1 GHz 64-bit high-speed comparator using ANT dynamic logic with two-phase clocking

C.-C. Wang
C.-F. Wu
K.-C. Tsai

Indexing terms: Noninverting block, Parallel comparator, Two-phase clocking

Abstract: A high-speed 64-bit comparator using two-phase clocking dynamic CMOS logic with modified noninverting all-N-transistor block is presented. The pull-up charging and pull-down discharging of a comparator unit are accelerated by inserting two feedback MOS transistors between the evaluation N-block and the output. Detailed simulation results reveal appropriate L/W guidelines for the all-N-transistor block design. To increase throughput a parallel tree structure with two-phase clocks is employed. The comparator units of two adjacent layers are triggered by two out-of-phase clocks so that their individual outputs are pipelined without using extra hardware, e.g. latches. The operating clock frequency is 1.0 GHz while the compared output of two 64-bit binary numbers is done in 3.5 cycles.

1 Introduction

High-speed operation has long been a target of circuit design owing to the speed demand of supercomputing, CPU, etc [1, 2]. One of the critical operations is the comparison of two binary data. For instance, the conditional JMP and MOV instructions of x86s, the priority encoders, BAM data retrieving [3], and so on. Theoretically, the fastest comparator is made of full combinational logic gates. However, the gate count and the area will be a problem when the length of the data is very large, e.g. $n = 64$. Meanwhile, the limited fan-in and fan-out is another impractical factor for the realisation of such a combinational comparator.

CMOS dynamic logic is one of the promising options to challenge GHz operations [4] for comparator design. Other logics suffer from different difficulties. For example, domino logic [1] cannot be noninverting; NORA [5] has the charge-sharing problem; all-N-logic [4] and robust single-phase clocking [6] cannot operate correctly under clocks with short rise or fall times, which cannot be easily integrated with other parts of the logic design; single-phase logic [7] and Zipper CMOS [8] contain slow P-logic blocks. We propose an all-N-transistor (ANT) noninverting function block for high-speed design. A parallel comparator using ANTs arranged in a tree structure and triggered by two out-of-phase clocks is implemented.

IEE Proceedings online no. 19982348

Paper first received 4th July 1997 and in revised form 3rd July 1998

The authors are with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan 80424, Republic of China

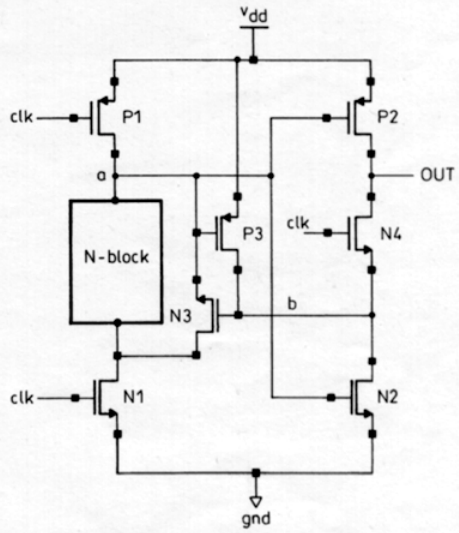

Fig. 1 *Schematic diagram of ANT logic*

2 High-speed 64-bit comparator

2.1 All-N-transistor function unit

Although the N-block dynamic logic intrinsically possesses high speed [4], it is not good enough for operation in the gigahertz range. The reasons are first, 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 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. In short, P3 and N3, respectively, provide an extra charge and an extra discharge path such that the speed of the evaluation can be accelerated. The ANT in Fig. 1 is noninverting. The detailed operation of the ANT is described as follows:

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

(ii) 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 the output is high, N3 will be turned on via N4. This means that N3 provides another fast discharge path for the charge at node a. When the voltage at node a falls below the threshold voltage of PMOS, P2 and P3 turn on. The output will then be charged to V_{dd} via paths P2 and P3, N4.

(iii) 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 falls. When $V_a - V_{dd} > V_{tp}$, P3 will be on such that the gate of N3 is charged to 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.

Summarising (ii) and (iii), the output will be high when the N-block is evaluated 'pass', i.e. '1', during clk = 1.

(iv) When clk = 1 and the N-block is evaluated to be 'stop', the charge at node a should be kept if the previous output state is low. There will be no discharge path for node a because N3 will be off via N4. If the previous state is high, the output will be at 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 the ANT logic block is thus conclusively correct. The feedback transistor pair N3 and P3 indeed provide extra paths for the operations.

