International Journal of advancement in electronics and computer engineering (IJAECE)
Volume 5, Issue 7, November 2016, pp.1101-1105, ISSN 2278 -1412

Result Analysis of Carry Skip Adder using High-Speed Skips Logic at Different Levels

Rajesh sahu [1], Deepak Kumar [2]
[1] Mtech. Scholar, ECE, VIST, Bhopal, rajsahu512@gmail.com, India;
[2,] H.O.D, ECE, VIST, Bhopal, E-mail, India;

Abstract *– A carry skip adder (CSKA) structure has the high speed and very low power consumption. The speed of the structure is achieved by concatenation of all the blocks. The incrimination blocks are used to improve the efficiency of the carry skip adder structure. In existing method multiplexer logic is used, the proposed structure uses the AND-OR-Invert (AOI) and OR-AND-Invert (OAI) for the skip logic. The carry skip adder structure is realized with both fixed stage size and variable stage size where the delay is reduced, and speed is improved. A hybrid variable latency extension lowers the power consumption without affecting the speed of the circuit. The results are obtained using XILINX 14.3 and it gives improvements in the delay and energy of the structures. In addition to this structure, the power–delay product was low among all the structures, while having its energy–delay product was almost same as that of the conventional structure. Simulations on the proposed structure by using hybrid variable latency CSKA reduces the power consumption compared with the previous works and it produces a high speed.*

Keywords*: Carry skip adder (CSKA), energy efficient, high performance, hybrid variable latency Adders,*

I. Introduction

One of the effective techniques to lower the power utilization of digital circuits is to reduce the supply voltage because of quadratic dependence of the switching energy on the voltage. Moreover, the sub threshold current, that is the main leakage element in OFF devices, has an exponential dependence on the supply voltage level through the drain-induced barrier lowering impact. Depending on the amount of the supply voltage reduction, the operation of ON devices may reside in the super threshold, near-threshold, or sub threshold regions. Working in the super threshold region provides us with lower delay and better switching and leakage powers compared with the near/sub threshold regions. In the sub threshold region, the 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 nano-scale technologies. The variations increase uncertainties in the said performance parameters. Additionally, the small sub threshold current causes a large delay for the circuits operating in the sub threshold region.

Recently, the near-threshold region has been considered as a section that gives an additional desirable trade-off purpose between delay and power dissipation Compared with that of the sub threshold one, as a result of it leads to lower delay compared with the sub threshold region and considerably lowers switching and leakage powers compared with the super threshold region. Additionally, near-threshold operation, that uses supply voltage, levels near the threshold voltage of transistors, suffers significantly less from the process and environmental variations compared with the sub threshold region.

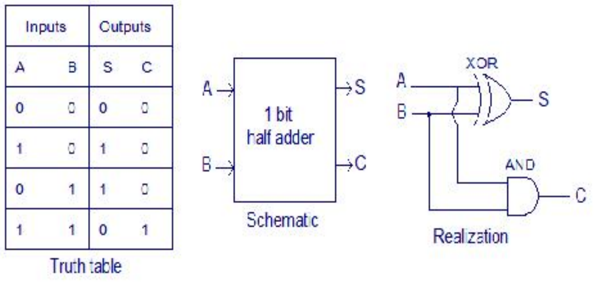
The dependence of the power (and performance) on the supply voltage has been the motivation for style of circuits with the feature of dynamic voltage and frequency scaling. In these circuits, to reduce the energy utilization, the system might change the voltage (and frequency) of the circuit based on the work requirement. For these systems, the circuit should be able to operate underneath a wide range of provide voltage levels. Of course, achieving higher speeds at lower provide voltages for the computational blocks, with the adder mutually the main components, could be crucial in the style of high-speed, yet energy efficient, processors. 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 utilization of processors. There are several works on the subject of optimizing the speed and power of these units that are reported in. Obviously, it's highly desirable to realize higher speeds at low-power/energy utilizations, which is a challenge for the designers of general purpose processors.

II. Theory

II.1. The Half adder

To understand what is a half adder you need to know what is an adder first. Adder circuit is a combinational digital circuit that is used for adding two numbers. A typical adder circuit produces a sum bit (denoted by S) and a carry bit (denoted by C) as the output. Typically adders are realized for adding binary numbers but they can be also realized for adding other formats like BCD (binary coded decimal, XS-3 etc. Besides addition, adder circuits can be used for a lot of other applications in digital electronics like address decoding, table index calculation etc.

Half adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit (S) and carry bit (C) as the output. If A and B are the input bits, then sum bit (S) is the X-OR of A and B and the carry bit (C) will be the AND of A and B. From this it is clear that a half adder circuit can be easily constructed using one X-OR gate and one AND gate. Half adder is the simplest of all adder circuit, but it has a major disadvantage. The half adder can add only two input bits (A and B) and has nothing to do with the carry if there is any in the input. So if the input to a half adder have a carry, then it will be neglected it and adds only the A and B bits. That means the binary addition process is not complete and that's why it is called a half adder. The truth table, schematic representation and XOR//AND realization of a half adder are shown in the figure below.



Inputs		Outputs	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Fig. 1 Truth Table, Schematic Representation And XOR//AND Realization Of A Half Adder

NAND gates or NOR gates can be used for realizing the half adder in universal logic and the relevant circuit diagrams are shown in the figure below.

