International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 14
October 2016

High Speed and Energy Efficient Carry Skip Adder Operating Under A Wide Range of Supply Voltages Levels

Mohammed Mujahed Ali Adeel
M.Tech Student Scholar
Department of Electronics & Communication Engineering,
VIF institute of technology and science, Moinabad, Rangareddy DT T.S, India. mohdadil67333@gmail.com

Imthiazunnisa Begum
HEAD OF THE DEPARTMENT
Department of Electronics & Communication Engineering,
VIF institute of technology and science, Moinabad,Rangareddy DT T.S, India. ecehodvifcet@gmail.com

ABSTRACT: In this paper, we present a carry skip adder (CSKA) structure that has a higher speed yet lower energy consumption compared with the conventional one. The speed enhancement is achieved by applying concatenation and incrementation schemes to improve the efficiency of the conventional CSKA (Conv-CSKA) structure. In addition, instead of utilizing multiplexer logic, the proposed structure makes use of AND-OR-Invert (AOI) and OR-AND-Invert (OAI) compound gates for the skip logic. The structure may be realized with both fixed stage size and variable stage size styles, wherein the latter further improves the speed and energy parameters of the adder. Finally, a hybrid variable latency extension of the proposed structure, which lowers the power consumption without considerably impacting the speed, is presented. This extension utilizes a modified parallel structure for increasing the slack time, and hence, enabling further voltage reduction. The proposed structures are assessed by comparing their speed, power, and energy parameters with those of other adders using a 45-nm static CMOS technology for a wide range of supply voltages. The results that are obtained using HSPICE simulations reveal, on average, 44% and 38% improvements in the delay and energy, respectively, compared with those of the Conv-CSKA. In addition, the power–delay product was the lowest among the structures considered in this paper, while its energy–delay product was almost the same as that of the Kogge–Stone parallel prefix adder with considerably smaller area and power consumption. Simulations on the proposed hybrid variable latency CSKA reveal reduction in the power consumption compared with the latest works in this field while having a reasonably high speed.

(I) INTRODUCTION

Adders are a key building block in arithmetic and logic units (alus) and hence increasing their speed and reducing their power/energy consumption strongly affect the speed and power consumption of processors. There are many works on the subject of optimizing the speed and power of these units, which have been reported. Obviously, it is highly desirable to achieve higher speeds at low-power/energy consumptions, which is a challenge for the designers of general purpose processors. One of the effective techniques to lower the power consumption of digital circuits is to reduce the supply voltage due to quadratic dependence of the switching energy on the voltage. Moreover, the subthreshold current, which is the main leakage component in OFF devices, has an exponential dependence on the supply voltage level through the drain-induced barrier lowering effect. Depending on the Amount of the supply voltage reduction, the operation of ON devices may reside in the superthreshold, near-threshold, or subthreshold regions. Working in the superthreshold region provides us with lower delay and higher switching and leakage powers compared with the near/subthreshold regions. In the subthreshold region, the logic gate delay and leakage power exhibit exponential dependences on the supply and threshold voltages. Moreover, these voltages are (potentially) subject to process and environmental variations in the nanoscale technologies. The variations increase uncertainties in the aforesaid performance parameters. In addition, the small subthreshold Current causes a large delay for the circuits operating in the subthreshold region. Recently, the near-threshold region has been considered as a region that provides a more desirable tradeoff point between delay and power dissipation compared with that of the subthreshold one, because it results in lower delay compared with the subthreshold region and significantly lowers switching and leakage powers compared with the superthreshold region. In addition, near-threshold operation, which uses supply voltage levels near the threshold voltage of transistors, suffers considerably less from the process and environmental variations compared with the subthreshold region.

(II) LITERATURE SURVEY:

(A) ENERGY-DELAY ESTIMATION TECHNIQUE FOR HIGH-PERFORMANCE MICROPROCESSOR VLSI ADDERS In this paper, we motivate the concept of comparing VLSI adders based on their energy-delay trade-offs and present a technique for estimating the energy-delay space of various high-performance VLSI adder topologies. Further, we show that our estimates accurately represent tradeoffs in the energy-delay

space for high-performance 32-bit and 64-bit processor adders in 0.13µm and 0.10µm CMOS technologies, with an accuracy of 8% in delay estimates and 20% in energy estimates, compared with simulated data.

