Layout Methodology for SDI Model Asynchronous Circuits

3E−3

Metehan ÖZCAN Masashi IMAI Hiroshi NAKAMURA Takashi NANYA

Research Center for Advanced Science and Technology, The University of Tokyo

1 Introduction

With advances in device technology interconnections are becoming responsible for most of the signal delay and distribution of clock signal becomes more difficult to attain. Due to its average-case behavior, timing reliability and nonexistence of clock skew, asynchronous systems are gaining interest for future high-speed systems. Semiconductor Industry Association suggests that in 2003 at 0.13μm technology asynchronous architecture will be prominent [1].

In asynchronous systems, delay model has a great impact on the design and analysis of the IC. Delay-Insensitive (DI) model assumes gate delays and wire delays are unbounded. In Quasi-Delay-Insensitive (QDI) model, "isochronic forks" are added to DI model. But these delay models may sometimes impose over-pessimistic delay assumptions on the circuit, resulting in slow circuits. Scalable Delay Insensitive (SDI) model is based on the idea that once a system is laid out and fabricated, elements in the design will be affected almost similarly by the changes in the operating environment [2]. This delay model guarantees that the circuit will function correctly if the circuit elements designated as SDI has delays bounded by a variance factor of K.

In this note, a layout methodology is proposed for SDI model circuits and initial experimental results are given.

2 SDI layout methodology

Completion generation is very crucial in asynchronous systems. With SDI model assumption, generation of completion signal may be enhanced by getting rid of unnecessary and over-pessimistic timing restrictions. Therefore, SDI model circuits are usually faster and more area efficient than QDI model circuits, but they introduce extra conditions to be met in every step of design methodology.

In the earlier phases of design, variance factor K is to be taken large enough to comfort harsh variations in the later phases of the design. But after the layout phase, it is expected that variations in the delays of elements can be covered by a K close to 1.

Figure 1 shows the layout methodology for SDI model circuits. Note that the overall circuit does not have to be based on the SDI model. For example, if we consider completion generation, we have to worry about critical paths from data part and control part for completion generation. For correct circuit operation, completion signal should be issued after the data part completes operation. For this purpose, start and end points for data and control circuit are specified in synthesis phase, and using this information racing paths are extracted and are input to layout engine with the desired variance factor K for each block. Later, this racing path information is used by layout engine to check whether the layout results comply to corresponding K's. If the resulting layout is not complying the SDI restrictions or needs performance enhancement, layout is re-performed by imposing restrictions on the critical paths.

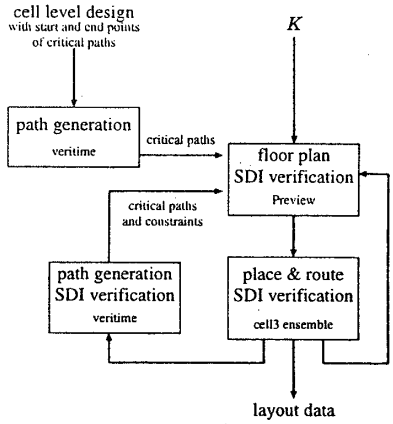

Figure 1: Layout methodology for SDI circuits.

3 Sample layouts

Register files based on QDI and SDI model are designed, laid out and simulated. All designs consist of 16 32-bit registers and contain one read and one write port for simplicity. Objective is to show the performance and area overhead difference of QDI and SDI model circuits, and the trade-off between variance factor K and circuit speed in SDI model circuits with same and different logic specifications.

Figure 2 shows the register files. Data and address are encoded as two-rail codes. In QDI design acknowledgment is generated by collecting acknowledgment signal from 16 registers, each of which are again generated from the acknowl-

