

# A 32-Bit Carry Lookahead Adder Design Using Complementary All-N-Transistor Logic

Gang-Neng Sung, *Student Member, IEEE*, Chun-Ying Juan, and Chua-Chin Wang, *Senior Member, IEEE*

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

**Abstract**—A complementary all-N-transistor (CANT) comprising the ANT logic and a novel inverted ANT logic is proposed in this paper. The threshold voltage of the transistors in the ANT logic's N-Block is variable depending upon the operation of the entire logic block. In the evaluation phase, the bulk voltage of the transistors in the N-Block is raised to  $V_{DD} - V_{thn}$  such that the drain current therein is increased to enhance operation speed. In the pre-charge phase, the bulk voltage of those transistors in the N-Block is reduced to its normal voltage level such that the subthreshold leakage current is dropped to reduce power consumption. By utilizing such a variable bulk voltage scheme in the CANT, a 32-bit CLA is designed to justify the low power and high speed performance. The power dissipation is 143 mW at 5.4 GHz clock rate given the worst PVT (SS, 1.08 V, 75°C) condition.

**Keywords**—Complementary all-N-transistor (CANT), tree-structure, carry lookahead adder (CLA), “o” cell, ANT.

## I. INTRODUCTION

The high-speed logic operation is one of the major demands for CPUs, DSPs, 3D display processors, etc. CMOS dynamic logic has been recognized as one of promising options to challenge over 10 GHz operation regarding adder designs. The pipeline structure design is welcomed to increase the operation speed as well as throughput. All-N-logic (ANL) [1] shows a high speed performance by a simple structure driven by a single-phase clock. However, if NMOS transistors in the N-Block of ANL stack up too long (level > 3), the operation speed is substantially dropped. In order to resolve this problem, the all-N-transistor (ANT) logic is then proposed [2]. ANL and ANT structures are proven to be able to realize inverted and non-inverted logics. By integrating these two logics alternatively, the logic complexity of the pipeline structure will be simplified besides the feature of operating at both rising and falling clock edges. Unfortunately, the transition time of the inverted logic is far larger than that of the non-inverted logic. If we combine these two logics directly, the clock rate will be compromised because of the inverted logic. Therefore, only the non-inverted logic is utilized in prior designs with non-overlapping dual clock signals to achieve high-speed pipelining. As a consequence, the clocking complexity is increased and extra glue logic is

needed to convert the non-inverted logic into inverted logic functions.

Besides, as the CMOS power dissipation is known to be  $P_{diss} = f \times C \times V^2$ , where  $f$  is the frequency of transitions,  $C$  is the load,  $V$  denotes the voltage swing, higher operation speed demands higher power dissipation. Therefore, dynamic threshold voltage for the bulk of transistors has been proposed to provide high speed and low power consumption in CMOS circuits. [3] applied the bulk dynamic threshold in the true-single-phase-clocking (TSPC) logic. However, the TSPC logic contains PMOS which is not suitable in high speed circuits. [4] and [5] used the bulk dynamic threshold technique in domino-like dynamic logic. However, the bulk bias is driven by the clock signal to drastically increase the load of the clock. Moreover, the domino-like dynamic logic can not be integrated in TSPC or ANL structures. Therefore, we propose a complementary all-N-transistor (CANT) structure to resolve all of the mentioned difficulties. To justify the performance, a 32-bit tree-structure carry lookahead adder (CLA) design utilizing the proposed inverted and non-inverted logics alternatively to simplify the circuit complexity is implemented.

## II. COMPLEMENTARY ALL-N-TRANSISTOR (CANT) LOGIC

### A. Non-inverted ANT Logic (ANTP)

The structure of a non-inverted ANT logic (ANTP) is shown in Fig. 1. If the N-Block logic in the ANTP is turned on, the output signal **OUTP** will be pulled up high. Otherwise it will be pulled down low. Therefore, the output logic state and the N-Block logic state are the same, namely “non-inverted” logic. The detailed operation is described as follows.

