

# Design of 32-Bit Carry Skip Adder Using Binary to Excess-1 Converter

Biruntha.R, Govindaraj.V

PG Scholar, Department of ECE, KPR Institute of Engg. &Tech., Coimbatore, biruntha.r92@gmail.com  
Assistance Professor(Sr.G), Department of ECE, KPR Institute of Engg. &Tech., Coimbatore, [see1govind@gmail.com](mailto:see1govind@gmail.com)

**Abstract**— Adders are fundamental arithmetic component in many computer systems, since most of the research in the last few decades has concentrated on reducing the delay of addition. One of the most efficient adder architectures in terms of delay and power dissipation is the carry-skip adder. The existing CSKA structure uses concatenation and incrementation schemes to improve the speed and AOI (AND OR Invert) and OAI (OR AND Invert) compound gates are used instead of mux. From the structure of the CSKA, it is clear that there is scope for reducing the area and power consumption significantly from the existing structure. In this project, an area efficient 32-bit carry-skip adder to achieve lower power consumption is designed. This work uses a simple and efficient gate-level modification to lower the power and area of the CSKA. The optimum sizes for the skip logic blocks are decided by considering the delay of critical path. Based on this modification CSKA architecture have been developed and compared with the conventional CSKA architecture. The proposed design has reduced area and power by 18.7% and 87% respectively as compared with the regular CSKA with only a slight increase in the delay (0.5%). The proposed CSKA can be applied to any application where the adder is needed. In this project, the proposed CSKA was applied in FIR filter to reduce the area and the power. This work evaluates the performance using TANNER 0.25 $\mu$ m CMOS process technology.

**Keywords**— Carry Skip Adder(CSKA), CI-CSKA, Area-Efficient, Low Power, BEC, TANNER tool

## 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. 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 may choose between different adder structures/families for optimizing power and speed. There are many adder families with different delays, power consumptions, and area usages. Examples include ripple carry adder (RCA), carry increment adder (CIA), carry skip adder (CSKA), carry select adder (CSLA), and parallel prefix adders (PPAs). The RCA has the simplest structure with the smallest area and power consumption but with the worst critical path delay. In the CSLA, the speed, power consumption, and area usages are considerably larger than those of the RCA. The PPAs, which are also called carry look-ahead adders, exploit direct parallel prefix structures to generate the carry as fast as possible. There are different types of the parallel prefix algorithms that lead to different PPA structures with different performances. As an example, the Kogge–Stone adder (KSA) is one of the fastest structures but results in large power consumption and area usage. It should be noted that the structure complexities of PPAs are more than those of other adder schemes.

The CSKA is an efficient adder in terms of power consumption and area usage. The critical path delay of the CSKA is much smaller than the one in the RCA, whereas its area and power consumption are similar to those of the RCA. In addition, the power-delay product (PDP) of the CSKA is smaller than those of the CSLA and PPA structures. In addition, due to the small number of transistors, the CSKA benefits from relatively short wiring lengths as well as a regular and simple layout. The comparatively lower speed of this adder structure, however, limits its use for high-speed applications.

In this paper, given the attractive features of the CSKA structure, we have focused on reducing its delay by modifying its implementation based on the static CMOS logic. The proposed modification increases the speed considerably while maintaining the low area and power consumption features of the CSKA. In addition, an adjustment of the structure, based on the variable latency technique, which in turn lowers the power consumption without considerably impacting the CSKA speed, is also presented. The design of (hybrid) variable latency CSKA structures have been reported in the literature. Hence, the contributions of this paper can be summarized as follows.

- 1) Proposing a modified CSKA structure by combining the concatenation and the incrementation schemes to the conventional CSKA (Conv-CSKA) structure for enhancing the speed and power efficiency of the adder. The modification provides us with the ability to use simpler carry skip logics based on the AOI/OAI compound gates instead of the multiplexer.
- 2) Providing a design strategy for constructing an efficient CSKA structure based on analytically expressions presented for the critical path delay.
- 3) Proposing a hybrid variable latency CSKA structure based on the extension of the suggested CSKA, by replacing some of the middle stages in its structure with a PPA, which is modified in this paper.

