ISSN 23499842(Online), Volume 1, Special Issue 1(NCEE'15), May 2015

International Journal of Innovative Trends and Emerging Technologies

HIGH SPEED DIGITAL CMOS COMPARATOR USING A PARALLEL PREFIX TREE

PRIYADHARSHINI.G[1] AND RAJENDRAN.S[2]
[1]PG student, M.E VLSI design, [2]Professor, ECE,
[1, 2] Sri Ramanujar Engineering college
[1] Email: priyadharshini.srkvg@mail.com

Abstract: We present a new comparator design featuring wide-range and high-speed operation using only conventional digital CMOS cells. Our comparator exploits a novel scalable parallel prefix structure that leverages the comparison outcome of the most significant bit, proceeding bitwise toward the least significant bit only when the compared bits are equal. This method reduces dynamic power dissipation by eliminating unnecessary transitions in a parallel prefix structure that generate the *N*-bit comparison result after $\log_4 N + \log_{16} N + 4$CMOS gate delays. Our comparator is composed of locally interconnected CMOS gates with a maximum fan-in and fan-out of five and four, respectively, independent of the comparator bit width. The main advantages of our design are high speed and power efficiency, maintained over a wide range. GDI (Gate diffusion input) and 0.65μm technology is used to design transistors.

***KEY WORDS*: High-speed arithmetic, high-speed wide-bit comparator architecture, parallel prefix tree structure.**

I. INTRODUCTION

Comparators are key design elements for a wide range of applications—scientific computation (graphics and image/signal processing [1]–[3]), test circuit applications(jitter measurements, signature analyzers, and built-in self-test circuits [4], [5]), and optimized equality-only comparators for general-purpose processor components (associative memories, load-store queue buffers, translation look-aside buffers, branch target buffers, and many other CPU argument comparison blocks [6]–[8]). Even though comparator logic design is straightforward, the extensive use of comparators in high-performance systems places a great importance on performance and power consumption optimizations. Some state-of-the-art comparator designs use dynamic gate logic circuit structures to enhance performance, while others leverage specialized arithmetic units for wide comparisons, along with custom logic circuits.

For example, Manuscript received January 5, 2012; revised July 16, 2012; accepted September 13, 2012. Date of publication December 3, 2012. date of current version September 23, 2013. This work was supported in part by the U.S. National Science Foundation under Grant CNS-0953447.

S. Abdel-Hafeez is with the Jordan University of Science and Technology, Irbid 22110, Jordan (e-mail: sabdel_99@yahoo.com).

Gordon-Ross is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: ann@chrec.org).

Parhami is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560 USA (e-mail: parhami@ece.ucsb.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org. Digital Object Identifier 10.1109/TVLSI.2012.2222453

Several prior designs [9]–[13] use subtracts in the form of flat adder components, but these designs are typically slow and area-intensive, even when implemented using fast adders [14]–[16]. Other comparator designs improve scalability and reduce comparison delays using a hierarchical prefix tree structure composed of 2-b comparators [17]. These structures require log2 N comparison levels, with each level consisting of several cascaded logic gates. However, the delay and area of these designs may be prohibitive for comparing wide operands.

The prefix tree structure's area and power consumption can be improved by leveraging two-input multiplexers (instead of 2-b comparator cells) at each level and generate-propagate logic cells on the first level (instead of 2-b adder cells), which takes advantage of one's complement addition [18]. Using this logic composition, a prefix tree requires six levels for the most common comparison bit width of 64 bits, but suffers from high power consumption due to every cell in the structure being active, regardless of the input operands' values. Furthermore, the structure can perform only "greater-than" or "less-than" comparisons and not equality.

To improve the speed and reduce power consumption, several designs rely on pipelining and power-down mechanisms [19] to reduce switching activity [20], [21] with respect to the actual input operands' bit values. One design uses all-N-transistor (ANT) circuits to compensate for high fan-in with high pipeline throughput [22]. A 64-b comparator requires only three pipeline cycles using a multiphase clocking scheme [23]. However, such a clocking scheme may be unsuitable for high-speed single-cycle processors because of several heavily loaded global clock signals that have high-power transition activity. Additionally, race conditions and a heavily con-strained clock jitter margin may make this design unsuitable for wide-range comparators.

An alternative architecture leverages priority-encoder magnitude decision logic with two pipelined operations that are triggered at both the falling and rising clock edges [24] to improve operating speed and eliminate long dynamic logic chains. However, 64-b and wider comparators require a multilevel cascade structure, with each logic level consisting of seven nMOS transistors connected in series that behave in saturating mode during operation. This structure leads to a large overall conductive resistance [16], with heavily loaded parasitic components on the clock signal, which severely limits the clock speed and jitter margin.

Other architectures use a multiplexer-based structure to split a 64-b comparator into two comparator stages [25]: the first stage consists of eight modules performing 8-b comparisons and the modules' outputs are input into a priority encoder and the second stage uses an 8-to-1 multiplexer to select the appropriate result from the eight modules in the first stage. This architecture uses two-phase domino clocking [14], [23], [26] to perform both stages in a single clock cycle. Since operations occur on the rising and falling clock edges, this further limits the operating speed and jitter margin and makes the design highly susceptible to race conditions [27].

Some comparators combine a tree structure with a two-phase domino clocking structure [28] for speed enhancement. These architectures add the two inputs, after negating one input via two's complement, using the carry-out signal as the “greater-than” or “less-than” indicator (equality is not supported). Since the critical signal is the carry-out, the tree structure's adder modules are optimized to compute only the carry signal. Because the adder module is implemented using a Manchester carry chain [19], this architecture reduces the tree structure's area, power consumption, and comparison delay. However, the heavy loading of the clock signal with 64×2 gates for the pre-charge and evaluate phases complicates routing, constrains the long clock cycle required for two-phase clocking, and necessitates large drivers for the clock signals.

Some architectures save power by dynamically eliminating unnecessary computations using novel ripple-based structures, such as those incorporating wide-range ripple-carry adders [29]–[31]. Similarly, other energy-efficient designs [32]–[34] leverage schemes to reduce switching activity. Compute-on-demand comparators compare two binary numbers one bit at a time, rippling from the most significant bit (MSB) to the least significant bit (LSB). The outcome of each bit comparison either enables the comparison of the next bit if the bits are equal, or represents the final comparison decision if the bits are different. Thus, a comparison cell is activated only if all bits of greater significance are equal. Although these designs reduce switching, they suffer from long worst case comparison delays for wide worst case operands.

To reduce the long delays suffered by bitwise ripple designs, an enhanced architecture incorporates an algorithm that uses no arithmetic operations. This scheme [35] detects the larger operand by determining which operand possesses the leftmost 1 bit after pre-encoding, before supplying the operands to bitwise competition logic (BCL) structure. The BCL structure partitions the operands into 8-b blocks and the result for each block is input into a multiplexer to determine the final comparison decision. Due to this BCL-based design's low transistor count, this design has the potential for low power consumption, but the pre-encoder logic modules preceding the BCL modules limit the maximum achievable operating frequency. In addition, special control logic is needed to enable the BCL units to switch dynamically in a synchronized fashion, thus increasing the power consumption and reducing the operating frequency.