Although the descriptions of the (ii) and (iv) appear contradictory when the previous state of the output is high, this can be resolved by introducing the design ratioed to the size of the N-block transistors and N3. When the clock switches from low to high and the previous output state is high, the voltage at node b will be charged through N4, but discharged through N2 initially. If the voltage at node a drops quickly enough, N2 will be rapidly turned off. Then the voltage at node b is solely charged by the output node. Then N3 is turned on by the voltage at node b, which in turn becomes another discharge path for node a. In conclusion, the key factor for the proposed ANT logic to be functionally correct is that the initial discharge speed of node a must be 'fast'. This indicates that the resistance of the paths in the N-block must be smaller than that of N3 and N2. In other words, one 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 Section 2.2.

Table 1: Sizes of ANT logic block

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

2.2 Sizing problem

One of the reasons why other high-speed logic cannot run correctly given clocks with short rise or fall time is that the size of each transistor cannot be tuned properly. Both [4, 6] intrinsically possess this shortcoming, which is why they could not use normal square-wave clocks in the GHz range. The sizing problem of transistors in the ANT drastically affects speed.

Theoretically, wider transistors produce smaller resistance. We have made 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 as tabulated in Table 1.

2.3 Two-bit comparator unit

To enhance throughput we adopt the strategy of comparing two bits of each word simultaneously. Assume the bits to be compared are A_1A_0 and B_1B_0. The optimised boolean expressions for OUT_1OUT_0 are as follows:

$$OUT_1 = A_1A_0 + A_1\overline{B_1} + A_0\overline{B_1}$$

$$OUT_0 = B_1B_0 + \overline{A_1}B_0 + \overline{A_1}B_1$$

The notation of OUT_0OUT_1 denotes the following results:

Table 2: Function table of 2-bit comparator

OUT_1OUT_0	Result
00	$A = B$
01	$A < B$
10	$A > B$
11	$A = B$

The 2-bit comparator unit is shown in Fig. 2. Note that the 2-bit comparator unit can easily be changed to an equality checker by XORing OUT_1 and OUT_0.

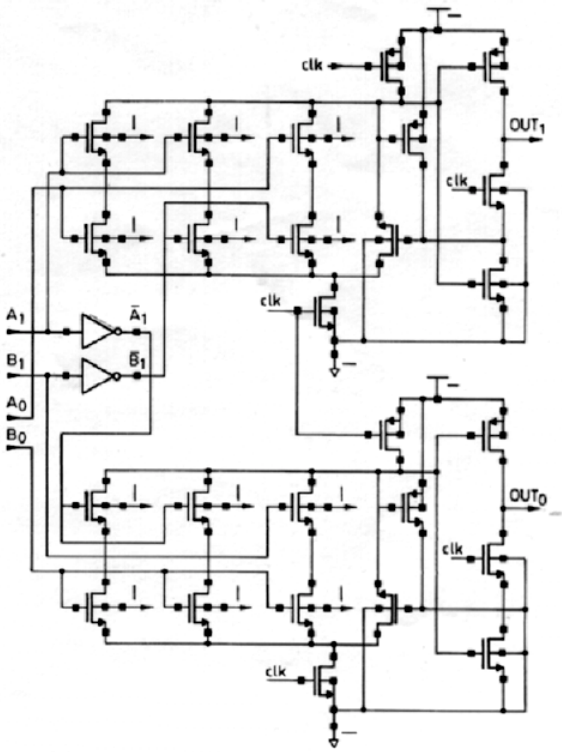

Fig. 2 *Schematic diagram of 2-bit comparator unit*

2.4 Two-phase clocking in cascaded units

In the final realisation of the 64-bit comparator the 4-bit comparator in fact is the basic building block. The 4-bit comparator is composed of three 2-bit comparator units as shown in Fig. 3. The operation is as follows:

— When clk = 1 or high, the inputs $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ are fed into the respective units of the first layer. Their individual outputs will be ready when clk turns low.

— When clk = 0, the outputs of the first layer units are fed into the second layer unit. The final result will be ready when clk turns high.

