A Pipelined Clock-Delayed Domino Carry-Lookahead Adder

Bhushan A. Shinkre and James E. Stine
VLSI Computer Architecture, Arithmetic, and CAD Research Laboratory
Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, Illinois 60616, USA
bshinkre,jstine@ece.iit.edu

ABSTRACT

Clock-delayed (CD) domino is a dynamic logic family developed to provide both inverting and non-inverting logic on single-rail gates. It is self-timed and can be easily pipelined for superior speed performance which makes it an attractive option in high-speed logic implementation. This paper presents the design of two different high-speed pipeline configurations of a 32-bit carry look-ahead adder using CD domino gates utilizing efficient clocking methodology to reduce the overall critical path delay.

Categories and Subject Descriptors

B.2.4 [**Arithmetic and Logic Structures**]: High-Speed Arithmetic; B.7.1 [**Integrated Circuits**]: Types and Design Styles—*algorithms implemented in hardware, VLSI*

General Terms

Algorithms, Performance, Design

Keywords

Arithmetic, VLSI, Dynamic Logic

1. INTRODUCTION

Dynamic gates are primarily designed for high-speed logic implementations [3]. Consequently, high-speed microprocessors are built using a wide variety of dynamic logic styles [5]. Advantages of using dynamic logic over static logic are that it has a reduced input capacitance and it uses fewer pMOS logic transistors. However, one of the inherent disadvantages of using dynamic gates is their inability of producing both inverting and non-inverting logic.

Most logic applications demand flexibility of generating both polarities of logic states. Therefore, practical implementation of dynamic gates becomes difficult and complex arithmetic or architectural modifications are necessary to facilitate their use. Although dual-rail and differential dynamic logic styles [2] are designed to provide a solution, they have intrinsic disadvantages, such as a large device overhead, large layout area, and vulnerability to discrepancies between input rates at external inputs to the circuit [1].

Clock-delayed (CD) domino logic [8] is one solution of integrating non-inverting logic and inverting logic by using a self-timed delay-matched clock tree for the precharge and evaluation clock in a pipeline stage. The clock to the gate is delayed until the inputs to the gate have been stabilized. This allows the gate to accept stable inputs without violating the monotonicity principle. The monotonicity principle is critical to domino gate design and states that dynamic gates should have monotonically rising inputs during the evaluation phase.

Similar domino designs, such as wave-domino logic [4], have been introduced, but the idea of generating both inverting and non-inverting outputs in wave-domino is never considered. CD domino take pipelining advantages of wave-domino gates and also provides both inverting and non-inverting outputs. The pipeline can be further effectively configured for high-speed operation by incorporating high-fanin and high-speed gates, and time borrowing. This reduces the critical path latency and increases the speed of operation remarkably.

In this paper, a 32-bit carry look-ahead adder (CLA) [6] is selected to demonstrate design flexibility and speed advantages of CD domino gates. As opposed to previous CLA designs, this paper employs efficient pipeline utilization of CD domino gates to improve the overall worst-case critical path delay. This paper is organized as follows. Section 2 introduces the fundamental concepts behind the CD domino gate design, the delay element design, and the basic pipeline structure. Section 3 discusses physical design issues, the occurrence of precharge glitches in the pipeline, and precharge circuits to solve the glitch problem. Section 4 presents two high-speed pipeline configurations and their respective block equations. Section 5 compares the critical path latency for worst-case addition, average power, gate count, and transistor count measured in both the designs. Finally, Section 6 presents our conclusions.

2. CLOCK-DELAYED DOMINO LOGIC

CD domino gates provide any logic function, inverting or non-inverting without any restrictions on specific polarity.

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

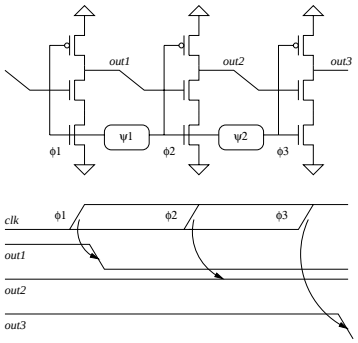

Figure 1: A typical CD domino chain

This is achieved through the employment of self-timed precharge and evaluation clocks. However, CD domino gates face disadvantages related to complexity in the design of the clock and the physical dimensions of the gate. A typical CD domino gate consists of a dynamic gate with a delay element. A typical CD domino chain and its clock transition are illustrated in Figure 1.

