine. The memory I/O is connected to digital I/O pins called channels.

General purpose bus emulators usually have three types of digital channels: an address field, a data field, and a number of timing or control channels. All channels within the address and data fields are controlled as a group: they all transition at the same time, and they all tristate at the same time. The timing channels consist of a small number of unidirectional (that is, input only or output only) signals, and are used to emulate bus signals such as the read, or write, or clock signals. A separate signal called the test strobe clocks the UUT's outputs into response memory. I/O logic levels are usually fixed.

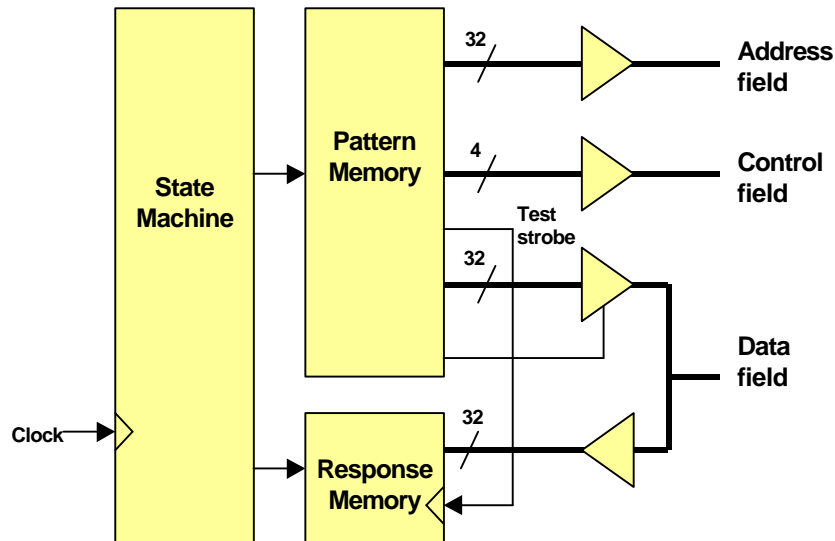


Figure 1: Bus Emulator Block Diagram

Bus emulators, especially those designed for a specific type of bus, enjoy the advantage of being easy to use. The use of dedicated signals for specific functions makes hookup to the UUT straightforward, and the emulator's controlling software usually has pre-programmed signal transition sequences for read cycles, write cycles, wait states, etc. But the ease of use comes at the expense of lack of flexibility. While the primary digital interface to a UUT is usually some type of bus, there are almost always some digital signals that may share only their basic operating frequency with the bus signals. These signals may include serial ports, specialized timing signals, ECL or PECL signals, control signals for external circuitry, and interfaces to off-board processors, memory, and the like. Trying to emulate these signals within the constraints of a bus emulator's small number of fixed-function I/O channels is frequently not possible.

Since bus emulators read their programmed data from memory, the data will "glitch" every time the memory address is changed. This can make the outputs unusable for edge-sensitive signals such as clocks. Some bus emulators have special channels with circuitry to remove glitches for edge-sensitive signals, or use an extra cycle to clock the memory outputs through a separate register.

Bus emulators rarely match the speed of modern buses. Every signal transition requires a memory location to store. This means that even a simple three-clock bus cycle such as the read cycle shown in Figure 2 requires six memory accesses to produce. Keeping the number of memory locations down to only six requires that non-clock signals transition only when the clock transitions, that is, on clock edges. This means that even a high-end 20 MHz bus emulator can produce one of these read cycles only at a 300 ns or 3.33 MHz rate. And of course, timing margining is only possible within the 50 ns resolution. The lack of speed and limited timing resolution means that bus emulators can be a good way to control a UUT, but are limited in their ability to test the digital interface of a UUT to its specifications.

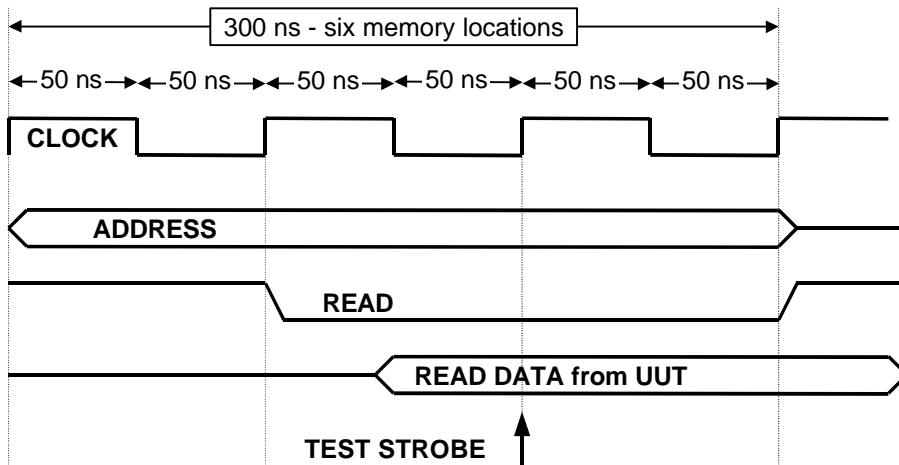


Figure 2: Bus Emulator Read Cycle

Digital Word Generators

Digital Word Generators, or DWGs, build upon the bus emulator's memory-and-state-machine architecture but make it more general-purpose. The local memory for storage of signal states remains, but the number and type of I/O channels is configurable. A DWG may have several types of fixed logic level and even programmable logic level channels available. The number of channels is usually configurable by buying modular channel options. Channels still have the bus emulator's "group constraints;" that is, functions like tristate control and timing are still fixed for a group of eight, sixteen, or even sixty-four channels. But each channel can be used either as part of a bus, or to control one of the specialized signals that a bus emulator cannot.