II.GATE DIFFUSION INPUT TECHNIQUE

Gate Diffusion Input technique is a new low power design style. This allows implementing complex logic functions using only two transistors. This technique results in reducing area, power consumption, propagation delay and number of devices. This GDI technique is best suitable for fast implementation and low power applications of combinatorial circuits. Even though GDI cell looks like CMOS inverter it differs from the inverter in the following way.

1. The GDI Cell contains three inputs: The Common Gate Input of nMOS and pMOS (G), input to source/drain of nMOS (N)
and input to source/drain of pMOS (P).

2. To arbitrarily bias the Bulks of both nMOS and pMOS are connected to N and P respectively.

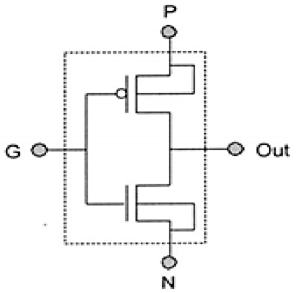


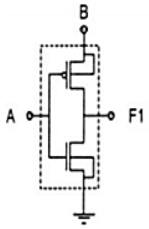
Fig.1. Basic GDI cell

The fastest comparators are made up of full combinatorial circuits. To reduce the switching activity of internal signals and to provide the parallelization parallel prefix tree structure is used. By implanting the design using static CMOS cells it results in large transistor count so that the area of the design in increased, by implementing using Gate Diffusion Input Cells power and transistor count, is reduced which result less area.

Gate Diffusion Input Technique which is based on a Shannon expansion is analyzed for minimizing the power consumption and delay of static digital circuits. This technique as compare to other currently used logic design style, allows less power consumption and reduced propagation delay for low-power design of combinatorial digital circuits with minimum number of transistors.

The most common problem of all design methods is the low swing of output signals due to the threshold drop across the single-channel pass transistors. Generally to overcome this problem, additional buffer circuit is used. In GDI cell, the effects of low swing problem can be understood by operational analysis of F1 function and it can be easily extend to other functions of GDI cell. Table 2 shows a full set of logic states and their related functionality modes of F1.

Table 1: Input Logic States versus Functionality F1



A	B	Functionality	F1
0	0	*pMOSTrans Gate*	V_{Tp}
0	1	*CMOSInverter*	1
1	0	*nMOS Trans Gate*	0
1	1	*CMOSInverter*	0

From the table, it can be seen that in half of the cases (B=1), the GDI cell operates as a regular CMOS inverter, which is widely used as a digital buffer for logic-level restoration. In the cases, when V_{dd}=1, without a swing drop from the previous stages, a GDI cell works as an inverter buffer and recovers the voltage swing but the only state where low swing occurs in the output value is A = 0, B= 0. In this case, the voltage level of F1 is VTp instead of expected 0 volt because of the poor high-to-low transition characteristics of the pMOS transistor. Among all the possible transitions, the only case where the effect of low swing occurs is the transition from A = 0, B= Vdd to A=0, B=0. The GDI cell allows a self-swing restoration in certain cases, but the worst case is also assumed in this analysis and additional circuitry is used for swing restoration in the implemented circuits.

To alleviate some of the drawbacks of previous designs (such as high power consumption, multi cycle computation, custom structures unsuitable for continued technology scaling, long time to market due to irregular VLSI structures, and irregular transistor geometry sizes), in this paper we leverage standard CMOS cells to architect fast, scalable, wide-range, and power-efficient algorithmic comparators with the following key features.

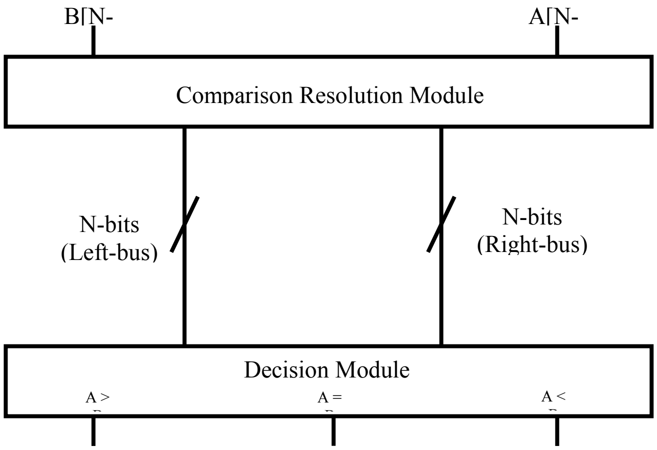


Fig 2. Block diagram of our comparator architecture, consisting of a comparison resolution module connected to a decision module.

1) Use of reconfigurable arithmetic algorithms, with total (input-to-output) hardware realization for both fully-custom and standard-cell approaches, improves the longevity of our design and makes our design ideal for technology scaling and short time to market.

2) A novel MSB-to-LSB parallel-prefix tree structure, based on a reduced switching paradigm and using parallelism at each level (as opposed to a sequential approach [32]), contributes to the speed and energy efficiency of our design.
3) Use of components built from simple single-gate-level logic, with maximum fan-in and fan-out of five and four, respectively, regardless of the comparator bit width, makes it easy to characterize and accurately model our comparator for arbitrary bit widths.
4) Use of combinatorial logic, with neither clock gating nor latency delay, enables global partitioning into two main pipelined stages or locally into several pipelined stages based on the number of levels. This flexibility provides area versus performance tradeoffs.
5) The remainder of this paper is organized as follows. Section II covers our comparator's operating principles and overall structure and Section III provides the design details. Section IV evaluates the area, operating speed, and power consumption of our comparator. Performance analysis and simulation results for input widths ranging from 16 to 256 bits, along with generalization to *N*-bit inputs, appear in Section V. Concluding remarks and suggestions for further work are provided in Section VI.

III. COMPARATOR ARCHITECTURAL OVERVIEW

The comparison resolution module in Fig. 2 (which depicts the high-level architecture of our proposed design) is a novel MSB-to-LSB parallel-prefix tree structure that performs bit-wise comparison of two N-bit operands A and B, denoted as A_{N-1}, A_{N-2}, . . ., A0 and B_{N-1}, B_{N-2}, . . ., B_0, where the subscripts range from N–1 for the MSB to 0 for the LSB. The comparison resolution module performs the bitwise comparison asynchronously from left to right, such that the comparison logic's computation is triggered only if all bits of greater significance are equal.