		QDI	SDI	SDI-2	SDI3	SDI3'
W	data min (ns)	1.701	1.651	1.581	1.599	1.626
	data max (ns)	1.881	1.721	1.655	1.681	1.719
	ack min (ns)	3.738	2.289	2.262	1.998	1.933
	ack max (ns)	4.075	2.289	2.262	1.998	1.933
	K	-	1.33	1.37	1.18	1.12
	normalized speed	100	61	61	53	52
I	data min (ns)	1.505	1.389	1.400	1.373	1.397
	data max (ns)	1.742	1.551	1.456	1.515	1.534
	ack min (ns)	3.699	2.754	2.848	2.367	2.283
	ack max (ns)	3.912	2.754	2.848	2.367	2.283
	K	-	1.78	1.96	1.56	1.49
	normalized speed	100	75	77	64	62
# of cells		3403	2127	2129	2091	2091
area (mm^2)		4.73	3.31	3.31	3.30	3.30

W: working phase, I: idle phase, $K = \text{ack}_{min}/\text{data}_{max}$

Table 1: Simulation results for register files

edgment signals of corresponding 32 1-bit registers. First SDI design generates acknowledgment signal by checking data input signals and register enable signals from register decoder. On the other hand, the second SDI (SDI-2) design generates acknowledgment signal by checking data (32) and address (4) input signals. In SDI-3 and SDI-3' only 8 of the 32 data input signals and 4 address input signals are checked, but SDI-3' is laid out after grouping control part.

For the generation of critical paths, veritime© engine of Cadence is used. First, start and end points of data and control part are specified in veritime format. Then critical paths are set as slowest data path and fastest control path. For the elimination of false paths, the functionality of C elements are used. Since a C element does not change outputs until all inputs make the same change, the slowest path to its inputs define the path delay through the C element. This feature is used for the elimination of false paths for the control part. Although static path analysis is used, generated results comply with the results in Table 1, which are generated by verilog simulation.

By looking the data in Table 1, it can be observed that SDI model circuits result in 40 % to 50 % performance improvement along with 30 % save in design area. SDI and SDI-2 have almost same values. But when we compare SDI-2 with SDI-3 and SDI-3', performance improvement is obtained at the expense of K. Having same logic definition, layout of SDI-3 and SDI-3' show that further performance

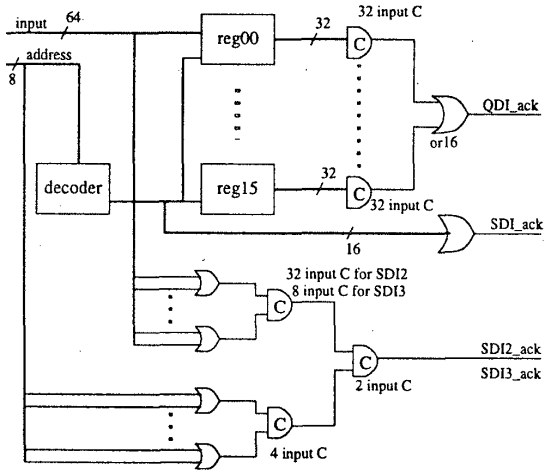

Figure 2: Register files of QDI and SDI models.

improvement can be obtained by updating layout considering critical paths.

4 Conclusion and further work

SDI model asynchronous circuits are both performance and area efficient when compared to corresponding QDI model asynchronous circuits. Furthermore, SDI model circuits can be further enhanced by decreasing desired variance factor K.

For speeding up layout of SDI model circuits, verification of critical paths against K should be done to some extent inside the layout engine, especially if layout of data part does not change drastically. In this way, the re-layout of the circuit can be done without an time-expensive back-annotating step to verification tools. In this paper, a methodology based on this idea is given and the generation of the extra information about critical paths by verification tools, inputing those paths into layout engine, and verification inside the layout is performed on register file circuits.

Further work includes how to optimize the re-layout phase and handling of multiple racing paths. Ultimate goal is to create a tool environment for the layout of SDI model asynchronous circuits, which includes placement, routing, and especially floor planning.

This work was supported in part by STARC commissioned research and by the Ministry of ESSC under Grant-in-aid for Scientific Research No.(B)09480049.

Bibliography

[1] "The national technology roadmap for semiconductors," **http://notes.sematech.org/roadmap5.pdf**, *Semiconductor Industry Association,* 1997.

[2] T. Nanya et al., "TITAC-2: A 32-bit asynchronous microprocessor based on scalable-delay-insensitive model," *Proc. ICCD'97,* , pp.288-294, Oct. 1997.