Using Dynamic Domino Circuits in Self-Timed Systems

Jung-Lin Yang
Dept. of Electrical and Computer Engineering
University of Utah
Salt Lake City, UT 84112
jyang@cs.utah.edu

Erik Brunvand
School of Computing
University of Utah
Salt Lake City, UT 84112
elb@cs.utah.edu

ABSTRACT
We introduce a simple hierarchical design technique for using dynamic domino circuits to build high-performance self-timed data path circuits. We wrap the dynamic domino circuit in a wrapper that communicates using a request/acknowledge protocol and mediates the pre-charge/evaluate cycle of the dynamic logic. We apply standard bundled delay matching for completion detection but add an early completion feature that can signal completion if function validity can be determined from the output value. We call the resulting wrapper semi-bundled because of this early acknowledge. The circuit overhead required for this semi-bundled feature is relatively small, but can provide measurable speedup in some situations. The technique is suitable for any dynamic logic family that has a pre-charge/evaluate cycle, and that produces monotonic output transitions.

Categories and Subject Descriptors
B.6.1 [**Logic Design**]: Design Styles – *Combinational Logic, Sequential Logic.*

General Terms
Performance, Design, Experimentation,

Keywords
Domino logic, self-timed circuits, asynchronous circuits

1. INTRODUCTION
If functional data path blocks are to be used in an asynchronous or self-timed system they should also use handshaking techniques to communicate with the other circuit blocks in the system. There are a variety of control protocols used in asynchronous or self-timed systems, but the most fundamental requirements for a self-timed data path element are that it know when to start executing its function, and that it report the completion of the function and the validity of the output values. It is this notification that the data path outputs are valid that makes them self-timed. [1,2].

GLSVLSI'03, April 28-29, 2003, Washington, DC, USA.

In general, self-timed data paths generate a completion signal in one of two ways: using a matched bundling delay that mimics the expected delay in the circuit [3], or using a multiple-wires-per-bit scheme that allows the data itself to encode completion [4,5]. The communication protocol used to signal initiation and completion of the function is typically some variation on standard four-phase or two-phase request/acknowledge schemes. [6]

Dynamic circuits such as domino, NP domino, DCVSL, etc. circuit styles, are widely used in high-performance systems [7,8]. However, the performance increase comes at a cost in design complexity and in sensitivity to electrical effects at the transistor level. Dynamic circuits generally use some sort of pre-charge/evaluate protocol that is controlled by the system clock. In addition to increasing the design complexity to deal with the separate phases of execution, this can also add extra loading to an already problematic global clock signal. Clock gating to reduce power is also made more complex when using dynamic logic. From the point of view of using dynamic circuits in a self-timed system we need to control the pre-charge/evaluate cycle without a clock, generate completion signals, and provide whatever latching is required by the dynamic logic structure.

2. SELF-TIMED WRAPPER
We would like to leverage the advantages of dynamic data path circuits for use in self-timed systems by developing support circuits that provide the necessary completion detection and handshaking functionality. Our wrapper circuit provides three basic functions:

1. Communication with the environment through the chosen handshake protocol
2. Control of the pre-charge/evaluate cycle of the dynamic function block, and completion detection on the function.
3. Latching the output data as appropriate

Because our completion detection circuit falls somewhere between bundling and actual completion detection we call our wrapper "semi-bundled." A diagram is shown in Figure 1.

2.1 Early Completion
A critical feature of the wrapper is to provide completion detection so that the circuit can produce the required acknowledge signal. A simple matched bundling delay can provide an acknowledge signal that corresponds to the worst-case timing of the circuit. However, because of the output behavior of many types of dynamic logic we can do better than that in some cases.

Our wrapper assumes two properties of the dynamic circuits: that they have a reset phase where all outputs are reset to a known value (typically a pre-charge phase), and that the output transitions from the reset state are monotonic. Many dynamic logic families, domino and DCVSL to name only two, exhibit these properties. Assuming these properties of the logic, our wrapper can monitor the output signals and determine that a function is complete if it makes a transition. If no transition is seen at the output by the time the bundling delay expires, then we can assume that no transition will occur and that the logic is at its final value.