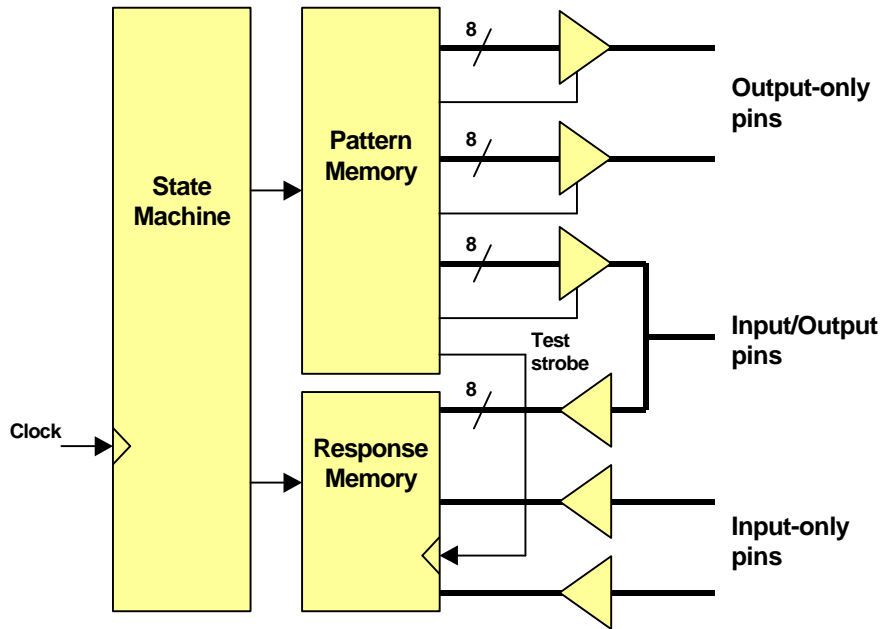


Figure 3: DWG Block Diagram

DWGs have a configuration flexibility advantage over bus emulators, but have the same limitations on timing flexibility and speed. Programmable timing resolution is still limited to the time required for one memory access cycle, and speed is similarly limited. DWGs also share the memory-and-state-machine disadvantage that the memory outputs will take some time to stabilize, so that outputs are not suitable for edge-sensitive applications without special processing or delay registers.

The capability of testing rather than simply controlling the digital portion of a UUT requires full control of timing and higher speeds. This capability has historically been found in a digital performance tester.

Performance Test

The most common way to perform final test on a circuit board or "black box" is to install it into its target system and then run it. This is usually known as "hot mock-up." Hot mock-up has the advantage of being simple to set up, but it has several drawbacks. The most obvious are that fault coverage is unknown, run times can be long, diagnosis of a bad UUT is difficult, and the performance margining needed to guarantee operation in other systems is simply not possible. And of course, if the final system is an air-to-air missile, a pacemaker, or a satellite, hot mock-up may be entirely impossible. A better solution is known as performance test.

Performance test systems are used to guarantee that an assembly will work in the final system without actually installing it in the system. A performance tester emulates the interface between the UUT and its final system as closely as possible. By precisely observing the behavior of the UUT, the tester can determine whether the UUT performs to specifications. The precise and flexible timing that a performance tester needs to emulate the interface is also used to test the interface at the limits of its specifications. This allows actual testing of the UUT's interface, instead of just controlling it. Performance testing also extends to synchronization between the tester's analog and digital instrumentation, so that all parts of a mixed-signal UUT interface can be emulated simultaneously.

