CMOS Comparators for High-Speed and Low-Power Applications

Eric R. Menendez* Dumezie K. Maduike‡ Rajesh Garg◇ Sunil P. Khatri◇
* Department of CSE, The Pennsylvania State University, University Park, PA 16802
‡ Department of ECE, Rutgers University, New Brunswick, NJ 08854
◇ Department of ECE, Texas A&M University, College Station TX 77843.

***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 5.0 V AMIS 0.5 μm process as well. For this experiment, the fastest design has a latency of 1.33 ns, which represents a 37% speed improvement over the best previously reported approach to date (which was implemented in a 0.5μm process).**

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]. Also, we performed our simulations in a 5.0 V AMIS 0.5 μm process [3] for a fair comparison with the best results from previous work (which were obtained for a 0.5μm process). **Our best hierarchical comparator is 37% faster than the best previously reported approach to date [4], which was done in a 0.5μm process.** For this comparison the delay for both approaches were obtained using spice without layout parasitics.

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 attached 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, as we will show in the sequel.

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. 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 of publication of [4].

In order to compare our design with the fastest reported comparator to date [4] (which was simulated in a 0.5μm process), we performed our simulations in a

1-4244-9707-X/06/$20.00 ©2006 IEEE

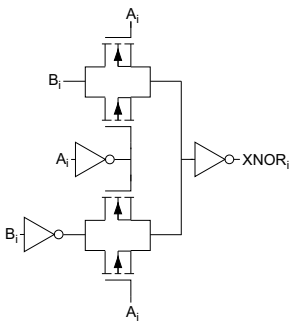

Fig. 2. XNOR gate to control pass-transistor

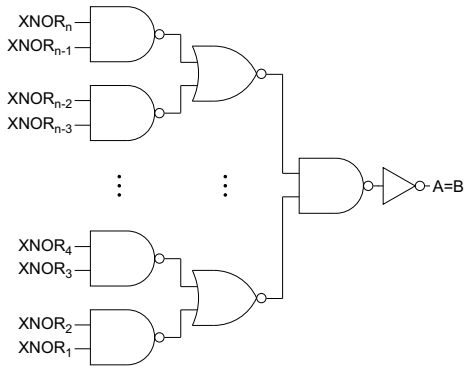

Fig. 3. Hierarchical NAND/NOR tree for equality signal

Bit:	8	7	6	5	4	3	2	1
A:	1	1	1	0	0	0	1	0
B:	1	1	1	1	0	1	0	1
LT:	0	0	0	1	0	1	0	1
$Equal$:	1	1	1	0	1	0	0	0
EQ:	1	1	1	1	0	0	0	0

TABLE I
EXAMPLE OF LT, $Equal$, AND EQ SIGNALS FOR AN 8-BIT COMPARATOR

0.5μm process. We show that our implementation is 37% faster than [4] under these conditions. We report results for our comparator using a 0.1μm process as well.

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 to $n-1$ 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. The reduced power of this design stems from the fact that it has smaller active area and fewer switching signals. The output of Figure 1 has a ripple nature, resulting in higher delay. 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:

- 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 $i \in \{1, 2, \cdots, n\}$, 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_1 and $Equal_1$. 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 $Equal_j$ ($i+1 \leq j \leq n$) 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 $Equal_j$ 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.

