An extension to JTAG for at-speed debug on a system.

Leon van de Logt, Frank van der Heyden, Tom Waayers
Philips Research Eindhoven, The Netherlands
{leon.van.de.logt, frank.van.der.heyden, tom.waayers}@philips.com

Abstract

When developing new designs, debugging the prototype is important to resolve application malfunction. During this board design debug, often a few pins of an IC are measured to check signals. Access to these pins is becoming more difficult due to packages like BGA. The JTAG port is an efficient mechanism to gain more access to the ICs. A method is presented to reconfigure the boundary scan chain to any desired length and to access pins involved in the debugging. The method is used asynchronously or synchronously to the test clock. In asynchronous mode high transfer frequencies are possible. For synchronous mode two different variants are described where the data throughput is determined by the intermediate logic. Both modes have proven to work on an FPGA and all implementations fully retain compliancy to the IEEE1149.1 standard.

Introduction

Development of an application or system consists of several stages. For volume production (for example consumer applications like TV or DVD) debugging the design is crucial. Otherwise, errors in the design will be present in every product of a million pieces production. Before full production starts, prototypes will be developed. Test and debug of these prototypes can be done using different techniques. A few of them are:

- Structural test. With the IEEE1149.1 standard, interconnections are validated on shorts and opens. Advantage of this method is, that it provides information about the integrity of the board without having the board functional. Hence, disadvantage is that it doesn't tell anything about the functionality
- Functional tests will validate board interconnections that cannot be resolved with structural tests. Often, these tests only indicate user level functionality. Intermediate signals cannot be explored.
- Flying probe test is often not used during prototyping. The small amount of prototypes does not justify the time needed to set up a test program
- Debugging. Typical for prototyping where the actual design is not fixed yet. Here, incorrectness's in the design can be solved using conventional techniques as a logic analyser, oscilloscope and so on.

So, during prototyping, board level debug or system debug is meant to resolve design errors. This is an important step before high volume production can start. The focus will be on aspects like:

- Design faults in the system,
- Board layout faults
- Software faults
- Chip design faults in the application (emc, dissipation).

All these aspects are important in the total design debug step to evaluate correct functioning.
For doing design debug fast and efficient access to the signals of ICs is of prime importance. Traditionally, contacting the available pins on the PCB to monitor the signals does this. As the use of small outline packages and BGA grows rapidly it is more and more difficult to do measurements on the PCB. Probe access to the pins is then not obvious at all as illustrated by figure 1.

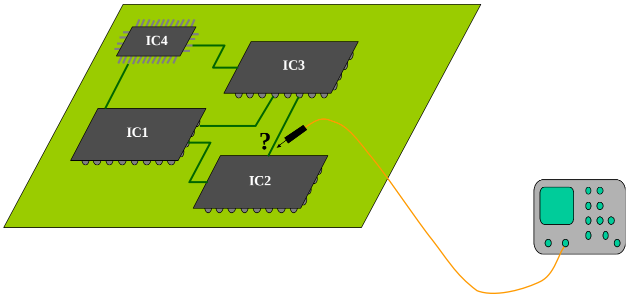

Figure 1: *An access problem when debugging modern PCBs. Normal probes cannot contact hidden pins*

The large pressure on material costs for producing PCBs causes the application to take fewer surfaces on the board, also for prototypes. Adding additional test spots and/or connectors for use with test needles is often not feasible because of space, signal integrity and major redesign issues. Now, designers are looking for a trade off between board space and the required debug access capabilities.

Currently, data streams and/or signals not accessible through standard devices cannot be accessed or it is at least very difficult.

This design debug stage is time consuming and therefore an expensive part in the total cycle of product creation. Much valuable time is lost when looking for errors in a design that cannot be located due to access problems. To monitor signals on these chips other means of access and monitoring must be created.

The Test Access Port (TAP or JTAG port, part of the IEEE1149.1 standard) [1], is an access mechanism which is more and more explored to be used for other means than just structural testing. For programming Flash devices and PLDs it is already common to use the TAP for access [2]. Using the TAP port for chip level debug has already been demonstrated as well as for in-circuit emulation (e.g. the EJTAG mechanism [10]). Using the TAP port for board level debug of the design functionality is a logical extension. The implemented boundary scan architecture, which has a connection to every IO pin of an IC, potentially enhances the access to functional signals. When adapted, it is possible to monitor signals to otherwise inaccessible pins without having to incorporate additional board design.