Our wrapper includes both a measured early-completion circuit that watches the outputs of the evaluation network, and a matched worst-case delay that "times out" if none of the inputs will change. This is similar in spirit to other early, or speculative completion schemes that have been proposed in other types of self-timed systems [9,10].

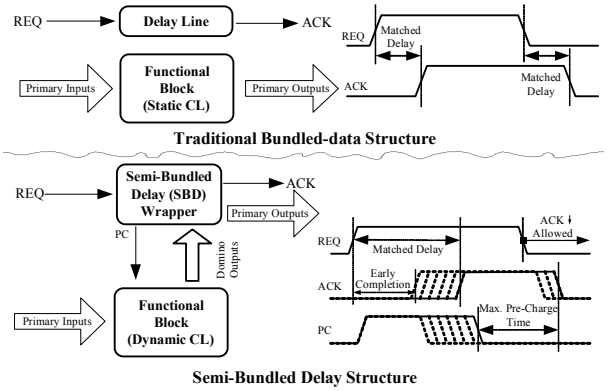

Figure 1 Semi-Bundled Delay Structure

3. WRAPPER IMPLEMENTATION

The wrapper circuit consists of 5 major sub-blocks: pre-charge/eval signal generator (SBD_PC), worst-case matched delay (MDelay), pre-charge matched delay (MD_PC), asynchronous latch (SBD_latch), and completion signal generator (SBD_ACK). Except the SBD_latch, every block in the wrapper is built with either a generalized C-element (gC)[11] or domino logic.

3.1 Pre-charge/Evaluation Signal (SBD_PC)

SBD_PC generates the control signal to the Domino evaluation network to tell when to evaluate and when to pre-charge (Figure 2). Our wrapper uses the incoming request signal to start the evaluation cycle, but then initiates pre-charge as soon as possible by looking at function completion and latch-completion rather than waiting for the falling handshake transition. The output of this block (PC) will be set to high as long as the request signal from the environment has been pulled up to '1'. This puts the domino function block into evaluation mode so that it can start to compute outputs through its evaluation network. Once the evaluation network's result is latched, PC will be pulled down to '0' putting the domino function block into pre-charge mode.

3.2 Delay Circuits (MDelay, MD_PC)

There are two places in the wrapper that require delays: the matched delay for the worst-case evaluation time for the domino function block (Mdelay), and the matched delay to model the time it takes to pre-charge the domino function block (MD_PC). Both of these delays are asymmetric: they want to delay one edge (we'll call this the triggering edge) but not the other (the reset edge). It is also desirable to use a circuit that provides a controllable and substantial delay in a small circuit area.

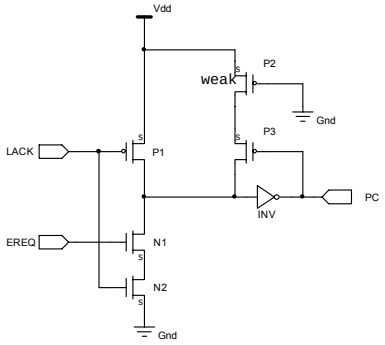

Figure 2: SBD_PC transistor-level implementation

One design for such a delay uses a tunable buffer circuit. This circuit, shown in Figure 3 uses transistors N2 and P2 as variable resistors. They must always be on to maintain correct operation of the controllable inverter, but by using a reference voltage they can provide a variable amount of current-limiting resistance. Using PSPICE with models for the TSMC 0.25um/2.5V CMOS technology we can delay the rising/falling edge from between 1 to 3 FO4 delays using a single stage tunable buffer. However, deep-sub-micron processes with sub-1V power supplies will make designing and predicting delays much more difficult. Future SBD circuits will have to address this delay issue.