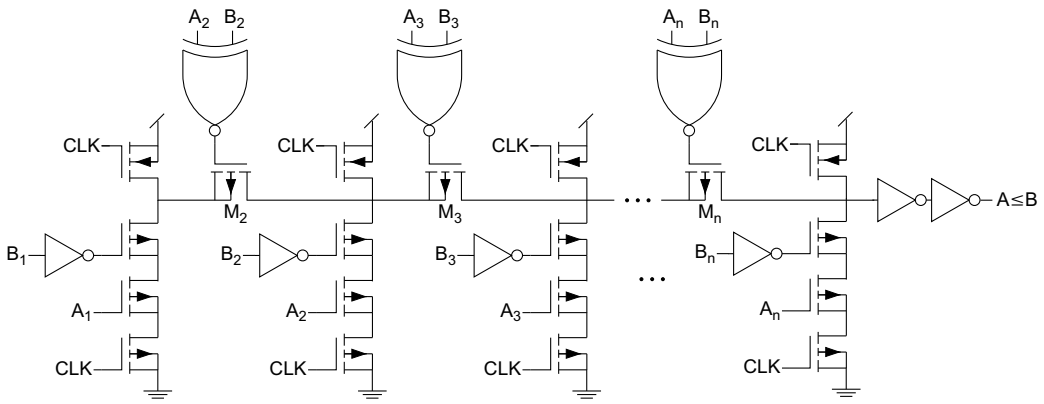

Fig. 1. Output stage of low-power design

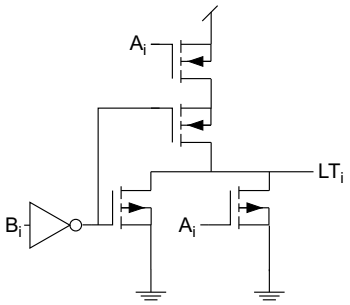

Fig. 4. LT_i signal for high-speed comparator

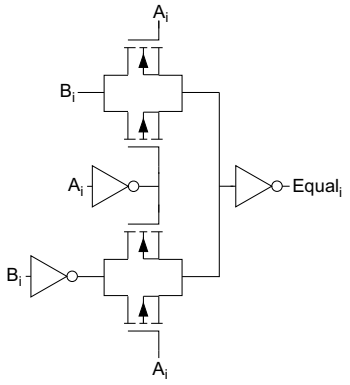

Fig. 5. $Equal_i$ signal for high-speed comparator

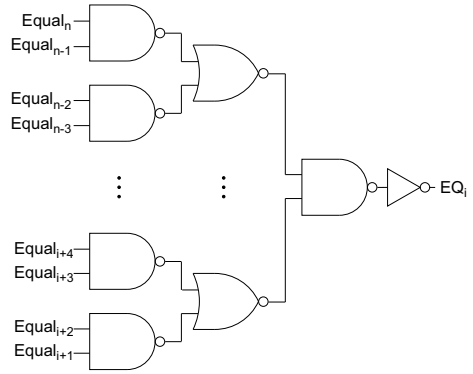

Fig. 6. Hierarchical NAND/NOR tree to compute EQ_i signal for high-speed comparator

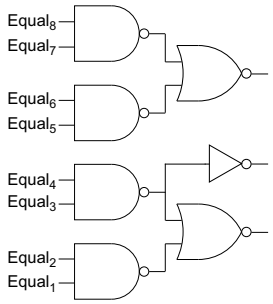

Fig. 7. Example for 8-bit comparator: EQ_2 is equivalent to the AND of the outputs of the gates shown in bold outline

The final output is computed with a large precharged NOR gate (Figure 8). The i^{th} pulldown stack ($i \in \{1, 2, \cdots, n\}$) 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, the output will go high. Therefore, the output is high if and only if $A < B$.

This design consumes higher power since all the signals LT_i, $Equal_i$ and EQ_i are independently computed in the pre-charge phase. During the evaluation phase all that remains to be done is to compute $A < B$ from the circuit in Figure 8, as well as the logical AND of EQ_1 and $Equal_1$ (since $A = B$ iff $EQ_1 \cdot Equal_1 = 1$). Hence the design has a low delay.

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 8^{th}, 7^{th}, 6^{th}, and 4^{th} bits of A and