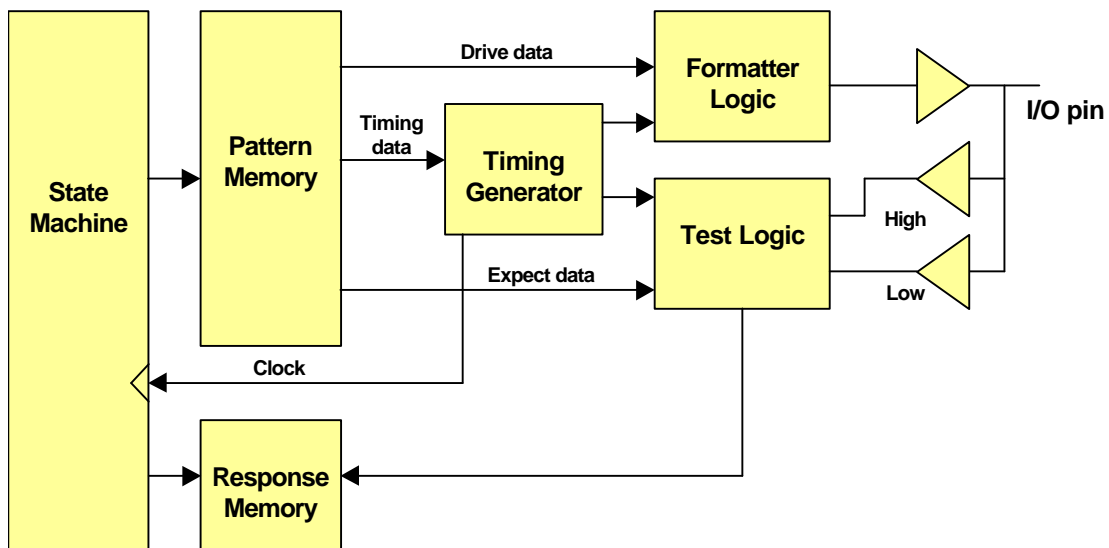


Figure 4: Performance Tester Block Diagram

Digital Performance Testers

Digital performance testers overcome the timing limitations of DWGs and bus emulators by adding specialized circuitry to control the timing of signals. The pattern memory still supplies the digital data, but does not directly control timing. A separate timing generation circuit allows precise placement of every signal transition and the generation of pulses or other “formatted” data from a single pattern memory location. Timing generation for the response side allows precise control of when UUT responses are observed. The timing generator also sources the basic clock for the system, so that this too can vary dynamically, cycle by cycle, based on the pattern memory data.

Because performance testers allow high-speed testing of a UUT’s timing performance, they also usually include additional features to aid in the task. These include specialized high-speed clock channels, a probe circuit to diagnose faults caused by aberrant signal behavior, and built-in synchronization with analog instruments to allow mixed-signal testing with the same high precision as the digital testing.

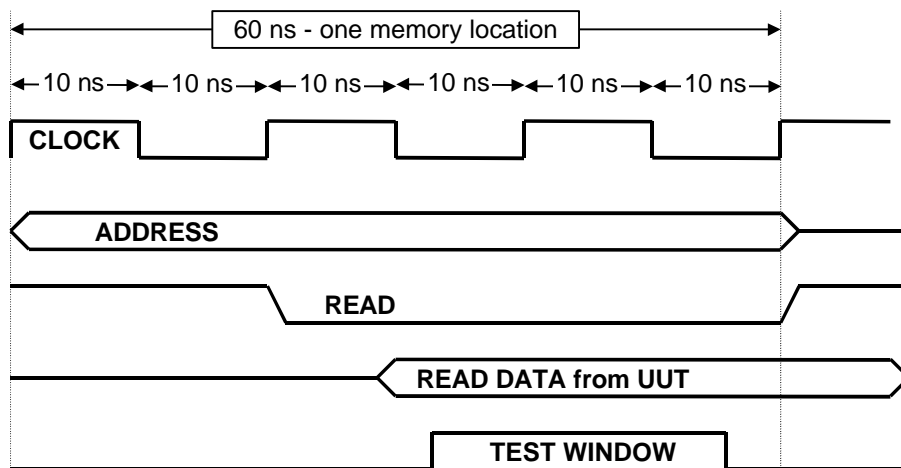


Figure 5: Performance Tester Read Cycle

A performance tester’s control over timing allows it to be programmed in cycles, rather than by signal transitions. This not only saves pattern memory, but also more closely matches the UUT’s operation. For example, the 20 MHz bus emulator’s sample read cycle of Figure 2 consists of six discrete events and consumes six pattern memory locations. It takes 300 ns to sequence through the six events. A 20 MHz performance tester’s version of this same read cycle is shown in Figure 5. This example makes use of the performance tester’s high-performance clock channel and its capability to produce multiple signal transitions per cycle. The entire read cycle not only takes just 60 ns to execute, but also consumes only one pattern memory location. This is because cycle based programming specifies the timing of the signals in a separate

memory field from the data of the signals. Both the data and the timing for each signal, including the multiple transitions, is specified in a single pattern memory location. The improved control over timing usually also includes the ability to use a *test window*, instead of just a *test strobe*. A test window monitors a UUT output for the expected behavior during a period rather than an instant in time. This allows performance such as setup and hold time to be guaranteed in a single measurement.

Digital performance testers are traditionally only available as a stand-alone test system. A system that offers a complete solution in a proprietary architecture will likely remain the way to get the ultimate in performance. But a proprietary architecture is not useful to a VXI systems integrator. The integrator who wants to test digital performance without buying an entire digital test system would like to have the core functions of a stand-alone system available in a general purpose VXI instrument. Designing such an instrument requires adapting the performance tester architecture to work in a VXI backplane.

Digital Performance Test Systems in VXI

The VXI specification was created specifically to allow implementation of the functions of benchtop instruments in a system framework. To implement the core functions of a digital test system in this benchtop environment requires careful selection of features. The goal is to keep the performance specifications of the test system while acquiring the modularity of a test instrument.

