

# Substrate-Bias Optimized 0.18um 2.5GHz 32-bit Adder with Post-Manufacture Tunable Clock

Qi-Wei Kuo<sup>1</sup>, Vikas Sharma<sup>2</sup> and Charlie Chung-Ping Chen<sup>1</sup>

<sup>1</sup>Graduate Institute of Electronic Engineering  
National Taiwan University

No.1, Sec. 4, Roosevelt Road,  
Taipei, Taiwan 106

R92943088@ntu.edu.tw, cchen@cc.ee.ntu.edu.tw

<sup>2</sup>University of Wisconsin-Madison  
Department of Electrical and Computer Engineering

1415 Engineering Drive  
Madison, WI 53706

vikassharma@wisc.edu

**Abstract**—In this paper, we present a 32bit Han-Carlson adder that operates at 2.56GHz and is based on TSMC 0.18um bulk CMOS technology. In this work, we optimize the substrate bias of the adder core to achieve a low power-delay product for low power and high speed purposes, and use a post-manufacture tunable clock structure that manipulates the clock at post-fabrication stage to compensate for the process dependent violation to the timing. Experimental results have shown that the substrate-bias optimization results in a 37% of power delay improvement and utilization of tunable delay elements achieve 50 ps of almost linear clock tunability.

## I. INTRODUCTION

The most important issues in designing an ALU are: 1) High throughput (high operating frequency) 2) Low delay (latency) 3) Low power 4) Robustness of the timing. In this work a substrate bias based power-delay product optimization is adopted to attack the first three issues. As technology scaling continues, it increasingly makes process parameters hard to control. A well-distributed tunable clock structure is used to provide a high speed operation i.e. 2.56GHz and a robust timing in the presence of process violation. In literature, the prefix adder has been the most popular one, due to its regular layout and better performance compared to the conventional adder architectures. It includes [1] [2] [3]. The last one i.e. [3] which was presented by Han and Carlson in 1987 has the lowest area-delay product [11]. For this reason, we use a novel way to implement Han and Carlson's work. In our design, an efficient circuitry for testing, and a phase-locked loop (PLL) for clock generation, has been included. Tunable clock buffers have also been inserted in the critical clock path to make the test circuit more reliable after fabrication. The remainder of this paper is organized as follows. Section II presents the adder architecture and its implementation. Section III discusses the clock distribution and testing issues of the circuit. Section IV discusses the effect of the substrate bias, and in Section V and VI, the layout considerations and the simulation results are given, respectively. Finally, this work is concluded in the Section VII.

## II. 32 BIT ADDER ARCHITECTURE

Basically, the ALU is composed of four parts, named Input Stage, P/G/Partial Sum Generation Stage, Carry Merge Stage, and CSG Stage, respectively. In the following subsection, we introduce the stage functions and their logical structures.

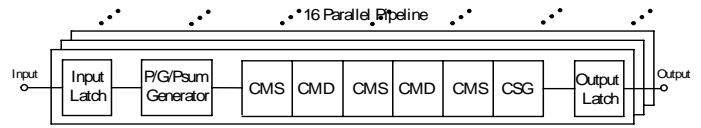


Figure 1 : 32bit Adder Architecture

### 1. P/G/PARTIAL SUM GENERATION

In order to do carry-merge action, Propagate ( $P$ ) and Generate ( $G$ ) bits are generated in this stage. Partial Sum ( $Psum$ ) bits have also been produced here. The equations need to be modified as given below, because inputs are in complementary format:

$$P = \overline{A \cdot B}$$

$$G = \overline{A + B}$$

$$Psum = \overline{A \oplus B} = P \oplus G = \overline{GP} = \overline{G + P}$$

As can be seen from the above equation that there exist only one way to implement Propagate ( $P$ ) and Generate ( $G$ ) bits, but more for Partial Sum ( $Psum$ ) bit. Since we want circuit to be in single rail, for  $Psum$ , the first implementation may increase the loading to input signal, and thus not a suitable choice. In some previous work,  $P \oplus G$  with pass transistors have been utilized to achieve this goal. Transmission gates, however, can degrade the input signal, and which may slower the pull-down speed due to additional routes and transistors existed in the circuit. Figure 2, shows schematic of the  $P$ ,  $G$  and  $Psum$  generating circuit. Since the  $Psum$  signal is not in the critical path, we used static logic to avoid the use of inverter that follows the dynamic  $P$  and  $G$  generating logic. This reduces the latency of the adder core.