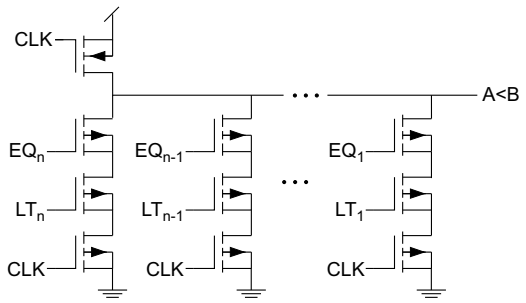

Fig. 8. Output stage of high-speed comparator

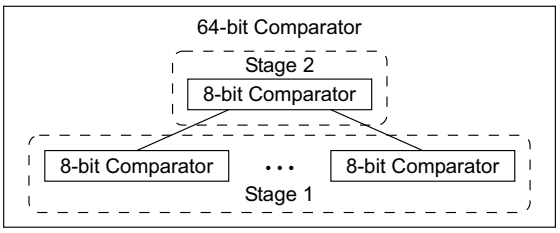

Fig. 9. Example of hierarchical comparator

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 hierarchical comparators to meet a wide variety of speed, power, and area requirements. Additionally, these comparators are pipelined for maximum throughput. Although the required 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 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.

Bits	Duty Cycle
4	0.681
8	0.570
16	0.587
32	0.633
64	0.631

TABLE II
Duty Cycle

VI. Performance Evaluation and Comparison

We simulated our designs in SPICE 3f5 [1] with a 1.2 V 100 nm process [2]. We used effective minimum sized devices for our design. For all the experiments we conducted layout parasitics were not utilized. The results for the low-power and high-speed designs are listed in Tables III and IV, respectively. Additionally, we simulated our comparators with a 5.0 V AMIS 0.5 μm process [3] to enable a fair comparison with the fastest previously reported comparator [4], which was also implemented in a 0.5μm process. These results are listed in Tables V and VI.

In all the experimental results reported in this paper, the delay of our comparator is the **worst-case** *delay (implicitly computed over all possible input transitions).* Also, recall that several signals are computed during the precharge phase in our approach. *The lengthening of the precharge phase due to the computation of these signals is accounted for in all the delay numbers that are reported in this paper.* As a consequence of the computation of signals during the precharge phase, the clock duty cycle[1] may deviate from 50% in our designs. For example, Table II reports the duty cycle ratio utilized for the high speed design, implemented in the 100nm process. Note that the duty cycle does not deviate substantially from 50%. It is well known that generation of clock signal with duty cycle other than 50% is an easy task. Tables IV and VI also report the delay numbers for 50% clock duty cycle.

Consider the comparator widths of 4 and 8. For these widths, the low power comparator has a delay of about 2$\times$ that of the high speed design, with an area utilization of about 60% of the high speed design.

Tables III and IV indicate that the Power-Delay Product (PDP) of the high-speed comparator is comparable to the low power comparator for widths 4 and 8. For higher widths, the low power comparator has a better PDP. However the Energy Delay Product (EDP) of the high speed comparator is consistently superior to that of the low power comparator. Tables V and VI indicate that the PDP and EDP of the low power comparator is better than that of the high-speed comparator.

[1]For this paper, duty cycle is defined as the ratio of the duration of evaluation phase to the total clock period of the design.

Bits	Delay (ps)	Avg. Power (mW)	PDP (fJ)	EDP (fJ.ps)	Area (μm^2)
4	222	0.0941	20.9	4.63×10^3	1.59
8	392	0.0824	32.3	1.26×10^4	3.15
16	900	0.0613	55.2	4.96×10^4	6.15
32	2571	0.0414	106.4	2.73×10^5	12.27
64	8471	0.0253	214.3	1.815×10^6	24.39

TABLE III
100 NM SIMULATION RESULTS FOR LOW-POWER DESIGN

Bits	Delay (ps)	Avg. Power (mW)	PDP (fJ)	EDP (fJ.ps)	Area (μm^2)	0.5 Duty Cycle Delay (ps)
4	116	0.142	16.5	1.91×10^3	2.43	158
8	175	0.200	35.0	6.12×10^3	5.23	199
16	220	0.367	80.7	1.77×10^4	11.43	258
32	300	0.679	203.7	6.11×10^4	25.07	379
64	462	1.251	577.9	2.67×10^5	54.47	583