In Figure 1, a chain composed of three CD domino gates is shown. The self-timed output of the delay elements tells the next gate when the data output is ready to be accepted. Thus, the delay element value to be set is always longer than the worst-case delay of the dynamic gate. A margin is generally given to the clock amount to take care of any imbalance. It is also seen from the clock waveforms that both forms of logic can be obtained by directly connecting the output of the gate to the input of the next gate without violating the monotonicity principle. Only a low to high transition or a stable low state is allowed, otherwise, it results in severe logic glitch.

In CD domino logic, the self-timed clock and carefully designed delay element takes care of this violation. The inputs are only accepted when they are stabilized and, hence, there is no transition at the beginning of the evaluation stage. Regular dynamic gates are vulnerable to this problem and a static inverter is necessary between the two dynamic gates to take care of this problem [3]. It is critical that the clock output rising edge occurs after the gate output has switched over and not before. This is illustrated in Figure 1 where ϕ_2 switches only after $out1$ is stabilized. CD domino gates are implemented on single-rail dynamic gates in contrast to the well-known dual-rail logic gates [7].

Delay elements are used to create delay in the clock used to precharge and evaluate dynamic gates. A typical delay element is built of inverters and transmission gates as illustrated in Figure 2. The delay value is adjusted by designing the gate widths appropriately. These individual delay elements are cascaded in the form of a chain to produce an array of delayed clocks. The primary clock enters the chain

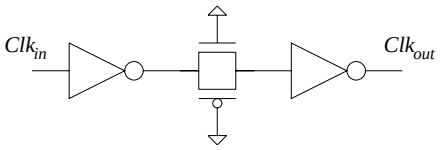

Figure 2: A typical delay element

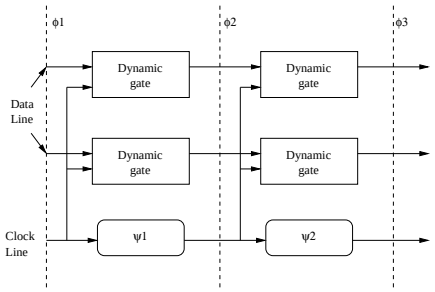

Figure 3: A simplest CD domino pipeline

and delayed clocks are generated which are further routed into the pipeline at appropriate locations. The delay generated for a particular dynamic gate should match its intrinsic gate delay, its output net wiring delay, its fanout gate load delay, and a margin for safe operation. Overall, the gate delay is set equal to the worst-case pull-down delay of the corresponding dynamic gate during the evaluation phase.

The margin given for the delay element is dependent on several factors such as the variations in the fabrication process, voltage and temperature discrepancies, fanout load, parasitic capacitances, and signal delay. A 10-20% margin is assigned while deriving the delay value, however, this is a conservative value which can be altered for better performance [8]. A simple CD domino pipeline scheme is illustrated in Figure 3. The clock output from the previous CD domino gate level in the pipeline is used as the clock output to the next gate level. The primary outputs provided by this clocking scheme must wait for the slowest gate at each gate level.

3. PHYSICAL DESIGN AND PRECHARGE GLITCH ISSUES

It is necessary to design CD domino gates carefully as the gate speed depends largely on its physical dimensions. Apart from the gate size, there are several other factors such as charge sharing, charge distribution, and parasitic capacitance that contribute to the design problem. In this design, the gate size of every transistor in the nMOS logic tree and the evaluation transistor is determined by the incremental proportions of the aspect ratio (W/L). The aspect ratio scaling factor α is determined by the designer through careful attention to fan-out, driving capability, and overall impedance of the circuit. Keepers are not applied in this design because they require precise sizing although they are an attractive option to prevent any logic glitch due to charge leakage if the precharge period is long.