- **Pre-charge phase:** When the input signal  $CLK = 0$ , NM1P and NM4P are turned off. Then the pass/stop result of N-Block can not affect the output signal **OUTP**. Node **BP** keeps in low via NM2P. NM3P is turned off and node **AP** remains high. This makes the output state to stay as the previous state.
- **Evaluation phase:** When the  $CLK = 1$ , PM1P is closed, NM1P and NM4P are turned on.

If the N-Block is evaluated to “stop”, node **AP** will stay high. In the meantime, PM2P will be turned off and

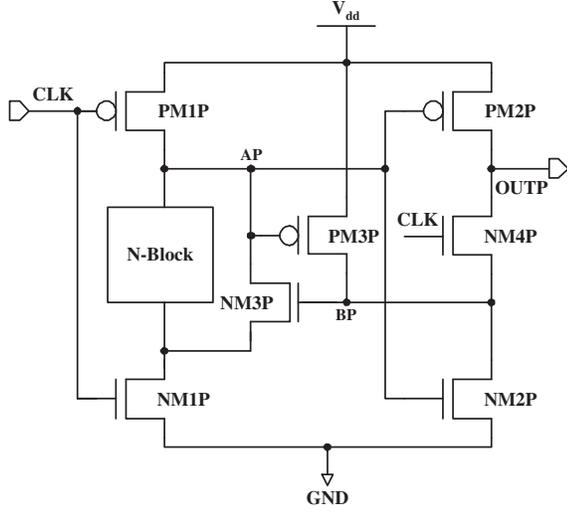


Fig. 1. Non-inverted ANT logic (ANTP).

NM2P will be turned on. The output will be pulled down via NM4P and NM2P.

If the N-Block is evaluated to “pass”, node **AP** will be pulled down via the N-Block and NM1P. When the voltage of node **AP** drops below  $(V_{dd} - V_{th})$ , where the  $V_{dd}$  is the supply voltage and the  $V_{th}$  is the threshold voltage of the MOS, respectively, PM2P and PM3P start to turn on slowly. The output signal **OUTP** and node **BP** will be pulled up high, and NM3P will be turned on because the voltage of node **BP** is larger than  $V_{th}$ . Therefore, not only will the charge at node **AP** be discharged faster via NM3P and NM1P, but also the output signal **OUTP** and node **BP** will be charged to high via PM2P and PM3P.

### B. Inverted ANT Logic (ANTN)

The inverted ANT logic (ANTN) is shown in Fig. 2. When the input clock signal  $CLK = 0$ , the output signal **OUTN** depends on the state of the N-Block. If the N-Block logic in the ANTN is turned on, the signal **OUTN** will be pulled down low. Otherwise, it will be pulled up high. Therefore, the output logic state and the N-Block logic state are opposite, namely “inverted” logic. The detailed operation is described as follows.

- **Pre-charge phase:** When the input signal  $CLK = 1$ , PM1N and PM4N are turned off. Then the pass/stop result of N-Block can not affect the output signal **OUTN**. NM2N is turned off because node **AN** is pulled down via NM1N. Node **BN** keeps in high via PM2N and forces PM3N to turn off. The output signal **OUTN** stays the same as the previous state because PM4N and NM2N are closed.
- **Evaluation phase:** When the  $CLK = 0$ , NM1N is closed, PM1N and PM4N are turned on.

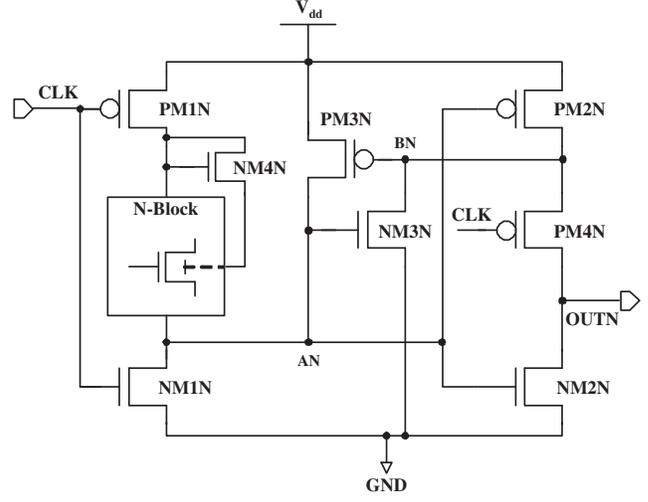


Fig. 2. Inverted ANT logic (ANTN).

