An Area-Efficient Multi-Level Single-Track Pipeline Template

Pankaj Golani and Peter A. Beerel
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
{pgolani, pabeerel} @ usc.edu

Abstract[1] — **This paper presents a new asynchronous design template using single-track handshaking that targets medium-to-high performance applications. Unlike other single-track templates, the proposed template supports multiple levels of logic per pipeline stage, improving area efficiency by sharing the control logic among more logic while at the same time providing higher robustness to timing variability. The template also yields higher throughput than most four-phase templates and lower latency than bundled-data templates. The template has been incorporated into the asynchronous ASIC flow *Proteus* and experiments on ISCAS benchmarks show significant improvement in achievable throughput per area.**

I. Introduction

Synchronous designs have dominated the market of VLSI and ASIC design for many years. However, as the VLSI industry approaches the "end of Moore's law", the challenges of designing circuits with traditional global clocking strategies have been steadily increasing. Variation is already forcing designers to put high margins in the designs that may wipe out the performance gains of a full technology generation [9]. At the same time, the practicality of asynchronous alternatives has substantially increased. In particular, template-based asynchronous design styles have demonstrated very high performance while at the same time being amenable to standard-cell-based ASIC flows and thereby fast design times (e.g., [6][14]). Different template-based designs, including STFB [2], MOUSETRAP [7], GasP [8], and PCHB [4], trade off robustness to delay variations with performance.

Previous single-track pipeline templates STFB [2], SSTFB [2][3] and STAPL [12] has been shown to be more timing robust than 6-4 GasP and have higher peak throughput and lower power compared to the well-known four-phase QDI PCHB template [4]. However, these templates are homogeneous in nature with only single-level of logic per pipeline stage. This leads to area inefficient designs in applications where the performance is limited by the *algorithmic cycle time* [13]. For example, many applications have implicit read-modify-write loops involving large memories or register files whose read latencies are significant. In these designs, the fine-grain pipelined nature of PCHB and GasP is over-kill because the latencies along the loop combined with the data-dependency in the design forces the pipelines to be often starved for tokens. In such applications, this over pipelining represents wasted area, latency, and power.

This paper presents a new pipeline template that follows the two-phase static single-track handshake protocol [1] and can have multiple levels of logic per pipeline stage. It uses 1-of-N data encoding and domino logic[2], which provides low latency making them an excellent choice for latency-critical, iterative applications (e.g., [14]). The flexibility of having multi-levels of logic per pipeline stages offers a new trade-off between performance and area by reducing the effective control area overhead without sacrificing latency. The remainder of this paper is organized as follows. In Section II we provide a short background on static single-track template. In Section III we present our proposed template and in Section IV we evaluate its throughput / area characteristics. Finally, we offer some conclusions in Section V.

II. Background

This section describes the two-phase static single-track full-buffer template (SSTFB) on which our research is primarily based. The SSTFB template uses two phase static single-track handshaking illustrated in Figure 1(a) [1]. The sender initiates the handshake by driving the handshake wires (channel) high thus sending a request and then releases the channel. The receiver is then responsible for holding the wire high until it acknowledges the input by driving the channel low and then releasing the channel. The sender is then responsible for holding the channel low until it drives it high again thus

[1] Peter A. Beerel is also Chief Scientist at Fulcrum Microsystems.

978-3-9810801-7-9/DATE11/©2011 EDAA

[2] It may be important to note that several start-up companies have commercialized domino-logic-based asynchronous designs in sub-65nm designs for a wide variety of applications [10][11] and have demonstrated that with careful design domino logic in deep sub micron designs is an attractive choice.

sending a new request. The transistor-level channel drivers are shown in Figure 1(a). For the sake of simplicity we also abstract the static single-track channel with its sender and receivers using special gate-level symbols SUP and SDOWN, as illustrated in Figure 1(b). The cycle time of the SSTFB template can be as low as 6 transitions (~1.2GHz in 180nm technology) [14] with a forward latency of 2 transitions per pipeline stage. The use of one-level of semi-weak conditioned domino logic per pipeline stage [13], however, makes them inefficient for gates with complex functionalities. As the number of inputs of the gate increases, the size of the NMOS stack increases which causes slower domino logic, increase in area as well as charge sharing problems. Because the pipeline template only supports a single-level of domino logic, more complicated functions must be decomposed into multiple pipeline stages.

