

Design of a 64-Bit Low-Energy High-Performance Adder using Dynamic Feedthrough Logic

Pierce Chuang, David Li, and Manoj Sachdev
Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada
{pichuang,d4li,msachdev}@uwaterloo.ca

Abstract—In this work, a new design approach in implementing low-energy, high-performance 64-bit adder using dynamic feedthrough logic (DFTL) is introduced and analyzed. Design issues of using DFTL in several logic depth are analyzed in order to achieve the best optimal balance between performance and power consumption. A “timing window” technique is also proposed to reduce the amount of excessive power dissipation in the DFTL approach. A 64-bit Sklansky carry-merge adder is used as a benchmark comparison between different logic styles including DFTL, CDL, dynamic, and static logic. Simulation results reveal that the proposed work achieves better performance and is more energy efficient than the other logic styles for high performance adder designs.

I. INTRODUCTION

In the past, various low-energy, high-performance adder design methodology on the carry-merge trees have been proposed. Consequently, compound domino logic (CDL) has become the most popular logic style in adder designs [1][2][3][4] owing to its reduced power consumption and performance superiority over pure dynamic and static gates respectively.

In this paper, a new approach in designing low-energy, high-performance adder using dynamic feedthrough logic (DFTL) is presented. Traditional compound domino logic (CDL) has been modified where the static logic gates have been replaced by FTL-based logic gates. While this approach results in superior performance, the amount of direct path current along with reliability issues are becoming a major concern. In this work, we attempt to address these shortcomings of using DFTL in logic operations with an analysis on the optimal sizing ratio and a “timing window” technique. For comparison purposes, the energy vs. delay (E-D) behavior of identical 64-bit Sklansky carry-merge tree implemented in DFTL, CDL, dynamic logic, and static logic gates is analyzed and compared.

The rest of this paper is organized as follows: Section II reviews previous work on feedthrough logic (FTL). Section III focuses on the design issues associated with dynamic feedthrough logic. Section IV presents the carry-merge architecture and introduces a timing window technique that further reduces the power consumption. Simulation results are shown in section V. Finally, Section VI concludes with some final remarks and comments.

II. PREVIOUS WORKS

The concept of FTL in CMOS technology was first proposed in [5][6], and the basic structure is shown in Fig. 1. The principle of FTL works as follows: During the high phase of CLK (reset period), the output node is pulled to ground through transistor Q2. When CLK becomes low (evaluation period), the output node conditionally evaluates to either logic high or low, depending on the inputs to NMOS pull-down network (PDN). FTL is considered much faster than traditional

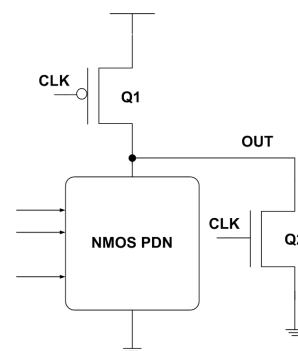


Fig. 1. Pure FTL Structure

logic gates due to the following reasons.

- It only requires logic expression using NMOS transistors and hence the load is gradually reduced.
- The critical path is constant. In this case, the critical path is always a PMOS transistor regardless of the logic expression.
- The output is pre-evaluated (ratioed logic) before the inputs from the preceding stage is ready.

Despite its performance advantage, FTL suffers from reduced noise margin, direct path current, and non-zero nominal low output voltage which are all caused by the contention between PMOS and NMOS during the evaluation period. Moreover, cascading multiple FTL stages together to perform complicated logic evaluations, such as addition, are not practical. When CLK is low, transistor Q1 of every stage turns on and the output begins to rise. This will result in false logic evaluation since initially there is no contention between Q1 and NMOS PDN because all inputs to NMOS are reset to low during the reset period. This not only diminishes FTL's speed advantage but also raises a serious reliability problem.

To demonstrate this problem, a chain of inverters implemented with FTL are simulated and the output waveform is shown in Fig. 2. At even number of stages, the output node will be initially pulled up, and then settle back to the non-zero output low voltage. As the number of stages increase, the unwanted glitch also aggravates and takes longer time to settle back. Clearly, further improvements are required to make FTL a practical circuit design. Therefore, in the following section, we proposed a new type of logic style called dynamic feedthrough logic which aims to solve the aforementioned problems.