If the N-Block is evaluated to “stop”, node **AN** keeps in low. In the meantime, NM2N will be turned off and PM2N will be turned on. The output will be pulled up via PM4N and PM2N.

If the N-Block is evaluated to “pass”, node **AN** will be pulled up via the N-Block and PM1N. When the voltage of node **AN** increases over  $V_{th}$ , NM2N and NM3N start to turn on slowly. Notably, node **AN** can not be fully pulled up to  $V_{dd}$  through the NMOS stack and PM1N and the charging speed is very slow. However, **OUTN** and node **BN** are pulled down via NM2N and PM4N. PM3N will be turned on as soon as node **BN** drops below  $(V_{dd} - V_{th})$ . Then, node **AN** can charge fast via PM3N and fully pull up to  $V_{dd}$ . NM2N and NM3N can turn on very fast because node **AN** is on and consequently speed up discharging **OUTN** and node **BN**.

Though node **AN** can not directly pull up to  $V_{dd}$  via the N-Block, it will be fully pulled up to  $V_{dd}$  via the feedback path composed of NM3N and PM3N. By contrast, even though the node **AN** in ANTN logic can reach the voltage of  $V_{dd}$ , the charging speed is much slower than that of the node **AP** in ANTP logic [6]. This problem demands long computing time for the N-Block in ANTN logic to turn on. When the ANTP logic and ANTN logic are integrated in a pipeline structure, the speed of ANTN logic is much slower than that of ANTP logic such that the overall operating speed is reduced. In order to resolve this problem, we propose to utilize the bulk dynamic voltage technique to the N-Block in the ANTN logic.

A simple thought to enhance the speed of the ANTN logic is to change the bulk voltage of the transistors in its N-Block when the evaluation is carried out.

**Evaluation Phase by dynamic bulk voltage:** When the ANTN logic enters this phase, the N-Block has to decide “pass” or “stop”. In either case, a large current is required to speed up the whole evaluation. According to the well known

Eqn. (1).

$$V_{th} = V_{th0} + \gamma(\sqrt{|-2\Phi_F + V_{SB}|} - \sqrt{|-2\Phi_F|}) \quad (1)$$

where,  $V_{th0}$  is the threshold voltage when  $V_{SB} = 0$ ,  $\gamma$  is the body-effect coefficient,  $2\Phi_F$  is the silicon surface potential at the onset of strong inversion which is equal to -0.6 V for typical p-type substrates,  $V_{SB}$  is the source to body voltage. When the MOS has a positive  $V_{SB}$ ,  $V_{th}$  will be decreased and the D-S current of the MOS transistor will be increased. Therefore, we use NM4N to increase the  $V_{SB}$  of NMOS when the N-Block is expected to evaluate such that the node AN of ANTN can be pulled up faster. When the ANTN logic operate in the evaluation phase ( $CLK = 0$ ), NM4N is turned on and the bulk of the NMOSs will be raised up to a  $V_{dd} - V_{th}$ . The current in such a scenario will be larger than that of the N-Block of ANTP blocks in the same phase.

**Pre-charge Phase by dynamic bulk voltage:** In this phase, the speed is no longer a primary consideration. By contrast, the power is the major factor. By considering the subthreshold current equation of MOS transistor:

$$I_{sub} = I_o \frac{W}{L} e^{\frac{V_{GS} - V_{th}}{n v_t}} \quad (2)$$

where  $V_{th}$  is the MOS threshold voltage,  $v_t$  is the thermal voltage ( $= KT/q$ ) which is equal to 26 mV at 300°K,  $n$  is the subthreshold slope parameter. A lower  $V_{th}$  will dissipate less subthreshold leakage current which is why we propose to restore the bulk voltage to its normal level in this phase.