One of the natural problems with CD domino pipeline is the occurrence of precharge logic glitch at the end of a clock phase in the pipeline [9]. For example, consider a specific clock stage in the pipeline in which the gate remains charged during the evaluation phase. Since the precharge and evaluation phases are propagated in a definite clock sequence in a CD domino pipeline, the gate lying in the clock phase preceding this stage goes into early precharge. If the previous gate's output is an input to the subsequent gate, then there is a direct path between the output node and ground. As a result, all the charge at the output node retained during the evaluation stage of this gate is lost to ground. Thus, a

high to low transition occurs intermediately during the evaluation period, which is contrary to the basic dynamic gate operation. This is clearly illustrated by the waveforms in Figure 4, where $out1$ and $out2$ are the outputs of the gates operating on clock phases ϕ_1 and ϕ_2 respectively. A glitch occurs at $out2$ due to the race between ϕ_1 and ϕ_2.

This problem is also defined as precharge race [9]. The race condition is prevented by using inverting dynamic gates that precharge low, like in Figure 4. A high skew static $NAND$ gate is used to force out to precharge low and provide an inverting function during the evaluation phase. The sizing of the nMOS devices in the $NAND$ gate must be done so that its delay is always greater than the delay of the dynamic gate. Combining a precharge low inverting dynamic gate with a non-inverting one allows designers to implement dual-rail output dynamic gates that require single-rail logic. This design requires no delay element and can reduce area without sacrificing performance.

4. IMPLEMENTATION

A carry look-ahead adder can easily be configured and designed as a pipeline, which makes it an appropriate choice for CD domino implementation. Two efficient pipeline configurations are proposed that have a significant speed advantage over previous designs [10]. Both configurations are designed and simulated to measure the worst-case critical-path latency, average power, gate count, and transistor count.

4.1 Pipeline design 1: ADD4x8

In this design, a CD domino design is implemented for a 32-bit carry look-ahead adder that employs block carry look-ahead generation with r=4 blocks, as in [10]. The block diagram of this configuration is illustrated in Figure 5. This adder uses four 8-bit reduced full adder (RFA) blocks with a single level 4-bit block carry look-ahead generator (BCLG). A RFA is a normal full adder in which the sum, generate and propagate signals are produced. The carry-out signal, which is normally an output to a full adder, is eliminated in a RFA. Each 8-bit RFA generates 8-bit sums (s_{0-7}, s_{8-15}, s_{16-23}, s_{24-31}), 8-bit generates in both polarities (g_{0-7} $\overline{g}_{0-7}$, g_{8-15} $\overline{g}_{8-15}$, g_{16-23} $\overline{g}_{16-23}$, g_{24-31} $\overline{g}_{24-31}$), and 8-bit propagates ($\overline{p}_{0-7}$, $\overline{p}_{8-15}$, $\overline{p}_{16-23}$, $\overline{p}_{24-32}$).

Each 8-bit RFA also has eight internal carry generators (c_{0-7}, c_{8-15}, c_{16-23}, c_{24-31}), pipelined with four clock phases. The generate and propagate bits of each 8-bit RFA block enter the 4-bit BCLG along with the carry input c_0. The BCLG further generates 4-bit block propagate ($\overline{P}_1$, $\overline{P}_2$, $\overline{P}_3$, $\overline{P}_4$), 4-bit block generate ($\overline{G}_1$, $\overline{G}_2$, $\overline{G}_3$, $\overline{G}_4$), and finally computes 4-bit block carry ($\overline{C}_1$, $\overline{C}_2$, $\overline{C}_3$, $\overline{C}_4$) signals in a pipeline operating on four clock phases. $\overline{C}_1$, $\overline{C}_2$, and $\overline{C}_3$ are input into the second, third, and fourth 8-bit RFA block, respectively.

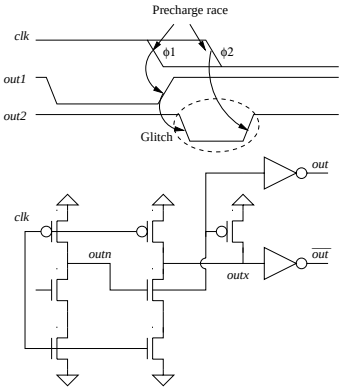

Figure 4: Glitch problem

The pipeline design is demonstrated in Figure 6. It is divided into three sections. Pipeline section 1 constitutes the computation of generate and propagate signals over all four 8-bit RFA blocks. Pipeline section 2 computes the sum bits of the first 8-bit RFA block and the block carry signals of the BCLG. Pipeline section 3 calculates the internal block carry signals and outputs the sum bits of the second, third, and fourth 8-bit RFA block, respectively. The entire addition operation completes in twelve clock phases.

