

CMOS Comparators for High-Speed and Low-Power Applications

Eric R. Menendez Dumezie K. Maduik Rajesh Garg Sunil P. Khatri

Abstract—In this paper, we present two designs for CMOS comparators: one which is targeted for high-speed applications and another for low-power applications. Additionally, we present hierarchical pipelined comparators which can be optimized for delay, area, or power consumption by using either design in different stages.

Simulation results for our fastest hierarchical 64-bit comparator with a 1.2 V 100 nm process demonstrate a worst-case delay of 440 ps. To enable a fair comparison with previously reported approaches, we also simulated our designs with a 3.3 V TSMC 0.35 μm process. For this experiment, the fastest design has a latency of 1.67 ns, which represents a 33% speed improvement over the best previously reported approach to date.

I. INTRODUCTION

Binary comparators are found in a wide variety of circuits, such as microprocessors, communications systems, encryption devices, and many others. A faster, more power efficient, or more compact comparator would be an advantage in any of these circuits.

In this paper, we present two CMOS unsigned binary comparators. Our first design is optimized for area and power efficiency, while our second design is geared towards maximum speed. The use of dynamic CMOS logic allows our designs to perform binary comparison of wide operands with increased speed and area efficiency. However, a downside of dynamic CMOS is that it requires a precharge period, which traditionally is wasted time. Our high-speed design takes advantage of the precharge time to compute several intermediate signals using static CMOS circuitry, which results in a faster design than previously reported results. From these two designs, hierarchical solutions can be created to meet a variety of delay, power, and area requirements. Our hierarchical designs are pipelined for maximum throughput.

Our simulations were performed in SPICE [1], and results are reported for both a 1.2 V 100 nm process [2] and a 3.3 V TSMC 0.35 μm process [3] for a fair comparison with previous work. Our best hierarchical comparator is 33% faster than the best previously reported approach to date [4].

The rest of this paper is organized as follows: Section II is a review of several previously reported approaches, Section III describes the low-power comparator design in detail, Section IV describes the high-speed design, Section V presents our hierarchical, pipelined comparators, and Section VI contains all of our simulation results as well as comparisons to previously reported designs.

II. PREVIOUS WORK

Several previous high-speed comparator designs have been proposed. In [5], an all-N precharged function block is at-

tached to several feedback transistors which add extra discharge paths, thus reducing the comparator's delay. However, the precharge period is not utilized for any computation, so the design is not as fast as our high-speed design.

In [6], a specialized priority-encoding algorithm is realized in a "magnitude decision module" to compare the operands, but this module contains many series transistors in critical discharge paths, so it suffers from increased delay and lack of suitability for wide operands.

The fastest comparator previously reported to date is found in [4]. However, this paper does not present a true less-than, equal-to, or greater-than comparator; instead, it only discusses equality, mutual, and zero/one comparators. Additionally, the single-cycle comparators presented are not suitable for wide operands. However, the authors of [4] compare their work with the designs in [5] and [6] and demonstrate their approach to be the fastest currently available at the time.

III. DESIGN OF A SMALL, LOW-POWER COMPARATOR

Our low-power comparator design (Figure 1) consists of a precharged gate with n pulldown stages connected by $n - 1$ intermediate pass-transistors, where n is the number of input bits. During the precharge period, (when the clock is low) each stage is precharged to V_{DD} . In the evaluate period, (when the clock is high) the i^{th} pulldown stack in the circuit will form a discharge path if $A_i > B_i$. The XNOR gates (Figure 2) attached to the intermediate pass-transistors allow pulldown stack $i - 1$ to discharge the output if $A_i = B_i$. The XNOR gate outputs are computed during the precharge period to avoid any potential race condition caused by the pass-transistors being in the wrong state. The result is that the output discharges if and only if $A > B$. Therefore, the output is high if and only if $A \leq B$.

To determine if $A = B$, the outputs of all the XNOR gates are ANDed together. This is realized using a hierarchical tree with alternating levels of NAND and NOR gates (Figure 3). The output of this tree is identical to the AND of all the XNOR gates. The final output of this tree is high if and only if $A = B$.

While not the fastest design possible, this comparator is small and simple, resulting in low power consumption and active circuit area. For small n , this comparator is reasonably fast, while for larger n , the large number of pass-gates in the critical delay path results in a significant delay.

