REMOTE DIAGNOSTICS AND UPGRADES

Tim Pender - Eastman Kodak Company
timothy.pender@kodak.com

1 ABSTRACT

Developing a product with remote diagnostic and upgrade capabilities drastically improves time to market, reduces field service cost, and prevents product downtime.

By employing a system level JTAG architecture into a product, it is possible to remotely perform comprehensive boundary scan tests, re-program Electrically Programmable Logic Devices (EPLD) and In-System Programmable (ISP) configuration proms, or even re-configure a Field Programmable Gate Array (FPGA) as a Built In Self Test (BIST) controller to test peripherals at speed.

2 INTRODUCTION

System Level JTAG can be thought of as a strategy that allows access to all board JTAG chains in a system. An optimum strategy would allow any board chain to be accessed via software control. A remote JTAG enabled product has a dedicated JTAG controller that can access any JTAG chain. Simply providing the latest programming files, boundary scan test vectors, and a sequence file enables self-upgrades and diagnostics.

Without a system level JTAG strategy field service would need to board swap or connect a programming cable to each board to upgrade products. A system level JTAG strategy eliminates the need for on site field service.

This paper will discuss practical system level JTAG concepts such as scan chain partitioning and BIST Controller Intellectual Property (IP) used in conjunction with Altera's Jam Byte Code (JBC) JTAG controller software. In addition, detailed signal-conditioning techniques will show how to improve JTAG signal integrity.

To help avoid confusion, this paper will be consistent when using the terms re-programming and re-configuration. Re-programming will be limited to the use of the JTAG interface to change the contents of a non-volatile EPLD, configuration prom, or Flash memory. Re-configuration, on the other hand, relates to the act of changing the contents of volatile SRAM based FPGAs.

3 BENEFITS

Test economics determine which JTAG architecture best suites your product. For instance most products can’t absorb the cost of embedding a Commercially Off The Self (COTS) JTAG tester because that would increase the cost of the product by thousands of dollars. A cost-effective alternative is to use Altera’s Byte code software as an embedded JTAG controller. This source code can be modified to support, Linux, Addressable Scan Ports (ASP), and custom JTAG partitioning devices.

Boundary scans test can be reused throughout the entire lifecycle of the product from prototype, to manufacturing, to system integration, and finally field service. This is a substantial reduction in test development costs.

For products with short shelf lives, time to market is key. A product with remote upgrade ability can allow the sale of the product before final firmware is released as it can be upgraded in the field. This creates huge potential for larger market share and increased profit. Also if new features or enhancements to firmware are requested, it can be done almost instantaneously. This prevents product downtime and can also extend the life of the product.

The following example will compare tradition field service vs. remote upgrades of 5000 products in the field.

<u>Traditional Field service</u>
Update time = (5000 prod) (4 hrs/ prod)
= 20,000 hrs or 500 weeks
Update cost = (20,000 hrs) ($75/hr) =
= $1.5 Million (per upgrade)

<u>Remote Upgrade</u>
Update time = 1hr
Development cost =
(2 Engineer Yr.) (2000hr /yr.) ($100/hr) = $400k
Hardware cost = (5000 prod) ($20 / prod) = $100k
Upgrade cost = $500k (one time cost)

Traditionally it would cost $1.5 million and take 100 field service engineers 5 weeks. On the other hand Remote upgrades can happen in parallel and it only consumes about an hour of machine time to download and update the all of the products. The cost of the remote upgrade is mainly in the up-front development. It is important to understand that the remote upgrades cost is a one-time cost while tradition field service methods are reoccurring.

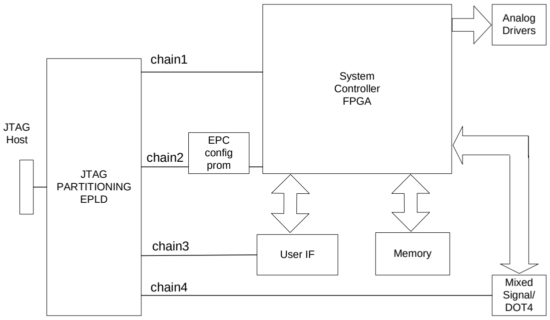


Figure 1- Remote Diagnostic Board Block Diagram

4 DESIGN OVERVIEW

Figure 1 is a block diagram of the Remote Diagnostics Board (RDB) design; it was developed for proof of concept. The board contains a system controller FPGA, JTAG partitioning EPLD, Memory, Mixed Signal, Analog, and User Interface blocks.

Two different methods of embedding test into a product will be discussed.

One method relies on embedding JTAG controller software, test vectors, programming files, and a JTAG interface within the product so it has dedicated resources. The JTAG controller can be remotely controlled to access any chain in the system. Altera's free JAM software (SW) can be used to execute test vectors converted from serial vector format (SVF). The emphasis of this approach is to provide JTAG partitioning support of several JTAG hosts and chains. An EPLD programmed with JTAG MUX VHDL code is used to provide the JTAG host with access to any chain(s). Section 4.1 will describe JTAG MUX IP in detail.

The second method utilizes an FPGA that is already in the board design to functionally test its peripherals. VHDL code for both the system controller and an Embedded BIST Controller can reside in two different pages of memory within an Altera EPC4 ISP configuration prom. Through a JTAG instruction one of eight pages of memory will configure the FPGA. Once the FPGA is configured as a BIST Controller, execution of individual BIST routines can be launched through the chain one interface known as the “soft” Test Access Port (TAP). The soft TAP is a JTAG TAP controller that has been programmed into the FPGA to provide a JTAG interface. This interface allows BIST execution and read-back of the BIST results. Section 4.2 will describe the FPGA BIST controller in detail.