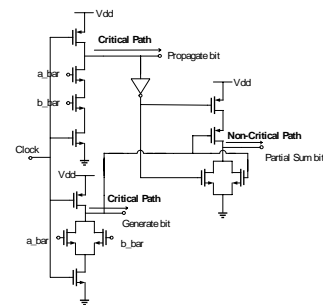


Figure 2 : Circuit Structures of P, G, and Psum

## 2. CARRY MERGE TREE

Carry merge tree is the key structure of the adder. As shown in figure 3, in total, there are five carry merge stages plus another one in Carry-Sum Generator (CSG). Table 1 shows the equations that dictate the carry merging actions. There are two possible ways in which carry merging actions can be executed. The first can be achieved by Static Carry-Merge Logic (CMS) which is a positive-in-negative-out element and another by Dynamic Carry-Merge Logic (CMD) which is a negative-in-positive-out. Based on these properties, we have used Static-Dynamic-Static architecture to implement the carry-merge tree. This avoids using a domino logic, which in turn requires an additional inverter, and results in a larger latency. Figure [4] shows the CMS and CMD. The duplicate pull-down network of the CMD is used to prevent the charge sharing effect. It is noticeable that the output of the CMD to the next CMS stage is not the inverter's output but dynamic logic's output. The inverter that follows the dynamic logic is just used as a keeper to a weakly turn-on PMOS.

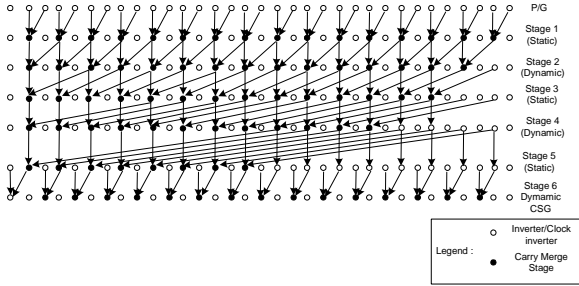


Figure 3 : Carry Merge tree of Han-Carlson Arithmetic

Carry-Merge Stage 1/3/5

$$\begin{aligned} \overline{P_{[i:j]}} &= \overline{P_{[i:k]} \bullet P_{[k:j]}} \\ \overline{G_{[i:j]}} &= \overline{G_{[i:k]} + P_{[i:k]} \bullet G_{[k:j]}} \\ \overline{P_{[i:0]}} &= \overline{P_{[i:0]}} \quad \overline{P_i} = \overline{P_i} \\ \overline{G_{[i:0]}} &= \overline{G_{[i:0]}} \quad \overline{G_i} = \overline{G_i} \end{aligned}$$

Carry-Merge Stage 2/4

$$\begin{aligned} \overline{P_{[i:j]}} &= \overline{\overline{P_{[i:k]}} + \overline{P_{[k:j]}}} \\ \overline{G_{[i:j]}} &= \overline{\overline{G_{[i:k]}} \bullet (\overline{P_{[i:k]}} \bullet \overline{G_{[k:j]}})} \\ \overline{P_{[i:0]}} &= \overline{\overline{P_{[i:0]}}} \quad \overline{P_i} = \overline{P_i} \\ \overline{G_{[i:0]}} &= \overline{\overline{G_{[i:0]}}} \quad \overline{G_i} = \overline{G_i} \end{aligned}$$

Table 1 : Han-Carlson Carry Merge Stage

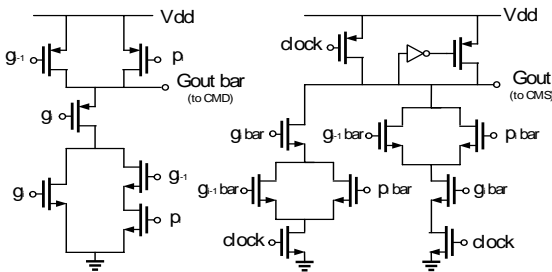


Figure 4 : CMS and CMD

## 3. CARRY-SUM GENERATOR

Once the carry bits are generated, the summations will be produced by executing  $Carry \oplus Psum$ . In order to save additional stage, we merge the last Static Carry-Merge element with the Sum Generating circuit. Figure 5 shows the structure of carry-sum generator. The left side of the circuit is a Static Carry-Merge Logic and the right side is a transmission gate logic which executes  $Carry \oplus Psum$ . Here, the only difference from carry merge stage is that an additional clock inverter with an attached P-latch is used to eliminate the noise for  $\overline{Cin_{even}}$ . Without the P-latch, the noise in the second stage will be huge and can make  $\overline{Cin_{even}}$  hard to recover to logic-1. Moreover, to be able to work even at lower frequencies, the clock in this stage is locally inverted from the initial clock to ensure the result is available at or after the next rising edge of the clock. After creating the true and complementary signal of  $Cin$  for each bit,  $sum$  and  $\overline{sum}$  bits are then created by using NMOS-XOR. In order to release the load to  $Cin$  and  $\overline{Cin_{even}}$  signal and have a full logic-1, pull-up PMOSs, as shown in figure 5, are attached to the output.