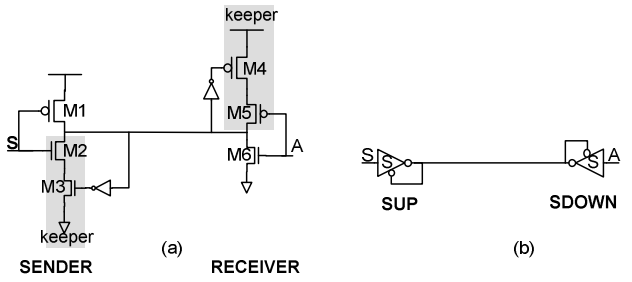

Figure 1 Static single-track handshake protocol

.The SSTFB template also has a rigid pulse-width constraint of 3 transitions on signals S0, S1 and A. During this pulse width the SUP and SDOWN gates must be strong enough to fully charge / discharge the input and output wires. Additionally for non-linear pipelines where a pipeline stage has more than one output, the SSTFB template has an implicit timing race between outputs. In particular, other outputs stop evaluating as soon as one of the dual-rail outputs becomes valid even if the others have not yet finished. This can cause a runt pulse at the output of the other output that can cause the circuit to malfunction. This race is addressed in [2] by constraining the maximum output channel length to be short enough to guarantee all outputs complete evaluation. However, in deep submicron ASIC design, the presence of process variations and crosstalk noise makes it increasingly difficult and inefficient to satisfy this constraint.

III. Multi-Level Single-Track (MLST)

This section proposes a new asynchronous template called multi-level single-track that uses two-phase static single-track protocol and targets medium-to-high performance applications. This section is organized as follows. Section III.III.A presents the block level diagram of the template; Section III.III.B presents the timing assumptions that need to be satisfied for this template.

A. Block level diagram

Figure 2 shows the block level diagram of the proposed MLST template. V_L is 1-of-1 single-track valid wire associated with the dual-rail inputs and V_R is 1-of-1 single-track valid wire associated with dual rail outputs. $Valid_k$ is the output valid of the k^{th} single-track output channel. *OCv* is the combined output channel valid signal generated by the precharged completion detector (PCCD). In case of non-linear pipelines (not shown here) C-element trees are used to combine 1-of-1 valid signals at the input and *OCv* signals at the output (not shown).

The main components of this template are explained below.

Multi-level single-track data path

The data path uses domino logic which yields lower forward latency than static logic. The last level of logic is controlled by the "go" signal generated by the pre-charged completion detector (PCCD) while the intermediate levels of logic are controlled by the "en" signal. The use of separate signals to control the evaluation of the last level and the remaining level of logic allows eager evaluation where the intermediate levels of logic do not have to wait for the handshake at the output to finish before being ready to re-evaluate. This strategy helps to improve the latency and throughput of the system. The domino logic is non-weak-conditioned [13] and thus does not have to wait for all inputs to arrive before it generates its output. This yields a smaller pull-down stack and higher performance.

Note that while intermediate levels of logic use conventional domino logic with inverters driving its output rails, the last level of logic has SUP gates driving its output rails because it is responsible for generating the dual-rail single-track outputs. The last level of logic also generates valid signals ($valid_i$) to indicate the validity of each pair of dual-rail single-track wires.

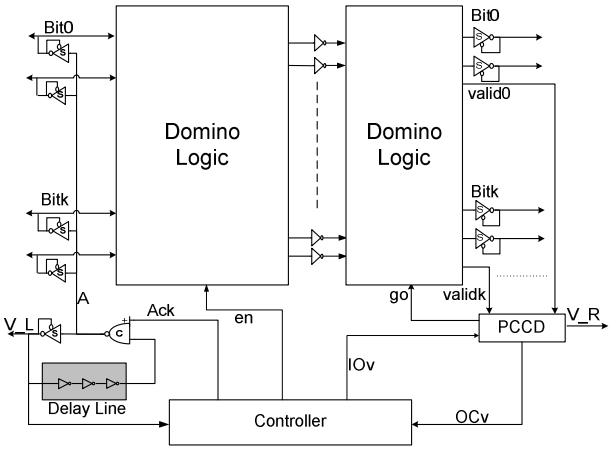

Figure 2 Block level diagram of the MLST

Pre-charged completion detector (PCCD)

The PCCD block is responsible for generating one 1of1 valid wire per output channel. This wire acts as a request signal to the associated receiver. The receiver lowers the wire to send the acknowledgement. Use of this valid wire eliminates the need of a completion detector at the input of the receiver.

Note that the request signal does not trigger evaluation of the data, evaluation is triggered a pulse-width after the acknowledgement signal. This is safe because the acknowledgement signal is triggered simultaneously with the input data being reset preventing re-evaluation with stale data. Additionally there is no bundled timing data constraint [13] associated with this wire as inputs are acknowledged only when the valid wire arrive and all outputs of the receiver are valid.

The PCCD is also responsible for generating the "go" signal which controls the evaluation / pre-charge of the last level of logic as well as evaluation / pre-charge of the PCCD. The pulse width on the go signal controls the evaluation and pre-charge period of the output signals and can be adjusted to account for their load by changing the number and strength of the inverters in the delay line.