The key characteristics to preserve in a VXI-based digital performance tester are the flexibility of the channel, the flexibility of the timing, and the timing accuracy. These features must be combined with the stand-alone tester's high throughput, diagnostic probe capability, and guarantee of timing and voltage specifications across the system.

The key characteristics of a digital performance tester are:

Channel flexibility

- User-selectable number of channels
- Programmable logic levels
- No "groups:" timing selectable per channel, logic levels selectable per channel, tristate control per channel
- Any channel can be an input or an output or both (bidirectional)

Timing flexibility

- Programmable drive and detect edge placement
- Fine edge placement resolution

High accuracy

- Timing guaranteed across system, across different types of channels, at fixture output
- Automatic timing alignment

High throughput

- 10,000 or more vectors per second sustained, so UUTs can be tested in seconds instead of minutes

Diagnostic capability

- Capability to detect glitches, ringing, etc. on every channel
- Dual threshold detectors for margining
- Hand held probe
- Support of data gathering for diagnostic software

The VXI backplane would not seem to accommodate many of these requirements very well. The VXI specification does offer good standards for mixed-signal triggering and a reasonably competent data transfer bus. But the VXI backplane's open architecture appears to be at odds with the traditional digital tester's need for end-to-end control over its environment. Performance testers usually control everything from the line cord to the end of the fixture wiring in order to guarantee specifications across all channels.

A few specific challenges are:

- Making a system with a variable number of channels. This requires a variable number of VXI modules, but the design must keep true to VXI's ease-of-use philosophy. The instrument should not use external

module-to-module interconnections or require the programmer to control the modules individually. The instrument should be controllable as a single system regardless of the number of modules.

- Getting enough power to run a dense channel module from the VXI backplane without resorting to external power supply connections.
- Achieving high throughput through a variety of interfaces between the VXI backplane and controlling computer. Throughput must be achieved using a VXI backplane bus that has a theoretical maximum transfer rate of only 10 million transfers per second.
- Achieving high accuracy across a variable number of channels in different modules through a variety of VXI backplanes and UUT fixturing schemes.

Implementation of a Digital Test System in VXI

This section describes how Teradyne designed the M910 VXI digital test instrument. The goal was to keep all of the key characteristics of a custom system while overcoming the obstacles posed by an open system. This was achieved by adapting a stand-alone tester's general-purpose, high performance architecture to work in a VXI backplane.

The major drawbacks of designing in an open VXI system can be summarized by three simple facts:

- You don't control anything outside of the module
- You have to make use of only VXI Local Bus and trigger bus interconnections
- You have to fit all of the features you want into a very small space

These simple problems also have simple solutions:

- Measure and compensate for what you can't control
- Decentralize high-performance and interconnect-intensive functions
- Integrate, integrate, integrate!

All of this is achieved by a creative use of the VXI architecture and good system design practices. System design is the key: the product must be designed to work as a single system without regard to the physical implementation. The most important system design decision is how to split the functions between modules.

Block Diagram

Figure 6 repeats the simple block diagram of a traditional stand-alone digital test system shown in Figure 4. This is the type of system implemented as the M910. Since all M910 channels have programmable drive and detect logic levels, the logic level generator has been added as a permanent part of the diagram. The colored areas show where there is a split between functions that are duplicated for every channel and functions that occur once per system or are shared across multiple channels.

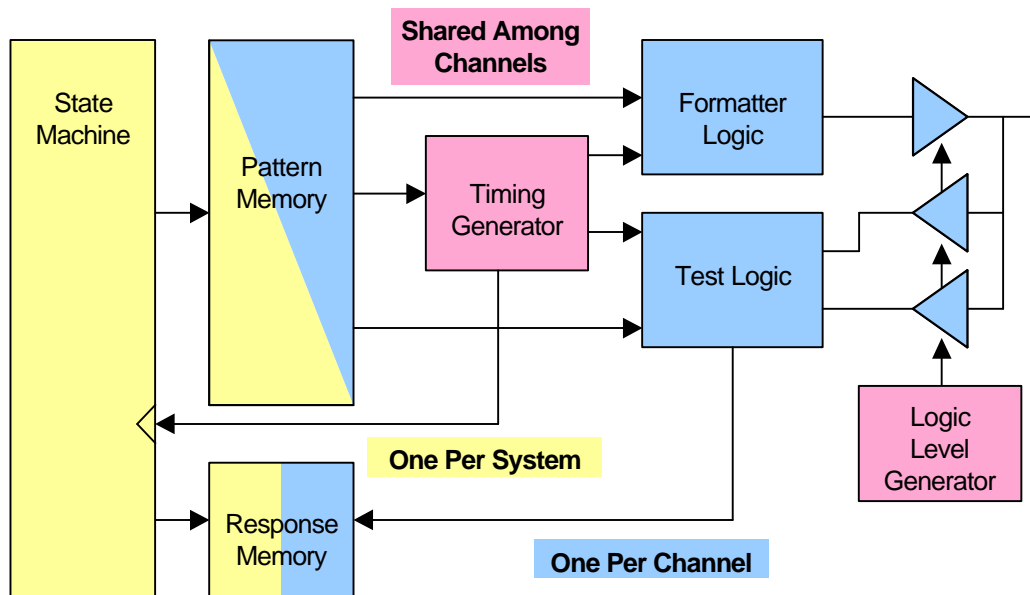


Figure 6: Performance Tester Allocation of Functions

The functions that occur once per channel are the driver and detector, the formatter logic, the test logic, the part of the pattern memory that holds the drive and expect data for each channel, and the parts of the response memory that captures test results from every channel.