Static inverters are necessary at the end of the BCLG pipeline to buffer the signal as dynamic gates have disadvantages related to limited fanout and load driving capability. The worst-case critical-path delay is the sum of individual gate delays falling on the path, which is approximately equal to the sum of the individual clock delays from ϕ_1 through ϕ_8. The pipeline sections are demonstrated in the block diagram in Figure 5, where bold arrows represent section 1 of the pipeline, dashed arrows represent section 2 of the pipeline, and dotted arrows represent section 3 of the pipeline.

4.2 Pipeline design 2: ADD1x32

The carry generation equations of the 4-bit BCLG in the previous configuration are rearranged to incorporate high-fanin and high-speed NOR gates. This a 32-bit CLA with a block size of $r = 1$ blocks. The block diagram of the configuration is illustrated in Figure 7. The CLG in this case is 32-bit, which computes carries from propagates and generates. Each propagate and generate are computed individually from each RFA. As opposed to the design in Figure 5, the carry equations are generated for each bit, therefore, there is no need for carry generation logic into a subsequent block.

The pipeline design is shown in Figure 8. Similar to the previous configuration, the pipeline is divided into three separate sections. Pipeline section 1 constitutes the computation of generate and propagate signals for all 32 bits that are further propagated into the 32-bit CLG in pipeline section 2. Pipeline section 2 computes the 32-bit carry outputs and the final carry output, and they are further routed back to the individual 1-bit reduced full adders. The final pipeline section is a single clock driven gate that computes the sum output. The critical path delay here is approximately equal to the sum of the clock delays from ϕ_1 through ϕ_5 since the final carry output c_{32} is generated at the end of the fifth clock phase in section 2. The pipeline sections are demonstrated in the block diagram in Figure 7, as in Figure 5.

5. RESULTS AND COMPARISON

Both the pipeline configurations, described above, are designed with an AMI 0.6 μm technology model and simulated with Synopsys' HSpice. Table 1 shows typical clock delays with 20% safety margin for each configuration. Safety margins are typically allocated in dynamic designs due to the invariability within clocking, however, as explained previously, tighter safety margins could be used to improve per-