(B) ENERGY–DELAY OPTIMIZATION OF 64-BIT CARRY-LOOKAHEAD ADDERS WITH A 240 PS 90 NM CMOS DESIGN EXAMPLE A methodology for energy–delay optimization of digital circuits is presented. This methodology is applied to minimizing the delay of representative carry-look ahead adders under energy constraints. Impact of various design choices, including the carrylookahead tree structure and logic style, are analyzed in the energy– delay space and verified through optimization. The result of the optimization is demonstrated on a design of the fastest adder found, a 240-ps Ling sparse domino adder in 1 V, 90 nm CMOS. The optimality of the results is assessed against the impact of technology scaling

(C)BTI-AWARE DESIGN USING VARIABLE LATENCY UNITS Circuit degradation due to bias temperature instability (BTI) can lead to timing failures in digital circuits. We develop variable latency unit (VLU) based BTI-aware designs, with a novel scheme for multioutput hold logic implementation for VLUs. A key observation is the identification and exploitation of specific super setting patterns in the two-dimensional space of frequency and aging of the circuit. The multioutput hold logic scheme is used in conjunction with an adaptive body bias framework to achieve high performance, allowing the design to be easily incorporated in traditional synthesis flows. As compared to conventional combinational BTI-resilience scheme, our design achieves an area reduction of 9.2%, with a significant throughput enhancement of 30.0%.

(D) NEAR-THRESHOLD COMPUTING: RECLAIMING MOORE'S LAW THROUGH ENERGY EFFICIENT INTEGRATED CIRCUITS Power has become the primary design constraint for chip designers today. While Moore's law continues to provide additional transistors, power budgets have begun to prohibit those devices from actually being used. To reduce energy consumption, voltage scaling techniques have proved a popular technique with subthreshold design representing the endpoint of voltage scaling. Although it is extremely energy efficient, subthreshold design has been relegated to niche markets due to its major performance penalties. This paper defines and explores near-threshold computing (NTC), a design space where the supply voltage is approximately equal to the threshold voltage of the transistors. This region retains much of the energy savings of subthreshold operation with more favorable performance and variability characteristics. This makes it applicable to a broad range of power-constrained computing segments from sensors to high performance servers. This paper explores the barriers to the widespread adoption of NTC and describes current work aimed at overcoming these obstacles

(III) MAIN OBJECTIVE

- In this paper, we present a carry skip adder (CSKA) structure that has a higher speed yet lower energy consumption compared with the conventional one.
- The speed enhancement is achieved by applying concatenation and incrementation schemes to improve the efficiency of the conventional CSKA (Conv-CSKA) structure. In addition, instead of utilizing multiplexer logic, the proposed structure =====makes use of AND-OR-Invert (AOI) and OR-AND-Invert (OAI) compound gates for the skip logic.
- The structure may be realized with both fixed stage size and variable stage size styles, wherein the latter further improves the speed and energy parameters of the adder. Finally, a hybrid variable latency extension of the proposed structure, which lowers the power consumption without considerably impacting the speed, is presented.
- This extension utilizes a modified parallel structure for increasing the slack time, and hence, enabling further voltage reduction.
- The proposed structures are assessed by comparing their speed, power, and energy parameters with those of other adders using a 45-nm static CMOS technology for a wide range of supply voltages. The results that are obtained using HSPICE simulations reveal, on average, 44% and 38% improvements in the delay and energy, respectively, compared with those of the Conv-CSKA.
- In addition, the power–delay product was the lowest among the structures considered in this paper, while its energy–delay product was almost the same as that of the Kogge–Stone parallel prefix adder with considerably smaller area and power consumption.
- Simulations on the proposed hybrid variable latency CSKA reveal reduction in the power consumption compared with the latest works

in this field while having a reasonably high speed.

(IV) EXISTING METHODS

- Conventional CSKA is a RCA-RCA configuration which generates a pair of sum words and output-carry bits corresponding the anticipated input-carry (cin= 0 and 1), and selects one out of each pair for final-sum and final-output-carry.
- Conventional CSKA has less CPD than RCA, but the design is not attractive, since it uses dual RCA. Few attempts have been made to avoid dual use of RCA in CSKA design.

