Designs of BCD Adder Based on Excess-3 Code In Quantumdot Cellular Automatta

A.Brindha[1],V.Suresh[2], K.Somu[3]
[1, 2, 3] Dept of ME VLSI
[1, 2, 3] Mahabarathi Engineering College

Abstract- *metal oxide semiconductor (CMOS) technology. Binary-Coded Decimal (BCD) adders are widely used in industrial computing. In this brief, we propose two types of excess-3 code (XS-3) based BCD adders (XS-3DAs). We use ripple-carry adders (RCA) and parallel binary adders (PBA) to construct XS-3DAs in QCA Designer tool, respectively. The PBA-based XS-3DA is constructed with a new correction logic. 4-bit, 8-bit, and 16-bit XS-3DAs are constructed based on the two proposed XS-3DAs, respectively. Comparisons show that, with the increase of design scaling, the delay and area-delay product (Quantum-dot cellular automata (QCA) is a novelNano electronic technology. QCA has attracted wide attention due to its extremely small feature sizes at the molecular or even atomic scale and ultra-low power consumption, making it a promising candidate to replace the complementary ADP) of the PBA-based XS-3Das can be significantly reduced in comparison with that of the RCA based XS-3DAs. Compared with the 16-digit RCA-based XS-3DA, the cell count, area, delay and ADP of the proposed 16-digit PBA-based XS-3DA are reduced by 37.88%, 25.99%, 37.68% and 53.88%, respectively.*

I. INTRODUCTION

Very-Large-Scale Integration (VLSI) is the process of creating an integrated circuit (IC) by combining thousands of transistors into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. Before the introduction of VLSI technology most ICs had a limited set of functions they could perform. An electronic circuit might consist of a CPU, ROM, RAM and other glue logic. VLSI lets IC makers add all of these into one chip.

The realization of Multiplication operation is essential for most of the scientific applications as well as in commercial applications like banking, accounting, financial analysis, tax calculation, currency conversion, and insurance etc. Multiplier is a key building block and almost obligatory component in all such applications. That is why reconfigurable PLDs are equipped with dedicated embedded multipliers. The prerequisite of the multiplier implementation is that it should be primarily fast and secondarily efficient in terms of power consumption and chip area. The multiplication involves two basic operations viz. generation of partial products and their accumulation. To speed up the multiplication process, one can reduce the number of partial products to be generated and later, accelerating their accumulation. The design of multiplier depends upon the type, range and precision of data to be processed by the multiplier block viz. fixed point and floating point numbers represented in binary and decimal format.

The major contributions of this Project are summarized as follows.

1) We propose and enable the application of hybrid high radix encodings for the generation of energy-efficient approximate multipliers, exceeding the increased hard-ware complexity of very high radix encodings.
2) The proposed technique can be applied to any multiplier architecture and is reconfigurable, enabling the user to select the optimal per application energy–error tradeoff.
3) An analytical error analysis is conducted, showing that the output error of the proposed technique is bounded and predictable. Such a rigorous error analysis leads to precise and a priori error estimation for any input distribution, without the need of time-consuming simulations.
4) We show that the proposed technique outperforms related state-of-the-art approximate signed multipliers in terms of hardware and accuracy, achieving up to 40% less energy dissipation for comparable error values. More specifically, the proposed technique is applied to a 16×16 bit multiplier and is evaluated using industrial strength tools, i.e., Synopsys Design Compiler, Prime Time, and Mentor Graphics Modalism. Compared with the accurate multiplier, the proposed technique delivers up to 55% energy and area reduction, for mean relative error up to 0.93%. Moreover, compared with related state-of- the-art approximate computing signed multipliers, i.e., our technique outperforms them significantly in terms of energy consumption and error. Finally, we examine the scalability of the proposed technique and show that for

the same error value, the delivered energy savings increase as the multiplier size increases. The rest of this paper is organized as follows. As far as the approximate adders are concerned, produces approximate adders by simplifying the logic used in the adder. Gupta et al. design imprecise full adder cells by approximating their logic function and then use them to build approximate adders. Nevertheless, it is not clear how these adders can be used in different tree architectures and how the error scales in the case of multi operand accumulation.