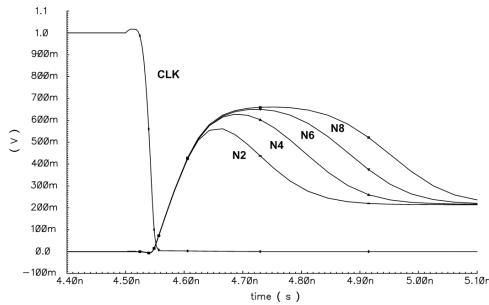


Fig. 2. Simulated Unwanted Glitch at Different Logic Depth in FTL Implementation

III. DYNAMIC FEEDTHROUGH LOGIC

In traditional CDL, complex static gates are placed at the output of dynamic gates instead of inverters. A dynamic feedthrough logic (DFTL) gate is shown in Fig. 3 where static gates are replaced by FTL. Transistor Q2 shown in Fig. 1 is no longer necessary in this design because during the precharge period (CLK is low), the inputs of NMOS PDN which come from the dynamics gates always charge to Vdd. Furthermore, DFTL eliminates the problem of false logic evaluation associated with cascaded FTL because the inputs to FTL's NMOS PDN are always high when first entering the evaluation period. Therefore, FTL gates enter the contention mode and conditionally make a low to high transition depending on the inputs during the evaluation period.

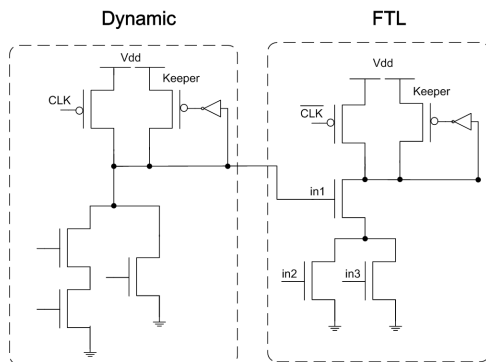


Fig. 3. DFTL Circuit Implementation

One concern regarding to DFTL is the reduced noise margin at the output of dynamic logic caused by the glitch at its inputs during the evaluation period. This glitch is caused by

the contention in the previous stage and cannot be eliminated. To maintain the same unity-gain dc noise (UGDN) level, the keeper of the dynamic stage therefore has to be upsized, which degrades the performance and also increases the power consumption. Hence, it is important to determine the optimal FTL driver/logic ratio that achieves the lowest delay and power dissipation while maintaining sufficient noise margin. The UGDN level defined in [7] is not applicable in this work because no inverter is presented between stages. Instead, the UGDN level in this work is defined as the dc-noise level at the input generating an equal level of voltage drop at the output of the same gate. A sufficient DC noise robustness of dynamic logic gates is 0.1 and is defined by the following equation:

$$\frac{UGDN}{1V} = 0.1, UGDN = 100mV \quad (1)$$

Based on the above equation, an acceptable noise margin for dynamic gates is 100mV. Since outputs of dynamic gates are precharged to Vdd = 1V initially, an acceptable output level then becomes 1V - 100mV = 0.9V

The effective size ratio between transistor Q1 and NMOS PDN from Fig. 1 that results in different glitch level is simulated in 90nm technology with 1V nominal supply voltage and is summarized in Table I. In order to determine proper keeper sizing for different glitch level, an additional 100mV DC noise is coupled with each glitch level and the total glitch is feed to the inputs of dynamic gates. Keepers of the dynamic gates are then upsized until the voltage of the dynamic output node drops less than 100mV. Table I also summaries the required keeper size to meet the target DC robustness for each glitch level.

TABLE I
SIMULATION RESULTS FOR GLITCH LEVEL VS. PROPER SIZING RATIO

Glitch (mV)	Size Ratio	Proper Keeper Sizing (μm)
118	1	0.12
150	1.2	0.12
220	1.6	0.41
263	1.8	1
312	2	1.75

A three stage circuit consisted of Dynamic-FTL-Dynamic gates shown in Fig. 4 is simulated. Sizing of the two dynamic circuits (except the keepers of the dynamic gates on the last stage) remains constant and the output of the third stage (dynamic) is driving a $20\mu\text{m}$ transistor load to mimic the actual adder implementation. Power, delay, PDP and EDP are measured and summarized in Table II. The effective size ratio of 1.6 is the most optimal ratio in terms of delay and EDP for DFTL while maintaining sufficient UGDN. Therefore, this size ratio will be used when designing the 64-bit adder.