(A) Existing system description; It is possible to create a logical circuit using multiple full adders to add N-bit numbers. Each full adder inputs a Cin, which is the Cout of the previous adder. This kind of adder is called a ripple-carry adder, since each carry bit "ripples" to the next full adder. Note that the first (and only the first) full adder may be replaced by a half adder (under the assumption that Cin = 0). The layout of a ripple-carry adder is simple, which allows for fast design time; however, the ripple-carry adder is relatively slow, since each full adder must wait for the carry bit to be calculated from the previous full adder.

(B) Existing system disadvantages: A ripple carry adder is a logic circuit in which the carry-out of each full adder is the carry in of the succeeding next most significant full adder. It is called a ripple carry adder because each carry bit gets rippled into the next stage.

- More area overhead system
- More power consumption
- Low speed architecture

(C)CONVENTIONAL CARRY SKIP ADDER: The structure of an N-bit Conv-CSKA, which is based on blocks of the RCA (RCA blocks), is shown in Fig.

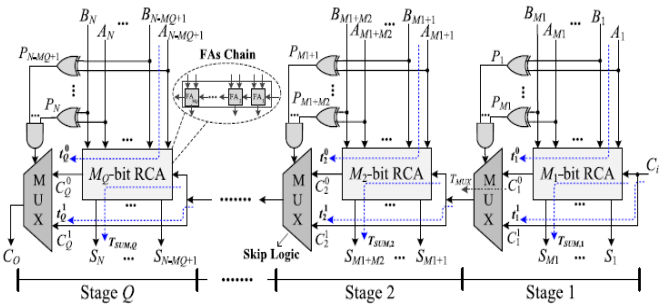

Figure 1 conventional carry skip adder

In addition to the chain of FAs in each stage, there is carry skip logic. For an RCA that contains N cascaded FAs, the worst propagation delay of the summation of two N-bit numbers, A and B, belongs to the case where all the FAs are in the propagation mode. It means that the worst case delay belongs to the case where

$$Pi = Ai \wedge Bi = 1 \text{ for } i = 1 \ldots N$$

Where Pi is the propagation signal related to Ai and Bi. This shows that the delay of the RCA is linearly related to N. In the case, where a group of cascaded FAs are in the propagate mode, the carry output of the chain is equal to the carry input. In the CSKA, the carry skip logic detects this situation, and makes the carry ready for the next stage without waiting for the operation of the FA chain to be completed. The skip operation is performed using the gates and the multiplexer shown in the figure. Based on this explanation, the N FAs of the CSKA are grouped in Q stages. Each stage contains an RCA block with Mj FAs (j = 1, . . . , Q) and a skip logic. In each stage, the inputs of the multiplexer (skip logic) are the carry input of the stage and the carry output of its RCA block (FA chain). In addition, the product of the propagation signals (P) of the stage is used as the selector signal of the multiplexer. The CSKA may be implemented using FSS and VSS where the highest speed may be obtained for the VSS structure.

(A) Fixed Stage Size CSKA: By assuming that each stage of the CSKA contains M FAs, there are Q = N/M stages where for the sake of simplicity, we assume Q is an integer. The input signals of the j th multiplexer are the carry output of the FAs chain in the j th stage denoted by C0j , the carry output of the previous stage (carry input of the j th stage) denoted by C1J.

The critical path of the CSKA contains three parts:

- The path of the FA chain of the first stage whose delay is equal to M × TCARRY;

- The path of the intermediate carry skip multiplexer whose delay is equal to the (Q – 1)× TMUX
- The path of the FA chain in the last stage whose its delay is equal to the (M −1) × TCARRY +TSUM. Note that TCARRY

TSUM, and TMUX are the propagation delays of the carry output of an FA, the sum output of an FA, and the output delay of a 2:1 multiplexer, respectively. Hence, the critical path delay of a FSS CSKA is formulated by

$$TD = [M \times TCARRY] + \_\_ N\ M - 1\ \_ \times TMUX\ \_ + [(M - 1) \times TCARRY + TSUM].$$

Based on (1), the optimum value of M (Mopt) that leads to optimum propagation delay may be calculated as (0.5Nα)1/2 where α is equal to TMUX/TCARRY. Therefore, the optimum propagation delay (TD,opt) is obtained from TD,opt = 2 _ 2NTCARRYTMUX + (TSUM − TCARRY − TMUX) = TSUM + (2 √ 2Nα−1−α) × TCARRY. Thus, the optimum delay of the FSS CSKA is almost proportional to the square root of the product of N and α.