Reference proposes an approximate adder that consists of an accurate and an inaccurate part. Reference designs a multiplier in which the LSBs of the additions are approximated by applying bitwise OR to the respective input bits. In , a fast approximate adder is produced by limiting the carry propagation, based on a proof that the longest carry chain in an n-bit adder is log n. However, despite the fact that all these techniques demonstrate the benefit of approximate computing, their fixed functionality and low-level design limit further improvements in efficiency.

Regarding the approximations in multiplication schemes, Kulkarni et al. propose an under designed 2 × 2 inaccurate multiplier block to produce the partial products, and then use it as a building block to design larger multipliers. This technique is characterized by high error and hardware overhead for the error detection and correction due to the small building blockpropose two approximate 4:2 compressors to accumulate the partial products by modifying the respective accurate truth table, and then use them to build approximate multipliers. The negative aspect of this technique is that there is no parameter to adjust the accuracy of the multiplier. Lin and Lin introduce a high accuracy approximate 4 × 4 Wallace tree multiplier by employing a 4:2 approximate counter that reduces the partial product stages, and then use it to build larger multipliers. However, the delay is large due to the array structure of the multiplier propose a multiplier that divides the operands in two parts: an accurate multiplication- based part that includes the MSBs and an approximate non multiplication-based part for the LSBs that does not generate partial products.

This design delivers significant reductions in power and delay, but only for specific input combinations. Moreover, Liu et al. propose an approximate multiplier with configurable error recovery that uses inaccurate fast adders for the partial product additions. Although this multiplier delivers low power, the error imposed cannot be predicted, as it depends on the carry propagation. split the operands in three m-bit segments and perform the multiplication utilizing the segment that contains the most significant nonzero bit. However, this approach exhibits small scalability, as m should be at least half the operand bit-width in order to keep the accuracy in acceptable limits. Reference extended this idea to enable dynamic range multiplications. The dynamic partition technique requires extra components for the signed multiplications, adding significant hardware overhead.

Recently, proposed a multiplier that rounds the input operands into the nearest exponent of two. Finally, replaces the floating-point operations with fixed-point ones, and by applying the proposed stochastic rounding, achieves good accuracy results in training deep neural networks while delivering high energy savings by limiting the data precision representation. The modified Booth encoding is commonly used in signed multipliers.

Although these techniques perform fast multiplications, the number of the partial products is not reduced in most cases, in contrast with our design. introduce the partial product perforation technique, where they omit the generation of some partial products based on the modified Booth encoding. Jiang et al. [17] propose an approximate radix-8 booth multiplier that uses an approximate adder for producing ±3 A, and combine this idea with the truncation method. Recently, Liu et al. [18] designed approximate modified Booth encoders by modifying its K-Map, and combined them with an approximate compressor.

In this paper, an approximate hybrid high radix encoding for designing energy- error efficient inexact multipliers is proposed. The difference with the existing related work is that it concerns approximations on the generation of the partial products, and can be combined with any accumulation technique, approximate or not. Another significant aspect of this paper is that the error imposed depends only on the configuration parameter k, and as a result, it can be calculated without the need for exhaustive simulations. Consequently, a precise estimation of the output quality can be extracted for the application’s inputs, giving the flexibility to target the maximum energy reduction for a specific error bound.

1.2. APPROXIMATE HYBRID HIGH RADIX MULTIPLIERS

High radix encodings offer partial products reduction, and as a result, their accumulation requires smaller trees, leading to energy, area, and/or delay savings. However, high radix encodings require complex encoding and partial product generation circuits, negating thus the benefits of the partial products reduction. In this section, the proposed hybrid high

radix encoding and the performed approximations for simplifying its circuit complexity are presented. In the proposed technique, the multiplicand B is encoded using the approximate high radix encoding, generating B˜, and the approximate multiplication A·B˜ is performed. Finally, its adaptation on inexact 16-bit hardware multipliers is described, and a qualitative analysis is conducted, targeting to estimate the potential area gains

II. LITERATURE SURVEY

2.1 APPROXIMATE COMPUTING: AN EMERGING PARADIGM FOR ENERGY- EFFICIENT DESIGN

ABSTRACT

Approximate computing has recently emerged as a promising approach to energy-efficient design of digital systems. Approximate computing relies on the ability of many systems and applications to tolerate some loss of quality or optimality in the computed result. By relaxing the need for fully precise or completely deterministic operations, approximate computing techniques allow substantially improved energy efficiency. This paper reviews recent progress in the area, including design of approximate arithmetic blocks, pertinent error and quality measures, and algorithm-level techniques for approximate computing.

