VLSI DESIGN OF A 1.0 GHZ 0.6-μM 8-BIT CLA USING PLA-STYLED ALL-N-TRANSISTOR LOGIC

Chua-Chin Wang and Kun-Chu Tsai

Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424
email : ccwang@ee.nsysu.edu.tw

ABSTRACT

A high speed 8-bit carry-lookahead adder (CLA) using two-phase clocking dynamic CMOS logic with modified non-inverting all-N-transistor (ANT) blocks which are arranged in a PLA design style is presented in this paper. The pull-up charging and pull-down dis- charging of the transistor arrays of the PLA are accelerated by inserting two feedback MOS transistors between the evaluation NMOS blocks and the outputs. Detailed simulation results reveal appropriate L/W guidelines for the all-N-transistor (ANT) block design. The analysis of the area (transistor count) tradeoff is also provided in this work. The operating clock frequency is 1.0 GHz while the output of the addition of two 8-bit binary numbers is done in 2 cycles. The proposed design methodology is proved to be also suitable for the long adders, e.g., 64-b adders, while the correct output will be ready in 4 cycles if the 64-b adder is composed of hierarchical nine 8-b CLAs.

1. INTRODUCTION

Fast adders are key elements in digital circuits, including multipliers, and DSP chips. Many efforts have been focused on the improvement of adder designs [7]. CMOS dynamic logic is one of the promising options to challenge the GHz operations [3] for the adder design. Other logics suffer from different difficulties. We, hence, propose an all-N-transistor (ANT) non-inverting function block for the high-speed design. Then, an 8-b carry-lookahead adder (CLA) using ANTs which are arranged in a PLA-like structure and triggered by a single clock is implemented. The major advantage of this design methodology is that it is scalable such that long data words, e.g., 64-b binary data, can also be processed. The 8-b CLA using PLA-styled ANT logic is measured to be fully functional up to 1.0 GHz at 5.0 V power supply, while the correct result of addition is available after 2 cycles.

2. HIGH SPEED 8-BIT CLA

2.1. All-N-transistor (ANT) function unit

Although the N-block dynamic logic intrinsically possesses high speed [2], it is not good enough for the operation in the giga hertz range. The reasons are : firstly, the slopes of the clock's edges must be gentle, and secondly, the number of stacks in the evaluation N-block severely affects the size of all of the transistors in the unit. Hence, a modified dynamic logic is presented in Fig. 1. The feature of this modification is the feedback transistor pair, P3 and N3, between the evaluation block and the output.

1). When *clk* = 0, P1 is on and the gate of P2 is precharged to be Vdd. Then, P2 is off and N4 is off. This makes the output to stay at the previous state.

2). When *clk* = 1 and the N-block is evaluated to be "pass", the charge at node *a* should be ground through the N-block and N1 theoretically. Note that N4 is on and N2 is also on at the beginning. If the previous state of output is high, then N3 will be turned via N4. This means that N3 provides another fast discharging path for the charge at node *a*. When the voltage at node *a* is dropped below the threshold voltage of PMOS, P2 and P3 start to be on. The output will then be charged to be Vdd via paths P2 and P3-N4.

This research was partially supported by National Science Council under grant NSC 86-2622-E-009-009 and 87-2215-E-110-010.

3). When *clk* = 1 and the previous state of the output is low and the N-block is evaluated to be "pass," the voltage at node *a* starts to drop. When $V_a - V_{dd} > V_{tp}$, P3 will be on such that the gate of N3 will be charged to be V_{dd}. Not only the charge at node *a* will be discharged faster, but also the output will be charged to high via P2 and N4.

Summarized from 2). and 3). in the above, the output will be high when the N-block is evaluated "pass", i.e., "1", during *clk* = 1.

4). When *clk* = 1 and the N-block is evaluated to be "stop", the charge at node *a* should be kept if the previous state of output is low. There will be no discharging path for node *a* because N3 will be off via N4. If the previous state is high, the output will be ground via N4 and N2 before the voltage at node *a* starts to drop.

Hence, the output will be low when the N-block is evaluated "stop", i.e., "0", during *clk* = 1. The function of ANT logic block, thus, is conclusively correct and noninverting. Restated, P3 and N3, respectively, provide an extra charging path and an extra discharging path such that the speed of the evaluation can be accelerated.

2.2. Sizing problem

One of the reasons why other high-speed logic can not run correctly given clocks with short rise time or fall time is that the size of each transistor can not be tuned properly. Both [1] and [2] intrinsically possess this shortcoming. The sizing problem of the transistors in the ANT besides those in the N-block drastically affect the speed. We have been proceeded several simulations to find out the best figure of merit for the sizing of each transistor in Fig. 1 using TSMC 0.6 μm SPDM technology is tabulated in Table 1.

Transistor	L (μm)	W (μm)
N1	0.6	15
N2	0.6	10
N3	0.6	3
N4	0.6	10
P1, P2	0.6	20
P3	0.6	6
N-block	0.6	10

Table 1 : The sizes of ANT logic block.