### C. Tree-Structured CLA

A 32-bit tree-structured CLA using the proposed CANT is illustrated in Fig. 3. We use the ANTP to realize the **p, g generator**. The propagation and generation signals can be generated at the rising edge of the clock signal. The **5-Stage “o” cell Array** is composed of ANTN and ANTP alternately in a form of “N-P-N-P-N”. Every single stage can be calculated at the corresponding edges of the clock. In other words, the result of **5-Stage “o” cell Array** will generate outputs after 2.5 clock cycles. The **SUM Stage** consists of ANTP blocks. Therefore, the output of the proposed CLA design is available within a total of 3.5 clock cycles.

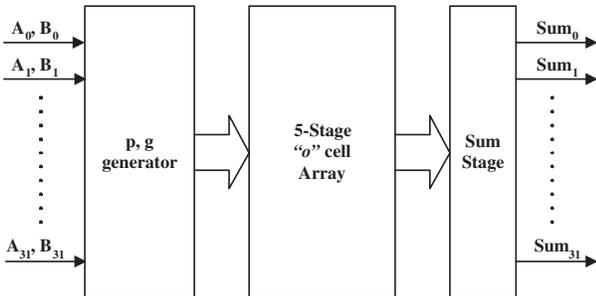


Fig. 3. Block diagram of the proposed design

The “o” cell is the basic component of the tree-structure adder and can produce the carry out signal parallelly [7]. Traditional carry lookahead logic will be very complicated if long data word addition is required. The tree-structure CLA brings the advantages of parallel computing and low circuit complexity [8]. In this work, we use CANT to implement the adding cell and produce the carry out signals in a pipeline structure. ANTP (non-inverted logic) and ANTN (inverted logic), works at rising and falling clock edges, respectively, to double the throughput. These two logics are cascaded alternatively to achieve high speed parallel computing.

Fig. 4 shows the structure of the **5-Stage “o” cell Array**. The hollow circular and black circular in Fig. 4 denote an “o” cell and a delay cell, respectively. The function of the “o” cell is shown in Fig. 5, where the “o” operator produces the propagation and generation signals.