IV. DESIGN OF A FAST "LOOK-BEHIND" COMPARATOR

Given two n -bit unsigned binary numbers, A and B , consider the following signals:

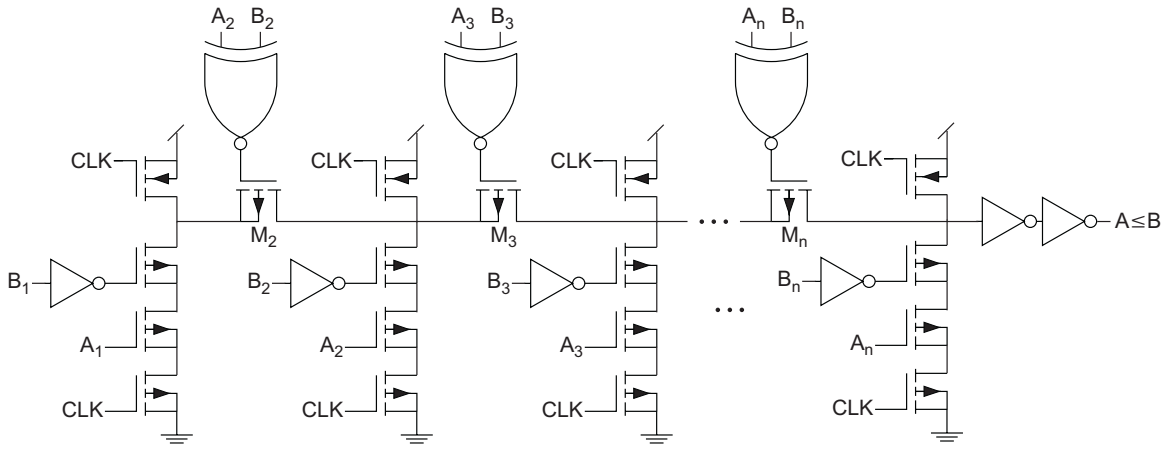


Fig. 1. Output stage of low-power design

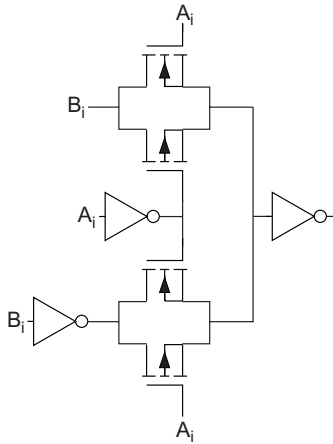


Fig. 2. XNOR gate to control pass-transistor

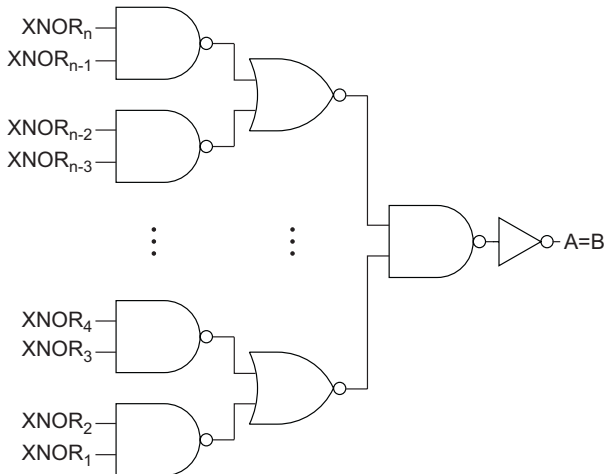


Fig. 3. Hierarchical NAND/NOR tree for equality signal

Bit:	8	7	6	5	4	3	2	1
<i>A</i> :	1	1	1	0	0	0	1	0
<i>B</i> :	1	1	1	1	0	1	0	1
<i>LT</i> :	0	0	0	1	0	1	0	1
<i>Equal</i> :	1	1	1	0	1	0	0	0
<i>EQ</i> :	1	1	1	1	0	0	0	0

TABLE I

EXAMPLE OF *LT*, *Equal*, AND *EQ* SIGNALS FOR AN 8-BIT COMPARATOR

- *LT_i*: High if $A_i < B_i$.
- *Equal_i*: High if $A_i = B_i$.
- *EQ_i*: High if $Equal_{i+1} \cdots Equal_n$ are all high. *Equal_n* is defined to be high.