4.1 JTAG PARTIONING

JTAG is a popular interface to debug hardware, it is used for boundary scan testing, in-system programming, digital signal processor (DSP)/microprocessor (uP) emulation, and is even now used for FPGA firmware (FW) debug. The question becomes, what are the limitations of the silicon vendor debug tools and how does this effect your JTAG access strategy?
For instance an FPGA vendor tool will only work if native devices are in the chain. Some devices have non-compliant JTAG interfaces, some want only one target device in the chain, while effective boundary scan tests need all JTAG devices in the chain.

Most silicon vendor JTAG software do not support JTAG partitioning ICs, therefore scan paths need to be externally controlled and must look transparent to the host. A proprietary solution was devised using programmable logic devices since no COTS solution was flexible enough to meet our needs. The intent is to incorporate a partitioning IC on the board that will allow all chains to be easily accessed while also providing a JTAG point of entry when integrated into a system.

The RDB contains four scan chains. Chain1 is connected to IO pins on the FPGA and depending on the configuration loaded may or may not have soft TAP functionally. Chain 2 has two devices, the Altera EPC4 and FPGA. Chain 3 consists of three 74abt18245 buffers. Chain 4 is a National Semiconductor STA400 Dot 4 chip. The RDB uses the remote version JTAG MUX to manage these chains.

The following sections will discuss the functionality of both the local and remote JTAG MUX. The following IP supports four hosts and four chains however these can be increased rather easily to suit the application.

4.1.1 JTAG MUX IP (LOCAL)

The local version JTAG MUX code is very small and can easily fit into an Altera Max 3032 (approx. $1 part). The code consists of up to four JTAG hosts and four JTAG chains. The local version of the JTAG MUX requires external signals to determine which host will be used and how to connect the chains. These bits are referred to as host present bits and chain present bits.

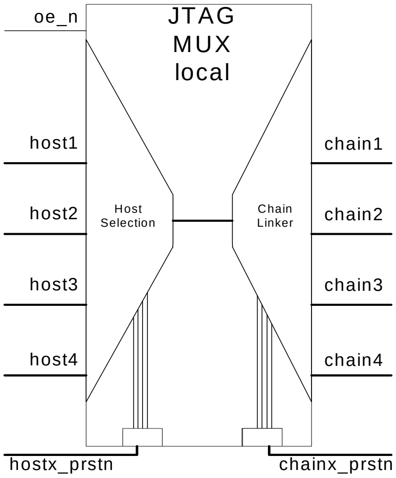


Figure 2- JTAG MUX IP (local) JTAG partitioning IP

4.1.1.1 HOST PORTS

Examples of JTAG hosts are Asset, Xilinx parallel IV, Altera Byteblaster MV, and DSP/uP Emulators. The JTAG MUX will select one host based on the host's present bits. The present bits should be pulled high on your board. When a low is detected it indicates that a JTAG Host is connected. If multiple hosts are connected, i.e. more than one present bit is low, the host with the highest priority will prevail. Host 1 has the highest priority and Host4 has the lowest. Typically a cable adapter is created so that by plugging into the JTAG header, the host present signal will be driven low automatically

4.1.1.2 CHAIN SELECTION

Chains can be isolated or daisy chained depending on the state of the chain present signal. The chain present signal should be pulled up on the board, so that if a low is detected on a chain present pin that chain will be inserted in the chain. If a single chain present bit is low then the corresponding chain will be connected to the JTAG host. If multiple chain present lines are low that indicates that the corresponding chains will be daisy-chained. There is a guaranteed order, Chain1 is closest to the host TDI and chain 4 is closest to the host TDO. Figures 3 and 4 illustrate chain selection. By installing a jumper the associated chain will be connected to the host. Figure 5 shows how the chain selection can be dynamically controlled by external test equipment.

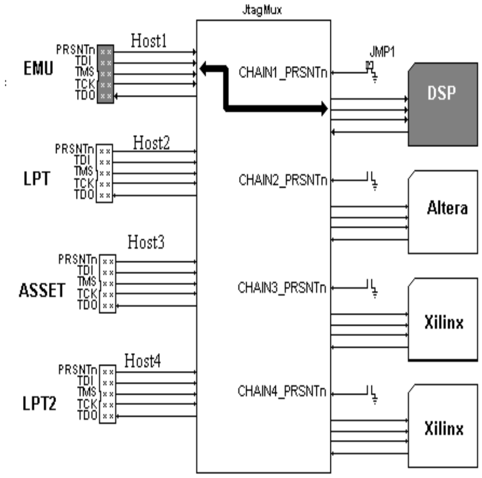


Figure 3 – Emulator Host drives the DSP, chain1 Present pin is connected to ground through JMP1.