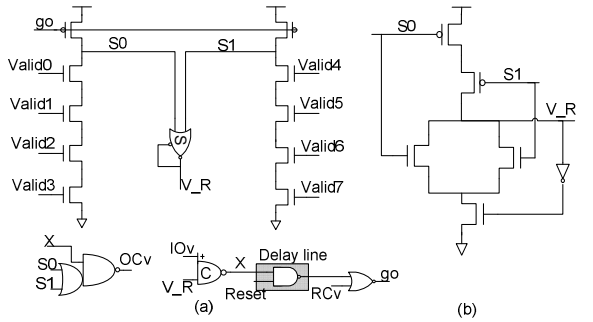

Figure 3 (a) Gate level diagram of the PCCD block (b) Transistor level diagram of the SNOR gate.

Controller

The controller is responsible for generating the acknowledgement signal Ack which drives the input channel low and en signal which controls the evaluation and pre-charge of intermediate logic. The pulse-width on signal A shown in Figure 2 controls the discharge period of the inputs wires and can be configured for a given wire load by changing the number and strength of the inverters in the delay line.

The controller is also responsible for generating the "input output valid" (*IOv*) signal which is driven low when all input and output channels are valid and is driven high when all output channel have been driven low by their receivers. More detailed explanation of the controller can be found in [14].

B. Timing constraints

The correct operation of the proposed template depends on satisfying the single-track handshake constraint [14] which ensures that the handshake wires (channels) are being driven by sender and receiver pipeline stage in a mutually exclusive fashion. This constraint can guaranteed by appropriate sizing of the delay lines in the PCCD block of the sender and top-level acknowledgement logic [14] of the receiver and can be verified using commercial timing analysis tools [6]. Additionally, there are two other relative timing assumptions that must be satisfied.

- Pre-charge of intermediate logic: Recall that the pre-charge of the intermediate levels of logic is caused by the "en" signal being driven low by the controller. The "en" signal is driven back high when the controller detects that left valid signal is driven low. The timing assumption is that this pulse on the en signal is long enough for the intermediate logic levels to pre-charge. In the case of a linear pipeline, this pulse is 5 gate delays and in case of non-linear pipelines this pulse will be longer.
- Reset of dual-rail inputs: Recall that we assumed that if the 1-of-1 valid wire associated with a channel is low, then all the dual rail inputs associated with that channel are also low. The timing assumption is that the dual rail inputs should be driven low by signal A through SDOWN gate before the receiver detects the neutrality of 1-of-1 valid wire and reasserts the en signal and the sender reasserts the go signal for evaluation. In case of linear pipeline this timing race essentially assumes that 1 gate delay is smaller than 5 gate delays and will only be more relaxed in case of non-linear pipelines.

Because these timing assumptions are quite relaxed (no tighter than 1 gate delay versus 5 gate delays) they can easily be satisfied in a standard cell flow with reasonable gate sizing.

IV. COMPARISONS

In this section we present comparison results of our multi-level single-track templates against four phase templates such as PCHB [4] and MLD [5]. We have compared all three templates on a variety of examples from the ISCAS benchmarks, that includes examples that are purely combinational and sequential circuits (include state elements and cycles). Since these different templates are designed to be more efficient at different target performance we select average throughput / area as an appropriate metric for comparisons.

For comparison we extended an EDA tool Proteus [5] to support two phase single-track handshake protocols to generate single-track asynchronous circuit from synchronous RTL netlist. The resulting circuit throughput is estimated through Verilog simulations of random input vectors using a unit-delay model in which each CMOS transition has the same unit delay. The area of the design is estimated by calculating the number of transistors in the design. For simplicity we ignore interconnect area.

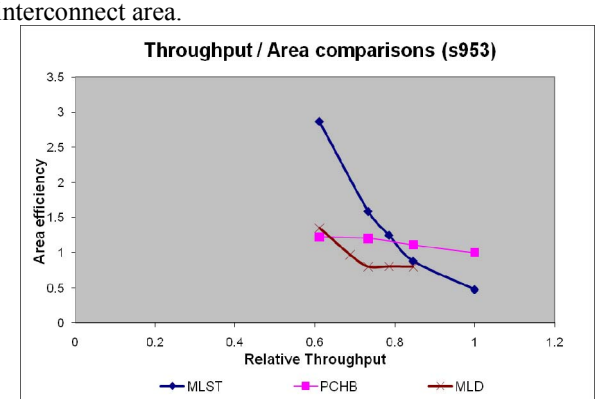

Figure 4 Throughput area tradeoff curves for the s953 example

Figure 4 illustrates the throughput area trade off curves for MLST versus the PCHB and MLD templates for the s953 example. The target throughput on the X axis is normalized by the algorithmic cycle time of the design and the area efficiency of MLST and MLD templates are calculated by comparing the number of transistors in the design to one implemented using PCHB template. We varied the target throughput and compared the resulting circuit area. For medium performance targets, designs implemented using PCHB can reduce area only by reducing the number of slack matching buffers and hence they suffer from large control area overhead. MLST templates, on the other hand, improve area efficiency by also taking advantage of clustering and the sharing of control logic across multiple levels of logic, thus effectively reducing the control area overhead. Compared to MLD, the inherently faster two-phase MLST template can support more levels of logic per pipeline stage for the same target throughput, making it comparably more area efficient.