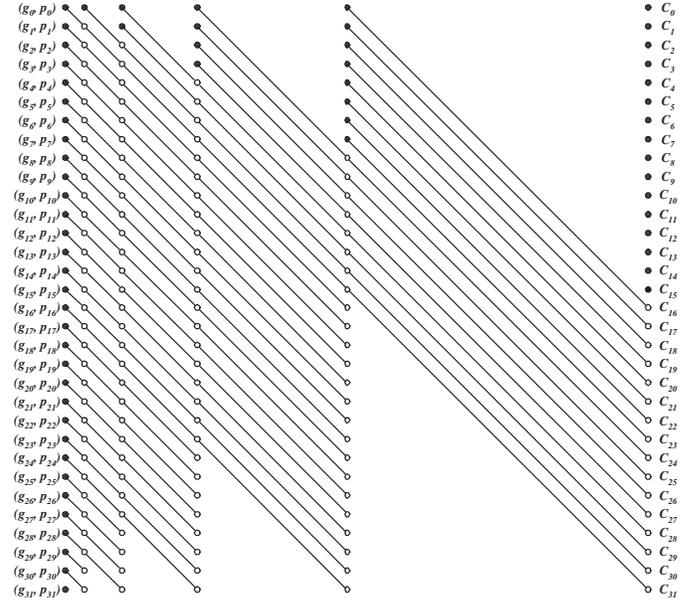


Fig. 4. Structure of the 5-Stage “o” cell Array

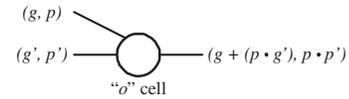


Fig. 5. The “o” cell operator

In CLA designs, the propagation signals and generation signals are needed to calculate sums and carry signals. Propagation signals are resulted from  $p_i = A_i \oplus B_i$ , while generation signals are  $g_i = A_i \cdot B_i$ . Thus, sum and carry signals are  $s_i = p_i \oplus c_{i-1}$  and  $c_i = g_i + p_i \cdot c_{i-1}$ , respectively, where  $A_i$  and  $B_i$  are input signals,  $\forall i, i = 1, \dots, 31$ . Assume that  $(G_1, P_1) = (g_1, p_1)$ ,  $i = 1$ ;  $(G_i, P_i) = (g_i, p_i) o (G_{i-1}, P_{i-1})$ ,  $2 \leq i \leq 31$ ;  $c_0 = 0$ . When  $i = 1$ ,  $c_1 = g_1 + (g_1 \cdot c_0) = g_1 = G_1$ . When  $i > 1$  and  $c_{i-1} = G_{i-1}$ , we can induce the following general form:

$$\begin{aligned}
(G_i, P_i) &= (g_i, p_i) o(G_{i-1}, P_{i-1}) \\
&= (g_i, p_i) o(c_{i-1}, P_{i-1}) \\
&= (g_i + (p_i \cdot c_{i-1}), p_i \cdot P_{i-1}) \quad (3)
\end{aligned}$$

Thus,  $G_i = g_i + (p_i \cdot c_{i-1})$ . According to Eqn. (3), we can get  $G_i = c_i$ . Therefore, the carry out signals can be computed by the “o” cell array.

### III. SIMULATION AND IMPLEMENTATION

TSMC (Taiwan Semiconductor Manufacturing Company) 0.13  $\mu\text{m}$  1P8M CMOS process is adopted to carry out the proposed 32-bit tree-structure CLA design using the CANT logic. Fig. 6 shows the simulation results of the proposed design which executes  $32'b(11\dots11) + 32'b(00\dots01)$  and  $32'b(00\dots00) + 32'b(11\dots11)$ . Fig. 7 shows the layout of the proposed design and the core area is  $0.08481 \text{ mm}^2$ . The power dissipation of the proposed design is less than 143 mW at 5.4 GHz clock rate. The comparison with prior designs is tabulated in Table I. The power delay product (PDP) of our CLA is reduced more than 23% relative to the prior work in [10] given the same clock rate. The saving in power alone is more than 18%.

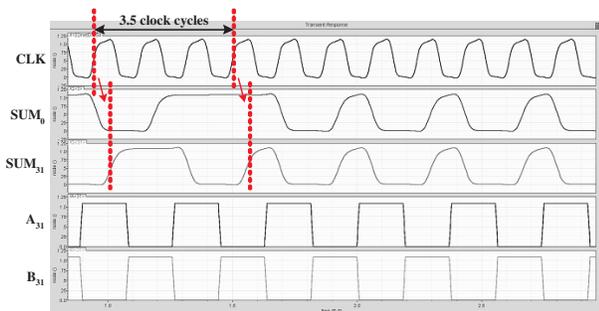


Fig. 6. Simulation waveform of the proposed design

	Ours	ANT [9]	DPANL [10]	DPANL [10]
Process ( $\mu\text{m}$ )	0.13	0.35	0.35	0.13
Clock Rate (GHz)	5.4	1.25	1.85	5.4
Power (mW)	143	N/A	1000	175
Adder Area ( $\text{mm}^2$ )	0.085	1.86	0.7	0.096
PDP ( $\text{J}^{-10}$ )	0.265	N/A	5.405	0.342

TABLE I  
COMPARISON WITH PRIOR DESIGNS

### IV. CONCLUSION

In this paper, we have proposed a high speed and low power CANT logic which is used for the implementation of CLAs. The technique of bulk dynamic threshold voltage is employed to increase the computing speed and reduce the power consumption. The CLA generates correct output after 3.5 cycles of delay given a 5.4 GHz clock. The post-layout simulation shows that 18% power is saved compared to the prior designs.