(B) Variable Stage Size CSKA :

As mentioned before, by assigning variable sizes to the stages, the speed of the CSKA may be improved. The speed improvement in this type is achieved by lowering the delays of the first and third terms . These delays are minimized by lowering sizes of first and last RCA blocks. For instance, the first RCA block size may be set to one, whereas sizes of the following blocks may increase. To determine the rate of increase, let us express the propagation delay of the C1j (t1j).

(V) Proposed system:

- The CBL based CSKA of involves significantly less logic resource than the conventional CSKA, but it has longer CPD which is almost equals to that of RCA. To overcome this problem, SQRT-CSKA based on CBL has been proposed.
- We observe that, logic optimization largely depends on availability of redundant operations in the formulation, where adder delays mainly depends on data-dependency.

The structure is based on combining the concatenation and the incrementation schemes with the Conv-CSKA structure, and hence, is denoted by CI-CSKA. It provides us with the ability to use simpler carry skip logics. The logic replaces 2:1 multiplexers by AOI/OAI compound gates.

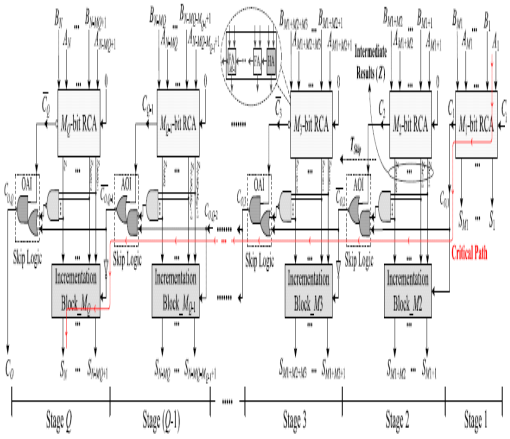

Figure 2: block of ripple carry adder

The gates, which consist of fewer transistors, have lower delay, area, and smaller power consumption compared with those of the 2:1 multiplexer . Note that, in this structure, as the carry propagates through the skip logics, it becomes complemented. Therefore, at the output of the skip logic of even stages, the complement of the carry is generated. The structure has a considerable lower propagation delay with a slightly smaller area compared with those of the conventional one. Note that while the power consumptions of the AOI (or OAI) gate are smaller than that of the multiplexer, the power consumption of the proposed CI-CSKA is a little more than that of the conventional one. This is due to the increase in the number of the gates, which imposes a higher wiring capacitance (in the noncritical paths).

Now, we describe the internal structure of the proposed CI-CSKA shown in Fig. 1 in more detail. The adder contains two N bits inputs, A and B, and Q stages. Each stage consists of an RCA block with the size of Mj (j = 1, . . . , Q). In this structure, the carry input of all the RCA blocks, except for the first block which is Ci , is zero (concatenation of the RCA blocks). Therefore, all the blocks execute their jobs simultaneously. In this structure, when the first block computes the summation of its corresponding input bits (i.e., SM1, . . . , S1), and C1, the other blocks simultaneously compute the intermediate results [i.e., {ZK j+Mj , . . . , ZK j+2, ZK j+1} for K j = _j−1 r=1 Mr (j = 2, . . . , Q)], and also Cj signals. In the proposed structure, the first stage has only one block, which is RCA. The stages 2 to Q consist of two blocks of RCA and incrementation. The incrementation block uses the intermediate results