The function that occurs once per system is the state machine that controls pattern sequencing and the recording of pass/fail information, and the associated pattern and response memory.

In addition to those described above, the M910 retains several other functions from a stand-alone tester. These are synchronized to the channels but occur only once per system. They include:

- The **User Clock**, which has several high performance differential outputs which are similar to, but have a higher performance than a standard channel.
- The **Sync Resources**, which source and receive synchronization pulses to coordinate testing with other instruments both inside and outside of the VXI backplane.
- The **Probe**, which can analyze responses with greater detail than a channel, and which can either be connected to any channel via a built-in matrix, or can accept data from a hand-held probe.

Lastly, two functions occur more often than once per system but less often than once per channel. These are:

- **Logic Level Generation.** These programmable D/A converters generate the analog voltage levels for a driven logic high, a driven logic low, a detected logic high, etc.
- **Phase and Window Generation.** These make up the “timing generator” function. The phase and window generators create the timing pulses that determine when a channel asserts its pattern data, when it stops asserting and returns to its programmed format, and when the input side of a channel is compared against expected data.

Implementation on Modules

The key to making all of these functions adapt to a VXI backplane without sacrificing features is to allocate the functions properly, first onto separate VXI modules, and then into ASICs.

Since allowing the user to select how many channels are in the system is a primary requirement, it seems obvious that the functions that occur once per channel or group of channels ought to be concentrated onto one module. In the M910, this is called a channel card. This allows for a varying number of channels by selecting the right number of channel cards. Concentration of channel features on to a single channel card reduces the number of interconnections required. This is critical to making the whole system work without external interconnections, since the VXI Local Bus (12 signal wires) is the only dedicated, non-shared interconnection available for this.

In the M910, the functions that are not on a channel card are located on a single overhead module, called the Central Resource Board (CRB). This allows a single board to implement functions with special interconnections, such as the mixed signal synchronization I/O. This too increases ease of use, and if the functions are split properly, interconnections can be minimized. The M910's allocation of functions across modules is shown in Figure 7.

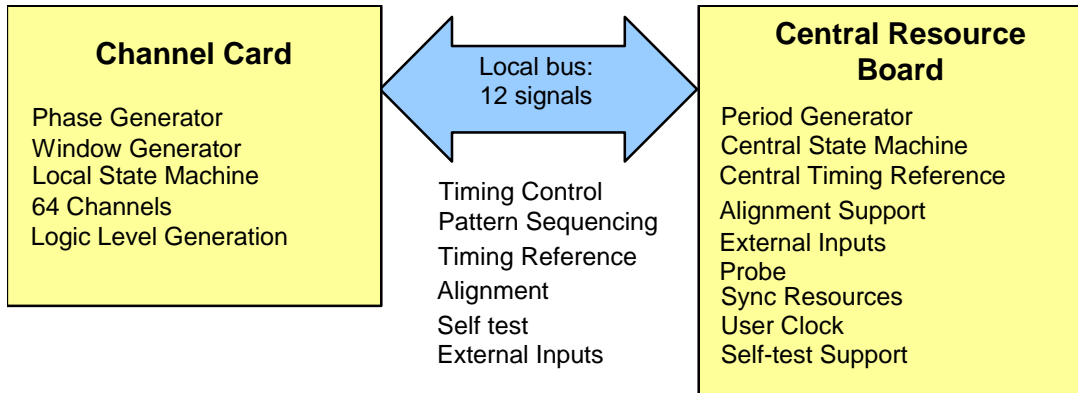


Figure 7: M910 Allocation of Functions

The interconnections between the CRB and channel card control sequencing, timing, and automatic timing adjustment across the system. There are a few signals left to pass analog signals from the channel cards back to the CRB for self-test, and to send a composite external input signal from the Central Resource Board to control the response of the channel cards.

The bandwidth and signal-to-signal phase of the daisy-chained Local Bus lines depend on the backplane, so the M910's design makes no assumptions about the Local Bus's performance. The M910 has to be able to overcome or compensate for limited backplane performance. For example, to operate within a limited Local Bus bandwidth, a high frequency reference clock from the CRB is divided down before being distributed down the backplane. Local Phase-Locked Loop (PLL) oscillators on the channel cards step the frequency back up for use by the local timing generators. Similarly, there are timing reference signals for which the phase of the signal is important. Using a Time Domain Reflectometer (TDR), the CRB automatically measures and compensates for phase differences on these Local Bus lines before use.

