Retiming Scan Circuit to Eliminate Timing Penalty

Ozgur Sinanoglu
New York University Abu Dhabi
Department of Computer Engineering
PO Box 129188, Abu Dhabi, UAE
ozgursin@nyu.edu

Vishwani D. Agrawal
Auburn University
Department of Electrical and Computer Engineering
Auburn, AL 36849, USA
vagrawal@eng.auburn.edu

Abstract–Scan design has a performance penalty that affects the critical path delay by an added fanout at the origin and a multiplexer at the destination. This problem is outlined in a recent paper [10], *which also proposes a solution. The purpose of the present work is to provide a retiming solution. Retiming of a synchronous sequential circuit is a transformation that moves flip-flops through combinational logic without altering the function. We move the destination flip-flop of a critical path backward through its scan multiplexer. This splits the flip-flop into three, one on each input of the multiplexer. First of these is the "original flip-flop" in the normal data path. The second, called "shadow flip-flop", appears only in the scan path. The third flip-flops from all critical paths are replaced by a single flip-flop that generates a delayed scan enable signal for controlling all retimed multiplexers. We further show how the fanout delay at the origin of a critical path can be eliminated by additional retiming. The use of the formally proven retiming transformations preserve both the function of the circuit and its scan operation without any change. The retimed scan, therefore, can test DC as well as delay faults. Benchmark results show further timing improvement and reduced hardware overhead compared to previously reported results* [10].

1 Introduction

A recent paper [10] describes a problem that occurs in scan testing. In scan design, every flip-flop is preceded by a multiplexer that adds approximately two gate delays to the combinational path. Consider Figure 1 (a) that pictures the critical path of a sequential circuit. The dotted line arrow represents the longest combinational path of the circuit whose delay determines the clock period of the circuit and, hence, the performance. Figure 1 (b) shows the same path after scan implementation. The multiplexers, controlled by a common scan enable (EN) signal, select either the normal mode data signal or the scan-in (S_IN) signal into flip-flops [2]. Scan also adds an additional fan-out, shown as scan-out (S_OUT), at the output of each flip-flop. Thus, the critical path slows down by the delays caused by one multiplexer and the fan-out.

One remedy often suggested to reduce the performance penalty is partial scan where the destination flip-flop of critical paths are excluded from scan. This, however, potentially reduces the fault coverage. An ingenious solution without having to resort to partial scan has been proposed in a recent paper [10]. That solution is outlined in Section 2. The present contribution provides a retiming solution, which has certain advantages over the previous method. Retiming is a graph theoretic technique [7, 8] with applications to digital design optimization. It is outlined in Section 3. Section 4 describes the new retiming solution and Section 5 gives some results for comparison.

2 Previous Work

A recently proposed method [10] to reduce the performance penalty of scan modifies the critical

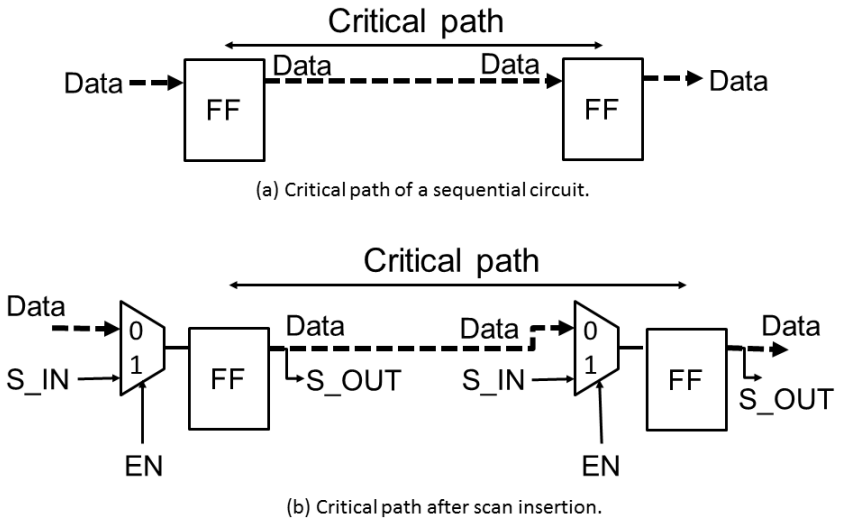

Figure 1. Slowing of critical path in conventional scan. S_IN: scan-in from previous flip-flop, S_OUT: scan-out to next flip-flop, EN: scan enable = 0 for normal mode and = 1 for scan mode. All flip-flops have a common clock.