generated by the RCA block and the carry output of the previous stage to calculate the final summation of the stage. The internal structure of the incrementation block, which contains a chain of half-adders (HAs), is shown in Fig. 3. In addition, note that, to reduce the delay considerably, for computing the carry output of the stage, the carry output of the incrementation block is not used. As shown in Fig. 1, the skip logic determines the carry output of the j th stage (CO, j) based on the intermediate results of the j th stage and the carry output of the previous stage (CO, j−1) as well as the carry output of the corresponding RCA block (Cj). When determining CO, j , these cases may be encountered. When Cj is equal to one, CO, j will be one. On the other hand, when Cj is equal to zero, if the product of the intermediate results is one (zero), the value of CO, j will be the same as CO, j−1 (zero). The reason for using both AOI and OAI compound gates as the skip logics is the inverting functions of these gates in standard cell libraries. This way the need for an inverter gate, which increases the power consumption and delay, is eliminated. As shown in Fig. 1, if an AOI is used as the skip logic, the next skip logic should use OAI gate. In addition, another point to mention is that the use of the proposed skipping structure in the Conv-CSKA structure increases the delay of the critical path considerably. This originates from the fact that, in the Conv-CSKA, the skip logic (AOI or OAI compound gates) is not able to bypass the zero carry input until the zero carry input propagates from the corresponding RCA block. To solve this problem, in the proposed structure, we have used an RCA block with a carry input of zero (using the concatenation approach). This way, since the RCA block of the stage does not need to wait for the carry output of the in parallel.

(A) Proposed Hybrid Variable Latency CSKA Structure:

The basic idea behind using VSS CSKA structures was based on almost balancing the delays of paths such that the delay of the critical path is minimized compared with that of the FSS structure. This deprives us from having the opportunity of using the slack time for the supply voltage scaling. To provide the variable latency feature for the VSS CSKA structure, we replace some of the middle stages in our proposed structure with a PPA modified in this paper. It should be noted that since the Conv-CSKA structure has a lower speed than that of the proposed one, in this section, we do not consider the conventional structure. The proposed hybrid variable latency CSKA structure is shown where an Mp-bit modified PPA is used for the pth stage (nucleus stage). Since the nucleus stage, which has the largest size (and delay) among the stages, is present in both SLP1 and SLP2, replacing it by the PPA reduces the delay of the longest off-critical paths.

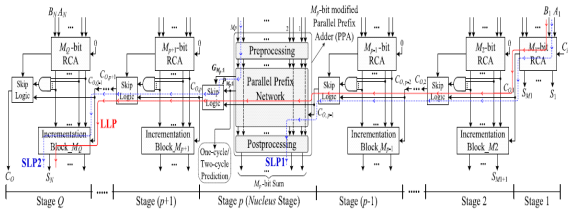

Figure 3: Hybrid Variable Latency CSKA Structure

Thus, the use of the fast PPA helps increasing the available slack time in the variable latency structure. It should be mentioned that since the input bits of the PPA block are used in the predictor block, this block becomes parts of both SLP1 and SLP2. In the proposed hybrid structure, the prefix network of the Brent–Kung adder is used for constructing the nucleus stage. One the advantages of the this adder compared with other prefix adders is that in this structure, using forward intermediate carries, which are computed by backward paths. In addition, the fan-out of adder is less than other parallel adders, while the length of its wiring is smaller. Finally, it has a simple and regular layout.

The internal structure of the stage p, including the modified PPA and skip logic, is shown. Note that, for this figure, the size of the PPA is assumed to be 8 (i.e., Mp = 8). As shown in the figure, in the preprocessing level, the propagate signals (Pi) and generate signals (Gi) for the inputs are calculated. In the next level, using Brent–Kung parallel prefix network, the longest carry (i.e., G8:1) of the prefix network along with P8:1, which is the product of the all propagate signals of the inputs, are calculated sooner than other intermediate signals in this network. The signal P8:1 is used in the skip logic to determine if the carry output of the previous stage (i.e., CO,p−1) should be skipped or not. In addition, this signal is exploited as the predictor signal in the variable latency adder. It should be mentioned that all of these operations are performed in parallel with other stages. In the case, where P8:1 is one, CO,p−1 should skip this stage predicting that some critical paths are activated. On the other hand, when P8:1 is zero, CO,p is equal to the G8:1. In addition, no critical path will be activated in this case.