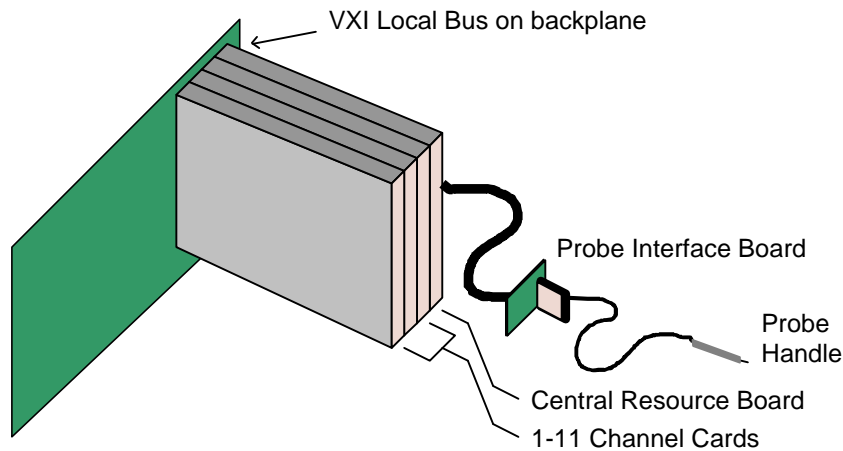


Figure 8: M910 Modules

The result of the division of functions across the CRB and channel cards is that the user can stack as many channel cards as fit in the backplane in adjacent slots next to the CRB. Every channel card adds another 64 channels, as well as another local set of phase and window generators and two sets of logic level DACs. The result is that as the number of channels increases, the timing and logic family flexibility increases too. The use of a small number of Local Bus signals also makes it possible for the channel cards to incorporate Local Bus disconnect relays, so that the last channel card in the row can automatically disconnect its left side Local Bus for protection against damage from adjacent instruments.

The CRB incorporates unique functions such as the user clock, sync resources, and probe. The probe function includes what is known as the probe handle. This is an oscilloscope-type probe used to acquire

signals from the UUT. Mounting the probe handle needs special care. This is because the front panels of the CRB and channel cards are occupied by connectors to carry their I/O to the fixture. The fixture wiring, once in place, would make it difficult for the operator to use a front panel mounted oscilloscope probe handle. A small external interface board implements enough buffering and timing alignment functions to extend the probe handle an arbitrary distance away from the CRB without affecting specifications. This is the only function not wholly contained in the VXI backplane.

System design starts with the division of functions. If carefully chosen, functions can be divided for maximum modularity within VXI's interconnect constraints, without compromising system performance.

Implementation in ASICs

The chosen split between a channel card and a CRB minimizes interconnections and maximizes flexibility, but requires a great deal of density on each module. Careful use of state of the art ASICs in a variety of technologies is necessary to get the requisite density. The channel card uses 23 ASICs of four different types:

- The two Gallium Arsenide **Timing ASICs** produce the timing signals for four drive phases and four test windows for the channels. A locally generated 1 GHz clock is used to implement an all-digital timing scheme with 1 ns programming resolution.
- The **Central Control ASIC** incorporates the local pattern sequencing. It accesses its own RAMs that contain pattern sequencing and timing selections.
- There are four **Channel ASICs** with 16 channels each, for a total of 64 channels. The channel ASICs access RAM that holds drive and expect data as well as test results data for every pattern. The channel ASIC also has built in timing adjustment circuitry and a Time Domain Reflectometer (TDR), used for adjustment of the channel's timing to the end of the fixture wiring. Most of the channel's digital self-test is also built into the channel ASIC.
- Sixteen **Analog Pin Electronics ASICs** implement sixty-four channels' worth of drivers, dual threshold detectors, and active loads. The use of a full custom ASIC allows inclusion of overhead functions such as level selection, timing alignment, and self-test in the same ASIC. A built in self-test matrix connection allows any channel's I/O pin to be connected back to self-test circuitry on the CRB.

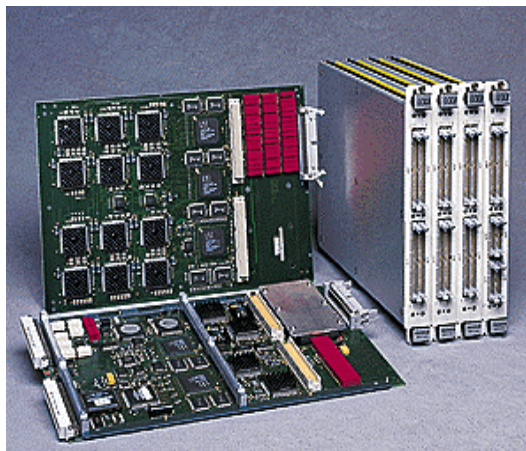


Figure 9: M910 Implementation

The four types of ASICs are also used as generic building blocks on the CRB. The separation of functions into ASIC building blocks also allows future performance upgrades by incrementally improving the performance on one ASIC. For example, changing the timing ASIC could allow finer resolution of drive phase timing, without affecting the rest of the system design.