$A < B$ if and only if $LT_i \cdot EQ_i$ is true for any $i \in \{1, 2, \dots, n\}$, since in that case all bits more significant than the i^{th} bit are equal ($EQ_i = 1$) and $A_i < B_i$ ($LT_i = 1$). To determine if $A = B$, we simply perform the AND of EQ_{Q1} and $Equal_{Q1}$. If the result is true, then $A = B$.

To take advantage of the traditionally wasted precharge time and to avoid any potential race conditions, all of the LT_i , $Equal_i$, and EQ_i signals are computed during the precharge period. When the clock goes high, all that remain to be computed are the $A < B$ and $A = B$ outputs. All of the LT_i and $Equal_i$ signals (Figures 4 and 5) are computed in parallel, so the EQ_i signals will take the longest time to compute. The EQ_i signals are computed by first ANDing together all of the $Equal_i$ signals with a hierarchical tree consisting of alternating NAND/NOR stages (Figure 6). The output of this tree is identical to the AND of all the $Equal_i$ signals. From this tree, each EQ_i signal is computed by ANDing together the proper nodes from this tree (Figure 7). This AND is performed by another hierarchical NAND/NOR tree. Each gate with a fanout of 16 or greater has a buffered output to minimize delay. Once all of the EQ_i signals have been computed, the clock goes high and the final output is computed.

The final output is computed with a large precharged NOR gate (Figure 8). The i^{th} pulldown stack ($i \in \{1, 2, \dots, n\}$) of the NOR gate will form a discharge path to ground if and only if $LT_i \cdot EQ_i$ is true. Since the gate is attached to an inverter,

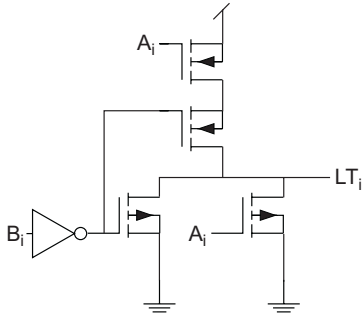


Fig. 4. LT_i signal for high-speed comparator

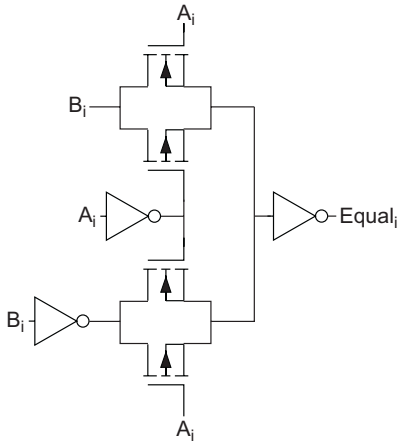


Fig. 5. $Equal_i$ signal for high-speed comparator

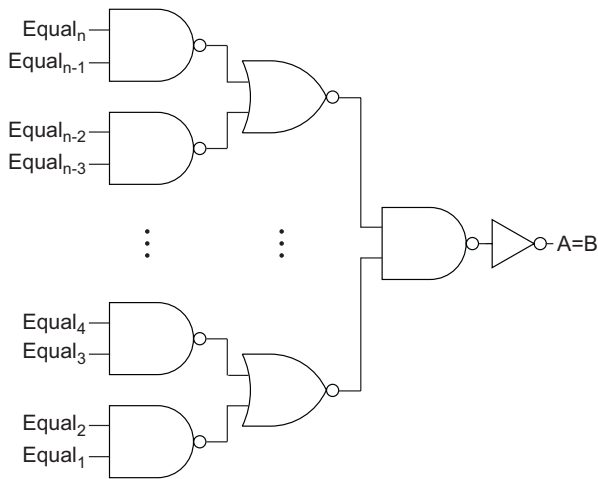


Fig. 6. Hierarchical NAND/NOR tree to compute EQ_i signal for high-speed comparator

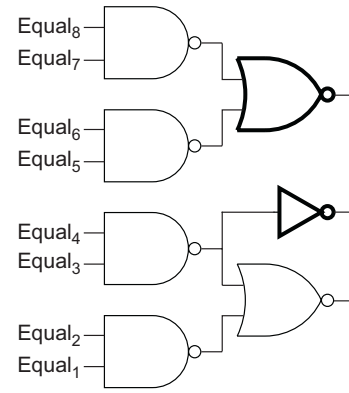


Fig. 7. Example for 8-bit comparator: EQ_2 is equivalent to the AND of the outputs of the gates shown in bold outline