Testing AC and DC parameters using JTAG demonstrates the use of the boundary scan architecture for other purposes than just interconnection testing [3]. Testing of AC-coupled nets for high speed interconnects is being investigated [4,5], and is standardized in IEEE1149.6. The mixed signal standard [6], is based on testing analogue interconnections. An interesting approach is shown by Nadeau et al. [7,8] who uses a separate control block to speed up the timing of normal boundary scan cells making high speed testing possible. Chakraborty [9], uses JTAG for fault injection in redundant architectures. Interesting is, that the paper describes the use of the additional chains, common in FPGAs, for pin selection and fault injection on the BS cell primary input.

This prior work, however, is focussed to interconnection testing of other nets than digital nets or is restricted to parameter testing. Hence, these methods are not focussed on enhancing system design debug or are specifically designed for FPGAs. This paper will focus on a method, not necessarily needing the timing of the boundary scan cells, to make design debug more flexible. It is focussed on reducing the contact problems in today's board designs. This paper will present a method making use of the IEEE1149.1 architecture.

Requirements and limitations when extending Boundary scan for design debug

Boundary scan is a static test method and as such not suitable for high-speed data transfer. However, its architecture can have additional value when it is going to be used for design debug. To meet the requirements for design debug the basic architecture need to be extended. Requirements are defined as the additional capabilities that the implementation must have for design debug. Also, limitations are defined by the scope of the IEEE1149.1 architecture and cost effectiveness. To facilitate design debug the following must be taken into account:

1) Signal measurements
    - Monitor, transfer and inject signals ranging in frequencies from DC -> HF (current applications use frequencies of tens of MHz).
    - Synchronous operation
    - Asynchronous operation
2) Access to pins (to increase the contact flexibility).
    - Read signals from one or more pins
    - Write signals to one or more pins
    - Transfer signals from one or more pins to other pins

When adapting the boundary scan implementation to meet the above requirements, additional requirements on board and chip level need to be considered.

3) The TAP connector is used to read and write test data to the application. This enables to route functional signals over an already available connector for data monitoring. This enhances the contact to the IC pins.
4) Silicon overhead; the adaptation takes as less as possible silicon area. The implementation must remain economical.
5) Minimal wiring or layout overhead at IC level. Layout and wiring are important issues during IC design.
6) Impact on functional data paths. Especially on high-speed signals, no possibly interfering logic can be present.
7) Total implementation must comply with the IEEE1149.1 standard.
8) Output behavior (CLAMP, Tristate, functional). To enhance debugging and routing possibilities, some pins need predefined values.

Figure 2 illustrates how signals can be routed for design debug when using the boundary scan architecture.

In this figure the JTAG port is used as external access mechanism. Through this port signals can be read or written. The figure shows a limited set of possibilities when an IEEE1149.1 device is present on the board. S1-S5 represents functional signals, which can be monitored through the boundary scan device. For example, high frequency signals S1 and S2 created by DUT2 can be observed through the JTAG port using the boundary scan cells of the IEEE1149.1 device. Another possibility provided by boundary scan is shown by the signals S3-S5. One or more of these signals can be used as stimulus to

DUT1. With, for example, a logic analyser, the output of DUT1 to the stimuli can be investigated.

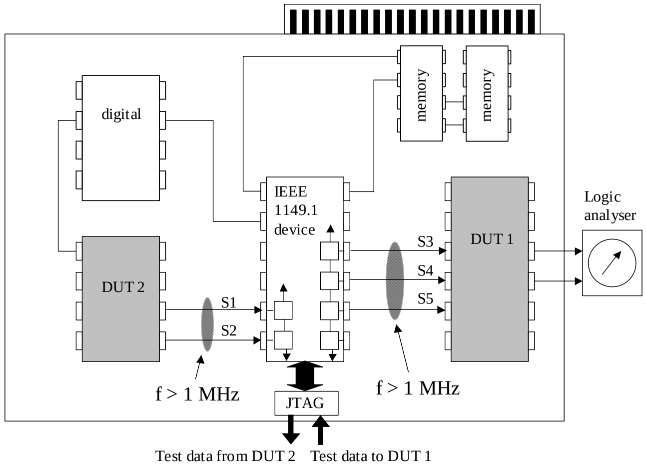

Figure 2: *Use of IEEE1149.1 to enhance design debug on prototypes. The figure shows possible routing paths for functional signals in combination with boundary scan architecture.*

With standard Boundary Scan the application is put in the test domain, not the functional domain. This avoids functional data to be passed through the chain towards the TAP connector or other pins in the application. However, when IC1 is put in EXTEST mode and all others ICs are in functional mode by using the BYPASS instruction, TDO of IC1 can be used as observer for functional data and TDI as driver.

However, the fundamentally static behavior of Boundary Scan restricts the use for functional driven test or debug. The serial nature of the boundary scan chain reduces the effective signal frequency on the digital pins. Although the test frequency (TCK) is generally in the range of 10-50 MHz, most ASIC designs will have chains of 1000 cells, reducing the speed on IC pins (data rate) effectively to 10 kHz. Such frequencies do not suffice for proper and reliable debugging of the functional features.