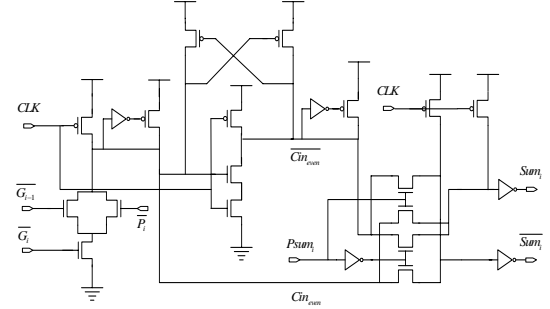


Figure 5 : Key structure of Complementary Signal Generation

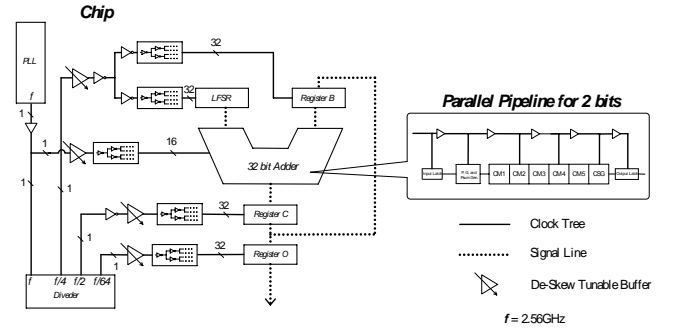


Figure 6 : Clock Distribution

## III. CLOCK DISTRIBUTION AND TESTING ISSUES

Figure 6 shows the clock distribution of our design. It is noticeable that all the elements are positive-edge triggered. An on-chip 2.56GHz clock signal is generated by the PLL, and then a divider divides it and generates the clock signals that are appropriate to each element. Three different frequencies are generated by the divider. The divide-4 clock signal is used to trigger Linear Feedback Shift Register (LFSR) and Register-B. After these two components being driven, the input signal

will enter the 32bit adder. The clock signal feeding the adder directly comes from the PLL. We used delay buffers inside the adder to generate different clock phases that will correctly drive the cells in each adder-parallel-pipeline. After two adder's clock cycles, the summing results will be produced and latched by Register-C which is triggered by the divide- 2 inverting clock. These results will again be latched when the Register-O is triggered by divide-64 clock signal. Based on this clock distribution scheme, the testing issues can be described as follows.

## 1. TESTING ISSUES

Two steps are required to test the circuit, namely as 1) Shifting Mode 2) Testing Mode. In the shifting mode, 32 bit logic-1 signals are shifted to the LFSR, and at the same time Register-B is connected to the ground. Whereas in the testing mode, 32bit logic-1 signals and 32 logic-0 signals are summed by the adder, and Register-C latches the results which will again be latched by Register-B in the next state. In other words, the adder in the testing mode serves as an accumulator. Register-O outputs the add-16-times result of the adder at the frequency of 40MHz, which is low enough to be measured by a logic analyzer

## 2. POST-MANUFACTURE TUNABLE CLOCK

In the testing mode, the synchronicity of the clocks that triggers the registers is important. Moreover, unavoidable clock skew will result after the fabrication. For this, we have utilized four Post-manufacture tunable Clocks to make the circuit more flexible and tunable at post-fabrication stage. Figure 7 shows the circuit architecture of the Post-manufacture tunable Clock. There are three RC branches between the input and output which are replaced by CMOS transmission gates and MOS capacitors. The gate terminals of the transmission gates are connected to the input pad of the chip. By feeding the Logic-1 or Logic-0 signals, we can easily control the on-off of the RC branches and it will change the latency of the delay element. Nonetheless, a careful sizing of the MOS capacitors has been done to achieve a linear delay variation. Figure 8 shows the tuning range of the buffer in SS, TT, and FF corner. The tuning range of the Post-manufacture Clock is about 50ps.