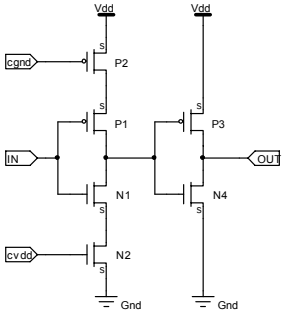

Figure 3: Tunable buffer

Another issue for the delay circuits in our wrapper is that they be asymmetric. In order to provide a delay for the triggering edge, but not for the reset edge, a fast-reset buffer can be used. Figure 4 shows one circuit for a fast-reset buffer. This circuit will reset the buffer chain in parallel during the time when the REQ signal is low, thus providing for a much shorter delay on the reset edge through the buffer chain than for the triggering edge. For small delays this may not be an issue, but for longer delays a fast-reset buffer can make a significant difference for the non-delayed edge.

The MDelay block (Figure 5) is used to postpone the rising edge of the request signal for the matched (worst-case) delay of the domino function block. However, the falling edge should propagate to its output (MD_ACK) without any additional delay. The "Worst-case Matched Delay" is implemented as a delay chain

of the type described above. The MD_PC pre-charge delay is designed in a similar way

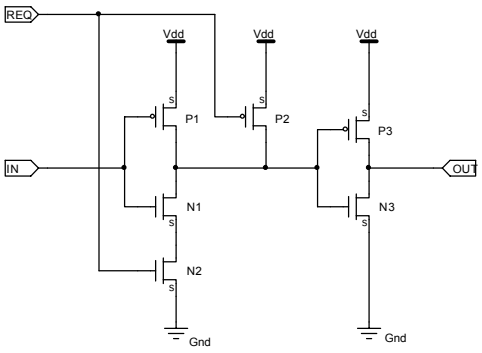

Figure 4: Fast-reset buffer

3.3 Asynchronous Latch (SBD_latch)

The SBD_latch block contains 2 sub-blocks: a positive edge triggered D flip-flop (PETDFF) and latch completion detection circuitry. This asynchronous latch is used only to generate a pipeline stage or at an SBD component boundary. It is not required for every SBD block. The PETDFF is implemented as a standard TSPC latch.

The completion detection technique for this flip flop is simply to monitor the values of input and output of the device. If the output (Dout) has the same value as the input (Din), we assume that the result is latched safely and the completion detection circuitry will be pulled high with quickly. Under such conditions there is no need to postpone the acknowledgement to account for the TSPC clock-to-latch propagation delay.

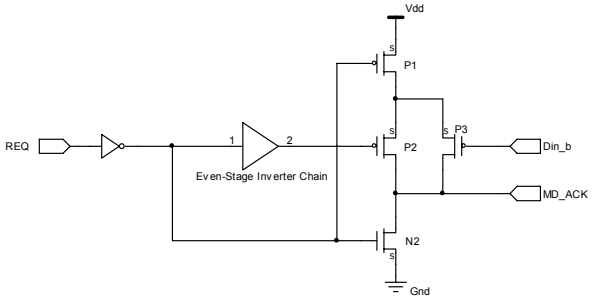

Figure 5: MDelay transistor-level implementation

3.4 Completion Signal (SBD_ACK)

The SBD_ACK is the final stage of our SBD handshaking wrapper. Its output ACK provides two important functions in the four-phase handshake with the rest of the self-timed system: the rising edge of ACK tells when the evaluated result of the domino functional block is valid, and the falling edge indicates that a new request can be issued. The transistor-level implementation is shown in the Figure 6.

4. DESIGN EXAMPLE

We use an extensible self-timed adder design as an example of using the SBD wrapper circuit with domino function blocks. This adder is meant only to demonstrate the use of the wrapper on a measurable circuit, not to be compared with state of the art addition circuits.