The heavy use of ASICs leaves room for other bulky functions. On the channel card, for example, there is enough room for a disconnect relay for all 64 channels. Besides removing the digital channel connection to the UUT when not needed, opening the relays also allows self-test to run without removing external UUT connections. There is also enough space for local DC/DC power conversion, which allows the whole channel card to run off of VXI backplane supplies without supplemental power inputs. The CRB also has enough

room to build in a variety of self-test functions. Included are a voltmeter, ammeter, and current source, which can be switched onto any channel I/O pin to self-test analog performance.

By integrating almost all core functions into ASICs, the full feature set of a traditional digital tester has been preserved in VXI's small envelope. ASICs also provide higher performance and higher reliability than discrete implementations.

Backplane Interface

In order to make the system as flexible as possible, the channel cards and the CRB each occupy their own slot in the VXI backplane, and are implemented as separate modules. This means that every module has its own backplane interface. The implementation of the backplane interface determines how easy the system is to use, and how fast it can run.

The typical VXI instrument is message-based, for ease of programming in SCPI or a similar ASCII language. But the message-based protocol allows the transfer of only a few thousand commands per second. Digital instruments require the transfer of millions of bits of data for each digital test. A message-based protocol would take many minutes to transfer this amount of data. Register-based interfaces are much faster, but require programming at the register level. A complex instrument may have several thousand registers spread across several modules. A register-based instrument of this complexity is hardly easy to program at the register level. Stand-alone systems use the equivalent of a register-based protocol, but hide the complexity from the programmer with a layer of control software. Thanks to *VXIplug&play*, this is also possible in VXI.

VXIplug&play instrument drivers, by implementing high-level functions on the host computer, allow the complexity of communicating with a register-based instrument to be contained within the driver. This has several advantages:

- The bulk of the processing is on the host computer. The computer industry offers faster host computers every day. Testing throughput increases with each generation.
- The instrument driver can read the system configuration and handle the control of multiple VXI modules from the same driver. The system of modules is controlled as a single instrument.
- The data sent to and from the modules can be compressed, further increasing throughput.
- Self-test can be run on the system as a whole, providing better fault coverage and diagnostics than individual module self-tests.
- Ease of use is controlled entirely by the *VXIplug&play* driver. Programming may be implemented as function calls, a programming language, a Soft Front Panel, or all of these simultaneously. The physical communication protocols become irrelevant to the programmer.

By using *VXIplug&play* to its best advantage, the M910 can use a simple register-based interface near the theoretical maximum VXI transfer rate. This combines to make a throughput of over 10,000 vectors per second entirely achievable with any moderately fast VXI backplane interface or embedded controller.

Maintaining System Accuracy

The greatest challenge in designing a multi-module system that operates in an unknown backplane and fixturing environment is that of maintaining accuracy of specifications. Even simple specifications such as drive voltage accuracy can become difficult to maintain when, for example, there are uncontrolled power supply and ground offsets across a backplane. Maintaining timing accuracy for channel I/O has similar problems. But specifications must be guaranteed in spite of this, so that using a high precision digital test instrument near its specification limits doesn't become difficult or impossible. Even when the instrument meets its speed and accuracy specifications at the front panel connectors, practical limitations in fixturing design cause enough signal degradation and timing skew for specifications to lose relevance.

One way to maintain specifications when faced with an unknown environment is to measure and adjust for things you can't control. The key specifications for a digital test instrument that are influenced by the environment are the driver/detector voltage accuracy and the overall timing accuracy.

The environment that affects these specifications includes the backplane (power supply voltages, ground offsets, Local Bus AC behavior), the UUT fixturing (DC and AC characteristics of the cabling), as well as the usual ambient temperature, altitude, etc.

The goal is to guarantee specifications not just relative to the module's backplane and cooling environment, and not just at the front panel, but in the face of all of the environmental variations listed above. Only then can the specifications of a high performance instrument be of use to the customer.

DC accuracy. Achieving freedom from backplane power supply variations is an old problem. The most common solution is to derive all voltages from a single voltage reference that is highly immune to power supply voltage and temperature variations. However, this is not enough in a digital test instrument. The amount of current sourced by a digital test instrument can be several amperes when summed across hundreds of channels. The resistance of the fixture wiring can cause a voltage offset between the channel and the UUT. If the UUT uses significant power from an external power supply, its ground can also have an offset from the VXI mainframe ground even without taking the digital source current into account. Digital inputs must be accurate relative to the UUT ground, not the backplane ground.

The M910's solution is to provide a ground reference input on the digital test instrument. When this input is connected to the UUT ground, all M910 digital signals are referenced to the UUT ground, not the backplane ground. Providing a separate input on each module also compensates for backplane ground offsets between modules. These separate inputs are all tied together to a common point on the UUT ground. An internal relay allows the ground reference to be switched to the backplane ground when less precision is required. This simple function compensates for significant unknowns across the system.