## 2. 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. 1

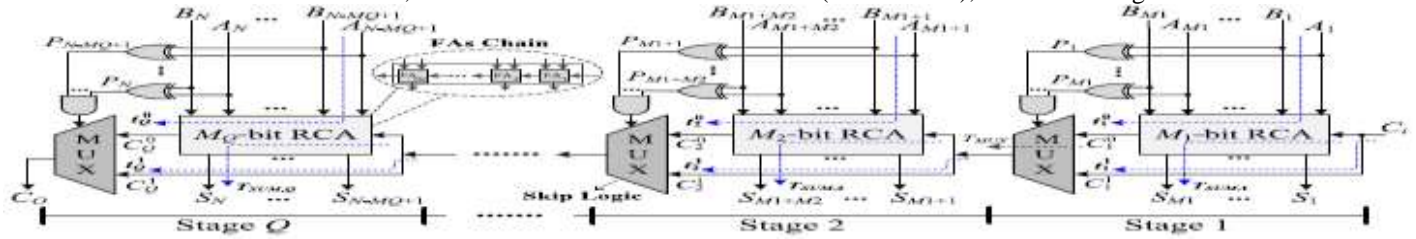


Fig -1: Conventional structure of the CSKA

In addition to the chain of FAs in each stage, there is a 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

$$P_i = A_i \oplus B_i = 1 \text{ for } i = 1, \dots, N$$

where  $P_i$  is the propagation signal related to  $A_i$  and  $B_i$ . This shows that the delay of the RCA is linearly related to  $N$  [1].

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  $M_j$  FAs ( $j = 1, \dots, Q$ ) 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 [2].

## 3. MODIFIED CSKA STRUCTURE

Based on the discussion presented in Section 2, it is concluded that by reducing the delay of the skip logic, one may lower the propagation delay of the CSKA significantly. Hence, in this paper, we present a modified CSKA structure that reduces this delay.

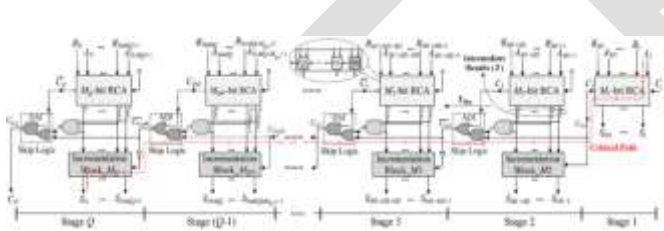


Fig -2: Proposed CI-CSKA Structure

### 3.1 General Description of the Proposed Structure

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 (Fig- 2). The gates, which consist of fewer transistors, have lower delay, area, and smaller power consumption compared with those of the 2:1 multiplexer [3]. 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-2 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  $M_j$  ( $j = 1, \dots, Q$ ). In this structure, the carry input of all the RCA blocks, except for the first block which is  $C_i$ , is zero (concatenation of the RCA blocks). Therefore, all the blocks execute their jobs simultaneously. 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.

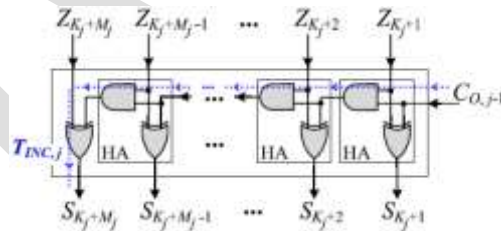


Fig-3: Internal structure of the  $j$ th incrementation block

As shown in Fig-2, if an AOI is used as the skip logic, the next skip logic should use OAI gate. In addition, 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 previous stage, the output carries of the blocks are calculated in parallel.

### 3.2 Area and Delay of the Proposed Structure

As mentioned before, the use of the static AOI and OAI gates (six transistors) compared with the static 2:1 multiplexer (12 transistors), leads to decreases in the area usage and delay of the skip logic[3],[4]. In addition, except for the first RCA block, this means that  $(Q - 1)$  FAs in the conventional structure are replaced with the same number of HAs in the suggested structure decreasing the area usage (Fig-2). In addition, note that the proposed structure utilizes incrementation blocks that do not exist in the conventional one. Therefore, the area usage of the proposed CI-CSKA structure is decreased compared with that of the conventional one.