IV. 64-BIT CARRY-MERGE ADDER ARCHITECTURE

In this work, a single cycle, sparse-4, radix-2 Sklansky architecture proposed in [4] (Fig. 5) is used as a test bench to demonstrate the performance advantage of the DFTL compared to other logic styles. The upper 32-bit block is not

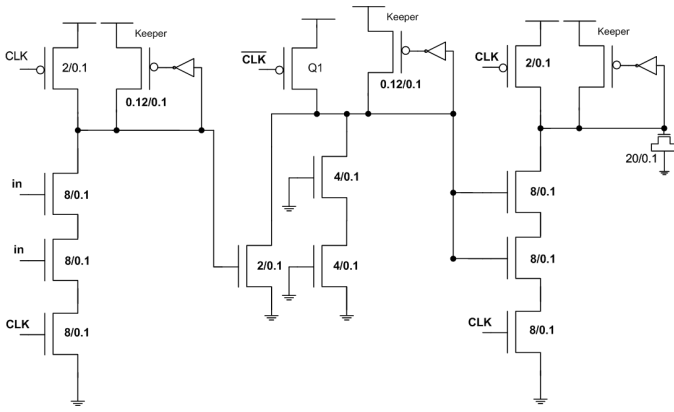


Fig. 4. Three Stage Dynamic-FTL-Dynamic Circuit

TABLE II
NORMALIZED POWER, DELAY, PDP, AND EDP FOR A THREE-STAGE
DYNAMIC-FTL-DYNAMIC CIRCUIT

PMOS/NMOS Ratio	Normalized Power	Normalized Delay	Normalized PDP	Normalized EDP
1	1	1	1	1
1.2	1.048	0.899	0.943	0.848
1.6	1.402	0.766	1.074	0.823
1.8	2.024	0.727	1.471	1.069
2	2.818	0.697	1.963	1.368

performance critical because of the two-stage delay (one dynamic and one static) between the lower and upper 32-bit. Therefore, the first two stages of the upper 32-bit are still implemented with traditional CDL while the rest of the tree including the lower 32-bit are implemented with DFTL to reduce the overall power consumption while not damper the performance. To further reduce the power consumption, a “timing window” technique (Fig. 6) is used to alleviate the problem of direct path current between NMOS PDN and Q1 during the contention.

This window can be generated by NAND two clock signals, and the width of this window can be easily manipulated. The “window” signal is only applied to the PMOS transistors on the FTL gates, since dynamic gates do not have the direct current path problem. However, if FTL gates do not evaluate during this window duration (inputs to NMOS PDN arrive after the window), then false logic evaluation occurs. Hence the window duration needs to be sufficient long to ensure proper logic evaluation under different process and temperature variations.

Fig. 7 shows the power consumption vs. window duration for the proposed DFTL adder, which indicates that the timing window technique is effective in reducing the overall power consumption. The same architecture with a window duration of 300ps consumes approximately 35% more power than the one with a window duration of 70ps. Therefore, it is extremely important to utilize the smallest window duration possible while ensuring reliability is not an issue under all corners. In this work, a window duration of five inverter delay, or an equivalent of 128ps, is created. Simulation results indicate that

