A Novel Structure of Dynamic Configurable Scan Chain Bypassing Unconcerned Segments on the Fly

Shengye Wang[1], Wei Cao[1*], Lingli Wang[1], Na Wang[2], Ping Tao[2]

[1] State Key Lab of ASIC and System, Fudan University, Shanghai 201203, China
[2] East China Institute of Computer Technology, Shanghai 200233, China
* Email: caow@fudan.edu.cn

Abstract

Most of the implementations of boundary scan chains are of fixed length, typically hundreds or thousands. Because the whole chain is scanned every time, many clock cycles are wasted when only a small part of it is concerned. In this paper, a novel structure of configurable boundary scan chain is proposed. Its length and content can be reconfigured without interrupting the chip's functionality. Experimental result shows that the maximum frequency can be as high as 510.4MHz for a full-configurable chain with 512 cells, under 32 nm process, which is 15.7x better than the intuitive method. The proposed structure has been applied to a processor prototype design, and is expected to meet requirements of different applications.

Keywords -- JTAG, Boundary Scan Chain, Reconfigurable, Segment Tree

1. Introduction

Boundary scan, also known as the IEEE 1149.1 [1], or JTAG standard, has burst into the IC industry since decades ago [2]. It appears to be one of the most successful testing standard approved by the IEEE. Initially targeting board-level testing for digital circuits, it is now widely used for various purposes, including on-board testing and diagnostics, programming [3], debugging, etc. As the development of system on chip (SoC), boundary scan is playing a more important role [4].

However, most of the implementations of scan chains are linear and of fixed length. As the performance of chips increases exponentially, the weaknesses of boundary scan began to emerge. Because the whole chain is scanned every time, many clock cycles are wasted when only a small part of it is concerned. There is a need that a chain be configured to appear like either a sub-chain in the middle, or an aggregation of a few sub-chains connected one after another, as shown in figure 1. Therefore, total scan time can be decreased to large extent.

In figure 1, one or more segments in the chain are selected. Test data input port, or TDI, is connected to the first segment of interest, whose end is tied to the head of next segment. The following segments of interest are chained up, connecting one's output to the successor's input. The end of last interested segment is connected to TDO, the test data output port. In general, the chain could

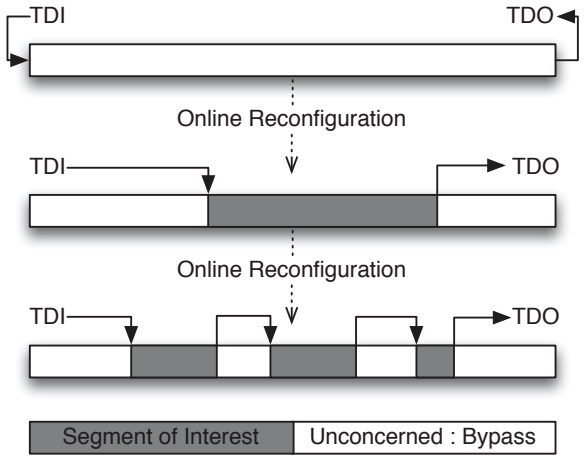

Figure 1. Configurable boundary scan chain.

be shortened on the fly to save scanning cycles while preserving information of interest. And it could also be easily recovered by reconfiguration. Besides the functionality, it is necessary to ensure that the chain could be easily designed, verified and exploited. That is, as well as the performance, the easiness of implementing the chain is also important.

1.1 The Intuitive Approach

An intuitive approach is to add a multiplexer at the output of each scan cell, as proposed in [5]. An illustration of this scheme is shown in figure 2.