### 3.3. Stage Sizes Consideration

Similar to the Conv-CSKA structure, the proposed CI-CSKA structure may be implemented with either FSS or VSS. Here, the stage size is the same as the RCA and incrementation blocks size. In the case of the FSS (FSS-CI-CSKA), there are  $Q = N/M$  stages with the size of  $M$ . The optimum value of  $M$ , which may be obtained using (1), is given by

$$M_{opt} = \sqrt{\frac{N(T_{AOI} + T_{OAI})}{2(T_{CARRY} + T_{AND})}} \quad (1)$$

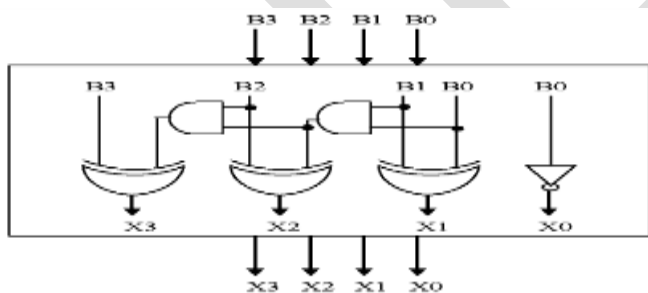
In the case of the VSS (VSS-CI-CSKA), the sizes of the stages obtained using some steps.

The size of the RCA block of the first stage is one. From the second stage to the nucleus stage, the size of stage is increased. The increase in the size is continued until the summation of all the sizes up to this stage becomes larger than  $N/2$ . The size of the last stage is one, and its RCA block contains a HA.

## 4. Proposed Area Efficient CSKA Structure

The basic work is to use Binary to Excess-1 Converter (BEC) in the CI-CSKA to get lower area and improved speed of operation. This logic is replaced in RCA with  $C_{in}=1$ . This logic can be implemented for different bits which are used in the modified design. The major advantage of this BEC logic comes from the fact that it uses lesser number of logic gates than the n-bit Full Adder (FA) structure. As stated above the main idea of this work is to use BEC instead of the RCA with  $C_{in}=1$  in order to decrease the area and increase the speed of operation in the regular CSKA to obtain modified CSKA. To replace the n-bit RCA, an n+1 bit BEC logic is required. The structure and the function table of a 4-bit BEC are shown in Figure 4 respectively.

Fig-4: Structure of the 4-Bit BEC



B[3:0]	X[3:0]
0000	0001
0001	0010
.	.
.	.
1110	1111
1111	0000

Fig-5: Function Table of 4-Bit BEC

Figure 5 shows the CI-CSKA structure with the modified ripple carry adder which consists of BEC. It is used instead of the RCA with  $C_{in}=1$ .

## 5. RESULT

### 5.1 Comparison of Different Parameters

Simulation was performed using the TANNER tool. Power and delay were directly obtained from software and area was measured in terms of number of transistors used in design.

**Table -1:** Comparison among different structures

Different structure/ Parameter	Conv- CSKA	CI-CSKA	Hybrid Variable Latency CSKA
Power(mW)	0.50	0.49	0.48
Delay(ns)	30.5	29.3	27.3
Area(transistor count)	1624	1762	1752

From this table we can conclude that the proposed area efficient CSKA consumes less power when compared to the other structures. The analysis also shows that the area is minimized.

### 5.2 Simulation Results

Using tanner software the output analysis for various structures of carry skip adder was obtained and it is shown in figure.

The input patterns applied to the carry skip adder structures are tabulated in table -2, 3, 4 and 5. By using these inputs the sum and carry values are generated.

**Table-2: Input pattern of A and B**

Bits	7	6	5	4	3	2	1	0
<b>A</b>	101 10	101 10	101 10	101 10	101 10	101 10	101 10	101 10
<b>B</b>	101 10	101 10	101 10	101 10	101 10	101 10	101 10	101 10

**Table-3: Input pattern of A and B**

Bits	15	14	13	12	11	10	9	8
<b>A</b>	101 10	101 10	101 10	101 10	101 10	101 10	101 10	101 10
<b>B</b>	111 11	111 11	111 11	111 11	111 11	111 11	111 11	111 11

**Table-4: Input pattern of A and B**

Bits	23	22	21	20	19	18	17	16
<b>A</b>	101 10	101 10	101 10	100 10	100 10	100 10	100 10	100 10
<b>B</b>	000 00	000 00	000 00	000 00	000 00	000 00	000 00	000 00

**Table 7.4: Input pattern of A and B**

Bits	31	30	29	28	27	26	25	24
<b>A</b>	000 00	000 00	000 00	000 00	000 00	101 10	101 10	101 10
<b>B</b>	110 01	110 01	110 01	110 01	110 01	110 01	110 01	110 01