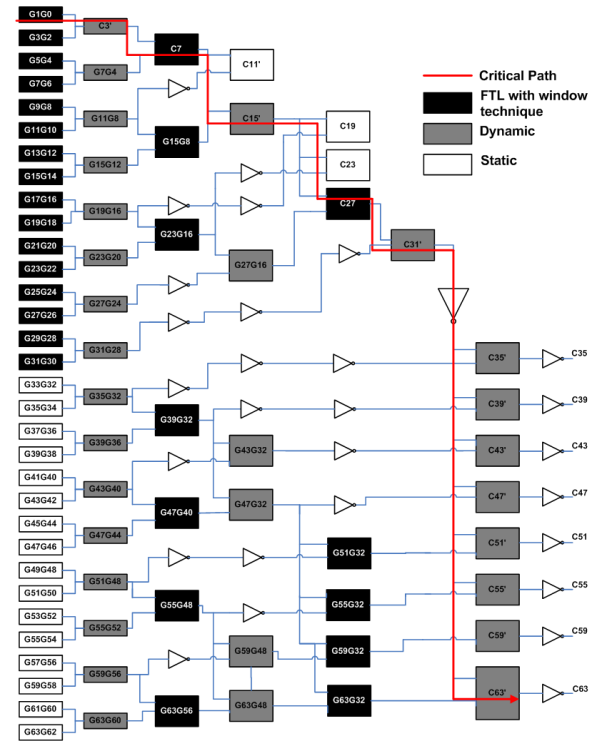


Fig. 5. Carry-Merge Tree for 64-bit Sklansky Adder

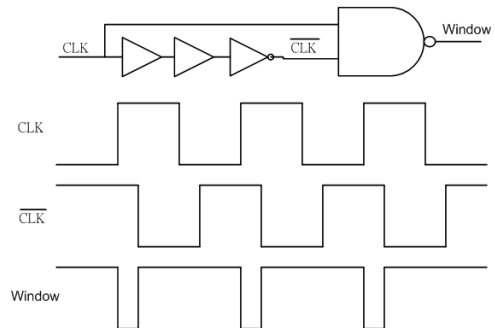


Fig. 6. Window Generation Circuitry and Waveforms

this duration is sufficient for all corner tests, and the resultant power consumption is only approximately 10% more than the duration of 70ps.

V. SIMULATION RESULTS

Performance comparison of 64-bit Sklansky carry-merge adder implemented using DFTL, CDL, dynamic, and static logic style is illustrated in Fig. 8. As the supply voltage is varied, DFTL-based adder is approximately 25% faster than the identical adder architecture implemented using CDL. Table III shows the performance comparison between this work and some of the previous high-performance adder design in terms of Fanout of 4 (FO4) delay [8]. Although only extensive simulation results are provided on the DFTL approach, its performance advantage over similar 90nm high-performance adder design is evident. Hence it is clear that the potential of DFTL-based design is enormous.

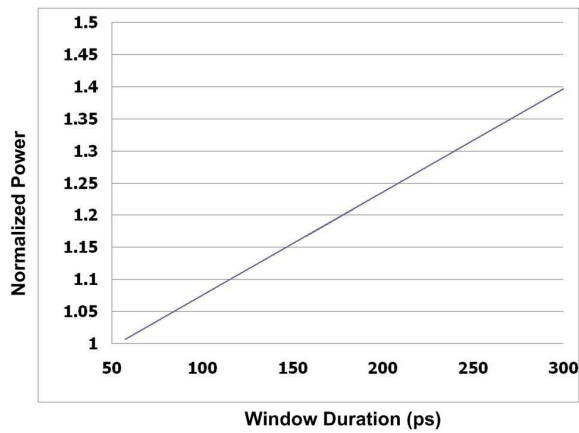


Fig. 7. Normalized Power vs. Window Duration for 64-Bit Sklansky Adder

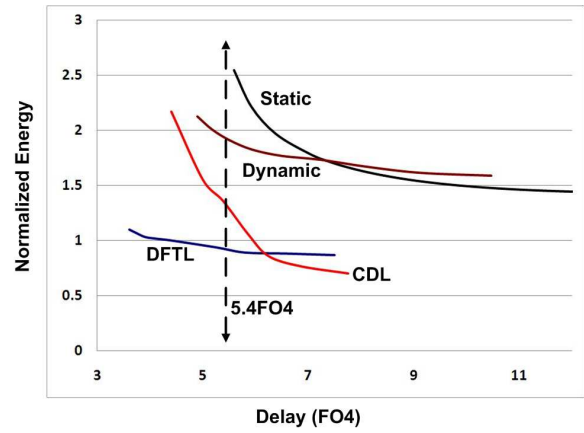


Fig. 9. Normalized Energy-Delay Curve of 64-Bit Sklansky Adder for Various Logic Styles

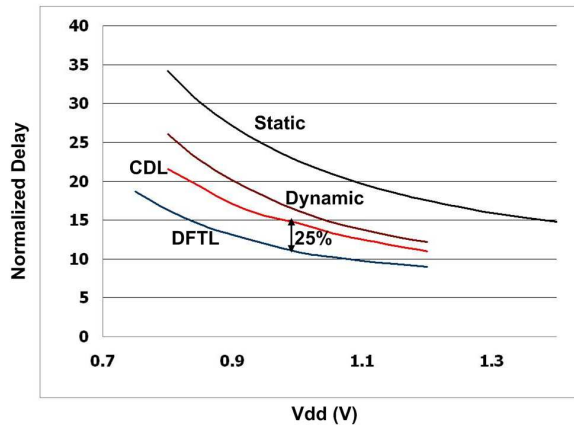


Fig. 8. Normalized Delay Comparison of 64-Bit Sklansky Adder for Various Logic Styles