The parallel structure encodes the bitwise comparison results into two N-bit buses, the left bus and the right bus,

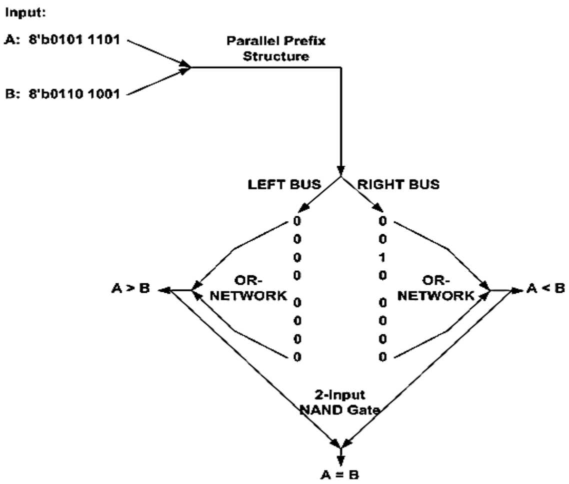


Fig 3. Example 8-b comparison

Table 2: Symbol notation and definitions

Symbol (Cells)	Definition
N	Operand bit width
A	First input operand
B	Second input operand
R	Right bus result bit
L	Left bus result bit
II	Bit wise AND
∑	Bit wise OR
T {*}	Logic function of cell type *
COMP {*}	Complement function of set *

Each of which store the partial comparison result as each bit position is evaluated, such that

$$\text{if } A_k > B_k, \quad \text{then left}_k = 1 \quad \text{and right}_k = 0$$
$$\text{if } A_k < B_k, \quad \text{then left}_k = 0 \quad \text{and right}_k = 1$$
$$\text{if } A_k = B_k, \quad \text{then left}_k = 0 \quad \text{andright}_k = 0.$$

In addition, to reduce switching activities, as soon as a bitwise comparison is not equal, the bitwise comparison of every bit of lower significance is terminated and all such positions are set to zero on both buses, thus, there is never more than one high bit on either bus.

The decision module uses two OR-networks to output the final comparison decision based on separate OR-scans of all of the bits on the left bus (producing the *L* bit) and all of the bits on the right bus (producing the *R* bit). If LR = 00, then *A* = *B*, if LR=10 then *A* > *B*, if

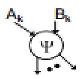
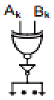
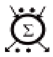
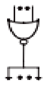
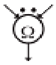
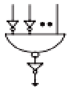
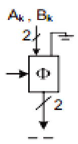
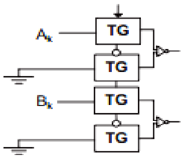
LR=01 then *A* < *B*, and LR = 11 is not possible.

An 8-b comparison of input operands *A*= 01011101 and *B*=01101001 is illustrated in Fig. 3. In the first step, a parallel prefix tree structure generates the encoded data on the left bus and right bus for each pair of corresponding bits from *A* and *B*.

In this example, A7=0 and B7=0 encodes asleft7= right7= 0, A6= 1, and B6= 1 encodes as left6= right6= 0, and A5= 0 and B5= 1 encodes left5= 0 and right5= 1. At this point, since the bits are unequal, the comparison terminates and a final comparison decision can be made based on the first three bits evaluated.

The parallel prefix structure forces all bits of lesser significance on each bus to 0, regardless of the remaining bit values in the operands. In the second step, the OR-networks perform the bus OR-scans, resulting in 0 and 1, respectively, and the final comparison decision is A>B.

Table 3: Logic gate representations for symbols used in fig. 3

Symbols (Cells)	Logic Gate	Maximum Fan-in/Fan-out And (Transistor Counts)
		2 / 4 (12)
		4 / 4 (8)
		5 / 1 (20)
 MUX-Logic	 TG: Transmission Gate	3 / 2 (12)

We partition the structure into five hierarchical prefixing sets, as depicted in Fig. 4, with the associated symbol representations in Tables 2 and 3, where each set performs a specific function whose output serves as input to the next set, until the fifth set produces the output on the left bus and the right bus. All cells (components) within each set operate in parallel, which is a key feature to increase operating speed while minimizing the transitions to a minimal set of left-most bits needed for a correct decision.

This prefixing set structure bounds the components' fan-in and fan-out regardless of comparator bit width and eliminates heavily loaded global signals with parasitic components, thus improving the operating speed and reducing power consumption. Additionally, the OR-network's fan-in and fan-out is limited by partitioning the buses into 4-b groupings of the input operands, thus reducing the capacitive load of each bus

Each set or group of cells produces outputs that serve as inputs to the next set in the hierarchy, with the exception of set 1, whose outputs serve as inputs to several sets. Set 1 - Compares the N-bit operands A and B bit-by-bit, using a single level of N ψ-type cells. The ψ-type cells provide a termination flag D_k to cells in sets 2 and 4, indicating whether the computation should terminate. These cells compute (where $0 \leq k \leq N - 1$)

$$\Psi : D_k = A_k \oplus B_k. \qquad (1)$$

Set 2 - Consists of $\sum_2$ - type cells, which combine the termination flags for each of the four Ψ - type cells from set 1 (each $\sum_2$ - type cell combines the termination flags of one 4-b partition) using NOR-logic to limit the fan-in and fan-out to a maximum of four. The $\sum_2$ - type cells either continue the comparison for bits of lesser. significance if all four inputs are 0s, or terminate the comparison if a final decision can be made. For $0 \leq m \leq N/4 - 1$, there is a total of N/4 $\sum_2$ -type cells, all functioning in parallel

$$\sum\nolimits_2 : C_{2,m} = \mathrm{COMP}\left(\sum_{i=4m}^{4m+3} D_i\right). \qquad (2)$$

Set 3 - Consists of $\sum_3$ - type cells, which are similar to $\sum_2$ - type cells, but can have more logic levels, different inputs, and carry different triggering points. A$\sum_3$ - type cell provides no comparison functionality; the cell's sole purpose is to limit the fan-in and fan-out regardless of operand bit width. To limit the $\sum_3$ - type cell's local interconnect to four, the number of levels in set 3 increases if the fan-in exceeds four. Set 3 provides functionality similar to set 2 using the same NOR logic to continue or terminate the bitwise comparison

IV. COMPARATOR DESIGN DETAILS

In this section, we detail our comparator's design (Fig. 4), which is based on using a novel parallel prefix tree (Tables 2 and 3 contain symbols and definitions).