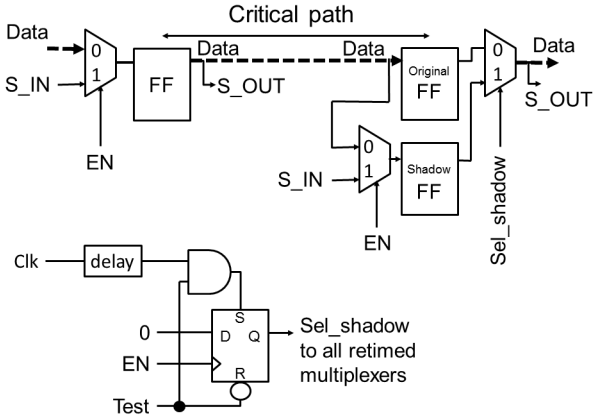

Figure 2. A previously proposed design for eliminating the performance penalty of scan by removing the scan multiplexer from the critical path [10]. All flip-flops have a common clock (Clk).

path of Figure 1 (b) as shown in Figure 2. The multiplexer is moved forward to the output of the destination flip-flop (original FF) and is replaced by a fanout that feeds into an added flip-flop (shadow FF) through an additional multiplexer. Thus, in the normal mode (EN = 0) both original FF and shadow FF contain same data. In the scan mode (EN = 1) the scan data is transferred through the shadow FF using the two multiplexers, which completely isolate the original FF.

In general, there can be several critical paths in a circuit. Each critical path is modified with a shadow FF and an additional multiplexer at its input. These multiplexers are controlled by the original scan enable signal (EN). However, all original multiplexers after being moved to the outputs of the original FFs are controlled by a common Sel_shadow signal generated by a single SR latch and AND gate arrangement shown at the bottom of Figure 2. Besides the scan enable signal, EN, a new "Test" signal is required. The clock (Clk) signal is the same for all flip-flops, though an adjustable delay is needed to balance any clock skew at the SR latch. "Test" ensures that Sel_shadow = 0 during the normal mode. The purpose of this circuit is to synchronize the Sel_shadow signal with the clock.

The design of Figure 2 is shown to perform both normal and scan modes correctly [10]. The cost in hardware overhead is one shadow FF and one multiplexer per critical path, and a single Sel_shadow generation circuit. In addition a new "Test" signal is required. The timing penalty of scan multi-

Figure 3. Retiming moves flip-flops across combinational logic.

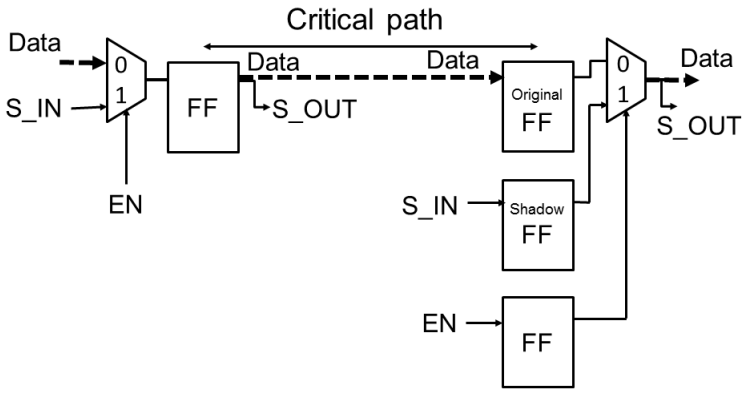

Figure 4. Removing multiplexer penalty through retiming of critical path in Figure 1 (b) by moving the flip-flop at critical path destination backward across the multiplexer. All flip-flops have a common clock.

plexer is reduced to that of a single fan-out. In the present work, we will give a simpler design using less hardware and eliminate the fan-out and the "Test" signal. The main idea used is the retiming transformation described next.