PROBLEM STATEMENT

- Perceptual limitations: these are determined by the ability of the human brain to ‘fill in’ missing information and filter out high-frequency patterns
- Redundant input data: this redundancy means that an algorithm can be lossy and still be adequate
- Noisy inputs.

ADVANTAGE

- It can be modified to exhibit the incremental refinement property and allow favorable energy-quality trade-offs.
- Reducing the number of support vectors reduces the number of dot product computations per classification

2.2 MODELLING AND SYNTHESIS OF QUALITY-ENERGY OPTIMAL APPROXIMATE ADDERS

ABSTRACT

A formal model to prove that for signal processing applications using a quadratic signal- to-noise ratio error measure, reducing bit-wise error frequency is sub-optimal. Instead, energy- optimal approximate addition requires limiting maximum error magnitude. Intriguingly, due to possible error patterns, this is achieved by reducing carry chains significantly below what is allowed by the timing budget for a large fraction of sum bits, using an aligned, fixed internal- carry structure for higher significance bits. We further demonstrate that remaining approximation error is reduced by retaliation of conditional bounding (CB) logic for lower significance bits. A key contribution is the formalization of an approximate CB logic synthesis problem that produces a rich space of Pareto-optimal adders with a range of quality-energy tradeoffs. We show how CB logic can be customized to result in over-and under-estimating approximate adders, and how a dithering adder that mixes them produces zero-centered error distributions, and, in accumulation, a reduced-variance error. We demonstrate synthesized approximate adders with energy up to 60% smaller than that of a conventional timing-starved adder, where a 30% reduction is due to the superior synthesis of inexact CB logic. When used in a larger system implementing an image-processing algorithm, energy savings of 40% are possible.

PROBLEM STATEMENT

In approximate computation is driven by its potential to achieve large energy savings. This paper formally demonstrates an optimal way to reduce energy via voltage over-scaling at the cost of errors due to timing starvation in addition. We identify a fundamental trade-off between error frequency and error magnitude in a timing-starved adder.

ADVANTAGE

- Approximation error is reduced by realization of conditional bounding (CB) logic for lower significance bits.
- A 30% reduction is due to the superior synthesis of inexact CB logic.
- Reducing bit-wise error frequency is sub-optimal. Instead, energy-optimal approximate addition requires limiting maximum error magnitude. Intriguingly, due to possible error patterns.

2.2 NEW METRICS FOR THE RELIABILITY OF APPROXIMATE ANDPROBABILISTIC ADDERS

ABSTRACT

New metrics are proposed for evaluating the reliability as well as the power efficiency of approximate and

probabilistic adders. Reliability is analyzed using the so-called sequential probability transition matrices (SPTMs). Error distance (ED) is initially defined as the arithmetic distance between an erroneous output and the correct output for a given input. The mean error distance (MED) and normalized error distance (NED) are then proposed as unified figures that consider the averaging effect of multiple inputs and the normalization of multiple-bit adders. It is shown that the MED is an effective metric for measuring the implementation accuracy of a multiple-bit adder and that the NED is a nearly invariant metric independent of the size of an adder. The MED is, therefore, useful in assessing the effectiveness of an approximate or probabilistic adder implementation, while the NED is useful in characterizing the reliability of a specific design.

PROBLEM STATEMENT

Addition is a fundamental function in arithmetic operation; several adder designs have been proposed for implementations in inexact computing. These adders show different operational profiles; some of them are approximate in nature while others rely on probabilistic features of Nano scale circuits. However, there has been a lack of appropriate metrics to evaluate the efficacy of various inexact designs.

ADVANTAGES

2.4 LOW-POWER HIGH-SPEED MULTIPLIER FOR ERROR-TOLERANT APPLICATION

ABSTRACT

In this paper, a new design concept that engaged accuracy as a design parameter is proposed. By introducing accuracy as a design parameter, the bottleneck of conventional digital design techniques can be breakthrough to improve on the performances of power consumption and speed. The aim is to fulfill the need for high performance basic sequential elements with low-power dissipation which is steadily growing.