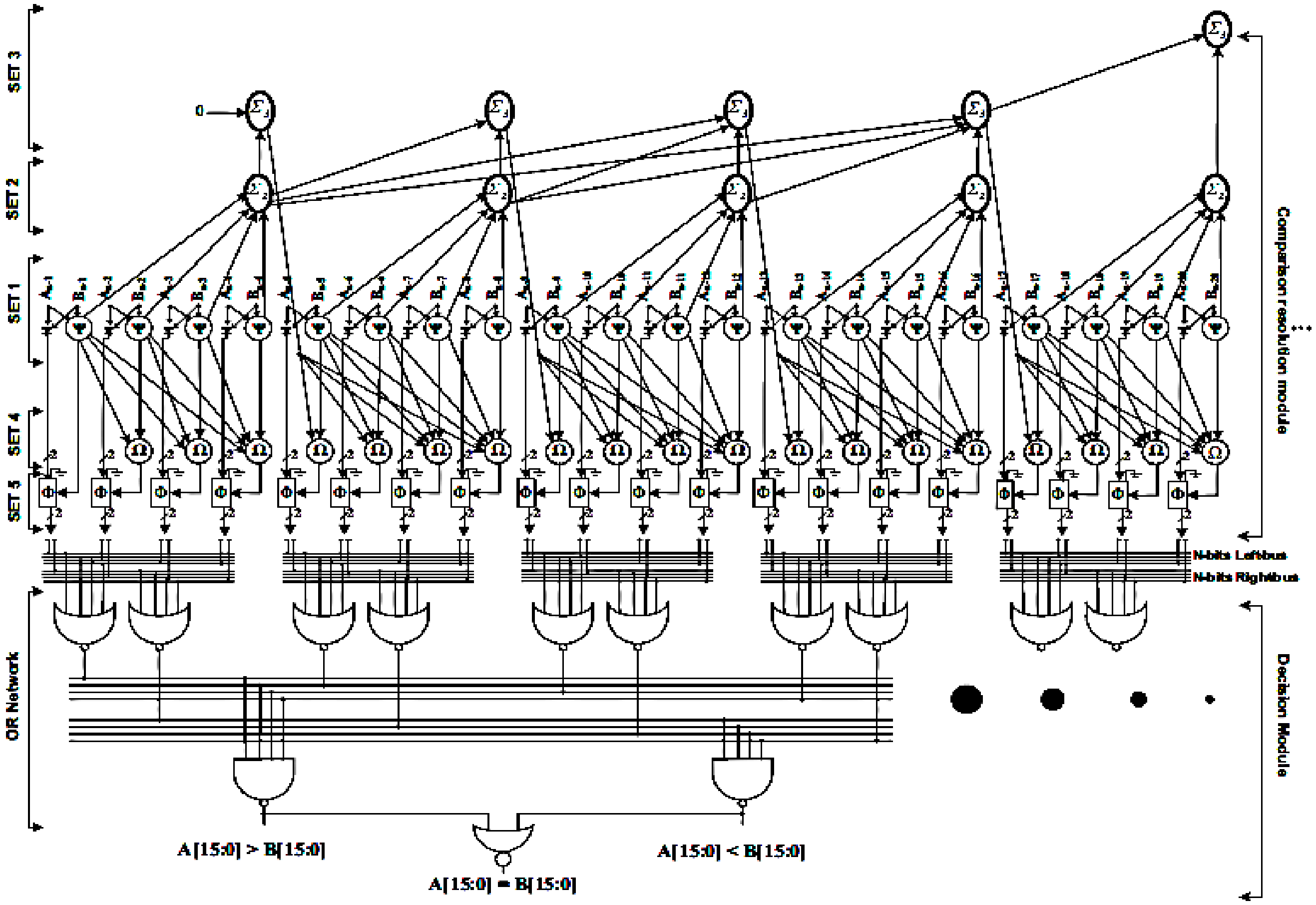


Fig.4 Implementation details for the comparison resolution module (sets 1 through 5) and the decision module

activity. If the comparison is terminated, set 3 signals set 4 to set the left bus and right bus bits to 0 for all bits of lower significance. For 0 ≤ m ≤ N/4 − 1, there is a total of N/4 $\sum_3$ - type cells per level, with cell function and number of levels as

$$\sum\nolimits_3 : C_{3,m} = \mathrm{COMP}\left(\sum_{0}^{m} C_{2,i}\right) \qquad (3)$$

$$\mathrm{Levels}_{\mathrm{set3}} = \left(\lceil \log_{16}(N) \rceil\right). \qquad (4)$$

From left to right, the first four $\sum_3$ - type cells in set 3 combine the 4 - b partition comparison outcomes from the one, two, three, and four 4-b partitions of set 2. Since the fourth $\sum_3$ - type cell has a fan-in of four, the number of levels in set 3 increases and set 3's fifth $\sum_3$ - type cell combines the comparison outcomes of the first 16 MSBs with a fan-in of only two and a fan-out of one.

Table 4: outcome of Ω - type cells in set 4 for a 16-bit comparison

$\Re$-Type Cell	Input Driving $\Re$-Type Cell Output
Y_{15}	D_{15}
Y_{14}	$\overline{D}_{15}D_{14}$
Y_{13}	$\overline{D}_{15}\overline{D}_{14}D_{13}$
Y_{12}	$\overline{D}_{15}\overline{D}_{14}\overline{D}_{13}D_{12}$
Y_{11}	$C_{3,0}D_{11}$
Y_{10}	$C_{3,0}\overline{D}_{11}D_{10}$
Y_9	$C_{3,0}\overline{D}_{11}\overline{D}_{10}D_9$
Y_8	$C_{3,0}\overline{D}_{11}\overline{D}_{10}\overline{D}_9D_8$
Y_7	$C_{3,1}D_7$
Y_6	$C_{3,1}\overline{D}_7D_6$
Y_5	$C_{3,1}\overline{D}_7\overline{D}_6D_5$
Y_4	$C_{3,1}\overline{D}_7\overline{D}_6\overline{D}_5D_4$
Y_3	$C_{3,2}D_3$
Y_2	$C_{3,2}\overline{D}_3D_2$
Y_1	$C_{3,2}\overline{D}_3\overline{D}_2D_1$
Y_0	$C_{3,2}\overline{D}_3\overline{D}_2\overline{D}_1D_0$

Set 4 consists of Ω-type cells, whose outputs control the select inputs of Ω -type cells (two-input multiplexors) in set 5, which in turn drive both the left bus and the right bus. For an Ω -type cell and the 4-b partition to which the cell belongs, bitwise comparison outcomes from set 1 provide information about the more significant bits in the cell's Ω-type cells, which compute ($0 \le k \le N-1$)

$$\Omega : Y_k = C_{3,\lfloor k/4\rfloor -1} D_k \prod_{i=4\lfloor K/4\rfloor -1}^{k-1} \overline{D}_i. \qquad (5)$$

The number of inputs in the Ω -type cells increases from left to right in each partition, ending with a fan-in of five. Thus, the Ω -type cells in set 4 determine whether set 5 propagates the bitwise comparison codes. Table 4 shows a sample 16-b comparison to clarify (5) using (1) – (4).