The intuitive scheme, simple and effective, satisfied the demand to some extent. However, there is one major weakness, which is the combinational delay could be extremely large if the chain is long. In an extreme case where the whole chain needs to be bypassed, this shortcoming could be fatal. Suppose the length of the chain is n, the maximum delay is the delay of n multiplexers. It is a disaster for performance-efficient applications, such as exhausting testing tasks. The intolerable maximum delay

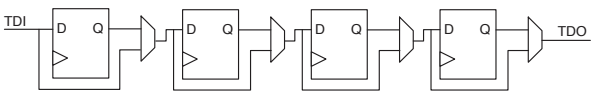

Figure 2. Intuitive approach for configurability in [5].

978-1-4673-6417-1/13/$31.00 ©2013 IEEE

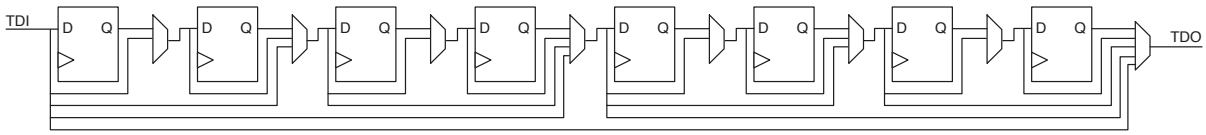

Figure 3. Boundary scan chain with segment-tree-like structure.

of this structure in the worst case is shown in section 3.

In this paper, a novel structure of configurable boundary scan chain is proposed. Its length and content can be reconfigured without interrupting chip's functionality. This design allows faster access through JTAG interface, and could be a support for many applications.

In the following section 2, the proposed architecture of boundary scan chains is presented. It exploits segment-tree-like arrangement to bypass unconcerned segments. Allowing only a few sub-chains in the middle be connected to TDI and TDO, the design bypass rest part of the chain. For a scan chain at the length of n, the maximum delay is guaranteed under $O(\log n)$ levels of logic, compared to $O(n)$ levels of the intuitive method. In section 3, the utilization of resources and maximum frequency of both designs in ASICs under 32nm process are compared. The structure has been applied to a prototype processor design. In section 4, the advantage of the proposed scheme is presented as conclusion.

2. Proposed Architecture

2.1 Segment Tree

The idea of building faster bypass chain relies on bypassing several cells at one time, inspired by a type of data structure in computer science called *segment tree*. Though widely used in computer science and algorithms, it is rare that an arrangement like this is applied to digital circuits.

Segment Tree is a data structure first raised by Bentley in [6]. It is a very effective data structure in computer science when dealing with segment problems. Querying some properties (i.e. sum, maximum or minimum item, etc.) of a segment only takes $O(\log n)$ time. The basic idea is to build nodes to represent a segment whose length is power of two. Any desired segment is then collapsed into an aggregation of these segments, which can be guaranteed to be less than $2\log(n)$ segments.

The idea of segment tree arrangement is brought to the scan chain in this paper. Paths need to be bypassed are regarded as the segments in the data structure. Therefore, the power of the sophisticated data structure is exploited.

2.2 Structure of Chain

Similar to original segment tree arrangement, for a chain of length n, let $k = \log_2 n$, there are $\sum_{i=0}^{k}(n/2^i) = 2n-1$ bypass paths. For each i, the corresponding path bypasses 2^i cells, whose range is $[2^i \cdot t, 2^i \cdot (t+1)-1]$, where t is an integer and $t \in [0,(n/2^i)-1]$. For example, if a scan chain of length 8 is to be built, bypass paths for $[0,0]$, $[1,1]$, $[2,2]$, $[3,3]$, $[4,4]$, $[5,5]$, $[6,6]$, $[7,7]$, $[0,1]$, $[2,3]$, $[4,5]$, $[6,7]$, $[0,3]$, $[4,7]$, $[0,7]$ are created respectively. Combination of these bypass paths is exploited to skip unconcerned segments. It is proved that any path that is a subset of $[0,n-1]$ can be represented by at most $2k$ segments as shown in [6]. Thus, the total combinational delay, determined by the longest bypass path used, is always less than or equal to $2k$ times of propagation delay of a multiplexer, where $k = \log_2 n$. An example of a chain of length 8 is shown in figure 3.