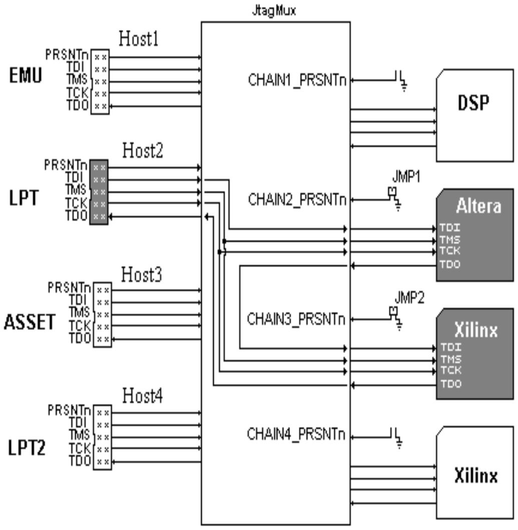


Figure 4 - Altera and Xilinx devices are daisy-chained, chain1&2 Present pins are connected to ground through JMP1 & JMP2.

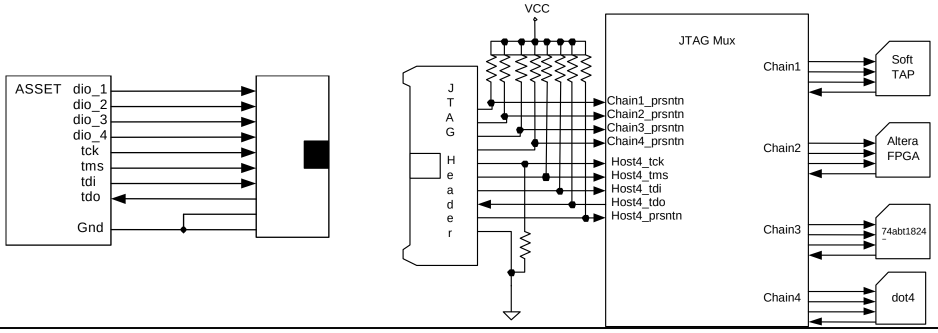


Figure 5 – Plugging in the host adapter cable will automatically ground the host present bit, the DIO from Asset will drive the chain present bits to dynamically select the targeted chain(s)

4.1.2 JTAG MUX (REMOTE)

The JTAG MUX remote version contains all the functionality of the local JTAG MUX however contains additional JTAG registers that allows the JTAG MUX to be put in software mode. Software mode is controlled by HOST4 and allows the chains to be selected by register values rather than chain present bits. Host4 is a special port that can be operated as a standard JTAG host or used to read/write JTAG registers. When the CMD_SELn line is asserted low the internal state machine will use host4 to read or write internal JTAG Registers. A software driver has been created that will generate the protocol necessary to access the registers, so the protocol does not need to be fully understood by the user. The JTAG MUX registers can be accessed by using the function: JTAG_reg(Rd_Wrn, &Address, &Data). This function will generate the appropriate signals on CMD_SEL and Host4 to read and write the JTAG registers.

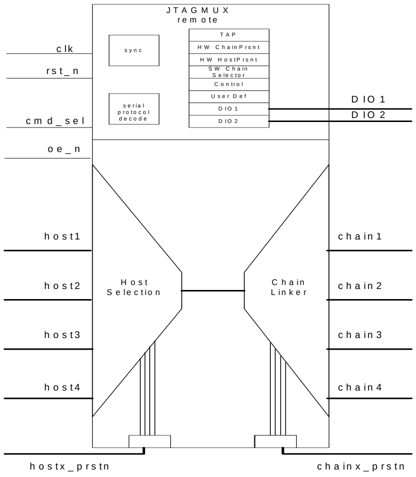


Figure 6 - JTAG MUX IP (remote) includes JTAG registers that can control chain selection

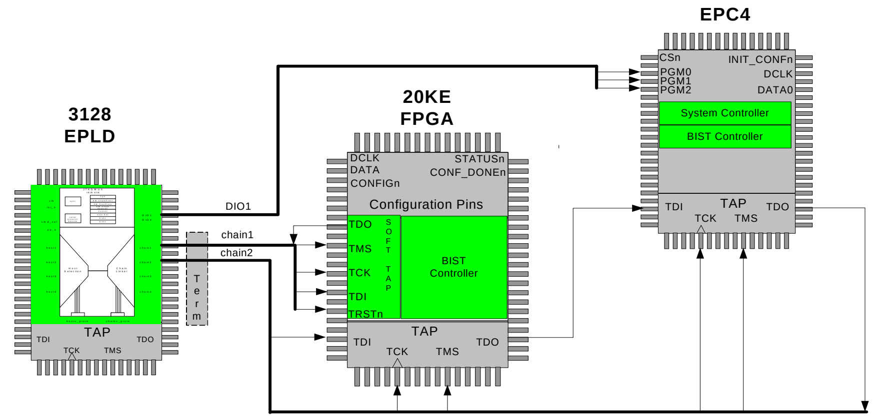
This section explained how to get access to all the chains in the design using JTAG MUX IP. By accessing any chain comprehensive boundary scan tests can be developed.

4.2 BIST

This section will describe an innovative way to implement built in self-test on a board. This approach utilizes a centralize FPGA on the RDB and configures test bench code to test its peripherals at speed. The test bench code and operational code can reside on the board but don't compete for the FPGA resources because they are mutually exclusive. This alternative approach has an FPGA designer develop VHDL BIST code. Traditionally SW engineers develop built in self-test to check the operation of the system. This approach gives flexibility to assign alternate personnel to the BIST task.