Set 5 consists of N φ -type cells (two-input, 2-b-wide multiplexers). One input is (A_k, B_k) and the other is hardwired to "00." The select control input is based on the Ω -type cell output from set 4. We define the 2-b as the left-bit code (Ak) and the right-bit code (Bk), where all left-bit codes and all right-bit codes combine to form the left bus and the right bus, respectively. The φ -type cells compute (where $0 \le k \le N - 1$)

$$\Phi : F_k^{1,0} = Y_k \times M_k + \overline{Y}_k \times (00). \qquad (6)$$

The output $F_k{}^{1,0}$ denotes the "greater-than," "less-than," or "equal to" final comparison decision

$$F_k^{1,0} \begin{cases} 00, & \text{for } A_k = B_k \\ 01, & \text{for } A_k < B_k \\ 10, & \text{for } A_k > B_k. \end{cases} \qquad (7)$$

Essentially, the 2-b code $F_k{}^{1,0}$ can be realized by OR-ing all left bit and all right bit separately, as shown in the decision module (Figs. 3 & 4), using an OR-gate network in the form of NOR-NAND gates yielding a more optimum gate structure

$$GL_{1,j}^{1} = \sum_{k=4j}^{4j+3} F_k^1 \qquad (8)$$

$$GR_{(1,j)}^{0} = \sum_{k=4j}^{4j+3} F_k^0. \qquad (9)$$

The superscripts "1" and "0" in (8) and (9) denote the summation of the left and right bits, respectively, and the subscript "1" denotes the first level of OR-logic in the decision module that receives data directly from set 5. If we limit the fan-in of each gate to four, the number L_{DM} of the OR- gate tree levels for the decision module is given by

$$L_{\text{DM}} = \lceil \log_4 N \rceil. \qquad (10)$$

V. AREA, SPEED AND POWER EVALUATION

A. Area Analysis

We begin by deriving the total number of cells required and use Table 5 to translate the cell counts into transistors for an N-bit comparator. Based on (1)–(10), the number of C_{CRM} cells required for the comparison resolution module and the numbers of C_{DM} cells in the decision module is, respectively

$$\begin{aligned} C_{\text{CRM}} = (N \times \Psi) + \left(\frac{N}{4} \times \sum\nolimits_2\right) \\ + \left(\lceil \log_{16} \rceil (N) \times \frac{N}{4} \times \sum\nolimits_3\right) \\ + (N \times \Omega) + (N \times \Phi) \end{aligned} \qquad (11)$$

$$C_{\text{DM}} = 2\left(\sum_{k=1}^{\lceil \log_4 N\rceil} \left\lceil \frac{N^k}{4} \right\rceil\right) \times \text{NOR-NAND}. \qquad (12)$$

Table 5 shows the total number of cells and the required number of levels per set for various comparator bit widths based on (11) and (12).

The cell counts in Table 5, along with the number of transistors per cell type (Table 2), allow us to derive the total number of transistors for various bit widths (Table 6). The results show an approximate linear growth in comparator size as a function of bit width.

B. Operating Speed

We analyze the critical path delay of our proposed comparator with N-bit inputs. The delay D_{CRM} for the comparison resolution module is

$$D_{\mathrm{CRM}} = D_{\mathrm{set1}} + D_{\mathrm{set2}} + D_{\mathrm{set3}} + D_{\mathrm{set4}} + D_{\mathrm{set5}}. \qquad (13)$$

Table 5: Total number of cells and circuit levels in each set for various comparator bit widths

Comparator Bit width	Set 1		Set 2		Set 3		Set 4		Set 5	
	Cells	Levels	Cells	Levels	Cells	Levels	Cells	Levels	Cells	Levels
16-b	16	1	42	1	43	1	16	1	16	1
32-b	32	1	82	1	83	1	32	1	32	1
64-b	64	1	162	1	163	1	64	1	64	1
128-b	128	1	322	1	323	2	128	1	128	1
256-b	256	1	642	1	643	2	256	1	256	1

Table 6: Total number of transistors for various comparator bit widths

Comparator Bit width	**Transistor Counts**					
	Set 1	**Set 2**	**Set 3**	**Set 4**	**Set 5**	**Total**
16-b	16 × 12	4 × 8	4 × 8	16 × 20	16 × 12	768
32-b	32 × 12	8 × 8	4 × 8	32 × 20	32 × 12	1424
64-b	64 × 12	16 × 8	4 × 8	64 × 20	64 × 12	2976
128-b	128 × 12	32 × 8	8 × 8	128 × 20	128 × 12	5952
256-b	256 × 12	64 × 8	8 × 8	256 × 20	256 × 12	11 840

All terms, except the third, on the right-hand side of (13) entail a single gate delay DU, resulting in

$$\begin{aligned} D_{\mathrm{CRM}} &= D_U + D_U + \left(\lceil \log_{16}(N) \rceil\right) D_U + D_U + D_U \\ &= 4D_U + \left(\lceil \log_{16}(N) \rceil\right) D_U. \end{aligned} \qquad (14)$$

The delay D_{DM} for the decision module's NOR-NAND gate network is

$$D_{\mathrm{DM}} = \log_4(N) D_U. \qquad (15)$$

The total (asynchronous) comparator delay D_T from input to output for an N-bit comparator is

$$D_T = 4D_U + \left(\lceil \log_{16}(N) \rceil\right) D_U + \left(\lceil \log_4(N) \rceil\right) D_U. \quad (16)$$

To the best of our knowledge, the total delay of (16) puts our design among the fastest comparators reported in the literature based on a basic CMOS gate circuit without any circuit level modifications. Detailed simulation-based comparisons will be provided in Section VI

C. Power Requirements

Minimizing the switching activity reduces the average power dissipation and is considered a key enabling technique
for modern low-power design [29]–[35]. In this subsection, we assess the impact of this method on power dissipation in our comparator design.

The operands activate all cells in set 1 in parallel, thus set 1provides no power savings. Table 6 shows that set 1 account for 25% of the total transistors, and thus power dissipation, for an arbitrary comparator size.

The cells of each partition in set 2 are selectively activated in parallel (except for the most significant partition, which is always active) if the previous partition's set 1 provides no comparison decision. However, to preserve parallelism and ensure high operating speed, set 2 does not limit activity to only one cell, and accounts for 4.2% of the transistor switching activity due to set 2's share of the total transistor count.

A partition in set 3, which is comprised of multilevel NOR-logic gates, is activated only if all bits of greater significance are equal. Thus, if the bitwise comparison is equal for all cells in set 1, a comparison request is sent to the next lower cells in set 1, a comparison request is sent to the next lower significant bit in set 3, otherwise, no gate activity occurs at this level. Set 3 achieves significant power savings, because this level. Set 3 achieves significant power savings, because final comparison decision, with only one cell per level being active. Table 6 shows that set 3 accounts for only 1.1% of the total switching activity. Set 4 combines the results of set 1 and the single active cell in set 3, which incorporates the comparison outcomes of all more significant sets to activate the cell at this bitwise position if all MSBs are unequal. Therefore, only one cell in set 4 is active, leading to a significant reduction in power dissipation. Table 6 shows that set 4 accounts for 41.6% of the total transistors for an arbitrary comparator size, but since only one cell in set 4 is active, set 4 only accounts for 2.6% of the total transistor switching activity, with this share decreasing as comparator bit width increases.

The single activated cell in set 4 triggers the multiplexer circuit in set 5 and provides an additional reduction in power consumption. Set 5 accounts for only 1.56% of the total transistor switching activity, with this share decreasing for wider comparators.