2.3 Configuration

The control signals can be generated by the JTAG host or other logic. To determine whether to bypass a specific path, the following algorithm is exploited.

First all the continuous cells-to-bypass are merged. Therefore, all the cells-to-bypass are represented by several segments like $[a,b]$, denoting cells $a, a+1, \cdots, b$ are to be bypassed. Note coexistence of $[a,b],[b+1,c]$ is not allowed, because they can be merged into $[a,c]$. Performance will be degraded in such case.

For every segment-to-bypass, $[a,b]$, it is divided into several implemented bypass paths. For bypass path $[2^i \cdot t, 2^i \cdot (t+1)-1]$ as stated above, it should be used if and only if $a \le 2^i \cdot t$ and $2^i \cdot (t+1)-1 \le b$. In this way, a subset of implemented bypass paths are selected, which will be configured to enable.

Every bypass path represented by $[x,y]$ is a multiplexer attached to data output of cell y, whose input contains final output from cell x. For each multiplexer connected to output of cell y, when multiple bypass paths are enabled, only the longest one should be used. So there would be some bypass paths enabled but not effective. This does no harm to the logic, but brings some extra information. Configuration bits for bypass path of unit length could be exploited (which bypasses exactly one cell) as the shift enable signal for that cell, preventing undesired modification during shifting.

The control logic of the chain needs to be implemented separately. A practical practice is to use a conventional chain to shift in the selection bit of the multiplexers. In this scheme, $2n-1$ configuration bits are necessary because there are $2n-1$ bypass paths to be controlled.

2.4 Potential Concerns

One thing to notice is that timing of the chain is hard to analyze with static timing analysis. EDA tools will use all the bypass paths of unit length to estimate the critical path in STA. However, with post-routing simulation rather than static analysis, the actual maximum delay can be retrieved. Either frequency of TCK, the JTAG clock should

Table 1. Area and delay comparison for both cases.

Area		
	Intuitive	Proposed
Combinational	9087.935	18612.744
Buf/Inv	5698.670	9162.145
Noncombinational	3383.165	3383.165
Net Interconnect	1322.868	3501.137
Total cell area	12471.100	21995.909
Total Area(μm^2)	13793.969	25497.047
Timing		
	Intuitive	Proposed
Max Delay(ns)	30.705	1.959
Frequency(MHz)	32.567	510.464

be loosely constrained or rule exceptions need to be set up during synthesizing, placing and routing.

An additional chain is necessary for configuration. However, because logic of the chain is merely a very little portion of the whole design, area occupied by additional $2n-1$ registers is not a burden to the design. The complexity of control logic even made the design dark-silicon friendly [7] because these logic is not often switched thus consuming little power. In addition, configuration information can be compressed to save area through encoding selection bits of the multiplexers.

In the worst case, the delay can still be as large as $2\log n$ levels of logic. It has been a leap compared to the intuitive approach, yet sometimes still remains to be an concern because it is a few nanoseconds when the chain is very long. Carefully planning enabled cells could resolve this problem. Enabling a dummy cell in the bypass paths is one of the solutions, since it will break the long bypass path to two. Another approach is to plan the bypass paths with care. In an extreme case, if all the used paths of length 2^x is aligned to boundary of 2^x, the delay can be reduced to one level of logic.

3. Implementation and Evaluation

This design has been implemented with 32nm process [8]. Industry standard EDA tools are exploited to evaluate resources usage and timing. To compare with the intuitive approach proposed in [5] and mentioned in section 1, the area and maximum delay are evaluated in both cases.

Take a chain of 512 scan cells as an example. Usage of area in details and maximum frequency is shown in table 1. All the data are obtained under the same condition (i.e. 32 nm process [8], very loose timing constraints and minimum area policy). Note the static timing analysis (STA) is not applicable in these cases because tools does not recognise optimal bypass path in static timing analysis. Gate-level simulations are exploited to retrieve actual delay timing information.