4.2.1 EPC ISP CONFIGURATION PROM

Altera's enhanced configuration proms are re-programmable via JTAG and have densities up to 16 Mbits. The EPC4 used on the RDB contains the code necessary to configure the FPGA. The EPC4 can store up to eight different configuration pages of memory. The page that is loaded depends on the logic levels on the Page Mode pins (PGM2:0) when configuration is enabled. On the RDB the JTAG MUX DIO register controls these pins, see Figure 7. Configuration occurs at power up or with the INIT_CONFIGURATION JTAG instruction. Programming times range from 2-4 minutes for these devices, however once they are programmed, the configuration can be changed on the fly in seconds. One configuration may be normal operation code when another



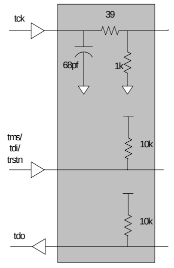


Figure 7– The JTAG MUX EPLD is used to isolate chains 1 and 2 as well drives the PGM pins on the EPC4 determining which configuration profile will be loaded into the FPGA. The FPGA is shown with the BIST Controller configuration.

configuration can turn the FPGA into embedded instrumentation to functionally test its stepper motors, solenoids, fan, ADC, DAC, memory, and interconnects.

When updating the EPC4 remotely it would be advantageous to program a new page of memory, this would ensure if an error occurred during an update your product would not be hung at next power up, it would simply load the last good configuration. Altera Quartus software does offer a “difference” programming file option that will only program the EPC with the difference between two revisions however it converts it directly into a JAM or JBC file. This would be acceptable if the EPC4 is the only device in the JTAG chain, however JAM and JBC files contain all chain info as well as the algorithm and data to program a chain. Outputting this difference directly to JAM/JBC does not give the user the chance to specify what the JTAG chain looks like. Possibly the next version of Quartus will fix this but in the short term you may want to leave your EPC4 in its own chain.

4.2.2 BIST CONTROLLER FPGA

The FPGA already in the design is going to serve two purposes, when in normal operating mode it will be a system controller and when in test mode it will be a BIST controller. MemBIST, DAC/ADC calibration, Stepper motor control, and monitoring voltage or temperature BISTs can be performed.
The MemBIST portion of code was created by Mentor Graphics Corporation (MGC) MemBist Architect tool. MemBIST Architect is intended to test internal ASIC memory. The same concept can be used to test external FPGA memory. The MemBIST VHDL routine can be created using many different algorithms.

Additionally MGC BSD Architect was used to generate a fully functional TAP controller. This soft TAP becomes the JTAG interface to the BIST Controller. The soft TAP can support additional BIST routines by modifying soft TAP VHDL and add new instructions to the instruction register decoder. An SVF file was created manually and converted to JBC so that the JTAG controller could execute the BIST routines.

When the FPGA is in normal mode, functioning as a system controller, chain1 has no function within the FPGA, however when configured as a BIST Controller chain1 is connected to the soft TAP. Care must be taken when selecting chains, note it would not be advisable to have both chain1 and chain2 daisy chained, since chain2 contains the FPGA “hard TAP”. If the hard TAP were put in a test mode such as EXTEST the chain would immediately be broken since the “soft TAP” core logic is isolated from the FPGA pins in this mode.

5 Physical Guidelines

5.1 TERMINATIONS

The Test Data Output (TDO) is normally tri-stated unless in the shift Instruction Register (IR) or shift Data Register (DR) states. When shifting data through the JTAG chain, TDO drives Test Data Input (TDI) in a daisy chain configuration. When TDO is tri-stated, the internal pull-up on TDI terminates the signal. When TDO drives programmable logic or a

CMOS buffer, TDO must be terminated to prevent damage to the CMOS inputs.

When terminating JTAG signals on the board it is important to know the approximate values of the internal terminators. Many devices that use the optional TRSTn pin are non-compliant for various reasons. The dot 1 standard says there should be an internal pull-up on TRSTn so that if TRSTn is left unconnected boundary scan operations will still work. If a device has an internal pull-down and is left unconnected the TAP is stuck in Test Logic Reset and will prevent boundary scan testing. The caveat of non-compliant internal terminators is voltage dividers can be formed leaving TRSTn in an unknown state. Suppose there are two devices on a board, one has an internal 5k pull up, and the other has a 5k internal pull-down. When these TRSTn pins are bussed together it forms a voltage divider leaving the TRSTn in the indeterminate region.

5.2 BUFFERING

The low voltage LVT and LVC drivers have been popular because of strong drive and 5 volt tolerance. Powerful drivers have very low internal impedance causing very quick rise times thereby creating reflections on lengthy nets. These drivers also have short propagation delays. If a glitch of noise appears on the input of the driver greater than the drivers propagation delay a false trigger will occur.

5.3 SIGNAL INTEGRITY

So why should JTAG need signal integrity, it is a slow bus right? Wrong, signal integrity of TCK is as essential as any clock signal. If you have signal integrity problems with your JTAG signals you will not be able to program EPLDs, use emulators to debug DSP / uP code, or perform boundary scan tests.

Below is a formula to estimate when a net will look like a transmission line. The rise time (Tr) of the driver chip is the single variable. If the rise time of a driver is 1nS then traces over 1.5 inches long with have noticeable reflections unless compensated.

$$\boxed{\text{Trace Length (inches)} > Tr * 1.5\text{inches/ns}}$$

The rise times from databooks will often be specified with a 50pf load, but your actual circuit load may be considerable less. The slower the rise time is, the longer the trace can be without worrying about reflections.