Our comparator's worst case cell activities occur when A = 00…01 and B = 00…00 (or vice versa) and Fig. 5 depicts the number of transitions versus comparator bit width. For each comparator bit width, the first bar shows the total number of transistors and the second bar shows the number of active transistors. We note that for all comparator bit widths, less than half of the transistors are active, making the power dissipation roughly one-third of the value if all of the transistors were

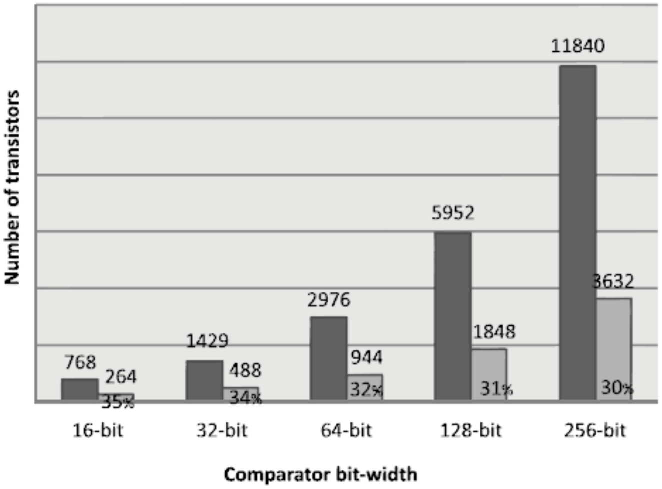


Fig. 5. Total number of transistors (dark shading) and number of active transistors (light shading) for various comparator bit widths. Percentages cited refer to the fraction of active transistors.

Table 8: Leakage power for our proposed comparator with 64 bits at different technology node factors measured at first-first corner and a temperature of 100^{O}C

	0.18 µm 1.95 V	0.15 µm 1.65 V	0.13 µm 1.5 V	0.09 µm 1 V
NAND CMOS 4 Transistors	11.583 nW	33.33 nW	657.3 nW	984.2 nW

Our design is thus competitive with other low-power comparators while offering the additional advantages of high-speed operation and scalability.

As technology scales further, the contribution of leakage current to the overall power consumption increases. Given that our design operates at the threshold voltage level and considering that dynamic power consumption has been reduced through circuit techniques, leakage power could become dominant (especially since every circuit component, not only the active components, contribute to the total leakage), thus overshadowing the savings achieved in dynamic power consumption via reduced activity. The worst leakage power is usually measured at the fast-fast corner with a severe temperature of 100 °C [37], [38] for a single NAND gate that is built using four CMOS transistors, as depicted in Table 5, for different technology node factors. Table 8 shows the results of HSPICE simulations for our proposed comparator with 64-b and reveals a leakage contribution of only 0.6%, 1.7%, and 4.3% with respect to the total power at 0.15 μm, 0.13 μm, and 90 nm, respectively, as compared to Table 6. This nominal increase in leakage power percentage is due to our design's small sizes and local cell interconnects with very limited fan-out and fan-in as well as the absence of global routing and ratioed dynamic sizes, and therefore, leakage power will not impact our power-saving method in near-future technologies.

The average power consumption values are significantly better, given that when the probability of reaching a decision at each bit position is 50%, the expected number of positions examined before reaching a decision is only two.

Table 7: Leakage power for CMOS NAND with four transistore at different technology node factors measured at first-first corner and a temperature of 100^{O}C

	0.18 µm 1.95 V	0.15 µm 1.65 V	0.13 µm 1.5 V	0.09 µm 1 V
64-b comparator 4000 transistor	0.0116 nW	0.0534 nW	0.626 nW	0.8619 nW

VI. SIMULATION – BASED COMPARISONS

To evaluate the functionality and performance of our comparator, we simulated the complete design with various inputs using the HSPICE simulator [39] with 0.15 μm-TSMC digital CMOS technology [40] for slow-slow corner (1.35 V at 125 °C). The worst case delay was evaluated by activating the maximum number of cells, including all the least significant cells (i.e., all input operand bits were equal, except at the least significant position). We limited the N-type transistor width to 2 μm and enlarged the P-type transistor width to a maximum of 5 μm, since all cells were locally interconnected and there were no global signals that required a large driver.

Since our key objective was to maximize the operating speed, both transistor types were chosen to have the minimum channel length (i.e., 0.15 μm), given the lack of restriction on the channel length modulation for our design. The maximum measured cell delay was 0.0847 ns for the Ω -type cell with a maximum fan-in of five and a maximum fan-out of one, as suggested by Table 2.

We evaluated our comparator against several state-of-the-art implementations, whose structures represent recently proposed topologies and circuits targeted for high-speed operation and power savings (i.e., objectives similar to ours). Simulation results for our 64-b comparator and reported results for several other comparators [25], [28], [32], [35], [41] are shown in Table 9. The maximum total input-to-output delay (in nanoseconds) versus input bit width for our comparator is shown in Fig. 6. The simulation results closely match the analytical model in Table 6, showing that the number of gate levels increases at ($\log_4 N$) + ($\log_{16} N$) + 4.

Independent of technology scaling, our comparator offers a 40% speed advantage over the design in [28], whose number of levels increases at ($\log_4 N$)+ two's complement_, with each level comprising of approximately three cascaded gates. Furthermore, the Cadence data sheet reported in [28] and [41] show that the design used 14 cascaded gates with a fan-out of four for a 64-b comparator, which operates at a slower speed as compared to our design that uses eight cascaded gates with a maximum fan-out of four. Additionally, for comparators wider than 64 bits in our design, the nonlinearity in the growth rate of the number of levels becomes less significant, as evident from Fig. 6. This is due to the second-order effect of logarithmic scaling for large parameter values [4], [16].

Fig. 7 shows the maximum power dissipation versus the number of bits that must be evaluated to reach a decision for a 64-b comparator based on our design operating at 1 GHz. For example, if the two input operands have the values 11111… 1 and 01111…1, only one bit needs to be evaluated for the Comparison decision. As expected, the power dissipation for our comparator is always higher than that in [32], which uses one logic level per cell to evaluate each bit sequentially, thereby trading off operating speed for low power. We also observed that our comparator dissipates more leakage power than all of the alternate comparator designs due to a larger number of transistors. Taking into consideration that leakage power is on the order of neon watts, while our savings is mainly with respect to dynamic activity, which is on the order of mill watts, the disadvantage is not critical. Essentially, our design trades low-order leakage for the cost of high-order dynamic activities and high operating speed.