PROBLEM STATEMENT

A migration of design emphasis from conventional delay and area optimization to power dissipation minimization, while preserving the desired performance. One common technique for energy efficiency CMOS circuits is the reduction of the supply voltage. However, there are two drawbacks: first is the increase in the gates delay.

ADVANTAGE

- For maximum area reduction, an n x n multiplier generates only n times of the most significant product bits and truncates the least significant half of the partial products in order to produce a final product with reduced precision.
- It achieves a lower average error, and truncation of the partial product bits, an error compensation circuit with an area overhead that is much lower than the truncated part is usually added.
- Such circuit is widely used in current digital signal processing applications.

2.5 APPROXIMATE ADDERS FOR APPROXIMATE MULTIPLICATIONABSTRACT

The gap between capabilities of CMOS technology scaling and requirements of future application workloads is increasing rapidly. There are several promising design approaches that jointly can reduce this gap significantly. Approximate computing is one of them and in recent years, has attracted the strongest attention of the scientific community. Approximate computing exploits inherent error salience of applications and features high-performance energy-efficient software and hardware implementations by trading-off computational quality (e.g., accuracy) for computational efforts (e.g., performance and energy). Over the decade, several research efforts have explored approximate computing throughout all the layers of computing stack, however, most of the work at hardware level of abstraction has been proposed on adders. In, a comparative survey of state-of-the-art approximate adders is provided. And it also provides comparison based on both conventional design metrics as well as approximate computing design metrics.

PROBLEM STATEMENT

Inexact computing is particularly interesting for computer arithmetic designs.

ADVANTAGE

The designs accomplish significant reductions in power dissipation, delay and transistor count compared to an exact design.

Moreover, two of the proposed multiplier designs provide excellent capabilities for image multiplication with respect to average normalized error distance and peak signal-to-noise ratio (more than 50dB for the considered image examples).

4.1 Introduction To Micro wind

Model wind is a useful tool that allows you to stimulate the inputs of your modules and view both outputs and internal signals. It allows you to do both behavioral and timing simulation; however, this document will focus on behavioral simulation. Keep in mind that these simulations are based on models and thus the results are only as accurate as the constituent models. Model Sims /VHDL, Model Sims /VLOG, Model Sims /LNL, and Model Sims /PLUS are produced by Model Technology Incorporated. The program described in this manual is furnished under a license agreement and may not be used or copied except in accordance with the terms of the agreement. The online documentation provided with this product may be printed by the end-user.

The number of copies that may be printed is limited to the number of licenses purchased. Model Sims is a registered trademark of Model Technology Incorporated. Model Technology is a trademark of Mentor Graphics Corporation. PostScript is a registered trademark of Adobe Systems Incorporated. UNIX is a registered trademark of AT&T in the USA and other countries. Model Sims the simulator of choice for both ASIC and FPGA designs. The best standards and platform support in the industry make it easy to adopt in the majority of process and tool flows
Major product features

- Leading RTL and gate performance with ASIC sign-off
- Native support of VHDL, Verilog, System Verilog, and System C

4.2 Advanced Code Coverage

Modalismvanced code coverage capabilities and ease of use lower the barriers for leveraging this valuable verification resource. The Model advanced code coverage capabilities provide valuable metrics for systematic verification. All coverage information is stored in the Unified Coverage Data Base (UCDB), which is used to collect and manage all coverage information in a highly efficient database. Coverage utilities that analyze code coverage data, such as merging and test ranking, are available. Coverage results can be viewed interactively, post-simulation, or after a merge of multiple simulation runs. Code coverage metrics can be reported by instance or by design unit, providing flexibility in managing coverage data.

The coverage types supported include:

Statement coverage: number of statements executed during a run

Branch coverage: expressions and case statements that affect the control flow of the HDL execution

Condition coverage: breaks down the condition on a branch into elements that make the result true or false

Expression coverage: the same as condition coverage, but covers concurrent signal assignments instead of branch decisions.

4.3 Methodologies For Functional Verification

Verification Methodologies form the backbone of a solid verification strategy. Mentor Graphics is actively driving advanced methodologies and their standardization across the industry.