3 The Retiming Transformation

Retiming transformation of a circuit moves all of the memory elements at the input of a combinational block to all of its outputs, or vice-versa. Provably, this procedure leaves the function of a synchronous circuit unchanged [7, 8]. Since its first publication in 1983, numerous applications in digital design automation have been found. They include minimization of state variables, reducing logic, reducing power consumption, improving testability and timing optimization [4, 9]. Figure 3 provides a simple illustration of the retiming transformation.

4 Retimed Scan Architecture

The two time penalties on the critical path, namely, multiplexer delay and fan-out delay, can be independently removed.

4.1 Eliminating Multiplexer Penalty

Figure 4 shows a retiming transformation of the circuit of Figure 1 (b) in which the flip-flop at the critical path destination has been moved across the multiplexer. Because multiplexer has three inputs, the flip-flop is triplicated. The first, shown as "original FF", directly receives the critical path data. The second, shadow FF,

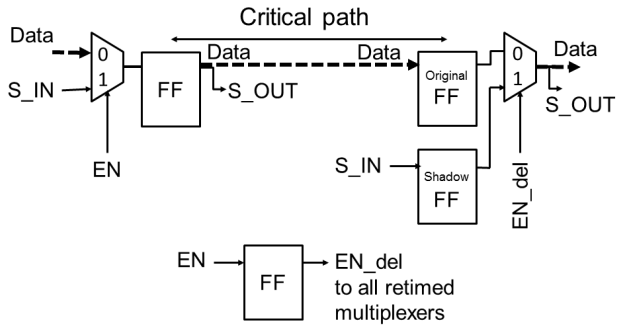

Figure 5. Generation of a common EN_del signal for controlling the multiplexer at the outputs of original and shadow FFs in Figure 4 for retimed multiple critical paths. All flip-flops have a common clock.

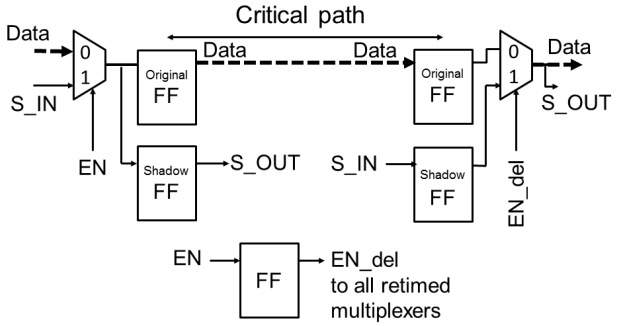

Figure 6. Removing fan-out penalty through further retiming of critical path in Figure 1 (b) by moving the flip-flop at the origin of the critical path forward across fan-out. All flip-flops have a common clock.

receives and forwards S_IN. The third flip-flop simply delays the EN signal by one clock cycle. According to the retiming rules [7, 8], all three flip-flops have the same clock as before. In general, a circuit may have several critical paths ending on separate flip-flops, all of which will be transformed as shown in Figure 4. However, the third flip-flops of all critical path destinations can be combined into just one flip-flop to generate an EN_del signal to control all "pushed out" multiplexers. This is shown in Figure 5.

4.2 Eliminating Fanout Penalty

The fan-out added to the source flip-flop of a critical path inserts some extra delay that may often be acceptable. If that is not the case then the source flip-flop can be moved forward across the fanout as shown in Figure 6. These flip-flops, named "original FF" and "shadow FF" receive same data and clock. The original FF feeds data directly to the critical path and shadow FF serves the scan path. Notice that the critical path in Figure 6 has exactly the same combinational delay as the original non-scan circuit of Figure 1 (a).

4.3 A Limitation of the Technique

Because our technique basically transfers the scan-induced delays in critical paths to their adjoining paths, it will not work in two specific cases:

1. Two adjoining critical paths, i.e., one critical path feeds into another critical path. Here the destination flip-flop of one path will be same as the origin flip-flop of the other path.