Design	Avg throughput (T)	Area (A)	T / A against PCHB	T/A against MLD
s298	23	2491	1.72	3.37
Counter	23	3072	1.199	2.24
s27	18	886	1.052	0.88
s344	25	4080	1.154	1.88
s349	25	4080	1.132	1.85
s386	26	1713	1.985	2.74
s400	24	3744	1.4	1.81
s420	25	3974	1.385	1.90
s444	23	4704	1.192	1.46
s510	25	2173	3.63	4.44
s526	21	4085	1.785	2.19
s641	23	9512	0.784	0.86
s713	25	9293	0.748	0.89
s820	27	9451	0.938	1.15
s832	24	7480	1.339	1.47
s838	27	8831	1.235	1.64
s953	26	11940	1.057	1.46
s1196	31	16472	1.235	1.55
s1238	33	11296	1.636	1.64
s1423	33	13819	1.331	1.38
c3540	28	36488	0.762	0.91
Hit detect	27	22746	2.524	3.59
MAC32	32	33727	1.133	1.13

Table 1 Throughput / Area comparisons for MLST vs PCHB and MLD

In Table 1, we compare the throughput per area of MLST template against PCHB template and MLD template over a larger set of benchmarks. We set the target cycle time constraint for PCHB template to 18 transitions as these templates are designed to be area efficient for 18 transitions. For MLST and MLD templates we have selected a representative target performance constraint to be a somewhat slower 26 transitions (~800 MHz in 65nm) It can be seen that at 26 transitions MLST provides 34% better throughput per area over PCHB templates and 75% better throughput per area over MLD templates. Note the throughput / area metric is a function of the target performance constraint we set. As we decrease the target performance requirement, the MLST template improves area efficiency by enabling larger pipeline stages and lower control overhead.

V. Conclusions

In this paper we propose novel multi-level single-track design template for medium-to-high performance applications. The design template has the flexibility of having different pulse widths on the input and an output driver which eases timing constraints compared to SSTFB and provides a trade-off between area and performance. It also has very good latency characteristics, using domino logic with no explicit latches or bundling constraints. Comparisons on several ISCAS benchmarks show that at lower frequencies MLST template yields higher throughput/area than known alternatives. At the same time, its two-phase nature yields far less switching and thus lower power than four-phase alternatives.

VI. References

[1] K. van Berkel and A. Bink. Single-track handshake signaling with application to micropipelines and handshake circuits. Proceedings of ASYNC'96

[2] M. Ferretti. Single-track Asynchronous Pipeline Template. *Ph.D. Thesis, University of Southern California, 2004.*

[3] P. Golani and P. A. Beerel. High performance noise robust asynchronous circuits. ISVLSI 2006

[4] A. M. Lines. Pipelined asynchronouscircuits. Master's thesis, California Institute of Technology, 1995.

[5] G. Dimou. Clustering and fanout optimizations of asynchronous circuits. *Ph.D. Thesis, University of Southern California, 2004.*

[6] P. Joshi. Static timing analysis of GasP M.S.Thesis, University of Southern California, 2008.

[7] M. Singh and S.M. Nowick. MOUSETRAP: Ultra-High speed Transition Signaling Asynchronous Pipelines. ACM/IEEE International workshop on timing issues in the specification and synthesis of digital systems. Dec 2000.

[8] I. E. Sutherland, and S. Fairbanks. GasP: a Minimal FIFO Control. *Seventh* International symposium on asynchronous circuits and systems, 46-52, 2001.

[9] K. Bowman., S. Duvall and J. Meindl. Impact of die-to-die and within die parameters fluctuations on the maximum clock frequency distribution for gigascale integration. IEEE Journal of Solid-State circuits, 37(2):183-190, 2002.

[10] Achronix Semiconductor Corporation. http://www.achronix.com.

[11] Fulcrum Microsystems Inc., http://www.fulcrummicro.com.

[12] M. Nystrom and A. J. Martin *Asynchronous Pulse Logic*, Kluwer Academic Publishers, Inc 2002

[13] P. A. Beerel, R. O. Ozdag and M. Ferretti, *A Designer's Guide to Asynchronous VLSI*, Cambridge University Press, 2010.

[14] P.Golani Theory, implementation and application of single-track designs. Ph.D Thesis, University of Southern California, May 2010