Assertion Based Verification

Assertion-based verification (ABV) is a methodology in which designers use assertions to capture specific design intent and, either through simulation, formal verification, or emulation of these assertions, verify that the design correctly implements that intent.

Universal Verification Methodology

The UVM is the first Industry-Standard verification methodology delivering an open and unified class library and methodology for interoperable verification IP and test benches. Based on OVM and written in System Verilog 1800, the UVM is the culmination of collaboration among user and vendor companies in Accelerate.

II. IDENTIFY, RESEARCH AND COLLECT IDEA

4.4 Introduction To Xilinx

Xilinx is a supplier of programmable logic devices. It is known for inventing the field programmable gate array (FPGA) and as the first semiconductor company with a fabless manufacturing model.

The Spartan-3 family architecture consists of five fundamental programmable functional elements:

Configurable Logic Blocks (CLBs) contain RAM-based Look-Up Tables (LUTs) to implement logic and storage elements that can be used as flip-flops or latches. CLBs can be programmed to perform a wide variety of logical functions as well as to store data.

Input/output Blocks (IOBs) Input/output control the flow of data between the I/O pins and the internal logic of the device. Each IOB supports bidirectional data flow plus 3-state operation. Double Data-Rate (DDR) registers are included. The Digitally Controlled Impedance (DCI) feature provides automatic on-chip terminations, simplifying board designs.
Block RAM - Block RAM provides data storage in the form of 18-Kbitdual-port blocks

Multiplier blocks - Multiplier Blocks accept two 18-bit binary numbers as inputs and calculate the product.

Digital Clock Manager (DCM) - DCM blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase shifting clock signals. These elements are organized as shown in Figure. A ring of IOBs surrounds a regular array of CLBs.

The DCMs are positioned at the ends of the outer block RAM columns. The Spartan-3 family features a rich network of traces and switches that interconnect all five functional elements, transmitting signals among them. Each functional element has an associated switch matrix that permits multiple connections to the routing.

5.1 HARDWARE DESCRIPTION LANGUAGE (HDL):

The dramatic increase in the logic density of silicon chips has made it possible to implement digital systems with multimillion gates on a single chip. The complexity of such systems makes it impractical to use traditional design descriptions (e.g., logic schematics) to provide a complete and accurate description of a design. Currently, all complex digital designs are expressed using a hardware description language (HDL). A major advantage of an HDL is that it provides a better and more concise documentation of a design than gate-level schematics. Two very popular HDLs are VHDL and VERILOG.

5.1.1 LEVELS OF REPRESENTATION AND ABSTRACTION

A digital system can be represented at different levels of abstraction. This keeps the description and design of complex systems manageable. Figure 5.1 shows different levels of abstraction.

5.1.1.1 BEHAVIORAL LEVEL

The highest level of abstraction is the behavioral level that describes a system in terms of what it does rather than in terms of its components and interconnection between them. A behavioral description specifies the relationship between the input and output signals. This could be a Boolean expression or a more abstract description such as the Register Transfer or Algorithm
Configuration

Spartan-3 FPGAs are programmed by loading configuration data into robust, reprogrammable, static CMOS configuration latches (CCLs) that collectively control all functional elements and routing resources. Before powering on the FPGA, configuration data is stored externally in a PROM or some other nonvolatile medium either on or off the board. After applying power, the configuration data is written to the FPGA using any of five

III. WRITEDOWNYOUR STUDIESANDFINDINGS

Digital Clock Manager (DCM) - DCM blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase shifting clock signals. These elements are organized as shown in Figure. A ring of IOBs surrounds a regular array of CLBs.

The DCMs are positioned at the ends of the outer block RAM columns. The Spartan-3 family features a rich network of traces and switches that interconnect all five functional elements, transmitting signals among them. Each functional element has an associated switch matrix that permits multiple connections to the routing.