2. Feedback critical path, i.e., a critical path that originates and ends at the same flip-flop.

We should remark, however, that in practical cases two adjoining paths, both near critical, may not have exact same delays. In those cases retiming can potentially improve the overall timing of the circuit by transferring excess delay from one path to the other.

5 Results

Results for retiming of several circuits are shown in Table 1. The performance penalty was eliminated for all circuits except s35932, which belongs to the categories listed in subsection 4.3. Last three circuits, Mickey-128, Trivim and Grain, are high speed circuits that implement *ecrypt stream cipher algorithms* [1]. In these results, both multiplexer and fanout penalties were eliminated. The previous method [10], used for comparison here, only eliminates the multiplexer penalty.

The number of transformations listed in the second column of Table 1 is related to the number of

Table 1. Retiming transformation to eliminate multiplexer delay penalty.

Circuit name	Retiming [this paper]						Previous work [10]	
	No. of transformations	CPU (s)	Area overhead (%)		Critical path delay reduction (%)		Area overhead (%)	Critical path delay reduction (%)
			Multiplexer only	Multiplexer and fanout	Multiplexer only	Multiplexer and fanout		
s713	7	< 1	7.8	10.4	2.5	3.1	11.0	1.9
s953	6	< 1	3.7	6.5	6.3	7.8	5.3	4.7
s1423	3	< 1	1.0	2.0	1.7	2.1	1.4	1.3
s5378	3	< 1	0.5	0.7	5.3	6.7	0.7	4.0
s9234	4	< 1	0.3	0.5	2.9	3.6	0.4	2.2
s13207	3	2	0.2	0.2	2.1	2.6	0.2	1.6
s15850	3	34	0.2	0.2	1.6	2.0	0.2	1.2
s35932	0	7	0.0	0.0	0.0	0.0	0.0	0.0
s38417	2	12	< 0.1	0.1	2.6	3.2	0.1	1.9
s38584	2	11	< 0.1	0.1	1.7	2.1	0.1	1.3
b17	16	44	0.1	0.2	0.7	0.9	0.2	0.5
b20	10	26	0.3	0.3	0.9	1.1	0.4	0.7
b21	11	25	0.3	0.4	0.9	1.1	0.4	0.7
b22	11	41	0.2	0.3	0.7	0.9	0.3	0.6
Mickey-128	2	5	0.5	0.6	5.5	7.0	0.7	4.5
Trivim	2	4	0.5	0.6	4.5	5.7	0.7	3.7
Grain	2	3	0.7	1.0	4.5	5.7	1.1	3.7
Range	0-16	0-44	0.0-7.8	0.0-10.4	0.0-6.3	0.0-7.8	0.0-11.0	0.0-4.7

critical paths in the circuit all of which were fixed. An iterative procedure is used [10]. Static timing analysis (STA) first identifies a critical path in the scan circuit. That path is retimed and STA is again applied to ascertain that the critical path delay of the circuit is indeed reduced and to find another critical path, which is now retimed. This process stops when the critical path delay of the circuit cannot be reduced. If, however, the critical path delay is found to increase after retiming then the process stops accepting the minimum delay solution from the previous iteration. The procedure was first carried out for removing the multiplexer penalty alone and then repeated for the fanout penalty. The number of iterations in the second column of the table shows the total number of iterations for removing both penalties.

The third column shows CPU times for transforming circuits on a Redhat Enterprise Linux 5 system running on Intel Xenon CPU E5520, 2.27GHz, 8 cores, 15.54GB real, 17.62GB virtual and 587.21GB local disk. The CPU time depends on the circuit size and the number of critical paths but does not seem to increase excessively.

The fourth column shows percent area overhead for the removal of multiplexer penalty alone, which includes one shadow flip-flop per critical path and one extra flip-flop per circuit for generating the delayed EN signal as shown in Figure 5. This overhead is proportional to the number of critical paths but reduces as the circuit becomes larger. The overheads for the previously reported technique [10], which also removed the multiplexer penalty only, are shown in column 8 and are higher because an additional multiplexer per critical path was required.