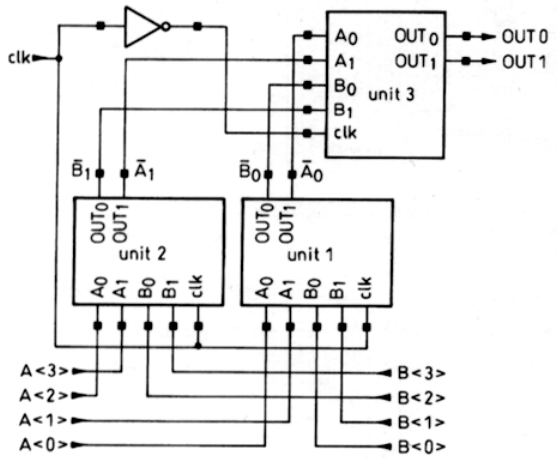

Fig. 3 *Block diagram of 4-bit comparator*

In short, the adjacent layers employ two out-of-phase clocks to prevent transparency of the data. This implies that no latches are required in the entire operation or in the pipeline operations. Meanwhile, since the input of the next layer and the output of the previous layer is on the same half cycle, it is twice as fast as using one clock.

An example of 4-bit comparator waveforms is shown in Fig. 4. Referring to Fig. 4, we set $B_3 = B_1 = 0$ and $A_2 = A_0 = B_2 = B_0 = 1$ and let A_1 and A_3 be periodic signals with frequency 500 and 250 MHz, respectively. At the high half of the first cycle of the clk the inputs are A = 1111 and B = 0101, and then the outputs of the comparator units are presented to the comparator unit of the second layer, which is $OUT1_1OUT1_0 = 10$ and $OUT2_1OUT2_0 = 10$. They are correct. Then in the low half of the first cycle, these outputs are fed into the comparator unit at the second layer. And the result $OUT_1OUT_0 = 10$ appears at the high half of the second cycle.

2.5 Parallel 64-bit comparator

Fully exploiting the 'half-cycle stealing' of the ANT logic, we construct the entire parallel 64-bit comparator by organising the 2-bit comparator units as in Fig. 5. The units in adjacent layers alternatively use the out-of-phase clocks. In total, 63 units are used and the correct result should be placed at the output after 3.5 cycles. The two out-of-phase clocks are in fact generated form one clock source. Only one inverter is required to flip the phase of the original clock such that the second clock is produced with 180° phase shift.

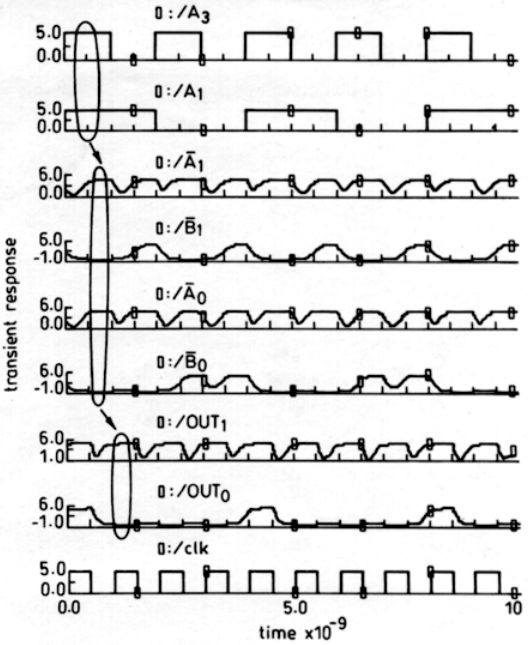

Fig. 4 *Waveform diagram of simulation of 4-bit comparator*

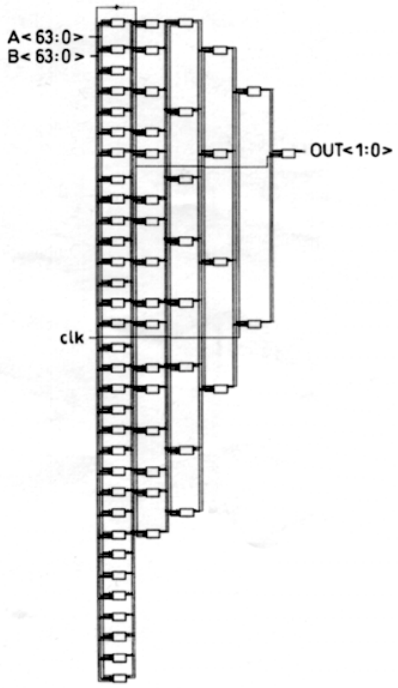

Fig. 5 *Schematic diagram of 64-bit comparator using ANT logic*

3 Performance simulations

To verify the performance of the 64-bit comparator using the ANT logic with two-phase clocking, we use the TSMC 0.6μm SPDM technology to simulate several comparator designs using different logics. The clock rate is 1.0GHz with 0.01ps rise and fall time. The results are tabulated in Table 3.