Whenever possible try to maintain point to point connections between the TCK driver and load. This will require one buffer, one trace, and one series compensation resistor for each TCK in the design. The series compensation resistor should be impedance matched so that the source driver impedance plus the series compensation resistor is equivalent to the PCB trace impedance.

If routing resources are available this approach is easiest to implement. This is similar to a how most clocks are managed, see Figure 8.

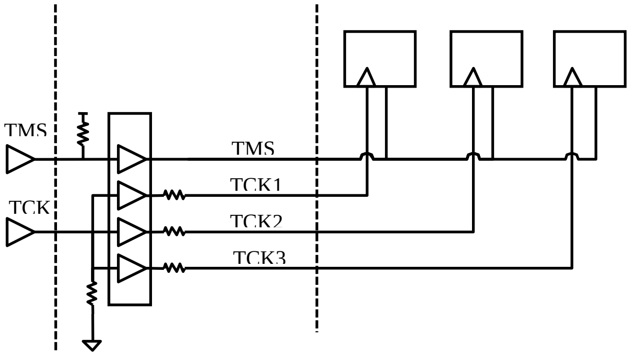


Figure 8 –Recommended TCK tree creates point to point connections. Reflections on TMS usually settle in a TCK half-cycle

When point to point connection are not possible there are other techniques that can clean up bussed signals. Two circuits with different routing methods will be analyzed to show how to improve the signal integrity of bussed TCKs. The circuits were simulated in Hyperlynx software and the resultant waveforms were probed at the receiver and displayed right of each compensation method.

Rise and fall times of the TCK driver, the length of the trace, the layout routing method, and the number of loads will all affect signal integrity. When performing a signal integrity analysis of TCK what you need to look for is whether any receivers will see irregularities on the edges or fluctuation at the rails that fall in the indeterminate region.

Circuit 1- An LVT244 buffer drives 3 ABT loads. The loads are 6inches from the driver and are distributed 1inch apart. Figure 9-11 shows different source side compensation methods. The 1K pull-down is simply a resistor to terminate TCK when the driver is tri-stated it has very little effect on the simulations

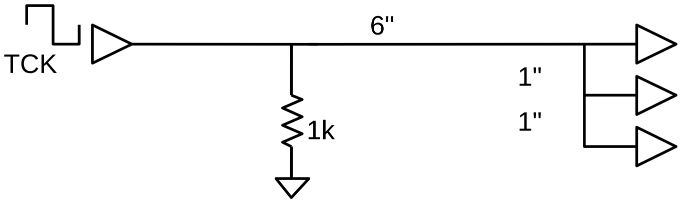
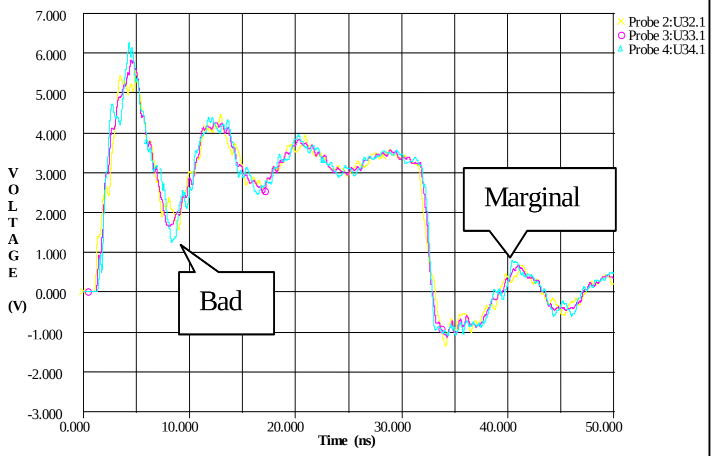


Figure 9 - NO COMPENSATION

The waveform shows fluctuations on TCK upper rail into intermediate area for about 3ns (BAD). Fluctuations at the lower rail are marginal since they exceed .8v for less than 1ns.



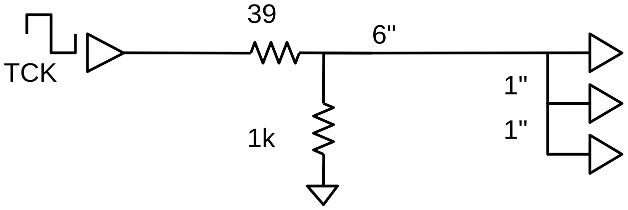
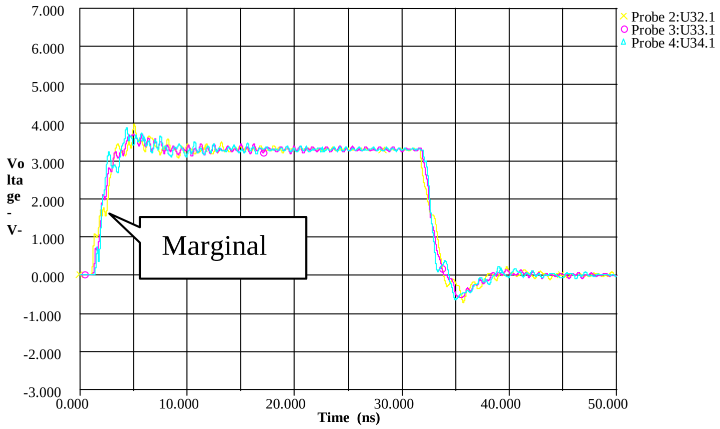