2.3. PLA-styled 8-bit CLA design

The formulation of a 8-b CLA is represented by the following equations:

$$\begin{aligned} S_i &= C_{i-1} \bigoplus P_i \\ C_i &= G_{i-1} + P_{i-1}G_{i-2} + P_{i-1}P_{i-2}G_{i-3} + \\ &\quad \ldots + P_{i-1}P_{i-2}\ldots P_1P_0C_0 \end{aligned} \qquad (1)$$

where $A_i, B_i, i = 0 \ldots 7$, are inputs, and P_i, G_i are ***propagate*** and ***generate*** signals, respectively,

$$P_i = A_i \bigoplus B_i, \qquad G_i = A_i \cdot B_i \qquad (2)$$

If the P_i's and G_i's are produced by combinatorial logic function blocks before they are fed into the function blocks for S_i's and C_i's, then Eqn. (1) implies that a two-level AND-OR logic function block is a possible solution to achieve high speed operations. Thus, the PLA-styled design is suitable for such a function block.

A conceptual PLA-styled design for CLA is shown in Fig. 2. A typical PLA consists of an AND array and an OR array. It is well known that the series NMOS in the evaluation block of NAND or AND gates will produce long discharging delays which subsequently slow down the entire circuit. We can take advantage of the non-inverting feature of the ANT logic to utilize a NOT-OR-NOT-OR configuration instead of the typical AND-OR style, where the two OR planes are made of ANT logic blocks. Meanwhile, it can also minimize the series transistor count in the evaluation block. The OR array is made of the ANT logic with a predefined evaluation block. The inputs to the first OR array is the inverted P_i's (propagate) and G_i's (generate) signals which are also produced by other ANT logic units as shown in Fig. 3. Note that we define the propagate signals in a different way from the traditional $P_i = A_i + B_i$ because the $P_i = A_i \bigoplus B_i$ can be reused to generate the sum term, i.e., S_i.

2.4. Speed and area analysis

Speed : The critical path of an adder resides on the generation of carry signals, i.e., C_7 in the 8-bit adder. After the binary data are ready, the generation of P_i's and G_i's by using the ANT logic takes the high half of a full cycle. That is, the results of GP blocks in Fig. 3 will be ready when the *clk* is low. The inverted P_i's and G_i's will then be fed into the first OR plane of the ANT-based PLA. The inverted outputs of the first OR plane will be presented to the second OR at the high half of the second cycle. The final C_i's results then are ready in the low half of the second cycle. Right after the generation of every C_i's, they are inverted and fed into the S_i's function blocks as shown in Fig. 4. Another half cycle then is required to produce all of the S_i's. The final result will be latched after 2 cycles.

Area : As for the transistor count of the PLA-styled implementation for CLA using ANT logic, an analytic form is obtained after careful derivations. In short, if an n-bit CLA is to be realized by our methodology, the transistor count can be computed as follows.

$$T_{total} = \frac{1}{6}(n+1)(n+2)(n+3)+5n(n+1)+50n+3 \quad (3)$$

Note that theoretically a 64-b adder can be done by using the same PLA-styled design methodology, and the delay of 2 ns is expected. However, the total transistor will be over 70,000 which is very large. Another design alternative is to use a hierarchical design employing nine 8-bit CLAs to realize it, as shown in Fig. 5. The trade-off is that the delay will be 4 cycles (ns).

The entire implementation of the 8-b PLA-styled ANT-based CLA is shown in Fig. 6 in which the details from A_i's and B_i's to P_i's and G_i's are ignored for the sake of clearness. In contrast, the detailed schematic diagram of the CLA implemented by TSMC 0.6 μm SPDM is shown in Fig. 7.

3. PERFORMANCE SIMULATIONS AND COMPARISON

In order to verify the performance of the 8-bit CLA using the ANT logic in a PLA-styled design we use the TSMC 0.6 μm SPDM technology to simulate several comparator designs using different logics. The clock rate is 1.0 GHz with 0.01 ps rise time and fall time. The results are tabulated in the following table.

Logic	delay	# transistors
8-b PLA-ANT CLA	2.0 ns	928
32-b PLA-ANT CLA	2.0 ns	13428
64-b PLA-ANT CLA	2.0 ns	71908
64-b PLA-ANT hierarchical CLA	4.0 ns	8352
32-b EMODL adder [9]	2.7 ns	1537 (gates)
8-b TSPC adder (1 μm) [7]	7.5 ns	1832
All-N-logic [3]	Failed	2062

Table 2 : The performance comparison of different designs

An example of the output waveform of 8-b PLA-styled CLA using ANT logic shown in Fig. 8 illustrates that the result of addition of A = 11010101 and B = 10101010 appears after two cycles given the clock = 1.0 GHz.

4. CONCLUSION

We propose a novel high speed PLA-styled ANT logic design for the adders' implementation. Not only the correctness of the function in the giga hertz range is verified, but also the proper size of each transistor is tuned such that a usual square-wave clock can be used to run the 64-b long adder. The PLA-styled ANT-based structure using only one clock makes the result of an 8-b adder appear in 2.0 cycles (ns if the 1.0 GHz clock is used), or a hierarchical 64-b adder in 4.0 cycles.