Many comparator operations in microprocessors are executed by subtractors. Hence we choose a recently

proposed enhanced domino logic (EDL) [9] for comparison, which is included in Table 3. Though the delay of the EDL is also fast, the area is much too large, i.e. roughly 3074 × 4 = 12296 transistors. Note that owing to intrinsic limitations, traditional dynamic logic cannot operate with 1GHz clock. We use a slower clock frequency, i.e. 500MHz, to estimate its delay which is tabulated in Table 3.

Table 3: Performance comparison of different logics

Logic	Delay	Transistors
ANT	3.5ns	1890
Dynamic (using 500MHz clocks)	14.0ns	1260
Adder/subtractor by enhanced domino [9]	5.4ns	3074 (gates)
All-N-logic [4]	Failed	2062

To show the correctness of our delay estimation, an output waveform in Fig. 6 shows a worst-case comparison of our design, i.e. when A = 000...001 and B = 000...000 are presented to the 64-bit ANT logic comparator. Note that the final result 10 appears after 3.5 cycles.

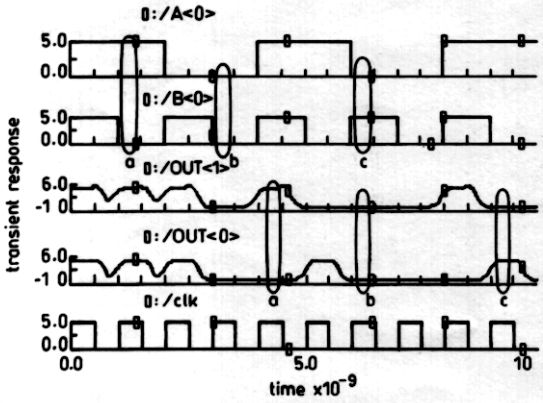

Fig. 6 *Worst-case simulation waveform*

4 Conclusion

We have proposed a novel high-speed ANT logic block for comparator design. Not only has the correctness of the function in the gigahertz range been verified, but also the proper size of each transistor has been tuned such that the usual square-wave clock can be used to run the 64-bit comparator. The pipeline-like half-cycle stealing structure using two out-of-phase clocks makes the result appear in 3.5 cycles (ns if 1GHz clocks are used).

5 Acknowledgment

This research was partially supported by the National Science Council under grant NSC 86-2622-E-009-009.

6 References

1 KRAMBECK, R.H., LEE, C.M., and LAW, H.-S.: 'High-speed compact circuits with CMOS', *IEEE J. Solid-State Circuits*, June 1982, **17**, pp. 614–619
2 PICKER, D., and FELLMAN, R.D.: 'A VLSI priority packet queue with inheritance and overwrite', *IEEE Trans. VLSI*, June 1995, **3**, (2), pp. 245–252
3 WANG, C.-C., and HORNG, I.-H.: 'Realization of bidirectional associative memory using a pseudo-parallel searching approach'. Proceedings of the IEEE international conference on *Neural networks*, Dec. 1995, Vol. 3, pp. 1502–1507
4 GU, R.X., and ELMASRY, M.I.: 'All-N-logic high-speed true-single-phase dynamic CMOS logic', *IEEE J. Solid-State Circuits*, Feb. 1996, **31**, (2), pp. 221–229
5 GONCALVES, N.F., and DE MAN, H.J.: 'NORA: A race-free dynamic CMOS technology for pipelined logic structures', *IEEE J. Solid-State Circuits*, June 1983, **18**, pp. 261–266
6 AFGHAHI, M.: 'A robust single phase clocking for low power, high-speed VLSI applications', *IEEE J. Solid-State Circuits*, Feb. 1996, **31**, (2), pp. 247–253
7 YUAN, J., and SVENSSON, C.: 'High-speed CMOS circuit technique', *IEEE J. Solid-State Circuits*, Feb. 1989, **24**, pp. 62–70
8 LEE, C.M., and SZETO, E.W.: 'Zipper CMOS', *IEEE Circuits Devices Mag.*, May 1986, pp. 10–16
9 WANG, Z., JULLIEN, G.A., MILLER, W.C., WANG, J., and BIZZAN, S.S.: 'Fast adders using enhanced multiple-output domino logic', *IEEE J. Solid-State Circuits*, Feb. 1997, **32**, (2), pp. 206–214