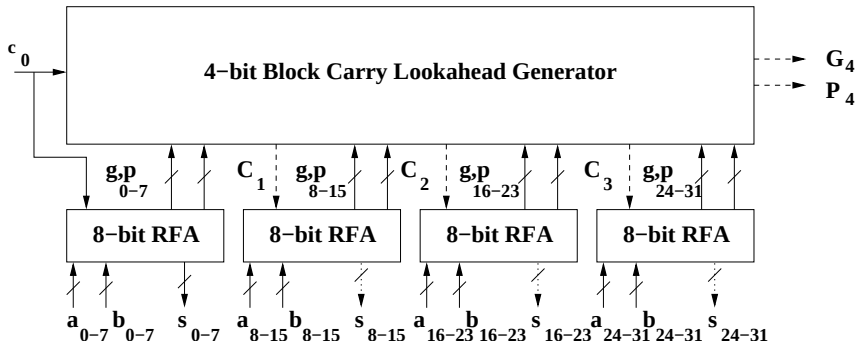

Figure 5: Block diagram of ADD4x8

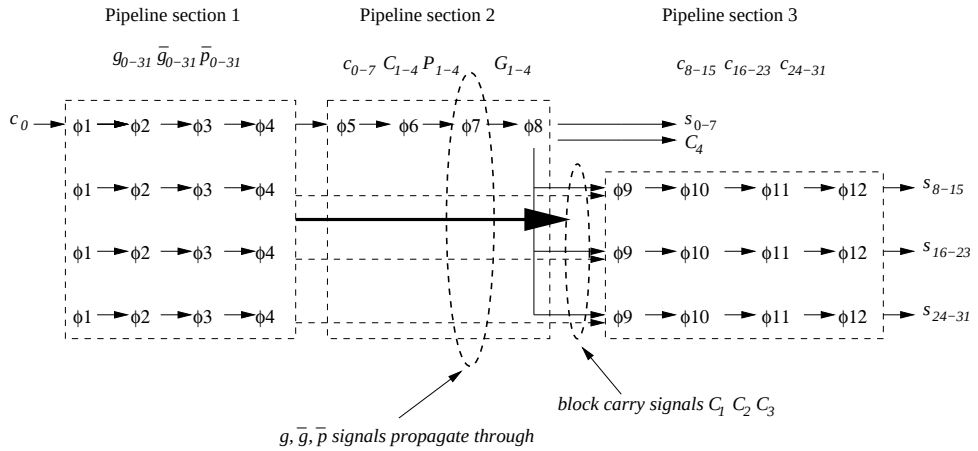

Figure 6: Pipeline design 1 - ADD4x8

formance. In the table, ψ is the difference between the two successive clock phases, such that ψ_1=ϕ_2-ϕ_1, ψ_2=ϕ_3-ϕ_2,...., ψ_{11}=ϕ_{12}-ϕ_{11}, as illustrated in Figure 3.

Comparison of the two configurations is done on the basis of critical path delay for the worst-case addition, average power consumption, gate count, and transistor count. The results are tabulated in Table 2. From Table 2, the critical path delay measurement implies that the ADD 1x32 configuration is 6% faster than the ADD4x8 configuration. The delay results from previous work are also shown [10]. Although the results obtained in this paper are compared with the results obtained for 1 μm and 0.25 μm, the design in this paper illustrates a more efficient pipelining scheme to reduce the overall critical-path latency.

Although the power consumption is not computed in [10], this paper computes the power consumption to illustrate the trade-off between power and speed. From the results, the average power consumption of ADD1x32 is almost double of that of ADD4x8 since the number of transistors and gates are more. It is estimated that the ADD1x32 configuration consumes 3.48 times the real estate of the ADD4x8 configuration.

Clock delays	ADD4x8 (ps)	ADD1x32 (ps)
ψ_1	110	110
ψ_2	190	540
ψ_3	270	200
ψ_4	150	200
ψ_5	110	110
ψ_6	110	-
ψ_7	150	-
ψ_8	470	-
ψ_9	110	-
ψ_{10}	110	-
ψ_{11}	150	-

Table 1: Clock delay values

6. CONCLUSION

Results shown in Section 5 illustrate that the ADD1x32 design has a considerable speed improvement over ADD4x8 design at the cost of increased power consumption and transistor count. This speed increase is obtained by using high-

Design	Process (μm)	Gates	Transistors	Delay (ns)	Power (mW)
ADD4x8	0.60	533	2,426	1.38	303.26
ADD1x32	0.60	883	8,447	1.30	656.77
ADD4x8 [10]	0.25	423	1,620	1.40	-
ADD1x32 [10]	0.25	897	8,737	1.10	-
ADD4x8 [10]	1.00	423	1,620	4.60	-
ADD1x32 [10]	1.00	897	8,737	3.50	-