TABLE III
RECENT HIGH-PERFORMANCE ADDER DESIGNS

Reference	Year	Technology	Supply Voltage	Delay [FO4]
[9]	1996	0.5 μ m	5V	5.56
[10]	2000	0.35 μ m	3.3V	14.3
[1]	2001	0.18 μ m	1.5V	7.44
[11]	2002	0.18 μ m	1.8V	7.4
[3]	2005	90nm	1.3V	7.72
[12]	2006	90nm	1V	7.4
This work	2008	90nm	1V	4.3

Fig. 9 illustrates the energy vs. delay tradeoff curve for the 64-bit Sklansky adder implemented using the aforementioned four logic style. Clearly CDL and DFTL are the superior logic styles over pure static and dynamic. If longer delay can be tolerated, CDL is about 15% more energy efficient than DFTL. However, DFTL becomes energy efficient when performance becomes important. For a delay of 5.4FO4 (Fig. 9), DFTL is approximately 45%, 100%, and 180% (extrapolation) more energy efficient than CDL, dynamic, and static respectively.

VI. CONCLUSION

In this work, we proposed a dynamic feedthrough logic for implementing 64-bit low-energy, high-performance carry-

merge adder. We have analyzed and discussed DFTL's advantages and drawbacks and determined its optimal size ratio. 64-bit Sklansky carry-merge architecture with timing window technique is implemented with DFTL and the performance and energy consumption superiority over other logic styles have been demonstrated through simulation results. For high performance application (5.4FO4 delay), the energy consumption is reduced by 45% with DFTL over CDL configuration. At a nominal voltage supply of 1V, DFTL based adder is approximately 25% faster than CDL configuration.

REFERENCES

- [1] S. Mathew, et.al "Sub-500-ps 64-b ALUs in 0.18- μ m SOI/Bulk CMOS: Design and Scaling Trends". *IEEE Journal of Solid-State Circuits*, Vol. 36(11): pp.1636-1646, November 2001.
- [2] S. Mathew, M. Anders, R. Krishnamurthy and S. Borkar "A 4GHz 130nm Address Generation Unit with 32-Bit Sparse-Tree Adder Core". *IEEE Journal of Solid-State Circuits*, Vol. 38(4): pp.689-695, May 2003.
- [3] S. Mathew et.al "A 4-GHz 300-mW 64-bit Integer Execution ALU With Dual Supply Voltages in 90nm CMOS". *IEEE Journal of Solid-State Circuits*, Vol. 40(1): pp.44-51, January 2005.
- [4] S. Wijeratne et.al "A 9-GHz 64-nm Intel Pentium 4 Processor Integer Execution Unit". *IEEE Journal of Solid-State Circuits*, Vol. 42(1): pp.26-37, January 2007.
- [5] V. Botello et.al "Low Power Arithmetic Circuit in Feedthrough Dynamic CMOS Logic". *49th IEEE International Midwest Symposium on Circuits and Systems*, Vol. 1: pp.709-712, August 2006.
- [6] V. Botello et.al "Analysis of High-Performance Fast Feedthrough Logic Families in CMOS". *IEEE Transactions on Circuits and Systems*, Vol. 54(6): pp.489-493, June 2007.
- [7] A. Alvandpour et.al "A Sub-130-nm Conditional Keeper Technique". *IEEE Journal of Solid-State Circuits*, Vol. 37(5): pp.633-638, May 2002.
- [8] M. Horowitz "VLSI Scaling for Architects". Stanford University.
- [9] S. Naffziger "A Sub-Nanosecond 0.5um 64b Adder Design". *IEEE International Solid-State Circuits Conference*, pp.363, February 1996.
- [10] D. Durate, J. Hezavei, and M. Irwin "Power Consumption & Performance Comparative Study of Logarithmic-Time CMOS Adders". *IEEE Workshop on SiGNAL Processing Systems*, pp.467-475, 2000.
- [11] H. Dao and V. Oklobdzija "Performance Comparison of VLSI Adders Using Logical Effort". *Proceedings of the 12th International Workshop on Integrated Circuit Design*, Vol. 2451: pp.25-34, 2002
- [12] S. Kao, R. Zlatanovici, and B. Nikolic "A 240ps 64b Carry-Lookahead Adder in 90nm CMOS". *IEEE International Solid-State Circuits Conference*, pp.1735-1744, February 2006.