VI. Conclusion

This brief introduced modifications and advancements to a previously proposed four-transistor SRAM cell array. A new write method where the advantages far outweigh the disadvantages has been used in conjunction with an alternative array structure to obtain improved immunity to process conditions. A design method for the deviations that need to be applied to the source nodes in order to effect control, based on noise margins analysis, has been introduced. The newly proposed write method together with its array structure is compared to the previous proposal shown in Table I. All specifications given were simulated using the typical mean model.

References

[1] T.-H. Joubert, E. Seevinck, and M. du Plessis, "A CMOS reduced-area SRAM cell," in *Proc. IEEE Int. Symp. Circuits and Systems ISCAS'00*, Geneva, Switzerland, May 28–31, 2000, pp. III-335–III-338.

[2] Austria Mikro Systems, "0.6 mm CMOS CUP process parameters," Austria Mikro Systems International AG, Surrey, U.K, Doc. 9933011, Rev. B, Oct. 1998.

[3] K. Anami *et al.*, "Design considerations of a static memory cell," *IEEE J. Solid-State Circuits*, vol. 18, pp. 414–417, Aug. 1983.

[4] E. Seevinck, F. List, and J. Lohstroh, "Static-noise margin analysis of MOS SRAM cells," *IEEE J. Solid-State Circuits*, vol. 22, pp. 748–754, Oct. 1987.

[5] J. Lohstroh, E. Seevinck, and J. De Groot, "Worst-case static noise margin criteria for logic circuits and their mathematical equivalence," *IEEE J. Solid-State Circuits*, vol. 18, pp. 803–807, Dec. 1983.

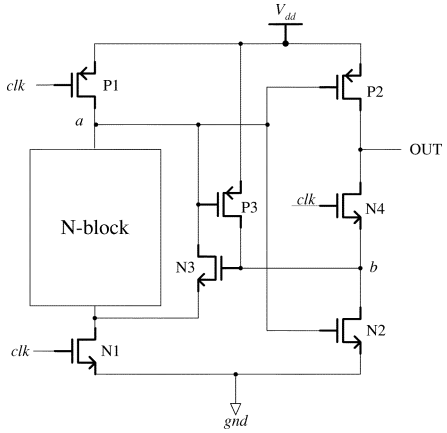

Fig. 1. Schematic diagram of the prior ANT logic.

TABLE I
SIZES OF ANT LOGIC BLOCK SHOWN IN FIG. 1

Transistor	Length	Width	Transistor	Length	Width
Transistor	Length	Width	Transistor	Length	Width
N1	0.35 μm	10 μm	N2	0.35 μm	10 μm
N3	0.35 μm	3 μm	N4	0.35 μm	10 μm
P1	0.35 μm	15 μm	P2	0.35 μm	15 μm
P3	0.35 μm	6 μm	N-block	0.35 μm	5 μm

A 1.25 GHz 32-Bit Tree-Structured Carry Lookahead Adder Using Modified ANT Logic

Chua-Chin Wang, Yih-Long Tseng, Po-Ming Lee, Rong-Chin Lee, and Chenn-Jung Huang

***Abstract*—In this brief, a 32-bit tree-structured carry lookahead adder (CLA) is proposed by using the modified all-N-transistor (ANT) design. The 32-bit CLA not only possesses few transistor count, but also occupies small area size. Moreover, the post-layout simulation results given by TimeMill show that the clock used in this 32-bit CLA can run up to 1.25 GHz at 3.3-V power supply. The output of the proposed CLA will be ready after 3.5 cycles. The proposed circuit is also easy to be expanded for long data additions. A physical chip is fabricated to verify the proposed circuit on silicon.**

***Index Terms*—Tree-structured carry lookahead adder (CLA), "*o*" cell, all-N-transistor (ANT).**

I. Introduction

The high-speed operation has long been a target of circuit designers owing to the speed demand of supercomputing, CPUs, etc. Hence, improving adder designs have received considerable attention [1]–[6]. CMOS dynamic logic is one of the promising options to challenge GHz operation regarding adder designs [5]. However, domino logic [1] cannot be noninverting; an adder in [4] can only process short data words; all-N-logic [5] and robust single phase clocking [6] cannot operate correctly under clocks with short rise time or fall time; single-phase logic [2] and Zipper CMOS [3] contain slow P-logic blocks. An all-N-transistor (ANT) noninverting function block was proposed to resolve the mentioned difficulties [7], which can be used to design a high-speed carry lookahead adder (CLA). However, a drawback of the programmable logic array (PLA)-styled CLA in [7] is that it can hardly run a pipelining operation owing to the demand of generating a back propagation signal. In this brief, we enhance the ANT logic by proposing an "*o*" cell and a tree-structured scheme for CLAs. The simulation results of the proposed design are also given to prove its high-speed performance. A physical chip is implemented by the Taiwan Semiconductor Manufacturing Company (TSMC) 0.35-μm 1P4M CMOS technology to prove the function of the proposed CLA design. The output is available with a total of 3.5 cycles of delay after the input data are fed.

Manuscript received March 20, 2001; revised November 20, 2001 and December 12, 2002. This work was supported in part by the National Science Council under Grant NSC 89-2218-E-110-014 and Grant 89-2218-E-110-015. This paper was recommended by Associate Editor Y. Ismail.

C.-C. Wang, Y.-L. Tseng , P.-M. Lee, and R.-C. Lee are with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan, R.O.C. (e-mail: ccwang@ee.nsysu.edu.tw).

C.-J. Huang is with the Department of Computer Science and Information Education, National Taitung Teachers College, Taitung 95004, Taiwan, R.O.C. (e-mail: cjh@cc.ntttc.edu.tw).

Digital Object Identifier 10.1109/TCSI.2003.816339

II. 32-Bit Tree-Structured CLA

A. Prior ANT Function Unit

An ANT logic is presented in Fig. 1. The main feature of this design is the presence of the feedback transistor pair, P3 and N3, between the evaluation block and the output. P3 and N3, respectively, provide an extra charging path and discharging path, thus accelerating the evaluation. Notably, the ANT in Fig. 1 is noninverting. The detailed operations of the ANT are described as follows [7].

1) When $\mathrm{clk} = 0$, P1 is on and the gate of P2 is precharged to be $\mathrm{V_{dd}}$. Then, P2 is off and N4 is off. This makes the output to stay at the previous state.