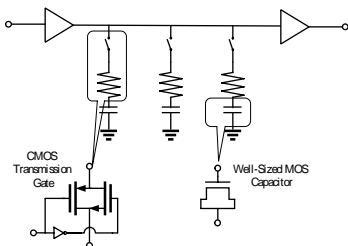


Figure 7 : De-Skew Tunable Buffer

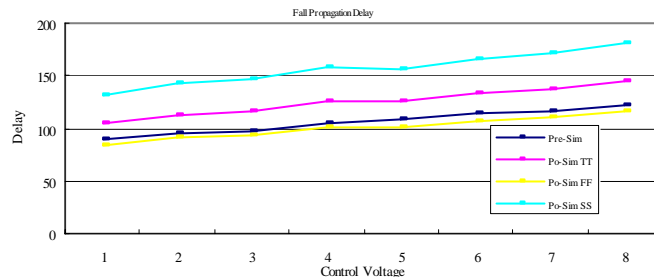
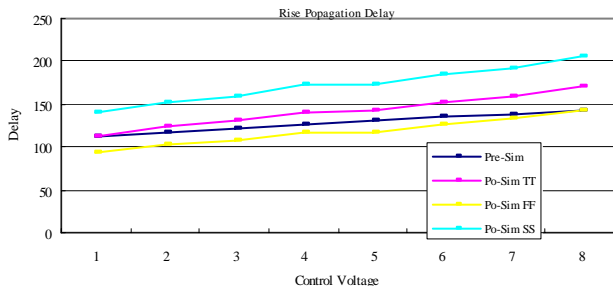
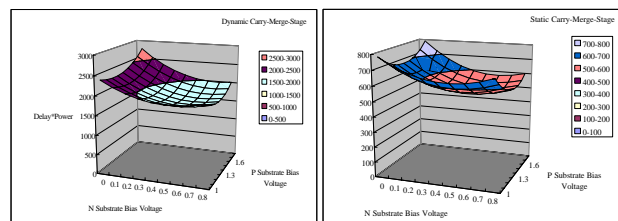


Figure 8 : Tuning range of the De-Skew Buffer.

## IV. SUBSTRATE BIAS

In the deep sub-micron devices, the threshold voltage of the transistor critically influences their speed and leakage current. In general, a higher threshold voltage will reduce the speed and leakage current of the transistor and a lower one will make the device faster with a larger leakage current. By varying the substrate bias, we can manipulate the threshold voltage and can get different delay and power dissipation values. In our design, Carry Merge Stages are the key cells that will affect the delay-power product of the adder. So we try to find an optimum substrate bias voltage of the Carry Merge Stages for a low power and high speed purposes. In Table 2, a comparison between the optimized substrate bias and non-substrate bias circuit has been shown. Row1 and Row2 show the lowest delay-power product in all substrate bias voltage combinations of the Static Carry-Merge (CMS) and Dynamic Carry-Merge (CMD), respectively. Since in total, there are three CMS and two CMD elements, we add their value together, and this result is shown in Row3 of the table. Experiment result shows that a 37% improvement in the delay-power product can be achieved by biasing the substrate voltage of NMOS to 0.55V and PMOS down to 1.45V. Figure 9 shows the relations between the delay power product of the Carry-Merge Stages and the substrate bias.



CMD's Delay-Power Product Mim. CMS's Delay-Power Product Mim.  
Value at  $V_n=0.6\text{ v}$   $V_p=1.4\text{ v}$  Value at  $V_n=0.5\text{ v}$   $V_p=1.5\text{ v}$

Figure 9 : The relations between substrate bias and the delay-power product

	No Substrate Bias			Substrate Bias		
	Delay	Power	D*P	Delay	Power	D*P
CMD	135.21	20.525	2775.2	131.79	13.073	1722
CMS	88.26	22.89	789.02	80.05	6.4341	515.1
C-M Tree	--	--	7917.4	--	--	4991
Impro.Ratio	0 %			37%		

Table 2 Delay-Power Product

## V. LAYOUT CONSIDERATION

Figure 10 shows the layout of the adder circuitry. It includes a phase-locked loop which is used for clock generation, a Han-Carlson Adder (HCA), the clock tree and the testing circuit. Since the power supply voltage predominantly influences the device speed, in this design, IR drop is an important issue that needs to be carefully dealt with. If the latency of the device varies too much, the timing violation will occur and the summing result will be wrong. In order to overcome the IR drop, we have inserted a well-scheduled power grid and placed several VDD pads around the adder core to prevent the power current from passing through too long distance. Besides, the PLL's power supply has been kept separate from the adder's power supply for a better noise immunity.