To overcome these limitations of normal boundary scan, the standard architecture need to be modified. In this paper boundary scan is used to define two debug methods:

1) Synchronous mode (SYNC) when multiple signals on pins are needed,
2) Asynchronous (ASYNC) mode when a single signal on a single pin is sufficient. This mode is independent of the test clock and allows the highest frequencies. These modes will be explained in the following section.

Extending the boundary scan architecture

While developing an extension to the boundary scan architecture, standard boundary scan cells as defined by IEEE1149.1 are used as far possible. The extension required is built around these standard cells. The concept of this extension will holds for synchronous mode and asynchronous mode. Differences occur at the level of implementation in the cells and will be explained in the following sections.

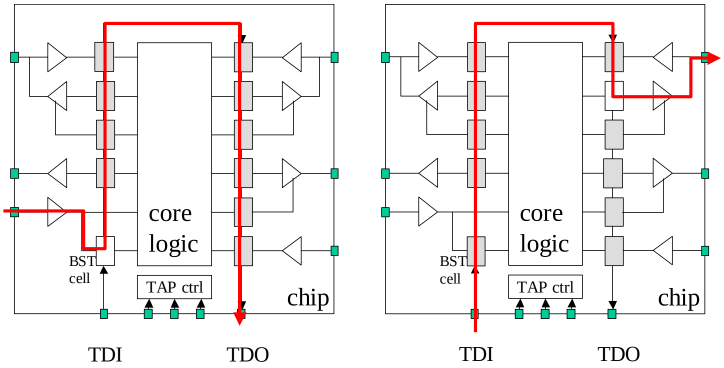

Figure 3: *Conceptual architecture used for design debug when IEEE1149.1 is used. The grey BST cells indicate that test signals bypasses this cell and that this cell does not read/write test signals to /from the pin.*

The extension is based on removing cells/pins from the boundary scan chain, which are not used for board level debug. Figure 3 demonstrates how this is realised. The figure shows a chip with a core and a diversity of input and output pins. Each pin has a boundary scan cell connected (BST cell). The buffers indicate the direction possibilities. The left picture shows a routing example from an input pin towards TDO bypassing the (grey) cells. Signals on the chip pins can be selectively transferred towards TDO. The grey BST cells indicate that the specific cell is not used for reading or writing data. This means that signals read at the selected input pin is directly transferred towards TDO for monitoring. The same concept is applicable for the right picture where a signal is injected at TDI and directly transferred towards the selected output pin. The BST cells are adapted to meet the requirements for design debug. Figure 4 shows figure 3 in more detail. The figure shows the concept when using the TAP port for access to the chip, and the boundary scan cells for access to the specific IC pins. The SYNC and ASYNC modes are both implemented in this concept. The figure shows a normal chip with a core and IEEE1149.1 implemented. The chip exists of a boundary scan TAP controller, boundary scan cells and a mixture of input/output/IO pins. In the figure, the adaptation of the basic architecture is based on the use of extensions to the standard boundary scan cells (BS-block) in combination with a configuration block (CFG block). The signal "stream" is a control signal required to set the device in design debug mode or the normal IEEE1149.1 mode. The configuration blocks determine which pins are selected for board level debug. This is done by logic which creates cell specific control signals (CFG(i) in which "i" denotes the specific cell). Therefore i ranges from 1 to n in which n is the total number of boundary scan cells. The selected pins, determined by CFG(i), are used for routing high speed signals or data streams. Not selected pins will be either connected to the core or set at a user-defined value. The implementation of the configuration block is explained in the following sections. The BS-block determines how signals are routed and implementation depends on the specific mode (SYNC/ASYNC) used. For synchronous mode, signals or data streams will be routed through the normal boundary scan cell. For this synchronous mode two variants have been used. For asynchronous mode the boundary scan cell is completely bypassed but existing paths are still used. Both modes will be elaborated more in the following sections.