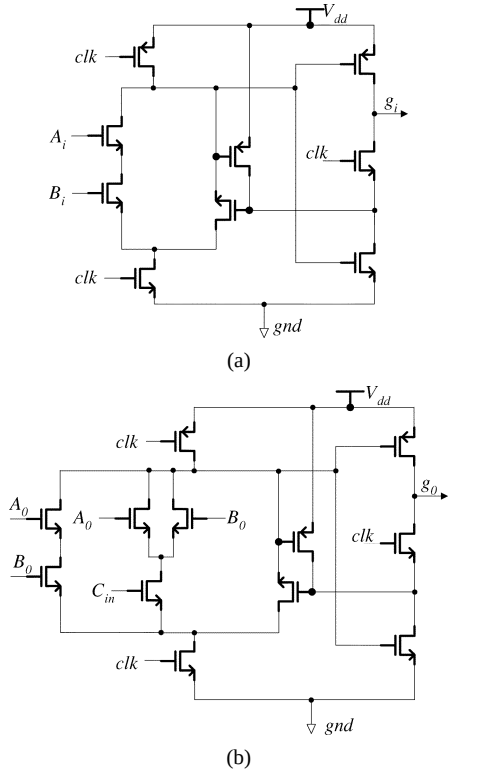

Fig. 2. Schematic diagram of $\boldsymbol{g}_0$ and $\boldsymbol{g}_i$.

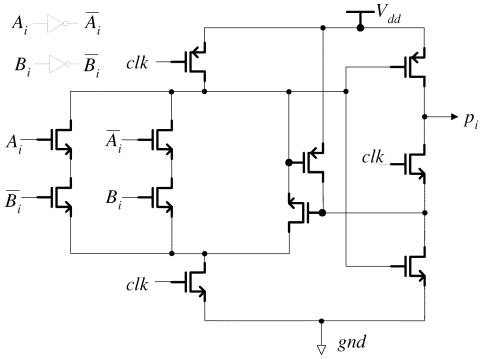

Fig. 3. Schematic diagram of $\boldsymbol{p}_i, \forall \boldsymbol{i}, \boldsymbol{i} = \mathbf{0}, \ldots, \mathbf{31}$.

2) When $\mathrm{clk} = 1$ and the N block is evaluated to be "pass," the charge at node a should be grounded 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 $\mathrm{V_{dd}}$ via paths P2 and P3–N4.
3) When $\mathrm{clk} = 1$ 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 $V_a - V_{\mathrm{dd}} > V_{\mathrm{tp}}$, P3 will be on such that the gate of N3 will be charged to be V_{dd}. Not only will the charge at node a be discharged faster, but also the output will be charged to high via P2 and N4.

Summarized from 2) and 3), the output will be high when the N block is evaluated "pass," i.e., "1," during $\mathrm{clk} = 1$.

4) When $\mathrm{clk} = 1$ 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.

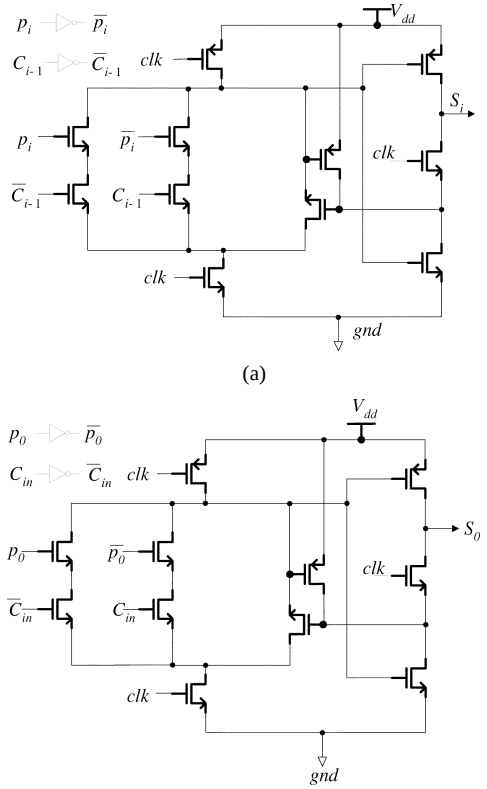

Fig. 4. Schematic diagram of $\boldsymbol{S}_0$ and $\boldsymbol{S}_i$.

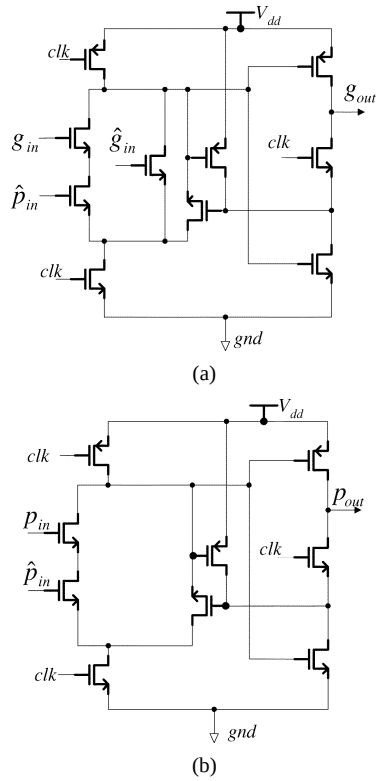

Fig. 5. Schematic diagram of the "$\boldsymbol{o}$" cell.