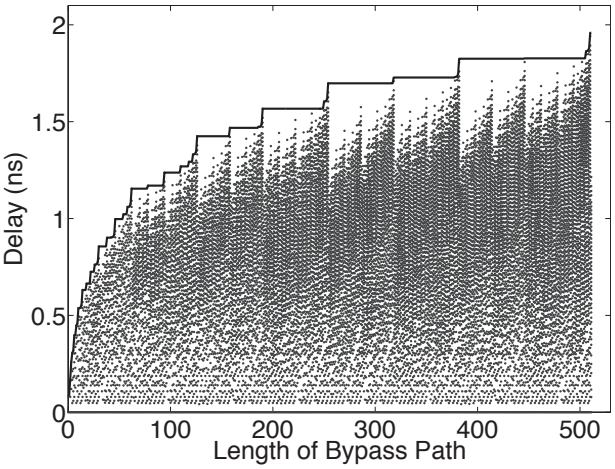

Figure 4. Delay v.s. length.

From table 1, it can be seen that at cost of more area, the proposed design decreased the unacceptable delay to a very low one. In other words, at the cost of approximately 2x area, the delay decreased by over 15x. For most of the designs, the achievable delay does meet the demand.

The result indicates with the proposed scheme, accessing the chain can be much faster. It does benefit applications like debugging and testing.

3.1 Delay and Bypass Length

Figure 4 is obtained from gate-level simulation. The x-axis is the length of the chain while y-axis is the delay distribution. The solid line indicates maximum delay increase as the chain lengthens.

From the figure it can be seen that as the length increases, the delay increase at approximately the order of $O(\log n)$. Usually bypass path starting with 1 and ending with 2^x-2 is likely to be time-consuming, because $2(\log n-1)$ multiplexers are chained up to create this bypass path. However it is not guaranteed to be the path with the greatest delay. For the graph shown above, a chain of length 512, the maximum delay appears when bypassing $[129,510]$, which is 1.959ns. The second long path, is path $[1,510]$, whose delay is 1.940ns. It is acceptable to use delay of $[1,2^x-2]$ as an estimation of maximum delay if a full simulation cannot be applied during the first stage of designing.

From this figure we can see the delay increase at the order of $O(\log n)$, which matches the theory of segment tree. This is important because we can ensure the performance is not decreasing as fast as $O(n)$, as the intuitive approach does. Chain length over 1,000 or 10,000 can benefit more from this structure.

3.2 Chain Inside a Sample Processor Design

The proposed scan chain has been applied into an infant multicore processor prototype design. Internal critical signals and some important registers can be accessed through JTAG interface. The scan chain for internal signals contains 1088 scan cells. Since only some signals are inter-

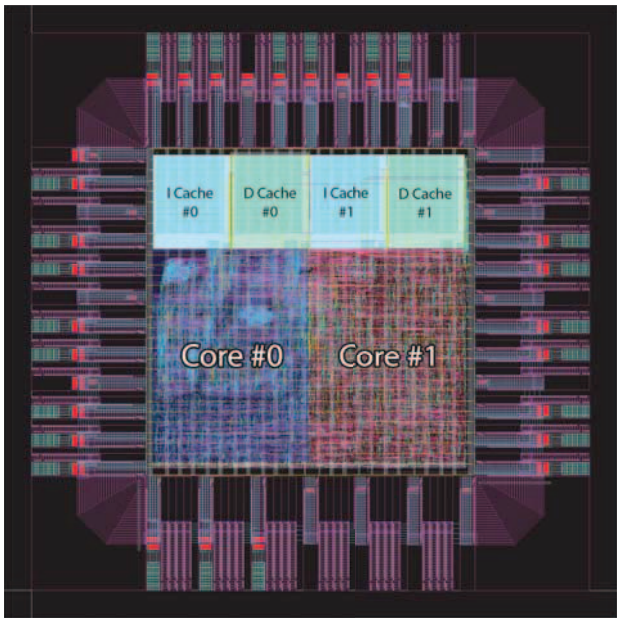

Figure 5. Layout of a multicore processor design with proposed scan chain.

ested while debugging, it is not necessary to do a full scan every time. The new structure decreases the amount of data transferred through JTAG interface, and increases the efficiency of debugging.