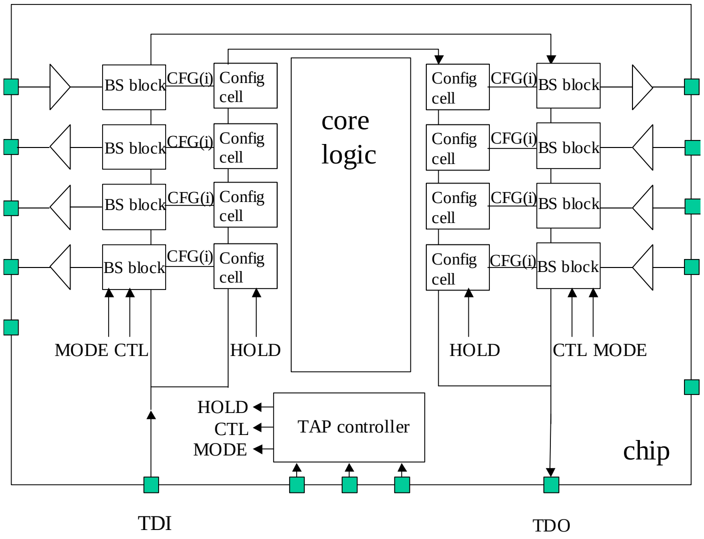

Figure 4: *Architecture used to implement a synchronous mode and asynchronous mode.*

Synchronous mode

When the synchronous mode is selected for debugging, the test clock is used. Two variants are considered here. The first variant is referred to as re configurable boundary scan. This variant makes it possible to transfer multiple data streams or different signals over chip pins at the same time.
The other variant is optimised for a minimal silicon solution. This variant transfers a single signal or single data stream over chip pins at high speed.

Re-configurable boundary scan - multiple data streaming.

The operation of the re-configurable boundary scan variant is similar to standard boundary scan except that the length of the chain can be defined by the user in-circuit. As a consequence, the shifting through the chain is many times shorter compared to a standard boundary scan chain, increasing the effective speed over chip pins.
Figure 5 shows how the implementation is realised for input pins and output pins. For the CFG block, a configuration register with the same length as the boundary scan chain is used. This register creates the setting for the selection of pins. CFG(i)="1" indicates that the specific BS block will read or write functional data from the IC pin (the pin is selected). A "0" indicates that the BS block does not read or write data from the IC pin. The figure shows a standard boundary scan cell where the required extension is in the grey box. C0-C3 are parameters which determine the specific boundary scan behavior as defined by IEEE1149.1. The parameter "stream" is a global parameter created by the TAP controller. CFG(i) is a cell specific parameter.

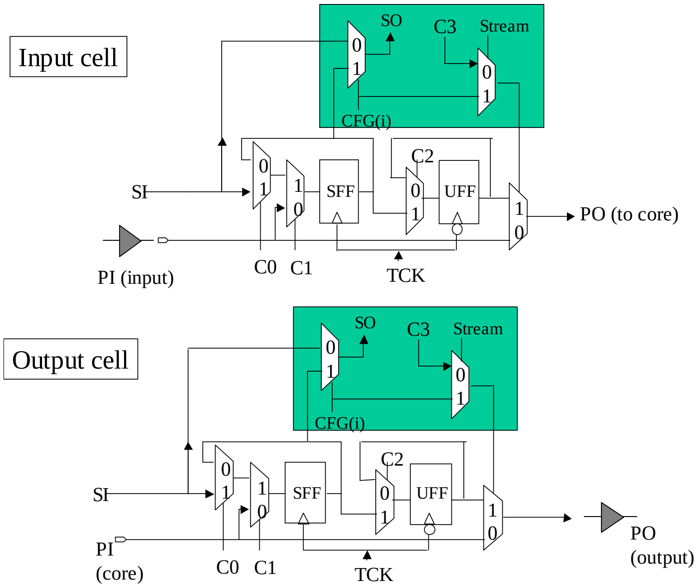

Figure 5: *Design of an input cell and output cell making synchronous mode for multiple data streaming (re-configurable boundary scan) possible. CFG(i) is a cell specific parameter which sets the configuration for this mode.*

Input cells							
Pin	*Cfg(i)*	*stream*	*Path*	*C0*	*C1*	*C2*	*C3*
Select	1	1	1149.1	1149.1			
Not select	0	1	SI-SO, PI-PO	1149.1			
EXTEST	1	0	Determined by 1149.1				
functional	-	0	PI-PO	-	-	-	0

Output cells							
Pin	*cfg(i)*	*stream*	*Path*	*C0*	*C1*	*C2*	*C3*
Select	1	1	1149.1	1149.1			
Not select	0	1	SI-SO, PI-PO	1149.1			
EXTEST	1	0	Determined by 1149.1				
functional	-	0	PI-PO	-	-	-	0

Table 1: *truth table for figure 5*

The operation is for input pins and output pins similar and indicated in table 1. When the pin is selected, CFG(i)=1 and the cell behaves as a normal boundary scan cell. When the pin is not selected (CFG(i)=0), a transparent path is created from SI to SO. Now, signals are not routed into the scan path for debugging. For not selected pins core data is routed to/from the pin (the path PI-PO). Normal boundary scan interconnection test is possible when CFG(i)=1 for all boundary scan cells.

The realisation for the configuration cell is shown in figure 6. Such cell controls each boundary scan cell.