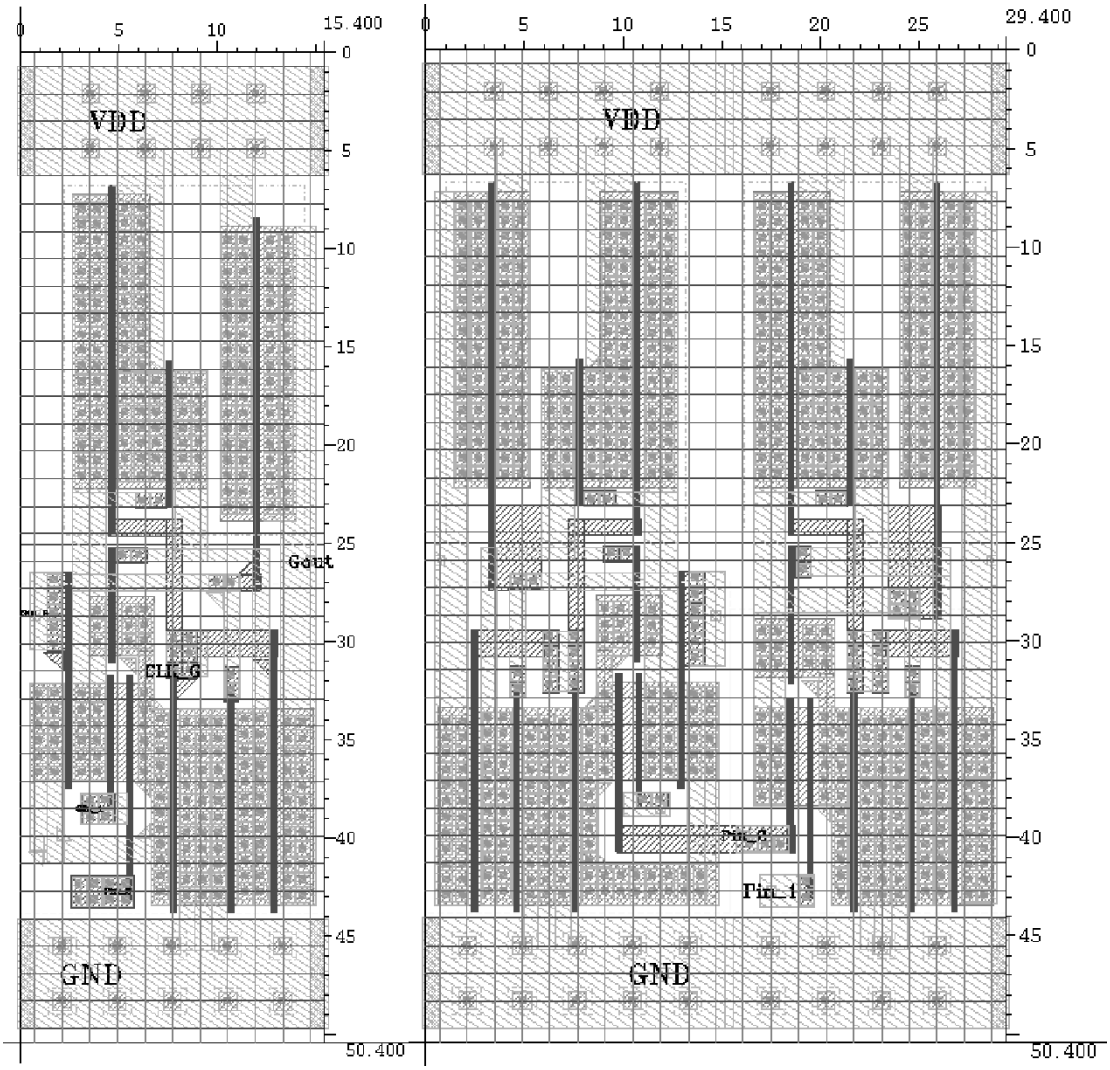

Fig. 6. Layout of the "**o**" cell.

Hence, the output will be low when the N block is evaluated "stop," i.e., "0," during $\mathrm{clk} = 1$. The function of ANT logic block, thus, is conclusively correct. The feedback transistor pair, N3 and P3, indeed provides extra paths for the operations.

Note that though the descriptions of 2) and 4), when the previous state of the output is high look contradictory, it can be resolved by introducing the ratioed design to the size of the N block transistors and the N3. When the clock switches from low to high and the previous state of the output is high, the voltage at node b will be charged through N4, but discharged through N2 initially. If the voltage at node a drops fast enough, N2 will be turned off very quickly. Then, the voltage at node b is solely charged by output node. Then, N3 is turned on by the voltage at node b, which in turn becomes another discharging path for node a. In conclusion, the key factor for the proposed ANT logic to be functionally correct is that the initial discharging speed of node a must be "fast." This indicates the resistance of the paths existing in N block must be smaller than that of N3 and N2. In other words, we can use only wide transistors in the N block. After running thorough simulations, we present the appropriate size ratio for each of the transistors in next section when the 0.35-μm 1P4M CMOS technology is employed.

B. Sizing Problem for 0.35-μm 1P4M CMOS Technology

A reason why other high-speed logics 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 [4] possess this inherent shortcoming, accounting for why they can not use normal square-wave clocks in the gigahertz range. Thus, the sizing problem of the transistors in the ANT, let alone those in the N block, significantly affects the speed. Theoretically, wider transistors have lower resistance. Herein, we conducted several simulations to obtain the optimal figure for the sizing of each transistor given in Fig. 1 using TSMC 0.35-μm 1P4M CMOS technology by using the optimization facility of HSPICE. Those results are summarized in Table I.

C. Cells for Sum, Generate and Propagate Functions

A CLA needs *propagate* signals as well as *generate* signals to calculate sums and the carry signals. The circuits to produce the *propagate* signals and the *generate* signals are described detailedly in the following subsections.

1) Generate Signals: The *generate* signal at stage i is formulated as $g_i = A_i \cdot B_i$, where A_i and B_i are input signals, $\forall i, i = 1, \ldots, 31$. Fig. 2(a) shows the schematic of the generation of $g_i, i = 1, \ldots, 31$. Notably, the g_0 used in our CLA is particularly formulated as $g_0 = A_0 \cdot B_0 + C_{\mathrm{in}}(A_0 + B_0)$. The schematic view of such a g_0 function is given in Fig. 2(b). The reason why the generation of g_0 is different from the rest of $g_i, \forall i, i = 1, \ldots, 31$, is that the carry can be produced by C_{in} and either one of A_0 and B_0.