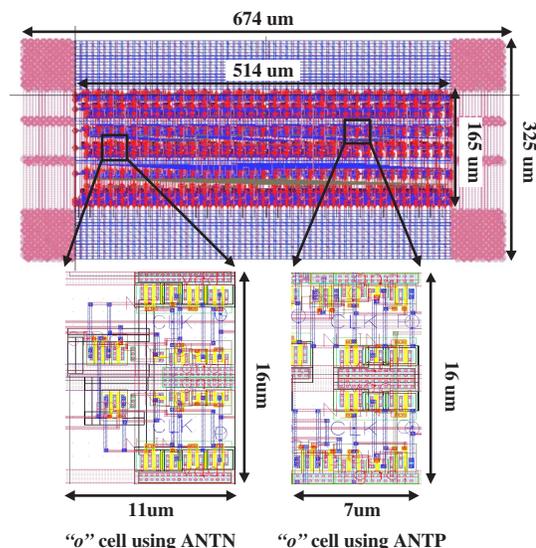


Fig. 7. Layout of the proposed design

### ACKNOWLEDGMENT

The authors would like to express their deepest gratefulness to CIC (Chip Implementation Center) of NAPL (National Applied Research Laboratories), Taiwan, for their thoughtful chip fabrication service. The authors also like to thank “Aim for Top University Plan” project of NSYSU and MOE, Taiwan, for partially supporting this investigation. This research was partially supported by National Science Council under grant NSC96-2628-E-110-018-MY3.

### REFERENCES

- [1] R. X. Gu and M. I. Elmasry, “All-N-logic high-speed true-single-phase dynamic CMOS logic,” *IEEE J. Solid-State Circuit*, vol. 31, no. 2, pp. 221-229, Feb. 1996.
- [2] Chua-Chin Wang, and Kun-Chu Tsai, “VLSI design of a 1.0 GHz 0.6- $\mu\text{m}$  8-bit CLA using PLA-styled all-N-transistor logic,” *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 236-239, May 1998.
- [3] K. Wu, S. Jia, Z. Chen and X. Gan, “Implementation of low-voltage true-single-phase-clocking (TSPC) logic using bulk dynamic threshold MOS technique,” *6<sup>th</sup> International Conference on ASIC 2005*, vol. 1, pp. 158-162, Oct. 2005.
- [4] W. Elgharbawy and M. Bayoumi, “New bulk dynamic threshold NMOS schemes for low-energy subthreshold domino-like circuits,” *IEEE Computer Society Annual Symposium on VLSI*, pp. 115-120, Feb. 2004.
- [5] W. Elgharbawy and M. Bayoumi, “B-DTNMOS: a novel bulk dynamic threshold NMOS scheme,” *IEEE International Symposium on Circuits and Systems*, vol 2, pp. 413-415, May 2004.
- [6] M. R. Casu, G. Masera, G. Piccinini, M. R. Ruo and M. Zamboni, “Comparative analysis of PD-SOI active body-biasing circuits,” *2000 IEEE International SOI Conference*, pp. 94-95, Oct. 2000.
- [7] R. P. Brent and H. T. Kung, “A regular layout for parallel adders,” *IEEE Trans. on Computers*, vol. C-31, no. 3, pp. 260-264, March 1982.
- [8] S. Knowle, “A family of adders,” *15<sup>th</sup> IEEE Symposium on Computer Arithmetic*, pp. 277-281, 2001.
- [9] C.-C. Wang, Y.-L. Tseng, P.-M. Lee, R.-C. Lee, and C.-J. Huang, “A 1.25 GHz 32-bit tree-structured carry lookahead adder using modified ANT logic,” *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 50, no. 9, pp. 1208-1216, Sep. 2003.
- [10] G. Yang, S.-O. Jung, K.-H. Baek, S. H. Kim and S.-M. Kang, “A 32-bit carry lookahead adder using dual-path all-N logic,” *IEEE Trans. on VLSI Systems*, vol. 13, no. 8, pp. 992-996, Aug. 2005.