Debugging starts with setting the correct configuration. via SI and SO with HOLD=0. After preloading HOLD=1 to guarantee that the configuration will not change during testing.

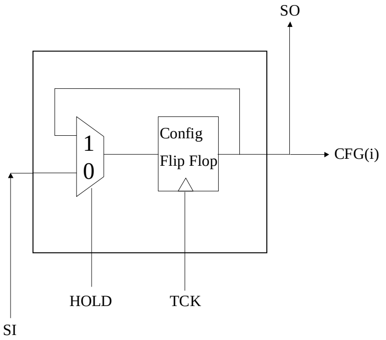

Figure 6: *Design of a configuration register cell. Each boundary scan cell is configured through such cell by means of CFG(i).*

Data rate over the IC pins

The advantage of adapting the chain in-circuit to a few cells is, that the effective data rate or transfer frequency on the IC pin is increased. Consider a chip with 100 boundary scan cells, a TCK frequency of 20 MHz and one single pin required for debug.

For standard EXTEST it takes 104 TCK cycles to put the first data bit on the pin. About 4 TCK cycles are used for the state machine and 100 TCK cycles are needed to shift through the chain. This equals a data transfer rate of 0.2MHz. In formula:

$$f_{effective\_at\_pin} = \frac{(4+\#cells)}{f_{TCK}}$$

With a re-configurable boundary scan chain, the configuration is such that only the specific cell/pin is in the chain. This reduces the chain length from 100 cells to one single cell and the effective frequency on the pin becomes 4MHz in this situation. The increase in the effective data rate is obvious.

It be must noticed that for each cell one additional multiplexer is introduced in the scan path. Although this does not limit the effective speed, it might be limiting when many consecutive cells are not selected. The delay of each multiplexer is added to the total delay between two scan flip-flops. For correct clocking the summed delay of these multiplexers cannot exceed 1/2 TCK.

Data streaming on IC pins

The normal JTAG protocol as defined by IEEE1149.1 is used. Similar to normal structural testing, the synchronous mode can read and write different signals on different pins. The use of a re configurable chain makes it possible that cells/pins that are not selected for testing are transparently routed from SI to SO. While the total length of the chain is reduced, each active pin will behave conform the standard protocol. The selected chain is loaded with data followed by an update stage. As a result, each selected pin can carry different data.

Single data streaming

In many cases design debug will focus on monitoring single signals. In that case only one single cell/pin will be selected in the chain. A re-configurable boundary scan chain shows important aspects that can be used for design debug. The additional configuration register, however, requires some more silicon overhead. Also, the multiplexer delay in the scan path puts a limit to the maximum data frequency. To circumvent these two issues a different approach can be followed.

Assuming that one single data stream often suffices, another variant can be developed. This one also is synchronous but reduces circumvents the effects of summed multiplexer delays. Here, data is transferred by clocking through all scan flip flops of the normal boundary scan cells. Because one chip pin is selected for debug, the update flip-flop is not necessarily needed. The selected cell for debug can simply transfer the data from the scan flip-flop to the chip pin or vice versa. The circuitry required for setting the selection of cell/pin for debug (figure 6) is the same as the circuitry around the update flip flop. Since the update flip flop is not needed for data transfer, it can be used for setting the selection. The implementation for input cells and output cells for this variant is shown in figure 7. Table 2 indicates the different paths created by figure 7. Similar to figure 5, a standard boundary scan cell is shown with C0-C3 determined by IEEE1149.1. The grey blocks indicate the extension required to make this mode possible.

The parameter "stream" is a global parameter generated by the TAP controller. With stream =1, the debug mode is selected. When a path SI-SO is created, this is always through the scan flip flop (SFF). The selection of cells is done using the update flip-flop (UFF). The configuration is preloaded into UFF after which after which a hold is generated by C2=0. When the TAP controller is in shift stage, C2 prevents that the setting is lost. During shift in debug mode the SFF can be used for reading and writing signals on the chip pins. With UFF=0 the pin is selected for debug and data is transferred to/from the chip. For UFF=1 the pin is not selected and core values are transferred to/from the chip pin. From the table it is clear that this implementation is compliant to IEEE1149.1.