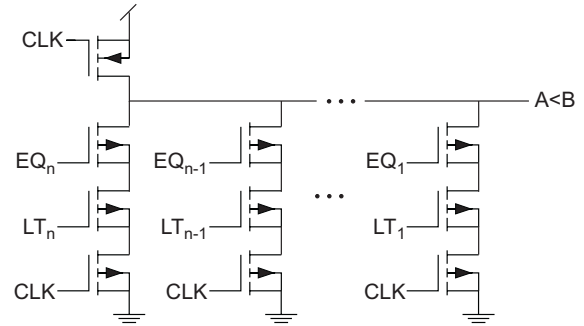


Fig. 8. Output stage of high-speed comparator

the output will go high. Therefore, the output is high if and only if $A < B$.

For example, in Table I, LT_i is true for bits 5, 3, and 1, since those are the only bits for which $A_i < B_i$. Likewise, since the 8th, 7th, 6th, and 4th bits of A and B are equal, $Equal_i$ is true when $i \in \{8, 7, 6, 4\}$. EQ_i is true for all $i \geq 5$, since $Equal_i$ is true for all $i > 5$. Therefore, $A < B$ since $EQ_5 \cdot LT_5$ is true.

This comparator has a very low delay, because the precharge time is used maximally and no series pass-gates are used in the critical delay path. This circuit has a larger area and power consumption since an EQ_i signal is computed for each bit.

V. HIERARCHICAL COMPARATORS FOR WIDE INPUTS

While the second design is suitable for wide comparators, faster designs are possible by combining several smaller comparators in a hierarchical fashion (Figure 9). By combining one or both of the designs with different widths in different stages, it is possible to take advantage of the best characteristics of both designs and create hybrid comparators to meet a wide variety of speed, power, and area requirements. Additionally, these comparators are pipelined for maximum throughput. Although the intermediate circuitry and flip-flops add some time to the delay, which we account for in our simulations, the results are still faster than a wide monolithic comparator.

For example, in the hierarchical comparator of Figure 9, the throughput is equal to the maximum clock period of the Stage

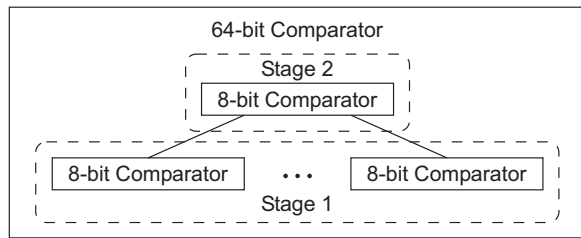


Fig. 9. Example of hierarchical comparator

1 and Stage 2 comparators. The latency is two clock periods. The throughput, area, and power consumption for large n are significantly smaller than those of a single stage comparator.

VI. PERFORMANCE EVALUATION AND COMPARISON

We simulated our designs in SPICE 3f5 [1] with a 1.2 V 100 nm process [2]. The results for the low-power and high-speed designs are listed in Tables II and III, respectively. Additionally, we simulated our comparators with a 3.3 V TSMC 0.35 μm process [3] to enable a fair comparison with previously reported work. These results are listed in Tables IV and V.

The 100 nm simulation results for 64- and 128-bit hierarchical comparators are listed in Tables VI and VII, respectively. The 0.35 μm results are listed in Tables VIII and IX.

Overall, our fastest 64-bit comparator is a hierarchical comparator consisting of two levels of 8-bit high-speed comparators. The latency of this comparator is 440 ps with the 100 nm process and 1.67 ns with the 0.35 μm process. The fastest previously reported approach has a latency of 2.5 ns [4]. Therefore, our design is 33% faster than the previously reported state of the art. This is due to the fact that we take advantage of the precharge time to compute intermediate signals. We can also construct low-power and low-area comparators in a similar manner.