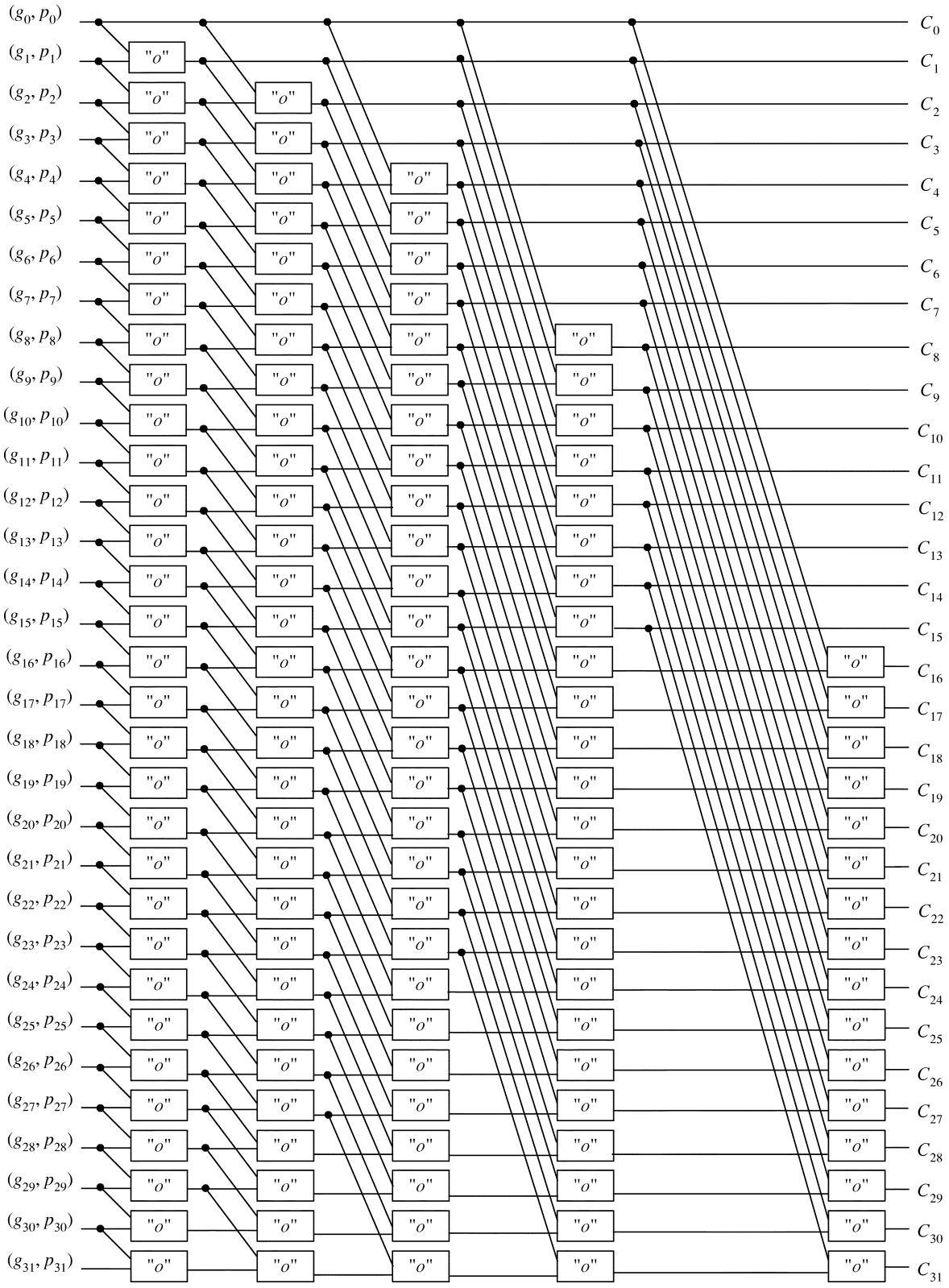

Fig. 7. 32-bit tree-structured CLA.

2) Propagate Signals: *Propagate* signals are resulted from the definition of $p_i = A_i \oplus B_i$. Notably, the *propagate* signals are defined differently from the traditional $p_i = A_i + B_i$, because $p_i = A_i \oplus B_i$ can be reused to compute the sum terms, $S_i, \forall i, i = 0 \ldots, 31$. The *propagate* signal-generation schematic is shown Fig. 3.

3) Sum Signals: The sum term is defined as $S_i = C_{i-1} \oplus p_i$, where C_{i-1} means $(i-1)$th carry signal, $\forall i, i = 1, \ldots, 31$. The schematic diagram of the sum signal S_i is shown in Fig. 4(a). Meanwhile, it should be noted that $S_0 = C_{\text{in}} \oplus P_0$ of which the schematic is shown in Fig. 4(b).

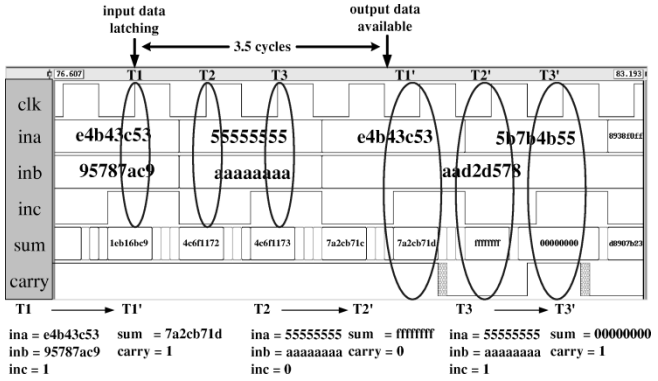

Fig. 8. Simulation result of the 32-bit CLA.

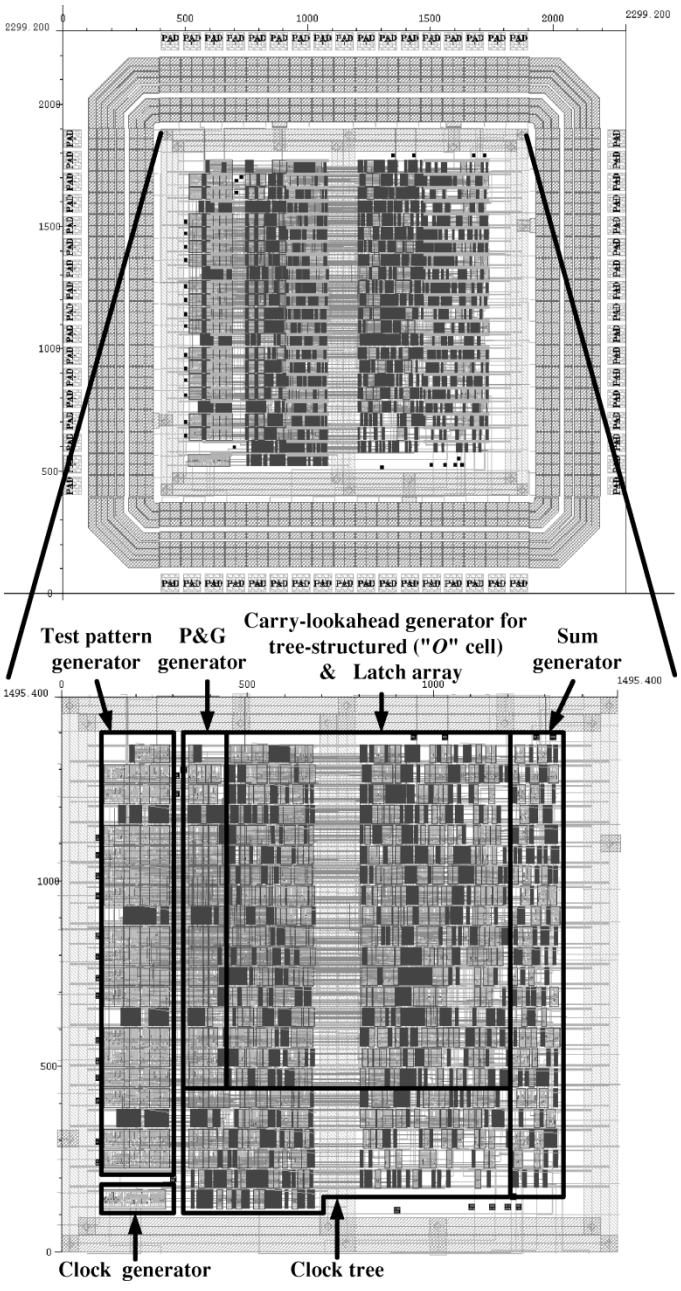

Fig. 9. Layout of the 32-bit CLA chip.