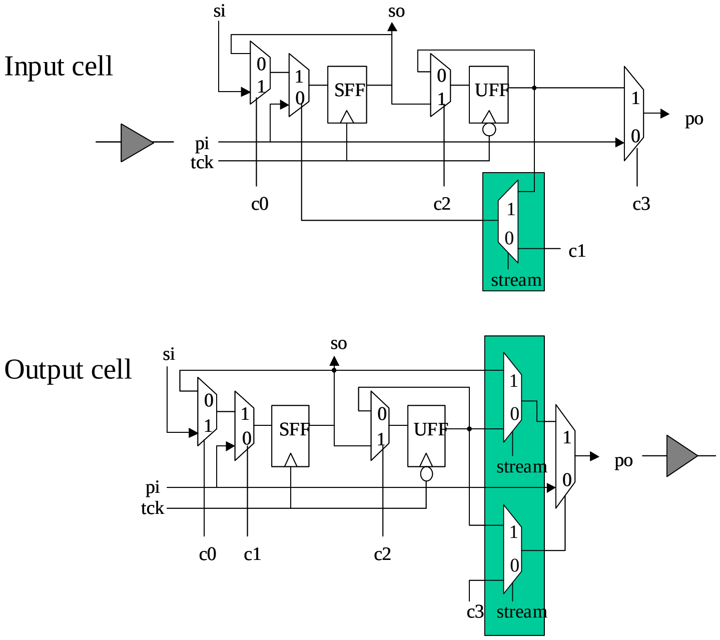

Figure 7: *Using the update flip-flop as alternative for setting the configuration. Data is transferred through SFF and, if selected, directly routed to/from PI/PO.*

Input cells							
Pin	*UFF*	*stream*	*path*	*C0*	*C1*	*C2*	*C3*
Select	0	1	PI-SO, PI-PO	1	-	0	0
Not select	1	1	SI-SO, PI-PO	1	-	0	0
EXTEST	-	0	Determined by 1149.1				
Functional	-	0	PI-PO	-	-	-	0

Output cells							
Pin	*UFF*	*stream*	*path*	*C0*	*C1*	*C2*	*C3*
Select	1	1	SI-PO, SI-SO	1	1	0	-
Not select	0	1	SI-SO, PI-PO	1	1	0	-
EXTEST	-	0	Determined by 1149.1				
Functional	-	0	PI-PO	-	-	-	0

Table2: *truth table for figure 7*

Data rate over the IC pins

Data transfer is done through the scan flip-flops. A summed multiplexer delay as with re-configurable boundary scan will then not occur. However, additional clock cycles are needed before data is available at an output pin or observed from an input pin. Although this will introduce an initial delay between stimulus and monitoring, this process does not hamper data rate. At every clock cycle a data bit becomes available. Signal frequencies higher than is possible with re-configurable boundary scan can be obtained. The highest sample rate is equal to the test clock frequency TCK.

Data streaming on IC pins

Not using the update flip-flop in this variant for data transfer reduces silicon overhead. On the other hand, multiple signals over chip pins are then not possible anymore. Only one single signal or data stream can be read or written to one chip pin. More pins are possible when output cells are selected for signals but those pins will all carry the same data with an additional phase shift determined by the number of in between SFF. When two output pins are selected, in between scan flip-flops will give a delay between these two signals.

Asynchronous mode

When a single signal is of importance for debug, the implementation can be done either synchronous (as described before) or asynchronous. For measuring a single chip pin, synchronisation with the measured data stream is not always needed. In those cases a simple real time measurement is sufficient. For this an asynchronous mode where speed is the key issue is developed. This mode is different from the synchronous mode since it does not use the boundary scan cell clocking mechanism. It does use, however, the already existing routing paths Data is directly read or written from the chip pin and directly transferred through the scan path. Any path created is a transparent path. The implementation is very similar to the single data streaming synchronous mode.

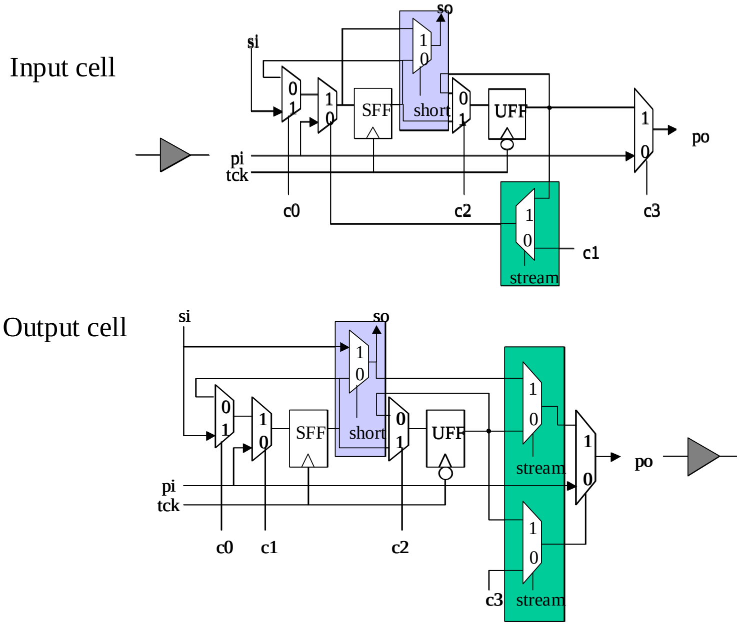

Figure 8: *An asynchronous implementation for transferring a single signal. "Short" controls the asynchronous data routing. With "short"=1, data is transferred directly from SI to SO without any flip-flop. The update flip-flop is used for setting the configuration similar to figure 7. Table2 applies for the other paths.*