After the parallel prefix network, the intermediate carries, which are functions of CO,p−1 and intermediate signals, are computed. Finally, in the postprocessing level, the output sums of this stage are

calculated. It should be noted that this implementation is based on the similar ideas of the concatenation and incrementation concepts used in the CI-CSKA discussed. It should be noted that the end part of the SPL1 path from CO,p−1 to final summation results of the PPA block and the beginning part of the SPL2 paths from inputs of this block to CO,p belong to the PPA block. In addition, similar to the proposed CI-CSKA structure, the first point of SPL1 is the first input bit of the first stage, and the last point of SPL2 is the last bit of the sum output of the incrementation block of the stage Q. The steps for determining the sizes of the stages in the hybrid variable latency CSKA structure are similar to the ones discussed. Since the PPA structure is more efficient when its size is equal to an integer power of two, we can select a larger size for the nucleus stage accordingly. This implies that the third step discussed in that section is modified. The larger size (number of bits), compared with that of the nucleus stage in the original CI-CSKA structure, leads to the decrease in the number of stages as well smaller delays for SLP1 and SLP2.

(VI) SIMULATION RESULTS

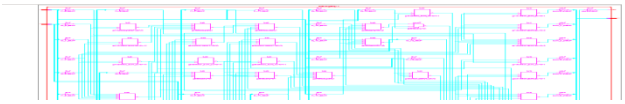

FIG 4 RTL Diagram Of Conventional Method

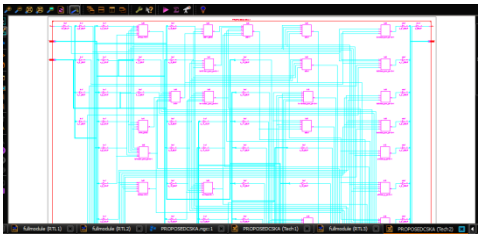

FIG 5 RTL Diagram Of Proposed Method

FIG 6 PERFORMANCE OF THE PROPOSED CONCEPT

(VII) CONCLUSIONS

In this paper, a static CMOS CSKA structure called CI-CSKA was proposed, which exhibits a higher speed and lower energy consumption compared with those of the conventional one. The speed enhancement was achieved by modifying the structure through the concatenation and incrementation techniques. In addition, AOI and OAI compound gates were exploited for the carry skip logics. The efficiency of the proposed structure for both FSS and VSS was studied by comparing its power and delay with those of the Conv-CSKA, RCA, CIA, SQRT-CSLA, and KSA structures. The results revealed considerably lower PDP for the VSS implementation of the CI-CSKA structure over a wide range of voltage from super-threshold to near threshold. The results also suggested the CI-CSKA structure as a very good adder for the applications where both the speed and energy consumption are critical. In addition, a hybrid variable latency extension of the structure was proposed. It exploited a modified parallel adder structure at the middle stage for increasing the slack time, which provided us with the opportunity for lowering the energy consumption by reducing the supply voltage. The efficacy of this structure was compared versus those of the variable latency RCA, C2SLA, and hybrid C2SLA structures. Again, the suggested structure showed the lowest delay and PDP making itself as a better candidate for high-speed low-energy applications.

REFERENCES

[1] I. Koren, Computer Arithmetic Algorithms, 2nd ed. Natick, MA, USA: A K Peters, Ltd., 2002.
[2] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy–delay optimization of 64-bit carry-lookahead adders with a 240 ps 90 nm CMOS designexample," IEEE J. Solid-State Circuits, vol. 44, no. 2, pp. 569–583, Feb. 2009.
[3] S. K. Mathew, M. A. Anders, B. Bloechel, T. Nguyen, R. K. Krishnamurthy, and S. Borkar, "A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS," IEEE J. Solid-State Circuits, vol. 40, no. 1, pp. 44–51, Jan. 2005.
[4] V. G. Oklobdzija, B. R. Zeydel, H. Q. Dao, S. Mathew, and R. Krishnamurthy, "Comparison of high-performance VLSI adders in the energy-delay space," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 754–758, Jun. 2005.
[5] B. Ramkumar and H. M. Kittur, "Low-power and area-efficient carry select adder," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 20, no. 2, pp. 371–375, Feb. 2012.
[6] M. Vratonjic, B. R. Zeydel, and V. G. Oklobdzija, "Low- and ultra low-power arithmetic units: Design and comparison," in Proc. IEEE Int. Conf. Comput. Design, VLSI Comput. Process. (ICCD), Oct. 2005, pp. 249–252.
[7] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 43, no. 10, pp. 689–702, Oct. 1996.
[8] Y. He and C.-H. Chang, "A power-delay efficient hybrid carrylookahead/ carry-select based redundant binary to two's complement converter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 1, pp. 336–346, Feb. 2008.