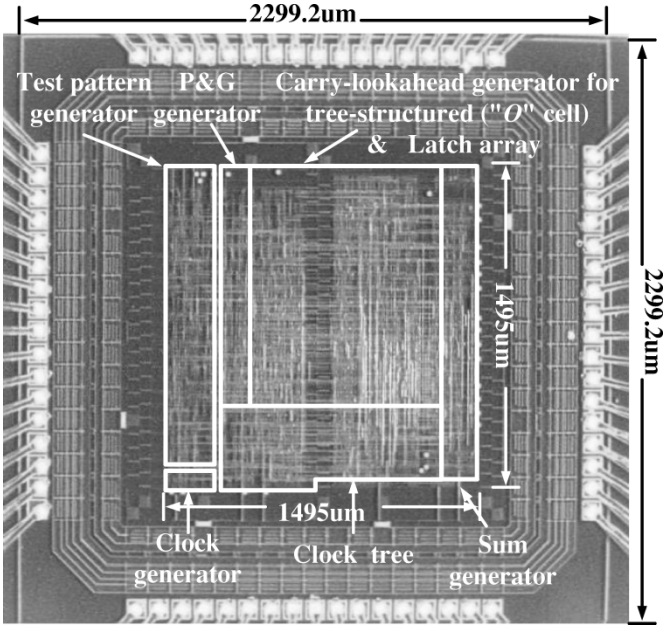

Fig. 10. Die photo of the 32-bit CLA chip.

D. Tree-Structured CLA

The 64-bit PLA-styled hierarchical CLA presented in [7] can not be pipelined because the carry-in signal for the first-level 8-bit CLA unit has to be fed by the second-level 8-bit CLA. This limitation diminishes the effectiveness of this circuit when large streams of data are operated such that higher data rates are required. Therefore, we propose an alternative architecture of CLAs based on a binary-tree structure in this subsection, which eliminates the generation of back propagation signals demanded in the PLA-styled hierarchical architecture.

1) Carry-Lookahead Generator Cell ("o" Cell): Brent and Kung [9] defined a so-called operator "o" as follows:

$$(g_{\text{out}}, p_{\text{out}}) = (\hat{g}_{\text{in}}, \hat{p}_{\text{in}}) o (g_{\text{in}}, p_{\text{in}}) = (\hat{g}_{\text{in}} + g_{\text{in}} \cdot \hat{p}_{\text{in}}, p_{\text{in}} \cdot \hat{p}_{\text{in}}) \quad (1)$$

Based on (1), the carry C_i can be expressed as

$$\begin{aligned} (G_0, P_0) &= (g_0, p_0) \\ (G_i, P_i) &= (g_i, p_i) o (G_{i-1}, P_{i-1}) o \ldots o (g_0, p_0) \\ C_i &= G_i \end{aligned} \quad (2)$$

where p_i's and g_i's are the *propagate* and *generate* signals as defined in the previous section, respectively. Note that $g_0 = A_0 B_0 + A_0 C_{\text{in}} + B_0 C_{\text{in}}$ is allowed to be used in both addition and subtraction. Fig. 5 shows the implementation of the "o" operator using ANT logic.

Due to the associativity property of the "o" operator, (G_i, P_i) can be evaluated in any order from the given p_i's and g_i's. Accordingly, Brent and Kung [9] proposed a tree structure to compute (G_i, P_i). Dozza *et al.* [8] reduced the critical delay of C_i's generations from $2(\log_2 n - 1)$ to $\log_2 n$ levels of the "o" cells by introducing more "o" cells in the binary tree. The layout of the "o" cell is illustrated in Fig. 6. An example of a 32-bit tree-structured CLA is illustrated in Fig. 7. Since this architecture is very regular and modular, this approach can be expanded to design an alternative 64-bit ANT CLA to fix the problem of the PLA-styled CLA [7]. Another advantage of such an arrangement of the "o" cells is that the 32-bit adder can be degenerated to be a small adder, e.g., 8-bit

TABLE II
COMPARISON OF DIFFERENT DESIGNS(n: NO. OF BITS, N/A: NOT AVAILABLE)

Adder Logic	Measured Delay	Cycle-based Delay Prediction	Transistors No.	Max. Freq.
8-bit PLA-styled CLA [7] 0.6 μm	2.0 ns	$\frac{2}{3}\log_2 8 = 2$ cycles	928	1 GHz
64-bit PLA-styled CLA [7] 0.6 μm	2.0 ns	2 cycles	71908	1 GHz
64-bit PLA-styled hierarchical CLA [7] 0.6 μm	4.0 ns	$2 \times \frac{2}{3}\log_2 8$ $= 4$ cycles	8352	1 GHz
32-bit adder [1] 1.2 μm	2.7 ns	N/A	1537(gates)	1 GHz
8-bit adder [4] 1 μm	7.5 ns	$11\frac{3}{4}\tau$ τ: unit transistor delay	1832	1 GHz
All-N-logic [5] 0.8 μm	Failed	$f_{max} = \frac{1}{2\max(t_r, t_f)}$ f_{max}: max. freq.	2062	1 GHz (Author Claimed Performance)
8-bit "o"-based CLA 0.35 μm	2.0 ns	$\frac{\log_2 n}{2} + 1$ cycles	577	1.25 GHz
32-bit "o"-based CLA 0.35 μm	2.8 ns	$\frac{\log_2 n}{2} + 1$ cycles	3425	1.25 GHz
64-bit "o"-based CLA 0.35 μm	3.2 ns	$\frac{\log_2 n}{2} + 1$ cycles	8033	1.25 GHz

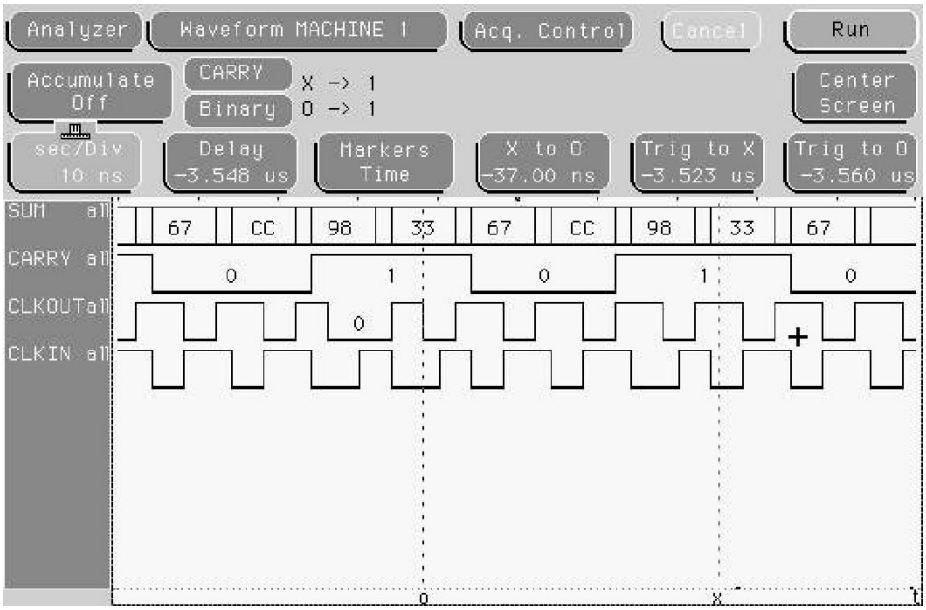

Fig. 11. Measurement of the chip given external testing vectors.

adder or 16-bit adder simply by grounding high bits of the input data. For instance, if the first 24 MSB bits are grounded, the output of the lowest byte addition will be available at the middle layer of the "o" cells in Fig. 7 with a delay of $((\log_2 8)/2) + 1 = 2.5$ cycles. The delay (or speed) analysis is given in the next section.