Table 9: Simulation and reported results for various 64-b comparator designs

Comparator Type	Technology / Power Supply	Transistor Count	Power Dissipation	Delay (ns)	Notes on Properties
Proposed (static type)	0.15 μm/1.5 V	4000 (64-b)	7.76 mW@1 GHz (64-b)	0.86 (64-b)	1) High transistor count
Hensley *et al.* [32] (static type)	0.18 μm /1.8 V	624 (24-b)	5.23 mW@100 MHz (24-b) 0.735 µW/MHz	4.16 (24-b)	1) Very slow
Perri*et al*. [28] (static type)	0.35 μm/3.3 V	1960 (64-b)	24 μW/MHz (64-b)	1.73 (64-b)	1) Supports only ">" or "<" 2) Not power efficient for the common case of data dependencies
Lam *et al*. [25]	0.35 μm/3.3 V	3386 (64-b)	14.2 mW@200 MHz 42 µW/MHz	2.82 (64-b)	1) Clock heavily loaded with large number of gated transistors 2) Not power efficient for the common case
Kim *et al*. [35]	0.18 μm/1.8 V	964 (32-b)	2.53 mW@200 MHz 12.65 µW/MHz	1.12 (32-b)	1) Pre-encoder and mux encoder output logic not included in the data measured 2) Dynamic clock is heavily loaded with
Cadence [41]	0.35 μm/3.3 V	2456 (64-b)	17.54 mW@200 MHz 34 µW/MHz	1.93 (64-b)	1) Not power efficient for the common case of data dependencies 2) High power dissipation in tree structure

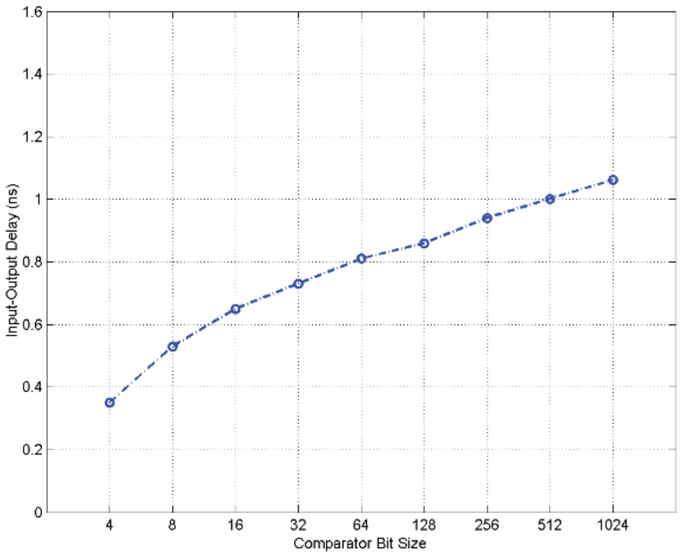


Fig.6. Maximum input-output delay versus input bit width for our proposed comparator design

rdin

ge (

fewer than 28 bits must be evaluated, which is the case with probability very close to 1 for random inputs, our comparator dissipates power at a rate of 0.9 µW / MHz. When the number of evaluated bits is greater than 32,

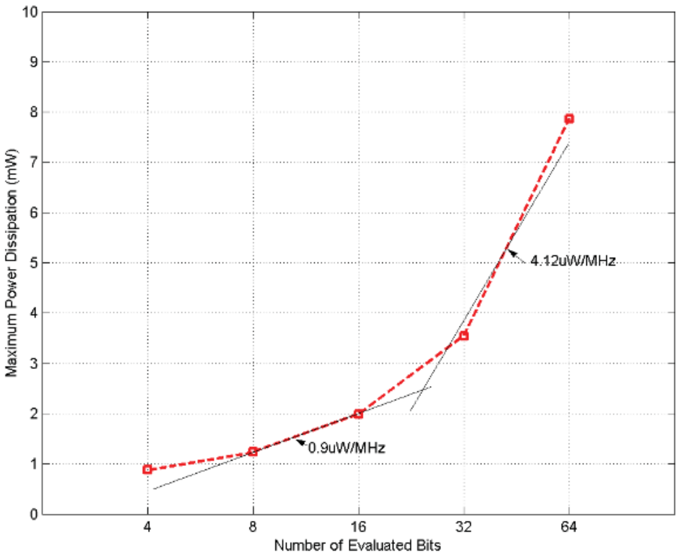


Fig.7. Maximum power dissipation versus number of bits that must be evaluated to reach a comparison decision for 64-b input at 1 GHz.

our comparator dissipates power at a rate of 4.12 μW / MHz. Our comparator operates at very low power when the number of evaluated bits ranges from 8 to 28, which makes our comparator suitable for applications with typical data-dependent completion time and a low average number of evaluated bits.

VI. CONCLUSION

In this paper, we presented a scalable high-speed low-power comparator using regular digital hardware structures consisting of two modules: the comparison resolution module and the decision module. These modules are structured as parallel prefix trees with repeated cells in the form of simple stages that are one gate level deep with a maximum fan-in of five and fan-out of four, independent of the input bit width. This regularity allows simple prediction of comparator characteristics for arbitrary bit widths and is attractive for continued technology scaling and logic synthesis.

Leveraging the parallel prefix tree structure [42] for our comparator design is novel in that this design performs the comparison operation from the most significant to the least significant bit, using parallel operation, rather than rippling. Regardless of the comparator bit width, our structure guarantees that less than 35% of all of the transistors used in the design are active during operation. Additionally, all cells are locally interconnected, which avoids the need for large cell drivers, thus balancing all cells to a uniform transistor size.

Simulation results with standard CMOS transistor cells revealed operating speeds of 1.2 and 1 GHz for 64- and 512-b comparators, respectively, under a 0.15-μm CMOS process and worst case operands. These results translate to a 40% speed advantage over state-of-the-art fast comparators. Furthermore, simulation results confirmed our comparator's power efficiency, with a power dissipation of 0.9 μW / MHz on average and 4.12 μW / MHz in the worst case when 32 bits or more of the inputs must be evaluated.

Our simulation-based analysis of leakage power dissipation showed that, whereas the percentage contribution of leakage power increases with each new technology generation, the increase effect is not significant enough to nullify the savings in dynamic power dissipation in near-future technologies.

Future work will include additional circuit optimizations to further reduce the power dissipation by adapting dynamic and analog implementations for the comparator resolution module and a high-speed zero-detector circuit for the decision module. Given that our comparator is composed of two balanced timing modules, the structure can be divided into two or more pipeline stages with balanced delays, based on a set structure, to effectively increase the comparison throughput at the expense of increased power and latency.

REFERENCES

[1] H. J. R. Liu and H. Yao, *High-Performance VLSI Signal Processing Innovative Architectures and Algorithms*, vol. 2. Piscataway, NJ: IEEEPress, 1998.

[2] Y. Sheng and W. Wang, "Design and implementation of compression algorithm comparator for digital image processing on component," in
Proc. 9th Int. Conf. Young Comput. Sci., Nov. 2008, pp. 1337–1341.

[3] B. Parhami, "Efficient hamming weight comparators for binary vectors based on accumulative and up/down parallel counters," *IEEE Trans. Circuits Syst.*, vol. 56, no. 2, pp. 167–171, Feb. 2009.

[4] A. H. Chan and G. W. Roberts, "A jitter characterization system using a component-invariant Vernier delay line," *IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst.*, vol. 12, no. 1, pp. 79–95, Jan. 2004.

[5] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*, Piscataway, NJ: IEEE Press, 1990.

[6] H. Suzuki, C. H. Kim, and K. Roy, "Fast tag comparator using diode partitioned domino for 64-bit microprocessor," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 2, pp. 322–328, Feb. 2007.