Figure 10- SERIES COMPENSATION
The waveform shows fluctuations on TCK edge near intermediate area. Considered marginal since the transitions are only about 1ns. Impedance matching with a series resistor did stop the ringing though. (A single load would have clean edges)



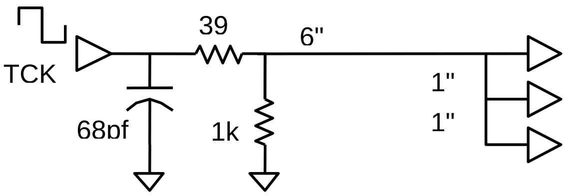
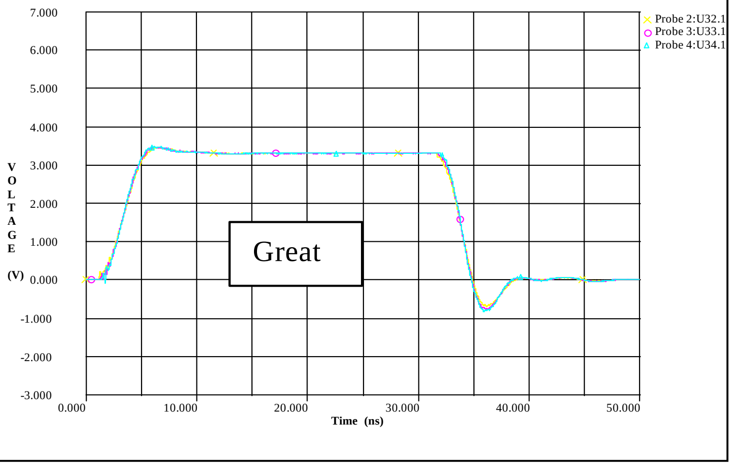


Figure 11 - SERIES COMPENSATION & CAP
The capacitor at the output of the driver slows the rise time down to from 1ns to 3.5 nS.



Circuit 2 - Source side compensation

An LVT driver is connected to two LVT loads in a star configuration. The star configuration itself tends to minimize reflections. Figures 12-14 show source side compensation while 15-17 shows load side compensation.

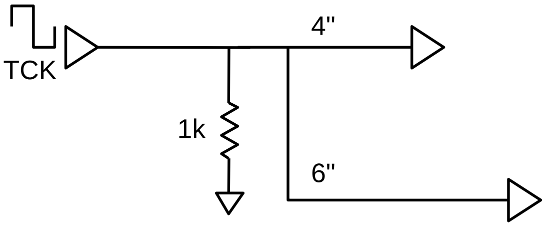
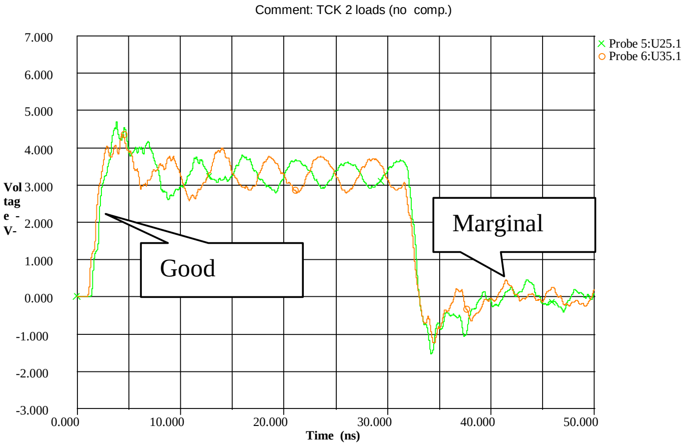


Figure 12 -NO COMPENSATION

Good edges, lots of ringing, and has fluctuations at the lower rail near 0.8v but for a short duration. Ok without compensation because the origin of both load traces is near the driver output.



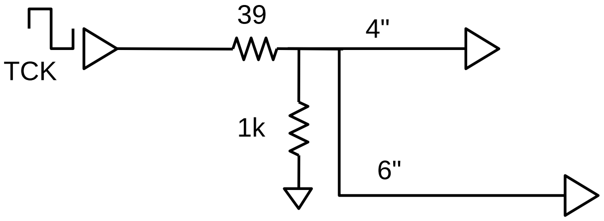
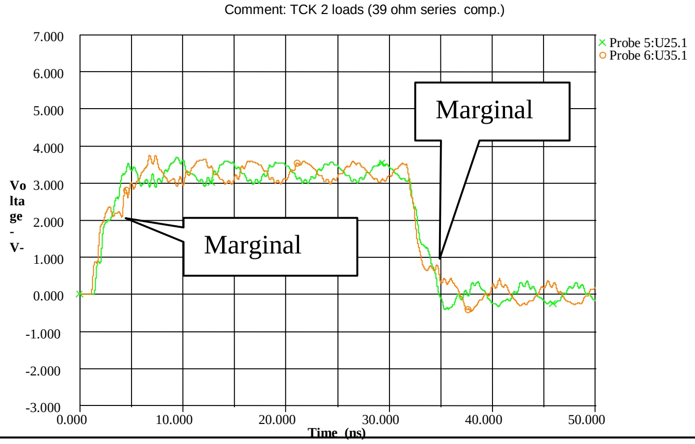


Figure 13 -SERIES COMPENSATION
Adding compensation made the both edges look worse, however it did help the ringing slightly.