Figure 8 shows the implementation. The same construction for setting the configuration is used for the asynchronous mode. The normal boundary scan cell with the required extension for setting the configuration is used again. The reader is referred to the previous section for an explanation of the operation and table 2. Additionally, for asynchronous mode, SFF is bypassed with an additional multiplexer. This multiplexer is used for both input and output cells and makes direct data transfer possible without clocking. When "short" =1, a direct path from SI to SO is created for both configuration settings. Table 2 applies for the other routing paths. With "short" =0 compliancy to IEEE1149.1 is guaranteed. To control this bypass, the global parameter "short" is used which in turn is controlled by the TAP controller.

Data rate over the IC pins

The absence of clocking during debugging provides an opportunity to go beyond the limits of the boundary scan test frequency. In fact, once the configuration is set, the scan path is free for any digital functional data. The data is read or written to the IC pin at real time; it acts as a digital wire. Transfer speed is equal to the test signal. When this signal comes from a JTAG tester it is generally limited to 20-30 MHz. But, when real time data from another chip on the board is tested, the limit is theoretically determined by functional design. Like the re-configurable boundary scan mode, delay introduced by the multiplexers in the scan path is present. Nevertheless, since clocking is not an issue this delay will not be critical for debug.

Data streaming on IC pins

High speed signals can be directly transferred to the chip all carrying the same signal. This is because the implementation consists of parallel connections. In practice, this mode will generally be used for testing data on a single pin for single data streams. After initialisation by the TAP controller, the selected pins are free to transfer data when the TAP controller is in the shift stage.

Experimental results on an FPGA

The two modes as described in the previous two sections have been validated on a prototyping system. This prototyping system is based on an FPGA. Both modes are combined within one boundary scan architecture. As a result it shows clearly that both modes can be combined in a single chip. Complexity of the total design is, however, increased because of the multiple modes. The design is made in VHDL using a standard TAP controller; the boundary scan cells are standard from a library. The required multiplexers are placed around the cells. Interfacing is done using a standard JTAG tester with a normal, user defined JTAG port on the chip. The design is downloaded to the FPGA using MAXplus software. The total setup used is demonstrated in figure 9.

Measurements have been done on a three input and three output device (total of six cells - see also figure 9). In the synchronous, multiple signal mode, it is shown that a random selection of pins can be chosen. Data on not selected input pins are not transferred into the scan path (no read out on TDO). Not selected output pins do not show any data patterns. The output pin is in this case a user defined "1" and can be measured as such. It is shown that with a function generator synchronized with TCK the input pin samples the data and transfers it to only the selected output pins. In asynchronous mode the real time character is demonstrated. Selecting one input pin and one output pin data is transferred (with a hardly measurable delay) towards the selected output pin. Frequency can be changed real time.

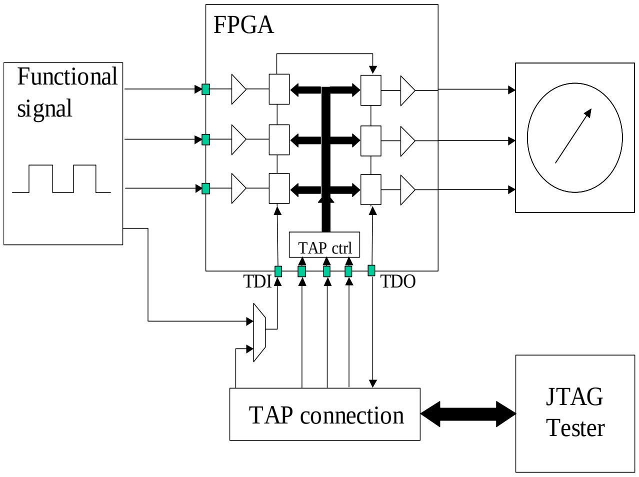

Figure 9: *Set up used for verifying both debug modes. Data streams are read selectively from the input pins and can be monitored selectively on the output pins.*

The impact of multiplexers in the scan path has been investigated for the synchronous mode. In a clock safe design, generally 1/2 TCK cycle is available between two cells. A simple calculation shows that for a data rate at 10 MHz on a chip pin, the total maximum of multiplexers between two flip-flops (2 cells) is <25 (2-3 ns estimated for multiplexer delay (including chip routing)). Above 25, clocking data will not be correct anymore implying that not more than 25 consecutive cells can be bypassed at once. To proof this situation a chip has been programmed in the FPGA with 100 output pins as shown by figure 10. Only the last cell is used for debug output and data is injected on TDI. Now, data from the tester has to pass 99 multiplexers before it will be clocked. Increasing TCK frequency on the JTAG tester indicates an upper limit of 1.7 MHz for correct clocking. This corresponds to a delay of (1/2*5.9E-7)/99=3ns per multiplexer in a non optimised FPGA implementation.