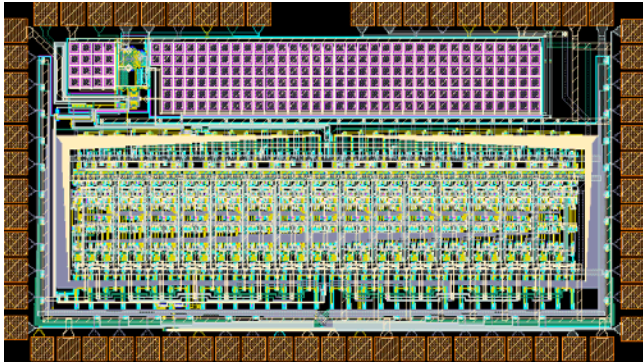


Figure 10: The Layout

Technology	Frequency	Reference
32bit 0.35um	1.25GHz	Wang and Tseng[5]
32bit 1.2um	400MHz	Wang and Willan[8]
32 bit 0.25um	1GHz	A. Goldovsky [6]
32bit 0.16um	1.42GHz	A. Geldovsky [7]
32bit 0.18um	2.56GHz	Our Work

Table 3 : Comparison of some related work

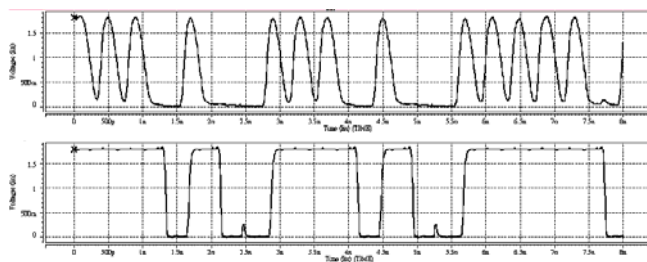


Figure 11 : The Summation result of one bit. The first waveform is the result from the CSG output. The CSG output signal is then latched by the output latch and is shown in the second row.

## VI. SIMULATION RESULT

An extensive Spice level Post-Layout simulation is done using TSMC 0.18 Bulk CMOS Technology Model file. Figure 11 shows one of the 32 summing results of the adder core versus time. The adder's P-substrate is biased at 1.45V and the N-substrate is biased at 0.55V. The result successfully confirms our adder core operating at the frequency of 2.56GHz. Some comparisons in different technology between our work and the related work are listed in Table 3.

## VII. CONCLUSIONS

A Power-Delay Optimized 32bit adder in TSMC 0.18um Bulk CMOS has been presented. At 2.56GHz, the adder core dissipates 357mW with the supply voltage of 1.8V. We improved the power-delay product by 37% and inserted Post-manufacture tunable Clock to overcome the process and temperature variation.

## REFERENCES

- [1] R. P. Brent, H. T. Kung, "A regular Layout for Parallel Adders" IEEE Trans., C-31(3):260-264, March 82.
- [2] P.M. Kogge, H. S. Stone, "A Parallel Algorithm for the Efficient Solution of a Genera Class of Recurrence Equations", IEEE Trans. on Computer Vol .C-22, No. 8, Aug., 1973.
- [3] T. Han, D. A. Carlson, "Fast Area-Efficient VLSI Adders" 8th IEEE Symp. Computer Arithmetic, Como Italy, pp. 49-56, May 87.
- [4] S. Vangal et al, "5GHz 32b Integer-Execution Core in 130nm Dual-Vt CMOS" pp. 334-335, ISSCC 2002
- [5] Chua-Chin Wang, Yih-Long Tseng, Po-Ming Lee, Rong-Chin Lee and Chenn-Jung Huang, "A 1.25GHz 32-Bit Tree-Structured Carry Lookahead Adder Using Modified ANT Logic" IEEE 2003.
- [6] A. Goldovsky, R. K. Kolagotla, C. J. Nicol and M. Besz, "A 1.0-nsec 32-bit Prefix Tree Adder in 0.25-um static CMOS" IEEE 1999.
- [7] Alexander Goldovsky, Hosahalli R. Srinivas, Ravi Kolagotla, and Rodney Hengst, "A folded 32-bit Prefix Tree Adder in 0.16-um static CMOS" Proc. 43rd IEEE Midwest Symp. On Circuit and System, Lansing MI, Aug 8-11, 2000.
- [8] Zhongde Wang, Graham A. Jullien, William C. Miller, Jinghong Wang and Sami S. Bizzan et al, "Fast Adders Using Enhanced Multiple-Output Domino Logic" IEEE 1997.
- [9] Utpal Desai1, Simon Tam, Robert Kim, Ji Zhang, Stefan Rusu, "ItaniumTM Processor Clock Design"
- [10] John G. Maneatis, "Low-Jitter Process-Independent DLL and PLL Based on Self-Biased Techniques"
- [11] Matthew Ziegler and Mircea Stan, "Optimal Logarithmic Adder Structures with a Fanout of Two for Minimizing the Area-Delay Product".