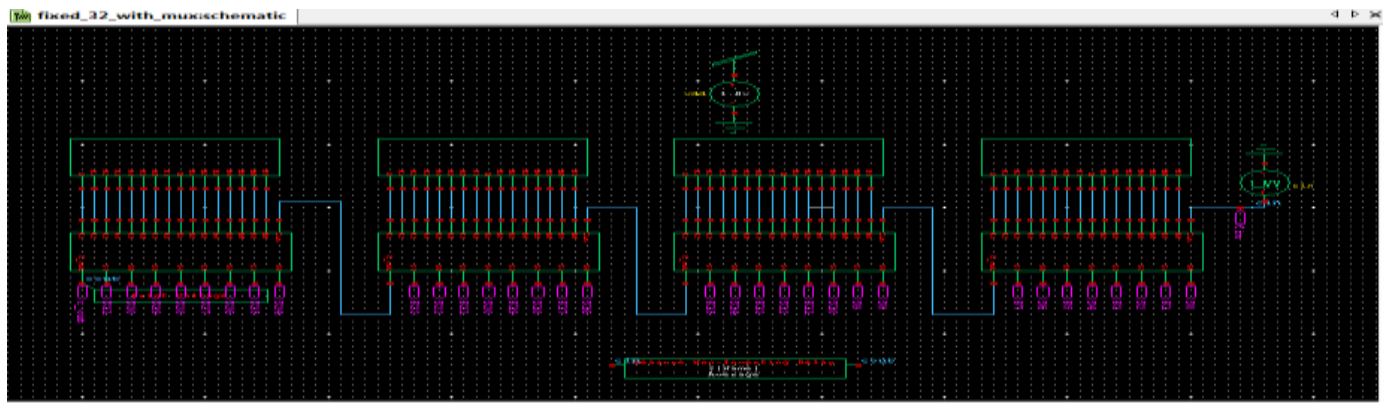


Fig -6: Conventional CSKA

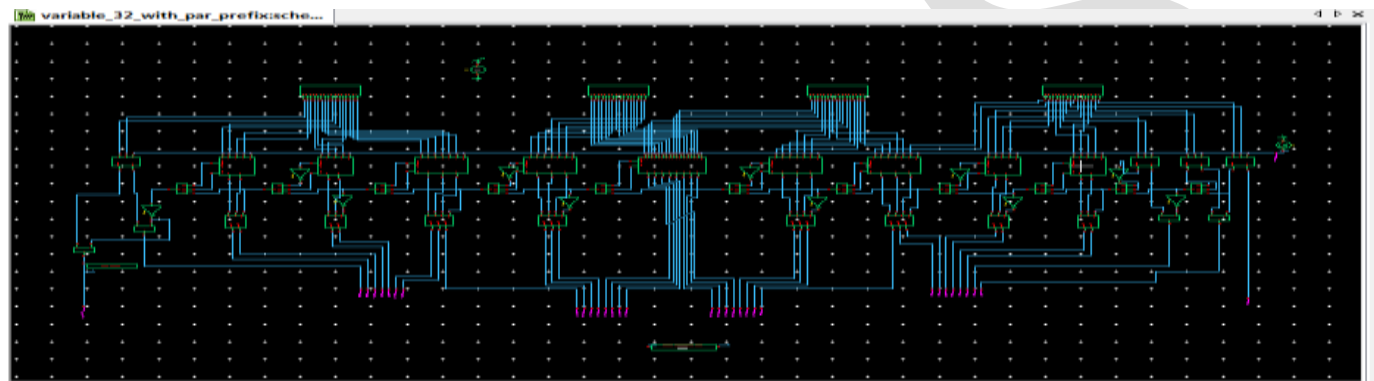


Fig -7: CI-CSKA. Stage size LSB to MSB {1,1,1,2,2,3,3,8,2,2,1}

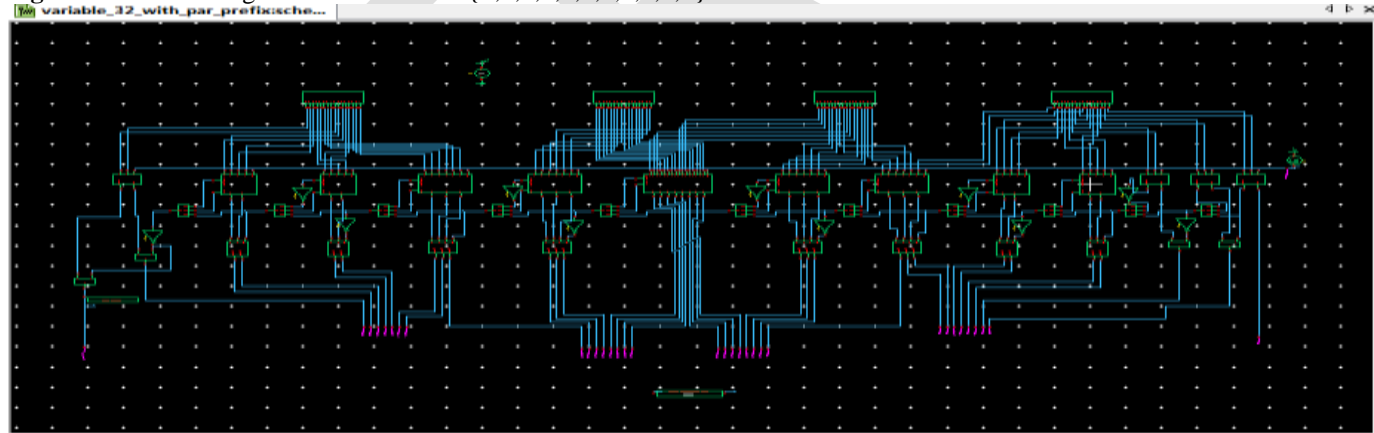


Fig -8: Proposed Area Efficient 32-bit CSKA

## CONCLUSION

A simple approach is proposed in this paper to reduce the area and power of CSKA architecture. The reduced number of gates of this work offers the great advantage in the reduction of area and also the total power. The compared results show that the modified CSKA has a slightly larger delay (only 0.5%), but the area and power of the 32-bit modified CSKA are significantly reduced by 18.7% and 87% respectively. The modified CSKA architecture is therefore, low area, low power, simple and efficient for VLSI hardware implementation.

## REFERENCES:

- [1] B. Parhami, "Computer Arithmetic Algorithms and Hardware Designs," Oxford Univ. Press, 2000.
- [2] S. Turrini, "Optimal group distribution in carry-skip adders," in Proc. 9th IEEE Symp. Comput. Arithmetic, Sep. 1989, pp. 96–103.
- [3] 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.
- [4] Y. Kim and L.-S. Kim, "64-bit carry-select adder with reduced area," Electron. Lett., vol. 37, no. 10, pp. 614–615, May 2001.
- [5] E. Gayles, R. M. Owens, and M. J. Irwin, "Low power circuit techniques for fast carry-skip adders," Proc. 1996 Midwest Symp. Circuits and Systems, pp. 87–90, Aug. 1996.
- [6] M. Alioto and G. Palumbo, "A simple strategy for optimized design of one-level carry-skip adders," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 1, pp. 141–148, Jan. 2003.