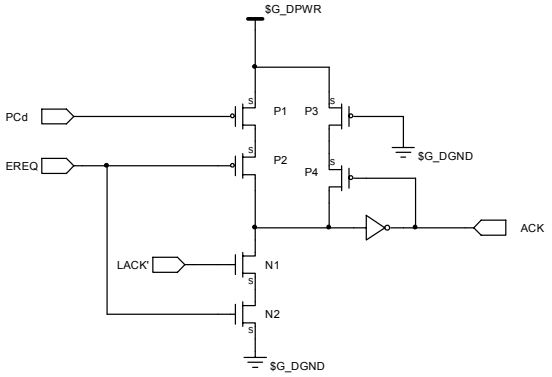

Figure 6: SBD_ACK transistor-level implementation

4.1 Top-level Block Diagram

Figure 7 shows a 12-bit self-timed adder built from three-bit adder blocks (ADD3) and three-bit plus-1 blocks (INC3). The ADD3 blocks compute a three-bit addition in a single domino gate. The INC3 blocks at the bottom of the figure use a chain of domino gates to generate the carry signal based on the results of the ADD3 blocks at the top. In this block diagram we show a four-stage design but the circuit is easily extended to make larger adders. Aside from its use as an example of the SBD wrapper this is the main advantage of this adder architecture. Each of the sub-blocks in Figure 7 is built using our SBD wrapper and a domino function block.

Using this design structure we can chain as many stages (ADD3 + INC3) as needed to form a larger self-timed adder. Note that there is no dependence between ADD3 components. Thus, all ADD3 components can compute simultaneously when the request signal arrives. The INC3 blocks generate a rippled carry output. The worst-case delay is thus one stage of ADD3 and the total delay of the INC3 chain. The delay of the INC3 chain varies according to the data. The collected latch-acknowledge signals (LACK) from

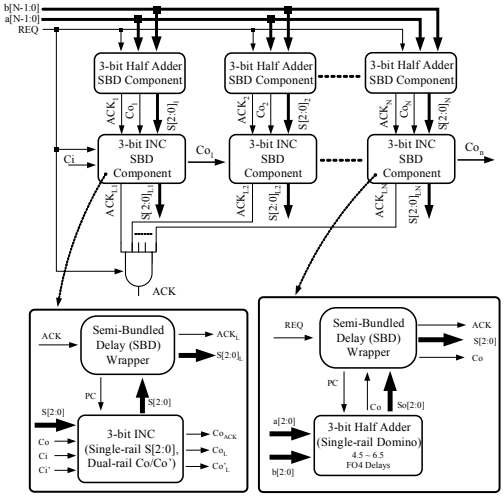

Figure 7: 12-bit Self-Timed Adder Example

the INC3 blocks determine when the final carry has been computed.

4.2 PSPICE Simulation

We simulated this 12-bit self-timed SBD adder in PSPICE using TSMC 0.25um 2.5V CMOS models. Each ADD3 component has its own domino SBD wrapper and completes its task in 1465ps to 1741ps (~ 17 FO4). This delay consists of the domino function evaluation delay, register (latch) delay, and handshaking overhead. The delay variation is small because the ADD3's Domino logic core takes only 470ps to 680ps of calculation time (~5 FO4). The overhead due specifically to the wrapper circuit (that is, additional overhead not found in a standard domino version) is around 450-500ps, or around 5 FO4 delays in this technology. High-performance microprocessors use more than 20 FO4 delays between latches [8] indicating that our ADD3 functional block is perhaps too small to be compared directly with realistic data path circuits. Larger domino circuits would suffer less overhead penalty from the SBD wrapper. However, it does demonstrate that dynamic domino circuits can be used in a straightforward way in self-timed designs.

A purely static design for exactly the same circuit using a full P-type pull-up stack and N-type pull-down stack with the same output latch and the same bundling delay margins simulates at 1915ps. So, even with the overhead of the wrapper circuit and the relatively small size of the domino function block, the domino version runs about 10% faster than the static version. Larger differences would be expected for larger domino function blocks.

We use a modified SBD wrapper for building the INC3 block. The critical input signal for an INC3 block is the carry out from the previous stage. Thus, we implement the carryout signal as a dual-rail domino circuit for easier completion detection. The other signals are still are implemented as single-rail. Using a dual-rail signal for the carryout allows faster INC3-chain completion by removing the output latch and by distributing the early–completion detection circuits in the INC3 chain.