5. REFERENCES

[1] M. Afghahi, "A robust single phase clocking for low power, high-speed VLSI applications," *IEEE J. of Solid-State Circuits*, vol. 31, no. 2, pp. 247-253, Feb. 1996.

[2] R. X. Gu, and M. I. Elmasry, "All-N-logic high-speed true-single-phase dynamic CMOS logic," *IEEE J. on Solid-State Circuits*, vol. 31, no. 2, pp. 221-229, Feb. 1996.

[3] R. Rogenmoser, and Q. Huang, "An 800-MHz 1mm CMOS pipelined 8-b adder using true phase clocked logic-flip-flops," *IEEE J. on Solid-State Circuits*, vol. 31, no. 3, pp. 401-409, Mar. 1996.

[4] Z. Wang, G. A. Jullien, W. C. Miller, J. Wang, and S. S. Bizzan, "Fast adders using enhanced multiple-output domino logic," *IEEE J. of Solid-State Circuits*, vol. 32, no. 2, pp. 206-214, Feb. 1997.

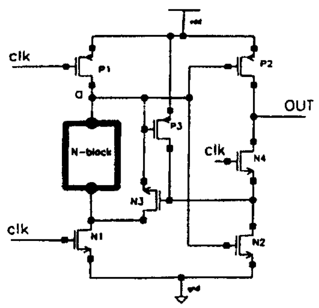

Fig. 1: Schematic diagram of the ANT logic.

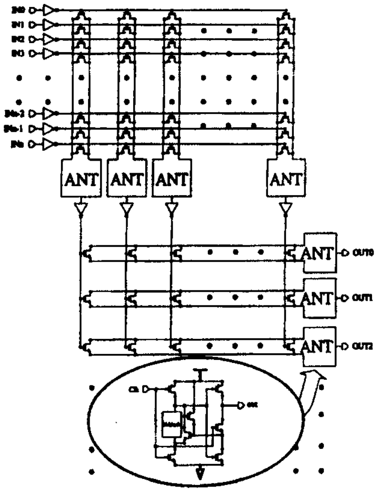

Fig. 2: Conceptual PLA-styled ANT-based CLA.

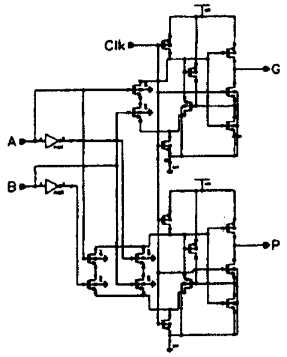

Fig. 3: Schematic diagram of P&G generation.

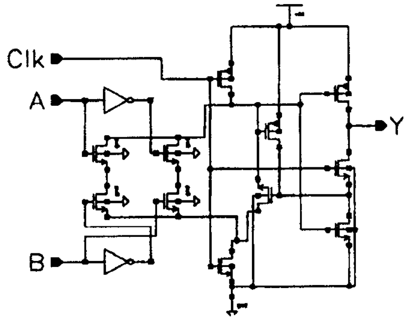

Fig. 4: Schematic diagram of SUM generation.

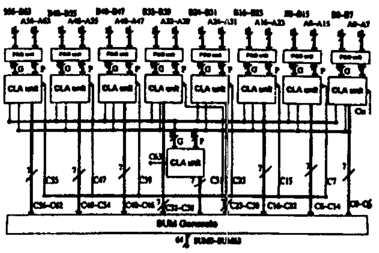

Fig. 5: Hierarchical 64-bit CLA.

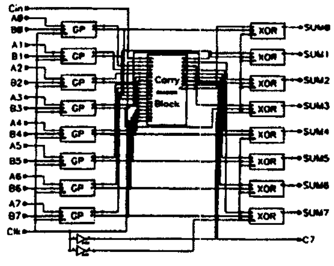

Fig. 6: Schematic diagram of PLA-styled ANT-based

8-b CLA.

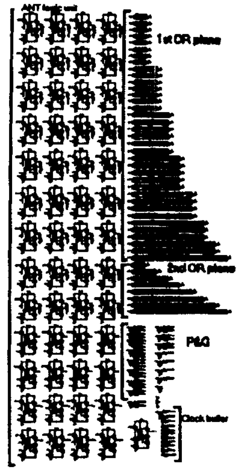

Fig. 7: Detailed circuits of 8-b CLA.

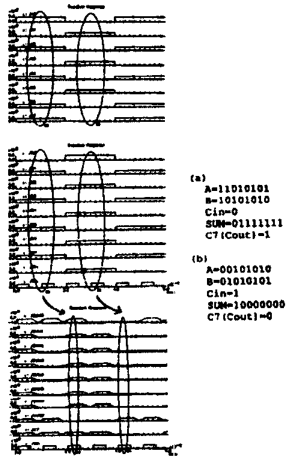

Fig. 8: Waveform diagram of PLA-styled ANT-based

8-b CLA.