Next, examine the multiplexer delay saving in columns 6 (retiming method) and 9 (previous method [10]). Retiming allows a complete elimination of the multiplexer delay while the previous method removes the multiplexer but adds a fanout in its place contributing to some delay. As a

result the critical path delay reduction for retiming is better than the previous technique.

For the calculation of path delays, a single-input gate or an inverter is assumed to have one unit of delay. The delay of a gate with fanin n_{in} and fanout n_{out} is computed as,

$$\text{Gate delay} = 2 \times \lceil \log_2 n_{in} \rceil + n_{out} - 1 \ \text{ units}$$

where $n_{in} \geq 2$. Thus, a two-input gate with a single fanout will have a delay of 2 units. The above formula estimates the delay by assuming that a gate with larger number of inputs is split into a balanced tree of two-input gates. Each fanout beyond one contributes one extra delay unit.

Columns 5 and 7 give the result after removing both multiplexer and fanout penalties. The overhead shown in Column 5 includes the overhead of Column 4 and an additional shadow flip-flop inserted in parallel with the flip-flop at the origin of each critical path as shown in Figure 6.

With the exception of s35932, the delay penalty of scan was completely eliminated for all circuits and Column 7 of Table 1 merely shows the total penalty of the conventional scan that is generally quoted as 5 to 10% [2].

6 Conclusion

Using the concept of retiming, we have shown how the performance penalty of scan can be completely eliminated in many circuits. There is a hardware overhead penalty that may increase with the number of critical paths. However, for several example circuits considered in this paper, the overhead remains small. In general, one has to choose between the hardware and delay penalties based on the application of the circuit. Because the new design is obtained by retiming transformations alone, the circuit function in both normal and test modes is guaranteed to remain unchanged. That means the new design will perform all types of scan tests, DC as well as delay (including launch off shift and launch off capture), without any change.

An often stated motivations for *partial scan*, in which only a subset of flip-flops is scanned, is to avoid the delay penalty in timing critical circuits [2]. Partial scan may reduce area and delay overheads but it results in higher test generation complexity and reduced fault coverage. The technique of this paper eliminates the need for such a trade off. Retiming can reduce the number of flip-flops in a circuit thereby reducing the hardware overhead and test time of full [5] or partial [3, 6] scan test. Those techniques globally retime the entire circuit. The retiming application of this paper is local and can be incorporated after any other optimizations have been done.

References

[1] "Hardware Implementations of Ecrypt Stream Ciphers." VHDL code available from *http://eeweb.poly.edu/faculty/karri/stream_ciphers/index.html*, accessed on Feb. 15, 2012.

[2] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*. Boston: Springer, 2000.

[3] S. T. Chakradhar and S. Dey, "Resynthesis and Retiming for Optimum Partal Scan," in *Proc. 31st Design Automation Conf.*, 1994, pp. 87–93.

[4] G. De Micheli, *Synthesis and Optimization of Digital Circuits*. New York: McGraw-Hill, 1994.

[5] Y. Higami, S. Kajihara, and K. Kinoshita, "Test Sequence Compaction by Reduced Scan Shift and Retiming," in *Proc. Fourth Asian Test Symp.*, 1995, pp. 169–175.

[6] D. Kagaris and S. Tragoudas, "Retiming-Based Partal Scan," *IEEE Trans. Computers*, vol. 45, no. 1, pp. 74–87, Jan. 1996.

[7] C. E. Leiserson, F. Rose, and J. B. Saxe, "Optimizing Synchronous Circuits by Retiming," in *Proc. 3rd Caltech Conf. on VLSI*, 1983, pp. 87–116.

[8] C. E. Leiserson and J. B. Saxe, "Retiming Synchronous Circuitry," *Algorithmica*, vol. 6, pp. 5–35, 1991.

[9] N. Maheshwari and S. S. Sapatnekar, *Timing Analysis and Optimization of Sequential Circuits*. Boston: Springer, 1999.

[10] O. Sinanoglu, "Eliminating Performance Penalty of Scan," in *Proc. 25th International Conf. VLSI Design*, Jan. 2012, pp. 346–351.