Bits	Delay (ps)	Avg. Power (mW)	Active Area (μm^2)
4	222	0.0941	1.59
8	392	0.0824	3.15
16	900	0.0613	6.15
32	2571	0.0414	12.27
64	8471	0.0253	24.39

TABLE II

100 NM SIMULATION RESULTS FOR LOW-POWER DESIGN

Bits	Delay (ps)	Avg. Power (mW)	Active Area (μm^2)
4	105	0.133	2.43
8	175	0.200	5.23
16	220	0.367	11.43
32	300	0.679	25.07
64	462	1.251	54.47

TABLE III

100 NM SIMULATION RESULTS FOR HIGH-SPEED DESIGN

Bits	Delay (ps)	Avg. Power (mW)	Active Area (μm^2)
4	766	1.46	19.45
8	1363	1.58	38.56
16	3086	1.40	75.31
32	8654	1.03	150.28
64	28335	0.656	298.75

TABLE IV

0.35 μm SIMULATION RESULTS FOR LOW-POWER DESIGN

Bits	Delay (ps)	Avg. Power (mW)	Active Area (μm^2)
4	350	2.09	29.77
8	680	2.91	64.07
16	849	5.63	140.02
32	1366	10.08	307.11
64	2063	18.74	667.26

TABLE V

0.35 μm SIMULATION RESULTS FOR HIGH-SPEED DESIGN

VII. CONCLUSION

We have presented two different designs for a CMOS unsigned binary comparator; one is slower but small and power efficient, and the other uses more power and transistors but is much faster. This is because it uses dynamic CMOS to compute the output during the evaluate period and static CMOS to do some computation during precharge. These designs may be combined and pipelined to meet a variety of speed, power, and area requirements. Our fastest design is about 33% faster than previously reported approaches.

REFERENCES

- [1] L. Nagel, "Spice: A computer program to simulate computer circuits," in *University of California, Berkeley UCB/ERL Memo M520*, May 1995.
- [2] "Bptm website," www-device.eecs.berkeley.edu/~ptm/.
- [3] "Mosis website," www.mosis.org.
- [4] C.-C. Wang, P.-M. Lee, C.-F. Wu, and H.-L. Wu, "High fan-in dynamic cmos comparators with low transistor count," *IEEE Transactions on Circuits and Systems I*, vol. 50, pp. 1216–1220, Sept. 2003.
- [5] C.-C. Wang, C.-F. Wu, and K.-C. Tsai, "1 ghz 64-bit high-speed comparator using ant dynamic logic with two-phase clocking," *IEE Proceedings on Computers and Digital Techniques*, vol. 145, pp. 433–436, Nov. 1998.
- [6] C.-H. Huang and J.-S. Wang, "High-performance and power-efficient cmos comparators," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 254–262, Feb. 2003.

Stage 1	Stage 2	Latency (ps)	Avg. Power (mW)	Active Area (μm^2)
8-bit High Speed	8-bit High Speed	440	1.80	47.07
16-bit High Speed	4-bit Low Power	534	1.56	47.31
4-bit Low Power	16-bit High Speed	534	1.87	36.87
16-bit High Speed	4-bit High Speed	530	1.56	48.15
4-bit High Speed	16-bit High Speed	530	1.86	50.31

TABLE VI
100 NM 64-BIT HIERARCHICAL COMPARATORS

Stage 1	Stage 2	Latency (ps)	Avg. Power (mW)	Active Area (μm^2)
16-bit High Speed	8-bit High Speed	530	3.10	96.67
8-bit High Speed	16-bit High Speed	530	3.08	95.11
16-bit High Speed	8-bit Low Power	874	1.83	94.59
8-bit Low Power	16-bit High Speed	874	1.54	61.83

TABLE VII
100 NM 128-BIT HIERARCHICAL COMPARATORS

Stage 1	Stage 2	Latency (ps)	Avg. Power (mW)	Active Area (μm^2)
8-bit High Speed	8-bit High Speed	1610	26.2	576.63
16-bit High Speed	4-bit Low Power	2008	23.8	579.53
4-bit Low Power	16-bit High Speed	2008	27.1	451.22
16-bit High Speed	4-bit High Speed	2008	23.7	589.85
4-bit High Speed	16-bit High Speed	2008	24.5	616.34

TABLE VIII
0.35 μm 64-BIT HIERARCHICAL COMPARATORS

Stage 1	Stage 2	Latency (ps)	Avg. Power (mW)	Active Area (μm^2)
16-bit High Speed	8-bit High Speed	2008	47.5	1184.23
8-bit High Speed	16-bit High Speed	2008	45.1	1165.14
16-bit High Speed	8-bit Low Power	3036	31.6	1158.72
8-bit Low Power	16-bit High Speed	3036	29.0	756.98

TABLE IX
0.35 μm 128-BIT HIERARCHICAL COMPARATORS