2) Speed Analysis: The critical path of an adder resides in the generation of carry signals in the tree-structured adder. When the binary data are ready, the generation of p_i's and g_i's using ANT logic requires the high half of a full cycle. The p_i's and g_i's are then fed into the array of "o" cells. The final C_i results are then ready after $((\log_2 n)/2)$ cycles. As soon as it is generated, each C_i is inverted and fed into the S_i's function blocks as shown in Fig. 4(a). Another half cycle is then required to produce all of the S_i's. The final result is available after $((\log_2 n)/2) + 1$ cycles.

3) Area Analysis: Area analysis includes the following aspects.

- Carry Out: The number of total "o" cells used in the generation of C_i's is summarized as

$$T_o = n \cdot \log_2 n - (1 + 2 + \cdots + 2^{\log_2 n - 1}) = n \cdot \log_2 n - n + 1. \quad (3)$$

The number of the transistors in the "o" cells becomes $19 \cdot (n \cdot \log_2 n - n + 1)$ since there are 19 transistors used in an "o" cell. However, all the *propagate* signals, P_i's, which reside in the "o" cells generating C_i's, are not involved in the computations. Consequently, they can be removed from the corresponding "o" cells for area saving. Hence, the total transistor count for the array

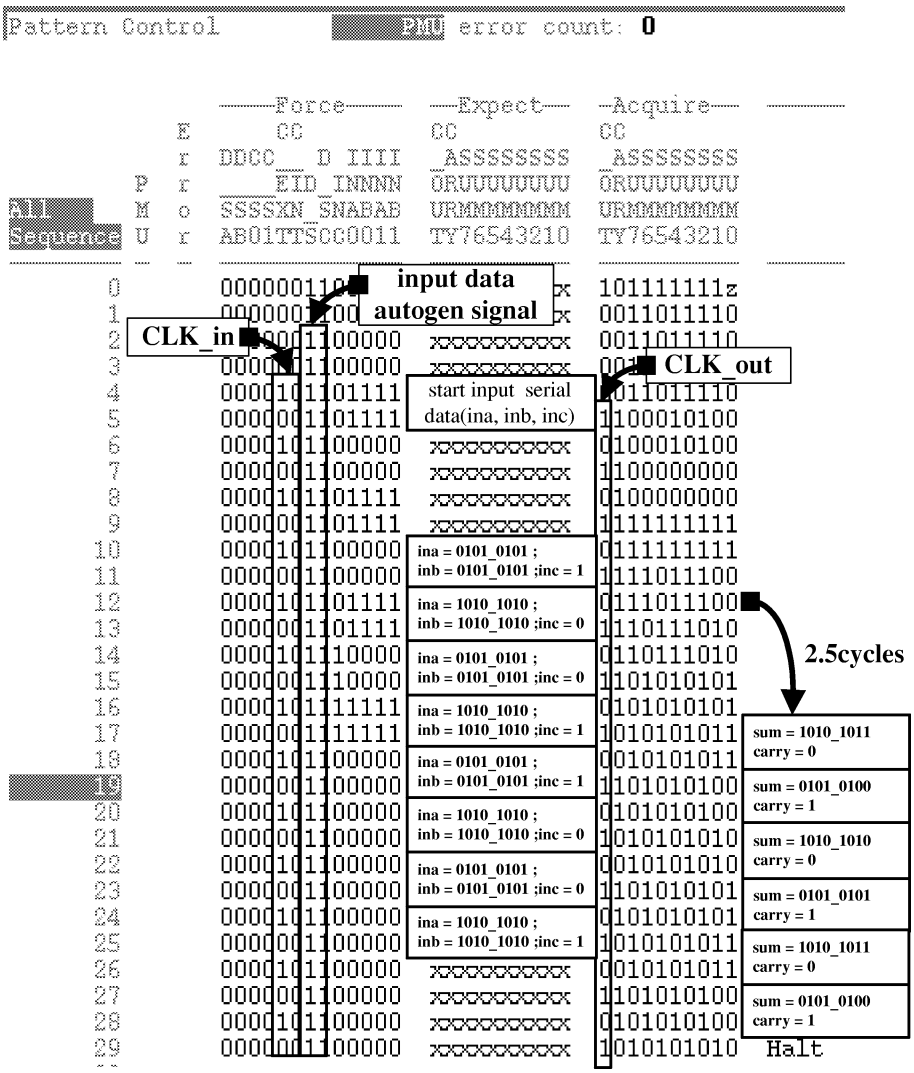

Fig. 12. Measurement of the chip given internal testing vectors (1).

of "o" cells is

$$T_{\text{Tree}} = 10 \cdot (n \cdot \log_2 n - n + 1) + 9 \cdot (n \cdot \log_2 n - 2n + 2) \quad (4)$$

- Inverters: Signals which must be inverted include the input data words, clock, C_i's, and p_i's. Therefore, the total number of transistors for the inverters is $T_I = 8n + 2$.
- p_i's and g_i's generation: The total number of transistors is $20n + 3$.
- S_i's generation: The total transistor count is $11n$.

In sum, the total number of transistors required to implement an n-bit CLA with tree-structured design using ANT logic is

$$\begin{aligned} T_{\text{Total}} &= 10 \cdot (n \cdot \log_2 n - n + 1) \\ &\quad + 9 \cdot (n \cdot \log_2 n - 2n + 2) \\ &\quad + 8n + 5 + 20n + 11n \\ &= 19n \cdot \log_2 n + 11n + 33. \end{aligned} \quad (5)$$

According to the above analysis, we can derive the number of transistors used in a 64-bit tree-structured CLA is 8034, which is less than that of 64-bit PLA-styled hierarchical CLA [7], 8352. The delay is also 4 cycles (3.2 ns if a 1.25 GHz clock is used). Moreover, the n-bit tree-structured CLA can be entirely pipelined by exploiting the 'half-cycle stealing' property of the ANT logic. Notably, no latches are required in the pipeline operations when two out-of-phase clocks are employed at the adjacent processing stages since the output can be always maintained in the previous state when the N block in the ANT logic is not evaluated. The clock rate of such a circuit can be as high as 1.25 GHz verified by TimeMill, which is shown in Fig. 8. The output of the consecutive additions of two n-bit binary numbers can be done in each cycle when the circuit is pipelinedly operated.