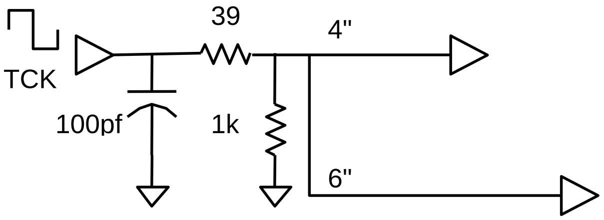
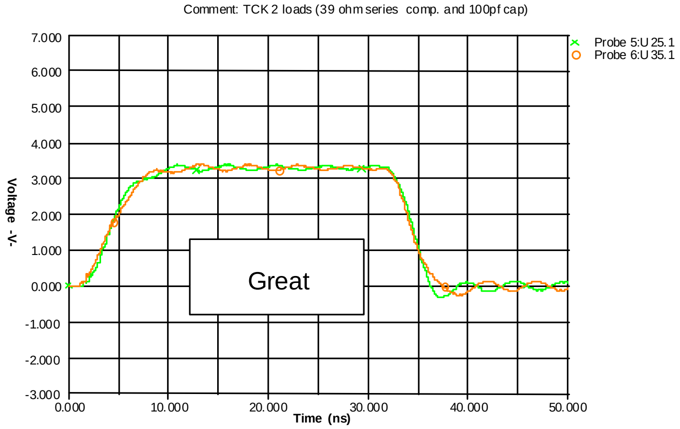
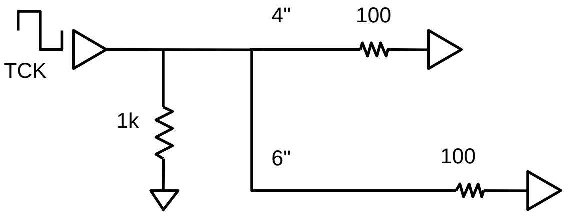


Figure 14 - CAP & SERIES COMPENSATION
Once again slowing the rise time really improved the waveform.



Circuit 2 - Load side Compensation
Alternatively adding series resistance at the input of the load will create an RC circuit where C is the input capacitance of the load. No source side compensation was used.



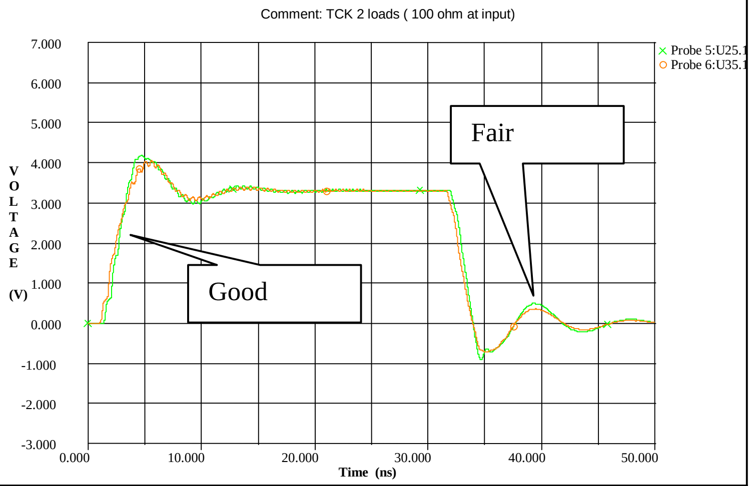
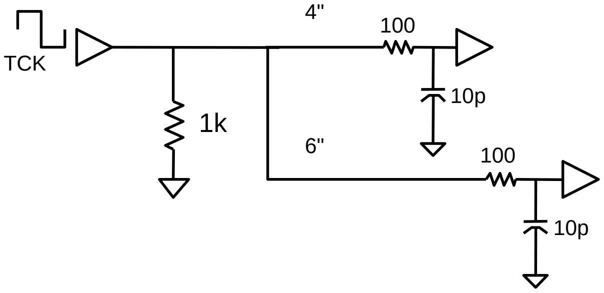


Figure 15 - INPUT RESISTOR
This also cleaned up the signal and uses less power. The lower rail peaks at 0.5 volts, which is ok.



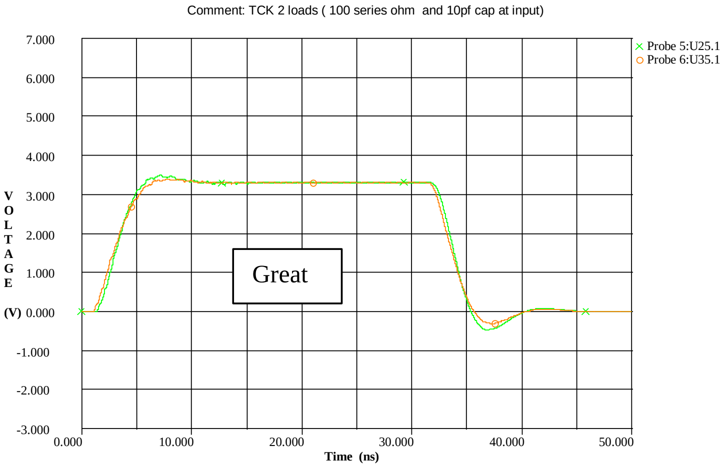
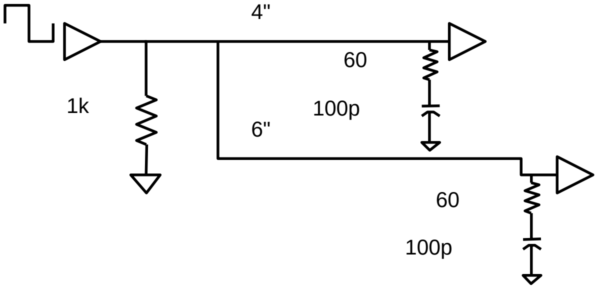


Figure 16 - INPUT RESISTOR & CAP
The new C value is 10pf + Cin which slows the edges down even further.