In asynchronous mode these multiplexers are also present in the scan path only resulting in slight delay of data on the output pin. Also now, the delay can be measured with 99 muliplexers in the scan path. A delay time of 360ns is measured which equals a delay per multiplexer of 3.6ns which is in close proximity of the 3ns found in the synchronous mode.

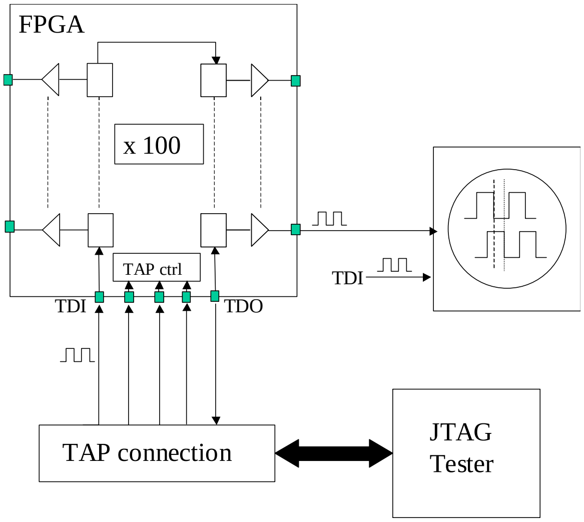

Figure 10: *Set up used to verify the delay introduced by the additional multiplexers.*

Future work

The multiplexers introduced in the scan path have an impact on the maximum speed for the re-configurable boundary scan mode. A substantial amount of multiplexers in the scan path between two flip-flops can decrease speed. Reducing the impact of the multiplexers is one of the things that need to be solved such that high speed is also possible in a multiple signal mode.

Routing inputs to outputs is possible in the current approach, however, only in shift direction. To make any possible routing from input to output possible a circular chain must be created. This means that TDO must be looped back to TDI.

Conclusions

An improvement for debugging prototype designs in the functional domain is described. It is demonstrated that the boundary scan architecture can be used for this debugging. The use of the TAP improves the access to the board for reading and writing functional signals on chip pins that are otherwise not accessible. Three different implementations for board design debug are presented. A synchronous mode is defined with two variants for implementation. One uses a re-configurable boundary scan chain, which can route multiple data streams over chip pins with a data rate of several MHz. The other implementation variant is optimised for a minimal silicon solution. It provides single data streaming at high speed. An asynchronous mode, which does not use the boundary scan clock, allows single data streaming at design speed at real time. For all three variants the boundary scan architecture is used to create a user defined selection of pins needed for debug.

Acknowledgement

This work has been partially funded through the European MEDEA+ (A207) program.

References

1) IEEE Computer Society. "IEEE Standard Test Access Port and Boundary-Scan architecture - IEEE Std. 1149.1-2001", IEEE, New York, 2001.
2) De Jong et al.: Testing and programming Flash memories on assemblies during high volume production. Proc. Of ITC, pp 470-479, 2001.
3) S. Sunter et al. "complete, contact less I/O testing-reaching the boundary in minimizing digital IC testing cost. Proc. Of ITC pp 446-455. 2002
4) S. Chung et al.: AC-JTAG: empowering JTAG beyond testing DC nets. Proc of ITC, pp 31-37, 2001.
5) Y. Kim et al.: frequency detection-based boundary scan testing of AC coupled nets. Proc. Of ITC pp 46-53, 2001
6) IEEE computer society: "IEEE standard for a mixed signal test bus". - IEEE Std.1149.4, 2000.
7) B. Nadeau-Dostie et al.: an embedded technique for at-speed interconnect testing. Proc. Of ITC pp 431-438, 1999.
8) Design for at-speed test, diagnosis and measurements by B. Nadeau-Dostie, Kluwer Academic Pub. USA, 2000, pp197-198.
9) Chakraborty TJ.: A novel fault injection method for system verification based on FPGA boundary scan architecture. Proc. of ITC pp 923-929, 2002.
10) MIPS technologies Inc., MIPS EJTAG Debug Solutions, 980818 Rev. 2.0.0, 1998. http://www.mips.com.