Table 2: Simulation results

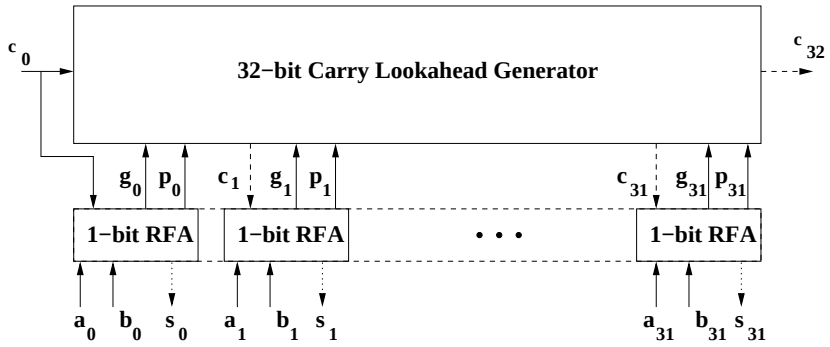

Figure 7: Block diagram of ADD1x32

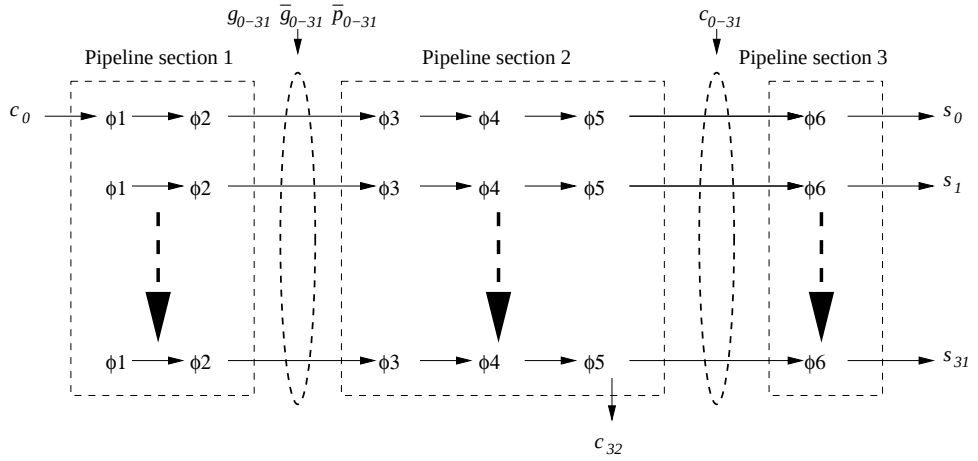

Figure 8: Pipeline design 2 - ADD1x32

fanin and high-speed *NOR* gates, and by implementing efficient pipelining methods like time borrowing. Power consumption in both configurations is less than in regular domino gates because the power consumption is more regularized due to the delaying of the clock signals. This paper shows that an efficient pipeline can enhance the use of CD domino for datapath elements such as adders.

CD domino provide high-speed, high-fanin, and compact gates with flexibility in design and logic output, ideal for high-speed adder design. Robust CD domino adder pipelines can be designed with meticulously designed clock generating circuits. CD domino is a suitable choice for datapath and pipeline circuit design, as demonstrated in this paper.

7. REFERENCES

[1] K. Gaj, E. G. Friedman, and M. J. Feldman. Choice of the optimum timing scheme for RSFQ digital circuits. In *Proceedings of 5th Proc. 5th Int. Workshop on High-Temp. Supercond. Electr. Dev.*, pages 39–40, May 1997.

[2] L. G. Heller. Cascode voltage switch logic: A differential CMOS logic family. In *IEEE International Solid-State Circuits Conference*, pages 16–17, 1995.

[3] R. H. Krambeck, C. M. Lee, and H. F. S. Law. High-speed compact circuits with CMOS. *IEEE Journal of Solid State Circuits*, 17(3):614–619, June 1982.

[4] W.-H. Lein and W. P. Burleson. Wave-domino logic: theory and applications. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 42(2):78–91, Feb 1995.

[5] S. D. Naffziger, G. Colon-Bonet, T. Fischer, R. Riedlinger, T. J. Sullivan, and T. Grutkowski. The implementation of the Itanium 2 microprocessor. *IEEE Journal of Solid State Circuits*, 37(11):1448–1460, 2002.

[6] A. Weinberger and J. L. Smith. A logic for high-speed addition. *National Bureau of Standards Circular 591*, pages 3–12, 1958.

[7] T. Williams and M. Horowitz. A zero-overhead self-timed 160-ns 54-b CMOS divider. *IEEE Journal of Solid State Circuits*, 26(11):1651 –1661, Nov 1991.

[8] G. Yee and C. Sechen. Clock-delayed domino for dynamic circuit design. *IEEE Transactions on VLSI Systems*, 8(4):425–430, Aug 2000.

[9] G. S. Yee. *Dynamic Logic Design and Synthesis Using Clock-Delayed Domino*. Ph.D. thesis, University of Washington, Seattle, 1999.

[10] G. S. Yee and C. Sechen. Clock-delayed domino for adder and combinational logic design. In *IEEE International Conference on Computer Design*, pages 332–337, 1996.