Not surprisingly, the delay variation versus the wrapper overhead is much better for the INC3 chain than for a single ADD3 circuit. The domino function blocks in the INC3 circuits chain together directly with no latch overhead in the individual stages. The 12-bit SBD INC component simulation evaluates between 860ps and 1460ps. The worst-case only happens when all INC3 blocks need to wait for the carryout from the previous stage. If any intermediate carry out completes earlier, the worst-case carry chain will break and the 12-bit INC also completes earlier.

This extensible adder example demonstrates two ways to use the SBD wrapper with domino function blocks. You can use either an SBD wrapper with an output latch for each function block and achieve deep pipelining at the expense of extra overhead, or you can use the SBD wrapper for completion detection without the output latches and let the domino function blocks chain together as a larger circuit.

5. CONCLUSIONS

We have shown a simple wrapper circuit that can make high-performance dynamic-logic function blocks usable in a self-timed system. The wrapper presents a standard self-timed req/ack protocol at its interface, and implements pre-charge/evaluate sequencing, variable-time completion detection, and possibly output latching for the dynamic function block inside the wrapper. The completion detection operates by either directly sensing the completion of the dynamic function block or by a matched delay. Because this has the potential to signal completion earlier than a purely bundled delay we call this approach Semi-Bundled Delay (SBD).

An SBD wrapper is suitable for any dynamic logic family that include a pre-charge phase that sets the outputs to a known level and has monotonic output behavior. Specifically we have shown an example using wrapper circuits around dynamic domino function blocks. The wrapper circuits themselves are either generalized C-elements or domino circuits. For the simple ADD3 circuit the overhead due to the wrapper itself is about 5 FO4 delays. Including this overhead, the domino circuit is about 10% faster than a fully static version of the circuit. Larger differences can be expected for more complex domino function blocks.

Of course, there are many other choices for implementing the wrapper itself. We have also designed and simulated DCVSL circuits for the wrapper, and are exploring domino function blocks that look more like finite state machines than just simple combinational functions.

Semi-bundled wrapper circuits allow a designer to take advantage of high-speed dynamic data path circuits that can be used with any self-timed or asynchronous design style that relies on an explicit completion signal.

6. REFERENCES

[1] Scott Hauck. Asynchronous design methodologies: An overview. *Proc. of the IEEE*, 83(1):69-93, January 1995.

[2] Charles L. Seitz. Self-timed VLSI systems. In Charles L. Seitz, editor, *Proceedings of the 1st Caltech Conference on Very Large Scale Integration*, pages 345-355, Pasadena, CA, January 1979. Caltech C.S. Dept.

[3] Ivan E. Sutherland. Micropipelines. *Communications of the ACM*, 32(6):720-738, June 1989.

[4] Tom Verhoeff. Delay-insensitive codes--an overview. *Distributed Computing*, 3(1):1-8, 1988.

[5] Jens Sparsø and Steve Furber, editors. *Principles of Asynchronous Circuit Design: A Systems Perspective*. Kluwer Academic Publishers, 2001.

[6] Erik Brunvand, Steven Nowick, and Kenneth Yun. Practical advances in asynchronous design and in asynchronous-synchronous interfaces. In *Proc. ACM/IEEE Design Automation Conference*, pages 104-109, 1999.

[7] Neil Weste and Kamran Eshraghian. *Principles of CMOS VLSI Design*. Addison-Wesley Publishers, 1993.

[8] David Chinnery and Kurt Keutzer. Clo*sing the Gap Between ASIC & Custom*. Klewer Academic Publishers, 2002.

[9] Mark E. Dean. *STRiP: A Self-Timed RISC Processor Architecture*. PhD thesis, Stanford University, 1992.

[10] S. M. Nowick. Design of a low-latency asynchronous adder using speculative completion. *IEE Proceedings, Computers and Digital Techniques*, 143(5):301-307, September 1996.

[11] Kenneth Y. Yun. Automatic synthesis of extended burst-mode circuits using generalized C-elements. In *Proc. European Design Automation Conference (EURO-DAC)*, pages 290-295, September 1996