IV. GET PEER REVIEWED

The following is the analysis of the proposed PBA-based XS-3DAs. The signals dA3:0 and dB3:0 are two decimal input numbers represented by excess-3 codes. It can be seen from Fig. 5 that, after adjusting the input of signal dcin when the signal bS3:0 is greater than or equal to “10”, the carry signal bCout is correct no matter the value of dCin; when bS3:0 is less than “9”, the carry signal bCout is correct no matter the value of dCin; when bS3:0 is equal to “9” and dCin is equal to “0”, the carry signal bCout is correct; when bS3:0 is equal to “9” and dCin is equal to “1”, the signal bCout is “0”, which is wrong and needs to be corrected. The output carry bCout of the XS-3DAs can be rewritten as formula (5).

dCout = bCout + (bS3:0 == 9)dCin (5)

The excess-3 code of “9” is (1111) XS-3. Formula (5) can be rewritten as formula (6).

dCout = bCout + -

bS3:0 == (1111)XS−3

dCin

dCout = bCout + bS3 · bS2 · bS1 · bS0 · dCin

dCout = bCout + (bS3 · bS2) · (bS1 · bS0) · dCin

V. IMPROVEMENT AS PER REVIEWER COMMENTS

Cases there could be chances where your paper receives number of critical remarks. In that cases don't get disheartened and try to improvise the maximum.

VI. CONCLUSION

Majority logic has been widely used in emerging computing nanotechnologies. The realization of arithmetic function based on majority logic has been studied quite extensively. In this brief, we make use of the advanced logic representation of three-input XOR and majority operations to derive new expressions of both 1-digit and multi-digit BCD adder. we have obtained promising results in both area and delay. For the 8-digit case, our proposed BCD adder has a 3.42× area-delay product improvement compared with the existing best designs.. Furthermore, we showed the efficiency of the proposed multipliers when applied in real-life applications. Finally, the proposed technique is scalable, delivering higher energy savings for the same error, as the multiplier's size increases. In this brief, we have proposed novel RCA-based and PBA based XS-3DAs in the QCA Designer tool. A new correction circuit is used to construct the PBA-based XS-3DAs. Our proposed RCA based and PBA-based XS-3DAs show excellent performance in terms of area and ADP. We have analyzed and compared the characteristics of the two types of XS-3DAs.The proposed CL circuit can significantly reduce the delay and ADP of the n-digit RCA-based XS-3DAs, thus saving the overall cost of QCA circuit designs.

VII. ACKNOWLEDGMENT

I consider my selves extremely fortunate and blessed for completing my project successfully.

I am express my heartfelt thanks to our Chairman Dr.Lion K.MOHAN, and owe my deep sense of gratitude to our blessed Secretary Dr.J.V.SEZHIYAN, M.E.,Ph.D., and Treasurer Mr.S.NALLIYAPPAN, B.sc., for their prayers. It is my privilege and pleasure to thank our Principal Dr.R.GURUMANI,M.E.,Ph.D., for granting necessary permission to accomplishthis project.

I am greatly vindebted to our Head of the Department Dr.K.SOMU,M.E.,Ph.D., for his immense help, invaluable guidance and forsharing his valuable times. I extend my gratitude to my project coordinator Mr.V.SURESH,M.E., for the guidance. He has bestowed upon us during theproject period. I have great pleasure to express my deep, sincere and whole heartedthanks to my project internal guide Mr .V. SURESH,ME.,

I take this opportunity to place on record of my gratitude to all the faculty and staffmembers of the department of Electronics and Communication Engineering for their co-operation during the course period and for completing the project.

Last but not least I extend my thanks to my FRIENDS and PARENT'S for their unceasing, prayerful encouragement and moral support.

REFERENCES

[1] Mittal, Anubhuti, Ashutosh Nandi, and Disha Yadav. "Comparative study of 16-order FIR filter design using different multiplication techniques." IET Circuits, Devices & Systems 11.3 (2017): 196-200.]

[2] M. Hasan, M. J. Hossein, M. Hossain, H. U. Zaman and S. Islam, "Design of a Scalable Low- Power 1-Bit Hybrid Full Adder for Fast Computation," in IIEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 8, pp. 1464-1468, Aug. 2020, doi: 10.1109/TCSII.2019.2940558.

[3] S. Goel, A. Kumar, and M. A. Bayoumi, "Design of robust, energy efficient full adders for deep-submicrometer design using hybrid-CMOS logic style," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 12, pp. 1309-1321, Dec. 2006.

[4] M. Zhang, J. Gu, and C.-H. Chang, "A novel hybrid pass logic with static CMOS output drive full-adder cell," in Proc. Int. Symp. Circuits Syst., May 2003, pp. 317-320.