Also since the design is pin count limited, JTAG interface also serves as IO interface. The proposed structure helped improve throughput of data exchange and IO operations by skipping unconcerned signals on the fly.

The microprocessor prototype is designed with 32nm process [8]. Its layout is shown above in figure 5. A thin bar to the bottom of the cores shows the test access port controller together with the proposed scan chain. They occupy very little area.

4. Conclusion

The circuit structure is simple and scalable. It is important that only n multiplexers (which are equivalent to $2n-1$ two-to-one multiplexers) are introduced and evenly distributed. The maximum delay can be ensured to be less than $\log n$ levels of logic, and could also be determined after configuration. Thus it has minor impact on the performance. Because the bypass paths are evenly distributed, the difficulty of routing is not a concern in the design. The total length of wires is of order $O(n \log n)$, which is acceptable since boundary scan cells are usually placed at boundary of a chip or a region.

The proposed chain structure, with an addition TAP controller, is fully compatible with IEEE JTAG standard [1]. It can be widely used in various applications. Injecting similar test pattern is a use of this structure. Suppose there are thousands of test patterns that share a lot in common, the efficiency of test can be improved with the proposed scan chain. Since cells out of the segment of interest will not change during shifting, those data is shifted in once and preserved by connecting capture input to data output. It saves time and cost in testing. This design also benefits circuit designing. Designers will not need to plan chains carefully to avoid time wasting in shifting unconcerned data. For an SoC integrator, he or she can connect all the scan chains provided by IP providers, without worrying about the debugging efficiency.

5. Acknowledgments

The authors would like to thank Xiaoxuan She, Xitian Fan, Chen Liang and Chenlu Wu from State Key Laboratory of ASIC and System Fudan University, Qian Yu, Huotian Zhang, Yijing Sun and Zimu Li from Department of Microelectronics Fudan University, Prof. Michael Taylor and Amer Sinha from Computer Science and Engineering, University of California San Diego for productive (and challenging) discussions. This paper is supported by National Natural Science Foundation of China (61131001, 61171011) and Fudan's Undergraduate Research Opportunities Program.

References

[1] IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-2013 (Revision of IEEE Std 1149.1-2001) (2013).

[2] R. Dellecker, Boundary-scan bursts into the modern production facility, Aerospace and Electronic Systems Magazine, IEEE, 16(6), p.21 (2001).

[3] T.E. Marchetti and T. Borroz, Programming Flash Memory with boundary scan using general purpose digital instrumentation, AUTOTESTCON, IEEE, p. 1 (2010).

[4] Insoo Kim and Hyoung Bok Min, Operation about multiple scan chains based on system-on-chip, International SoC Design Conference, 3, p. II-191 (2008).

[5] Chenguang Guo, Yanlong Zhang, Zhiping Wen, Lei Chen, Xuewu Li, Zengrong Liu and Min Wang, A novel configurable boundary-scan circuit design of SRAM-based FPGA, International Conference on Computer Science and Automation Engineering (CSAE) , 2, p. 335 (2011).

[6] Jon Louis Bentley, Solutions to Klee's rectangle problems, Technical Report, Carnegie-Mellon University, Pittsburgh (1977).

[7] Michael B Taylor, Is dark silicon useful? Harnessing the four horsemen of the coming dark silicon apocalypse, Design Automation Conference (DAC), p. 1131 (2012).

[8] Synopsys Armenia Educational Department, SAED 32/28nm Process (2012).