- [7] M. Lehman and N. Burla, "Skip techniques for high-speed carry-propagation in binary arithmetic units," IRE Trans. Electron. Comput., vol. EC-10, no. 4, pp. 691–698, Dec. 1961.
- [8] S. Majerski, "On determination of optimal distributions of carry skips in adders," IEEE Trans. Electron. Comput., vol. EC-16, no. 1, pp. 45–58, Feb. 1967.
- [9] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy-delay optimization of 64-bit carry-lookahead adders with a 240 ps 90 nm CMOS design example," IEEE J. Solid-State Circuits, vol. 44, no. 2, pp. 569–583, Feb. 2009.
- [10] 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.
- [11] Koren, Computer Arithmetic Algorithms, 2nd ed. Natick, MA, USA: A K Peters, Ltd., 2002.
- [12] M. Alioto and G. Palumbo, "A simple strategy for optimized design of one-level carry-skip adders," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 50, no. 1, pp. 141–148, Jan. 2003.
- [13] J. M. Rabaey, A. Chandrakasa, and B. Nikolic, Digital Integrated Circuits: A Design Perspective, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.
- [14] NanGate 45 nm Open Cell Library. [Online].
- [15] Available: <http://www.nangate.com/>, accessed Dec. 2010.
- [16] V. G. Oklobdzija, B. R. Zeydel, H. Dao, S. Mathew, and R. Krishnamurthy, "Energy-delay estimation technique for high-performance microprocessor VLSI adders," in Proc. 16th IEEE Symp. Comput. Arithmetic, Jun. 2003, pp. 272–279.
- [17] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," IEEE Trans. Comput., vol. C-22, no. 8, pp. 786–793, Aug. 1973.