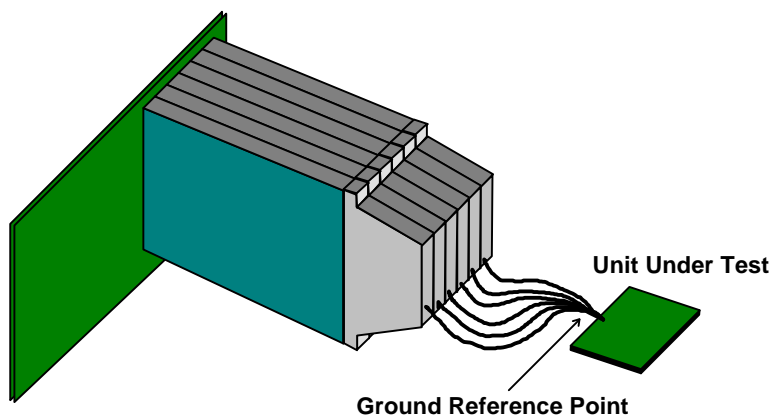


Figure 10: Compensating for ground offsets

Timing accuracy. The higher the speed of a digital test system, the more accurate its timing signals need to be. For example, if a system runs at 50 MHz (20 ns period), but two otherwise identical digital channels have a guaranteed "skew" of ± 5 ns relative their programmed position, there is only one possible placement of those signals inside of the 20 ns period that will guarantee that one precedes the other. This is illustrated in Figure 11. Maintaining equal fixture wire length can help control skews between channels, but this not only requires precision cabling, but also does not compensate for timing differences between modules caused by backplane Local Bus delays.

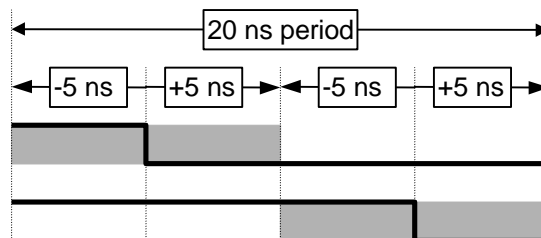


Figure 11: Timing Accuracy

Another problem is that of "round trip" time. This is best explained by example. Assume a system has perfect timing accuracy, but the wiring from the channel to the UUT is about 4 feet of coaxial cable with a propagation delay of about 8 ns. When a channel asserts at 0 ns into a cycle, the signal will not arrive at the UUT until 8 ns later. The UUT sees this as "time zero," even though it is 8 ns later than the test instrument's "time zero." If the UUT then responds to the stimulus after a delay of 10 ns, the response does not arrive back at the test instrument until $8 + 10 + 8 = 26$ ns after the channel assert time. So the test instrument must be programmed to expect the response at 26 ns, not at the 10 ns expected. The example's "round trip" time of 16 ns varies for every channel and for every fixture wire.

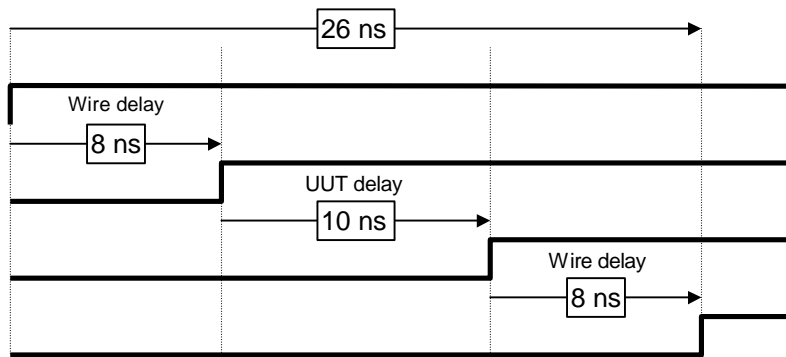


Figure 12: Round-trip Time

To correct for these types of timing errors one needs to measure and compensate for them. This requires the capability to

- Measure the electrical length of the backplane
- Measure the electrical length of the fixture wiring
- Compensate for the total timing difference between the fastest and slowest channel
- Compensate for the “round trip” delay for each channel

The M910 accomplishes all of these functions. The CRB has a built in Time Domain Reflectometer (TDR) that uses a patented method to measure the electrical length of the Local Bus lines on the backplane. A similar patented method is built in to each channel ASIC. As long as a 50 Ω fixture cabling environment is maintained, the channel TDR can measure the electrical length of an unloaded cable, even if this length is different for each channel. Patented timing adjustment circuitry in each channel ASIC then adjusts the driver and detector separately, compensating for fixture cabling delays, backplane delays, and “round-trip” delays. This automatic timing alignment can be invoked by the test programmer as a *VXIplug&play* driver function.

Timing specifications and timing alignment provide their value to the user only when actual use of the system is taken into account. The highest performance can only be realized when the system can adjust itself for real-world conditions.

Summary

The M910 is successful in transferring the key features of a stand-alone digital tester into a VXI form factor. This is achieved by:

- Adapting the architecture to fit within the limits of the VXI Local Bus
- High levels of integration in ASICs
- On-module power handling
- Measuring and adjusting for unknown operating conditions
- Concentrating the complexity of the control in the *VXIplug&play* driver

These methods are all part of a *systems* approach to design. The M910 is designed not as individual modules, but as a system that includes multiple modules, the backplane, the chassis, the fixturing, and the *VXIplug&play* driver. This allows the best features of a stand-alone digital tester to be implemented while sacrificing none of the advantages of VXI. In the M910, just as with VXI, the whole is truly greater than the sum of its parts.