TABLE IV
100 NM SIMULATION RESULTS FOR HIGH-SPEED DESIGN

Bits	Delay (ps)	Avg. Power (mW)	PDP (fJ)	EDP (fJ.ps)	Area (μm^2)
4	592	3.909	2.31×10^3	1.36×10^6	77.5
8	909	4.463	4.05×10^3	3.68×10^6	153.5
16	1827	4.394	8.02×10^3	1.46×10^7	299.5
32	3192	4.573	1.45×10^4	4.65×10^7	597.5
64	9194	3.725	3.42×10^4	3.14×10^8	1187.5

TABLE V
0.5 μM SIMULATION RESULTS FOR LOW-POWER DESIGN

Bits	Delay (ps)	Avg. Power (mW)	PDP (fJ)	EDP (fJ.ps)	Area (μm^2)	0.5 Duty Cycle Delay (ps)
4	370	13.57	5.02×10^3	1.85×10^6	116.50	380
8	535	19.66	1.05×10^4	5.61×10^6	244.50	607
16	854	28.02	2.39×10^4	2.04×10^7	530.50	1037
32	1455	40.30	5.86×10^4	8.52×10^7	1164.5	1821
64	2458	57.59	1.41×10^5	3.47×10^8	2538.5	2971

TABLE VI
0.5 μM SIMULATION RESULTS FOR HIGH-SPEED DESIGN

The 100 nm simulation results for 64- and 128-bit hierarchical comparators are listed in Tables VII and VIII, respectively. The 0.5 μm results are listed in Tables IX and X. These tables were constructed using the results from Tables III through VI.

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 1.33 ns with the 0.5 μm process. *The fastest previously reported approach (which is implemented in a 0.5μm process) has a latency of 2.12 ns [4]. Therefore, our design is 37% 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. For example, the 64-bit comparator with a first 16-bit high-speed stage, followed by a second 4-bit high-speed stage results in the lowest power consumption among all 64-bit comparator realizations for the 100nm process. In this manner, a designer can select the comparator realization which minimizes the cost function that they seek to optimize.

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 logic to compute the output during the evaluate period and static CMOS logic to do some computation during precharge. These designs may be combined and pipelined to meet a variety of speed, power, and area requirements. We compared our

Stage 1	Stage 2	Latency (ps)	Avg. Power (mW)	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 VII

100 NM 64-BIT HIERARCHICAL COMPARATORS

Stage 1	Stage 2	Latency (ps)	Avg. Power (mW)	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 VIII

100 NM 128-BIT HIERARCHICAL COMPARATORS

Stage 1	Stage 2	Latency (ps)	Avg. Power (mW)	Area (μm^2)
8-bit High Speed	8-bit High Speed	1328	176.9	2200.5
16-bit High Speed	4-bit Low Power	1968	115.9	2199.5
4-bit Low Power	16-bit High Speed	1968	90.6	1770.5
16-bit High Speed	4-bit High Speed	1968	125.6	2238.5
4-bit High Speed	16-bit High Speed	1968	245.1	2394.5

TABLE IX

0.5 μM 64-BIT HIERARCHICAL COMPARATORS

Stage 1	Stage 2	Latency (ps)	Avg. Power (mW)	Area (μm^2)
16-bit High Speed	8-bit High Speed	1968	243.8	4488.5
8-bit High Speed	16-bit High Speed	1968	342.5	4442.5
16-bit High Speed	8-bit Low Power	2078	228.6	4397.5
8-bit Low Power	16-bit High Speed	2078	99.4	2986.5

TABLE X

0.5 μM 128-BIT HIERARCHICAL COMPARATORS

design with the fastest previously reported approach, and our fastest design is about 37% faster than the best previously reported approach [4].

REFERENCES

[1] L. Nagel, "SPICE: A computer program to simulate computer circuits," in *University of California, Berkeley UCB/ERL Memo M520*, May 1995.

[2] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, and C. Hu, "New paradigm of predictive MOSFET and interconnect modeling for early circuit design," in *Proc. of IEEE Custom Integrated Circuit Conference*, pp. 201–204, Jun 2000. http://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, "1GHz 64-bit high-speed comparator using ANT dynamic logic with two-phase clocking," *IEE Procceedings 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.