[7] D. V. Ponomarev, G. Kucuk, O. Ergin, and K. Ghose, "Energy efficient comparators for superscalar datapaths," *IEEE Trans. Comput.*, vol. 53, no. 7, pp. 892–904, Jul. 2004.

[8] V. G. Oklobdzija, "An algorithmic and novel design of a leading zero detector circuit: Comparison with logic synthesis," *IEEE Trans. VeryLarge Scale Integr. (VLSI) Syst.*, vol. 2, no. 1, pp. 124–128, Mar. 1994.

[9] H. L. Helms, *High Speed (HC/HCT) CMOS Guide*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[10] *SN7485 4-bit Magnitude Comparators*, Texas Instruments, Dallas, TX,1999.

[11] K. W. Glass, "Digital comparator circuit," U.S. Patent 5 260 680, Feb. 13, 1992.

[12] D. norris, "Comparator circuit," U.S. Patent 5 534 844, Apr. 3, 1995.

[13] W. Guangjie, S. Shimin, and J. Lijiu, "New efficient design of digital comparator," in *Proc. 2nd Int. Conf. Appl. Specific Integr. Circuits*, 1996,
pp. 263–266.

[14] S. Abdel-Hafeez, "Single rail domino logic for four-phase clocking scheme," U.S. Patent 6 265 899, Oct. 20, 2001.

[15] M. D. Ercegovac and T. Lang, *Digital Arithmetic*, San Mateo, CA: Morgan Kaufmann, 2004.

[16] J. P. Uyemura, *CMOS Logic Circuit Design*, Norwood, MA: Kluwer, 1999.

[17] J. E. Stine and M. J. Schulte, "A combined two's complement and floating-point comparator," in *Proc. Int. Symp. Circuits Syst.*, vol. 1. 2005, pp. 89–92.

[18] S.-W. Cheng, "A high-speed magnitude comparator with small transistor count," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst.*, vol. 3. Dec. 2003, pp. 1168–1171.

[19] A. Bellaour and M. I. Elmasry, *Low-Power Digital VLSI Design Circuitsand Systems*. Norwood, MA: Kluwer, 1995.

[20] W. Belluomini, D. Jamsek, A. K. Nartin, C. McDowell, R. K. Montoye, H. C. Ngo, and J. Sawada, "Limited switch dynamic

logic circuits for high-speed low-power circuit design," *IBM J. Res. Develop.*, vol. 50, nos. 2–3, pp. 277–286, Mar.–May 2006.

[21] 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," in *IEEProc.-Comput. Digit. Tech.*, vol. 145, no. 6, pp. 433–436, Nov. 1998.

[22] C.-C. Wang, P.-M. Lee, C.-F. Wu, and H.-L. Wu, "High fan-in dynamic CMOS comparators with low transistor count," *IEEE Trans. CircuitsSyst. I*, vol. 50, no. 9, pp. 1216–1220, Sep. 2003.

[23] N. Maheshwari and S. S. Sapatnekar, "Optimizing large multiphase level-clocked circuits," *IEEE Trans. Comput.-Aided Design Integr. Cir-cuits Syst.*, vol. 18, no. 9, pp. 1249–1264, Nov. 1999.

[24] C.-H. Huang and J.-S. Wang, "High-performance and power-efficient CMOS comparators," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 254–262, Feb. 2003.

[25] H.-M. Lam and C.-Y. Tsui, "A mux-based high-performance single-cycle CMOS comparator," *IEEE Trans. Circuits Syst. II*, vol. 54, no. 7, pp. 591–595, Jul. 2007.

[26] F. Frustaci, S. Perri, M. Lanuzza, and P. Corsonello, "Energy-efficient single-clock-cycle binary comparator," *Int. J. Circuit Theory Appl.*, vol. 40, no. 3, pp. 237–246, Mar. 2012.

[27] P. Coussy and A. Morawiec, *High-Level Synthesis: From Algorithm toDigital Circuit*. New York: Springer-Verlag, 2008.

[28] S. Perri and P. Corsonello, "Fast low-cost implementation of single-clock-cycle binary comparator," *IEEE Trans. Circuits Syst. II*, vol. 55, no. 12, pp. 1239–1243, Dec. 2008.

[37] L. Yo-Sheng, C. Wu, C. Chang, R. Yang, W. Chen, J. Liaw, and C. H. Diaz, "Leakage scaling in deep submicron CMOS for SoC," *IEEETrans. Electron. Devices*, vol. 49, no. 6, pp. 2034–1041, Jun. 2002.

[38] W. K. Henson, N. Yang, S. Kubicek, E. M. Vogel, J. J. Wortman, K. De Meyer, and A. Naem, "Analysis of leakage currents and impact on offstate power consumption for CMOS technology in the 100-nm regime," *IEEE Trans. Electron. Devices*, vol. 47, no. 7, pp. 1393–1400, Jul. 2000.

[29] M. D. Ercegovac and T. Lang, "Sign detection and comparison networks with a small number of transitions," in *Proc. 12th IEEE Symp. Comput.Arithmetic*, Jul. 1995, pp. 59–66.

[30] J. D. Bruguera and T. Lang, "Multilevel reverse most-significant carry computation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 6, pp. 959–962, Dec. 2001.

[31] D. R. Lutz and D. N. Jayasimha, "The half-adder form and early branch condition resolution," in *Proc. 13th IEEE Symp. Comput. Arithmetic*, Jul. 1997, pp. 266–273.

[32] J. Hensley, M. Singh, and A. Lastra, "A fast, energy-efficient z-comparator," in *Proc. ACM Conf. Graph. Hardw.*, 2005, pp. 41–44.

[33] V. N. Ekanayake, I. K. Clinton, and R. Manohar, "Dynamic significance compression for a low-energy sensor network asynchronous processor," in *Proc. 11th IEEE Int. Symp. Asynchronous Circuits Syst.*, Mar. 2005,
pp. 144–154.

[34] H.-M. Lam and C.-Y. Tsui, "High-performance single clock cycle CMOS comparator," *Electron. Lett.*, vol. 42, no. 2, pp. 75–77, Jan. 2006.

[35] J.-Y. Kim and H.-J. Yoo, "Bitwise competition logic for compact digital comparator," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2007,
pp. 59–62.

[36] M. S. Schmookler and K. J. Nowka, "Leading zero anticipation and detection—a comparison of methods," in *Proc. 15th IEEE Symp. Com-put. Arithmetic*, Sep. 2001, pp. 7–12.

[39] Synopsys. (2010). *HSPICE*, Mountain View, CA [Online]. Available: http://www.synopsys.com

[40] *0.15 µm CMOS ASIC Process Digests*, Taiwan Semiconductor Manu-facturing Corporation, Hsinchu, Taiwan, 2002.

[41] *Cadence Online Documentation*. (2010) [Online]. Available:http://www.cadence.com

[42] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, 2nd ed. New York: Oxford, 2010.