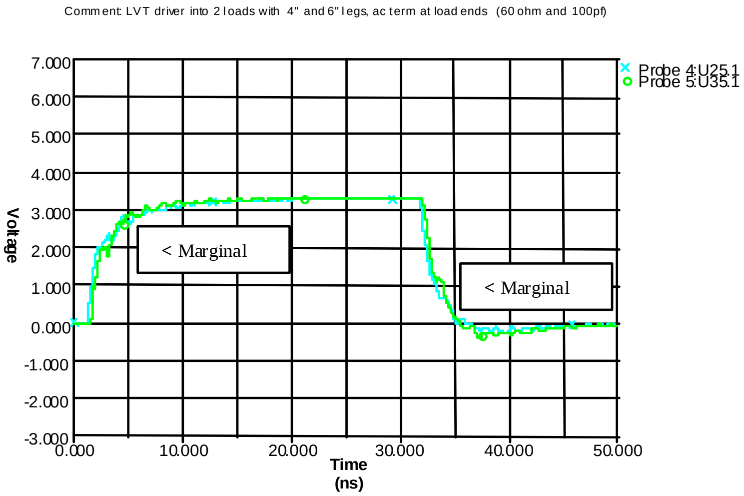


Figure 17 – AC Terminator
The AC terminator is considered marginal since both edges have glitches. AC terms are easy to apply for corrective measure since they are not in series.

6 CONCLUSION

The remote diagnostics and upgrades approach discussed in this paper uses a low cost JTAG controller to access any JTAG chain in the system. Because the JTAG controller can access any chain in the system through software, no on-site technical expertise is required to test or upgrade the system. By using COTS remote control software or remote access already with the product, the JTAG controller can be used to test or upgrade firmware remotely.

Accessing all chains in the system requires a system level JTAG strategy. There is a growing number of COTS JTAG partitioning solutions but most don't offer the flexibility of IP. FPGAs offer programmable logic levels, slew rate control, expandability, and the capability to solve unexpected problems with non-compliant devices. Yes this does require some investment to develop the IP but the cost savings associated with remote diagnostics and upgrades easily offsets the IP.

The JTAG Mux IP approach incorporates VHDL in an EPLD that supports multiple JTAG hosts, multiple scan chains, and accommodates the restrictions posed by non-compliant devices and software. This IP satisfies all host requirements for chain isolation or daisy chaining. The local JTAG Mux chain selection is controlled with digital inputs while the JTAG Mux remote IP can also use software mode chain selection.

The FPGA BIST controller is an example of how to embed a high-end tester into the product to test at speed. This method does not load any signal down as external test equipment would.

It is important to treat TCK as any other clock during board layout to ensure signal integrity.

Enormous benefits can be achieved by designing remote diagnostics and upgrades into your product; it can be the competitive edge.

7 Acknowledgement

I would like to acknowledge fellow Kodak team members George Simmonds, Bill Staub, Tom Fisher, Bob Costa, and Pat Keenan for helping to make the Remote Diagnostics project a success.

8 REFERENCES

1) IEEE Computer Society. "IEEE Standard Test Access Port and Boundary-Scan architecture - IEEE Std. 1149.1-1990", IEEE, New York, 1990.

2) "Enhanced Configuration Devices (EPC4, EPC8, & EPC16) Data Sheet", Altera Corp. 2002

3)" Configuration Devices for SRAM-Based LUT Devices Data Sheet", Altera Corp. 2002

4)"AN 122: Using Jam STAPL for ISP & ICR via an Embedded Processor " Application Note, Altera Corp. 2000

5) Leon van de Logt, Frans de Jong, Frank van der Heyden, "System level test on a prototyping board using a switch matrix", *International Test Conference BTW 2002*

9 ABBREVIATIONS

ASP	Addressable Scan Port
COTS	Commercially of the shelf
DIO	Digital input/output
DR	Data Register
EPLD	Electrically Programmable Logic Device
FPGA	Field Programmable Gate Array
FW	Firmware (contents of EPROMS or FPGAs)
HW	Hardware
IP	Intellectual Property (VHDL)
IR	Instruction Register
ISP	In-System Programmable
JAM	Altera developed Standard Test and Programming Language
JBC	Altera's Jam Byte Code PLD programming file format
JTAG	Joint Test Action Group - IEEE 1149.1
LPT	PC Parallel Port
LVC	Low Voltage CMOS
LVT	Low Voltage TTL
MUX	Multiplexer
PCB	Printed Circuit Board
RDB	Remote Diagnostics Board
STPL	Standard Test and Programming Language (JEDEC JESD71) developed by Altera
SVF	Serial Vector Format
SW	Software
TPD	Propagation delay of a gate
VHDL	Programming language for FPGAs and EPLDs