III. Simulations and Testing

The proposed 32-bit CLA using the modified ANT logic in a tree-structured style was designed and verified by electronic design automation (EDA) tools, including CADENCE, HSPICE, and TimeMill. After the "o" cell as shown in Fig. 6 is designed and verified, the Silicon Ensemble of CADENCE is employed to carry out the P&R to generate the layout of the whole chip. The parasitic R and C parameters of the layout, then, are extracted by LPE (layout parameter extractor) of DRACULA. Finally, TimeMill is used to measure the final post-layout delay of the circuit. Fig. 8 shows the post-layout simulation result given by TimeMill. Not only the functionality is proved, the cycles of the delay is identical to what we expected, i.e., 3.5 cycles. The maximal operating clock is 1.25 GHz. The layout view of the 32-bit CLA using tree-structured ANT is given in Fig. 9.

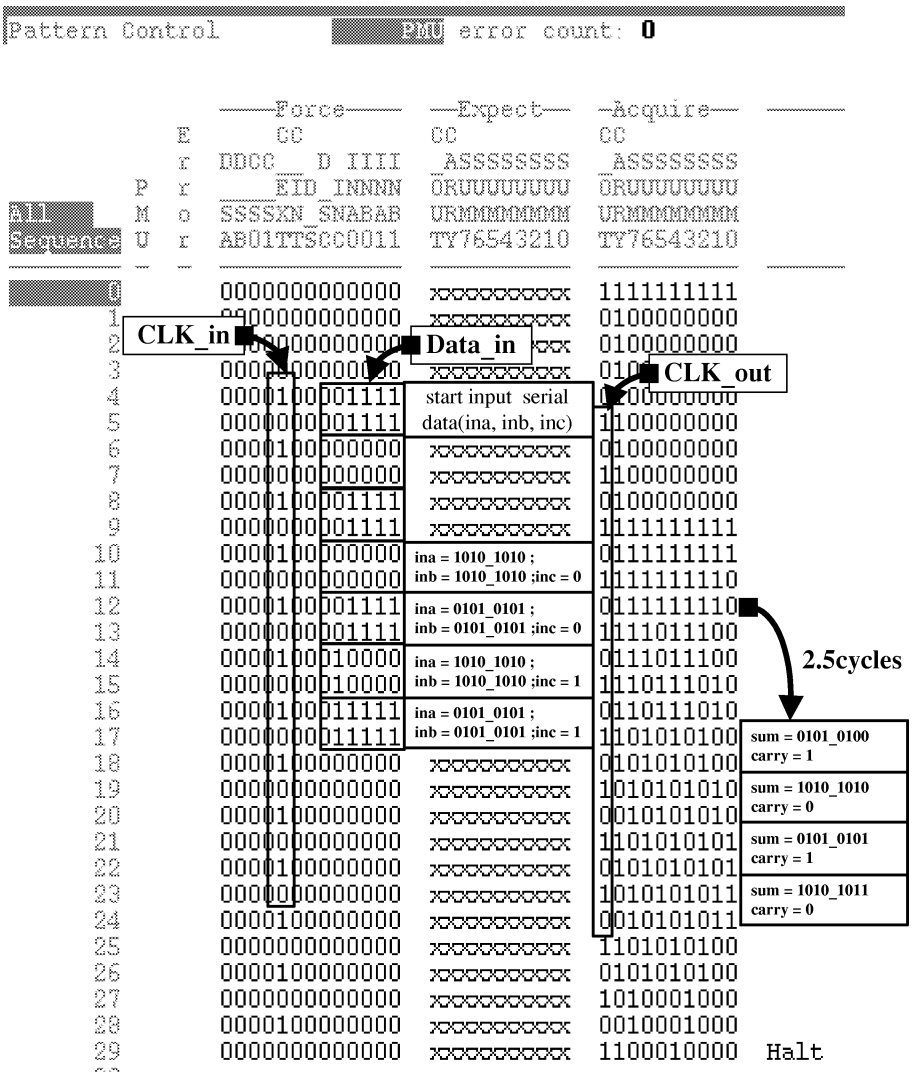

Fig. 13. Measurement of the chip given internal testing vectors (2).

In order to contrast the enhancement of the proposed CLA, the performance comparison between our CLA and prior several adder designs using different logics is revealed in Table II. Note that the clock rate is 1.25 or 1 GHz with a 0.01 ns rise time and the same fall time.

Fig. 10 is the die photo of the physical chip fabricated by TSMC. In order to make the testing results easy to be understood, the first 24 bits of the input data are grounded in Figs. 11–13. When fed with external testing vectors which are 8 valid bits in the lowest byte, the chip is also measured to be correct as shown in Fig. 11 which is a snapshot from HP 1660CP logic analyzer. Meanwhile, internal testing vectors are stored in a self-testing module of the chip. Figs. 12 and 13 are the measurement given by the IMS Logic Master of ATS Co. to reveal the scenarios when the lowest 8 bits are under test. Besides the correct functionality, the speed is physically proven to meet what we expected by delaying $((\log_2 8)/2) + 1 = 2.5$ cycles. This feature addresses the flexibility of our proposed design.

IV. Conclusion

In this brief, we have proposed a high-speed tree-structured ANT logic designs for the implementation of CLAs. Simulation results not only verify the accuracy of the function in the gigahertz range, but also indicate that the proper size of each transistor is adjusted such that a usual square-wave clock can be used. The tree-structured, using only one clock, causes the result of an 32-bit adder to be obtained in 2.8 ns when the 1.25 GHz clock is used. We also derived the number of transistors (area) for larger long adders using the proposed approach. At last, the result of consecutive additions can appear in each cycle when the tree-structured CLA is pipelinedly operated.

References

[1] Z. Wang, G. A. Jullien, W. C. Miller, J. Wang, and S. S. Bizzan, "Fast adders using enhanced multiple-output domino logic," *IEEE J. Solid-State Circuits*, vol. 32, pp. 206–214, Feb. 1997.

[2] J. Yuan and C. Svensson, "High-speed CMOS circuit technique," *IEEE J. Solid-State Circuits*, vol. 24, pp. 62–70, Feb. 1989.

[3] C. M. Lee and E. W. Szeto, "Zipper CMOS," *IEEE Circuits Devices Mag.*, vol. 4, pp. 10–16, May 1986.

[4] R. Rogenmoser and Q. Huang, "An 800-MHz 1 mm CMOS pipelined 8-b adder using true single phase clocked logic-flip-flops," *IEEE J. Solid-State Circuits*, vol. 31, pp. 401–409, Mar. 1996.

[5] R. X. Gu and M. I. Elmasry, "All-N-logic high-speed true-single-phase dynamic CMOS logic," *IEEE J. Solid-State Circuits*, vol. 31, pp. 221–229, Feb. 1996.

[6] M. Afghahi, "A robust single phase clocking for low power high-speed VLSI application," *IEEE J. Solid-State Circuits*, vol. 31, pp. 247–253, Feb. 1996.

[7] C.-C. Wang, C.-J. Huang, and K.-C. Tsai, "A 1.0 GHz 0.6-μm 8-bit carry lookahead adder using PLA-styled all-*N*-transistor logic," *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 133–135, Feb. 2000.
[8] D. Dozza, M. Gaddoni, and G. Baccarani, "A 3.5 ns, 64 bit, carry-lookahead adder," in *Proc. 1996 IEEE Int. Symp. Circuits and Systems*, vol. 2, June 1996, pp. 297–300.
[9] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," *IEEE Trans. Comput.*, vol. C-31, pp. 260–264, Mar. 1982.

High Fan-In Dynamic CMOS Comparators With Low Transistor Count

Chua-Chin Wang, Po-Ming Lee, Chi-Feng Wu, and Hsin-Long Wu

***Abstract*—In this brief, we propose several high fan-in dynamic CMOS comparators with low transistor count, high speed and low power. Major features of the proposed comparators are the rearrangement and reordering of transistors in the evaluation block of a dynamic cell. These comparators can be used as equality comparators, mutual comparators and zero/one detectors, which are widely used in build in self test and memory testing. Furthermore, a 64-bit fast dynamic CMOS comparator is implemented using the proposed dynamic comparator. The measured worst delay of the physical chip with pads is 12 ns.**

***Index Terms*—Dynamic comparator, high fan in, build in self test (BIST), zero/one detector, testing.**

I. Introduction

Wide bit comparators are required in modern circuits design and design for testability such as equality comparison, mutual comparison, and zero/one detection. The comparator is also a key component in the design of parallel testing, signature analyzer, and build in self test (BIST) circuits [1], etc. When a large fan in is required, serial static CMOS gates or the most common high fan-in circuit, i.e., dynamic logic, has to be used [2]–[4]. The prior comparators either need a large area overhead, have long delays, or possess high power consumption [5]. In order to resolve the difficulties introduced by the prior comparators, we propose three types of dynamic comparators to overcome the mentioned problems. The proposed circuits are simulated with 0.35-μ m CMOS 1P4M technology. Their simulation results reveal appealing outcomes. Moreover, a physical 64-bit comparator chip is implemented and fabricated on silicon using 0.5-μm 2P2M CMOS technology to verify the feasibility of the proposed circuits.

II. Equality Comparator

A. Prior Equality Comparators

1) XOR-Based Equality Comparator: An example is shown in Fig. 1(a). The advantage of this structure is that the *XOR*-based comparator can be built by using standard cells. However, if lots of inputs are required, a high fan-in NOR gate is needed. Otherwise, the high fan-in NOR gate must be composed of a tree of smaller NOR gates.

2) Pass-Gate Logic Comparator: An example is given in Fig. 1(b). Transmission gates might be used in low-power circuit designs. This structure does not draw any dc current, but it is slow for long comparators. Besides, the output current will be reduced by the long series of pass transistors if the number of inputs increases. If the output current needs to be kept at a given magnitude, buffers must be inserted in the long chain of the pass transistors.

3) Pseudo-nMOS Comparator: As shown in Fig. 1(c), another kind of the comparator was proposed. This gate draws the dc current, but it is small and fast. The major shortcoming of this circuit is that it can't provide a full-swing output voltage. One solution is to add a buffer at the output side. However, it will then increase the area of the circuit and seriously jeopardize the speed.

B. Proposed Dynamic Equality Comparator

Our proposed dynamic CMOS 4-bit equality comparator is shown in Fig. 1(d). When the clock (CLK) is low, NODE_1 is precharged to the voltage of the power supply (VDD). If $A\langle 0\rangle$ and $B\langle 0\rangle$ are both high, then N1 and N2 are ON and P1 and P2 are OFF. Thus, no current path exists during the evaluation period, and then NODE_1 will be kept high. If $A\langle 0\rangle$ is high and $B\langle 0\rangle$ is low, then N1 and P2 are on. Thus, a current path is formed between NODE_1 and ground through P2 and N1 during the evaluation period. Then NODE_1 will be pulled down. The operation for $A\langle 1\rangle B\langle 1\rangle$, $A\langle 2\rangle B\langle 2\rangle$, and $A\langle 3\rangle B\langle 3\rangle$ is similar. In short, when any pair of $A\langle i\rangle B\langle i\rangle$ is not equal, a current path will be formed and NODE_1 will be low. By contrast, if $A\langle i\rangle$ is equal to $B\langle i\rangle$ for all i, NODE_1 will keep high [6].

The pull-up time is determined only by the pull-up transistor P0, but the ground switch N0 will increase the pull-down time. Note that the ground switch may be omitted if the inputs of every pair are guaranteed at the same states during the precharge period [2], [5].

III. Mutual Comparator

By a similar thought, we can also improve the mutual comparators. The term "mutual comparator" means that all inputs must be of the same value. Mutual comparators are usually used in the parallel testing of memory where the outputs of the memory cell arrays are mutually compared.

A. Prior Mutual Comparator

Fig. 1(e) shows a typical 4-bit mutual comparator circuit. $D\langle 0\rangle$ is compared with $D\langle 1\rangle$, $D\langle 1\rangle$ is compared with $D\langle 2\rangle$, $D\langle 2\rangle$ with $D\langle 3\rangle$, and optionally $D\langle 3\rangle$ may be compared with $D\langle 0\rangle$. The output of the comparators, consisting of XOR gates, are fed into an NOR gate in order to generate the error signal [7].

B. Proposed Dynamic Mutual Comparator

Our 4-bit mutual comparator is shown in Fig. 1(f). When the CLK is low, NODE_1 is precharged to VDD. If $D\langle 0\rangle$, $D\langle 1\rangle$, $D\langle 2\rangle$, and $D\langle 3\rangle$ are all high, then N0, N1, N2, and N3 are on and P0, P1, P2, and P3 are all off. Thus, no current path exists during the evaluation period, and then NODE_1 will remain high. If $D\langle 0\rangle$, $D\langle 1\rangle$, $D\langle 2\rangle$, and $D\langle 3\rangle$ are all low, then N0, N1, N2, and N3 are off, and P0, P1, P2, and P3 are all on. Again, no current path exists during the evaluation period, either. Then, the NODE_1 will be kept high. If any input is different from the others, there will be some NMOS and some PMOS turn on

Manuscript received April 1, 2002; revised September 5, 2002.This work was supported in part by the National Science Council under Grant NSC 87-2215-E-110-010 and Grant NSC 86-2622-E-009-009, and in part by the Academic Foundation of Taiwan ERICSSON Co. Ltd. This paper was recommended by Associate Editor Y. Ismail.

C.-C. Wang, P.-M. Lee, and H.-L. Wu are with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan, R.O.C. (e-mail: ccwang@ee.nsysu.edu.tw).

C.-F. Wu is with Wafer Testing Factory of Philips Semiconductors, Kaohsiung 811, Taiwan, R.O.C., and also with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TCSI.2003.816338