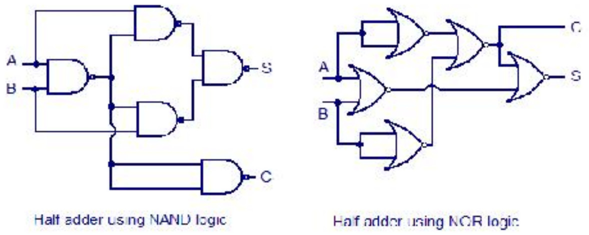


Fig. 2 Half Adder using NAND and NOR logic gate

II.2. Full Adder.

This type of adder is a little more difficult to implement than a half-adder. The main difference between a half-adder and a full-adder is that the full-adder has three inputs and two outputs. The first two inputs are A and B and the third input is an input carry designated as CIN. When full adder logic is designed we will be able to string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next. The output carry is designated as COUT and the normal output is designated as S. The truth-table is given below.

TABLE I Truth table of full adder

INPU TS		**OUTPUTS**		
A	**B**	**CIN**	**COUT**	**S**
0	**0**	**0**	**0**	**0**
0	**0**	**1**	**0**	**1**
0	**1**	**0**	**0**	**1**
0	**1**	**1**	**1**	**0**
1	**0**	**0**	**0**	**1**
1	**0**	**1**	**1**	**0**
1	**1**	**0**	**1**	**0**
1	**1**	**1**	**1**	**1**

From the above truth-table, the full adder logic can be implemented. We can see that the output S is an EXOR between the input A and the half-adder SUM output with B and CIN inputs. We must also note that the COUT will only be true if any of the two inputs out of the three are HIGH.

Thus, we can implement a full adder circuit with the help of two half adder circuits. The first will half adder will be used to add A and B to produce a partial Sum. The second half adder logic can be used to add CIN to the Sum produced by the first half adder to get the final S output. If any of the half adder logic produces a carry, there will be an output carry. Thus, COUT will be an OR function of the half-adder Carry outputs.

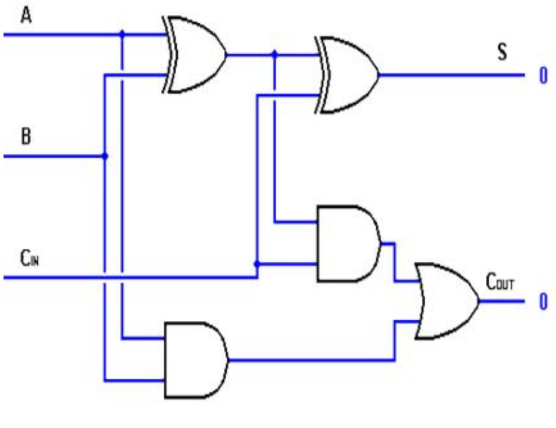


Fig. 3: Full Adder Circuit

II.3. Ripple carry adder

A ripple carry adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascaded, with the carry output from each full adder connected to the carry input of the next full adder in the chain. Figure 3 shows the interconnection of four full adder (FA) circuits to provide a 4-bit ripple carry adder. Notice from Figure 3 that the input is from the right side because the first cell traditionally represents the least significant bit (LSB). Bits a0 and b0 in the figure represent the least significant bits of the numbers to be added. The sum output is represented by the bits s0-s3.

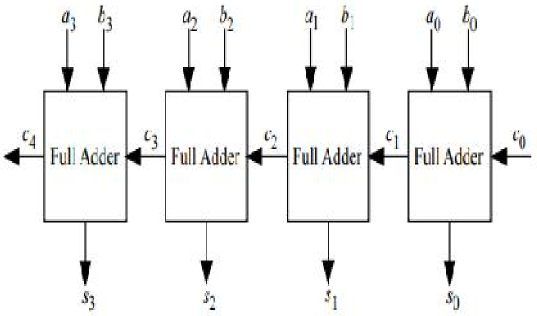


Fig: 4-Bit Full Adder.

II.4. Carry lookaheadadder (CLA)

The carry look ahead adder (CLA) solves the carry delay problem by calculating the carry signals in advance, based on the input signals. It is based on the fact that a carry signal will be generated in two cases: (1) when both bits a_i and b_i are 1, or (2) when one of the two bits is 1 and the carry-in is 1. Thus, one can write,

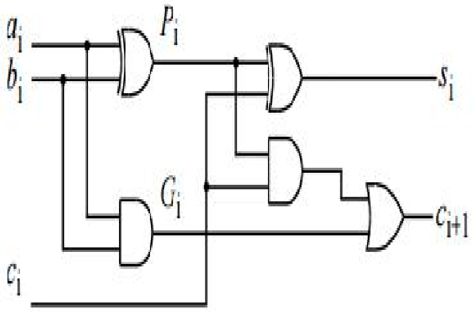


Fig 5: Full adder at stage i with p_i and g_i shown

$$c_{i+1} = a_i.b_i + (a_i \oplus b_i).c_i$$
$$s_i = (a_i \oplus b_i) \oplus c_i$$

The above two equations can be written in terms of two new signals p_i and g_i , which are shown in Figure 4:

III. Method

III.1. Proposed CSKA structure

III1.1. General description of the proposed structure

The structure is based on combining the concatenation and therefore the incrimination 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 skip logic contains the AOI (AND OR Invert) and OAI (OR AND Invert) compound gates for skip logic. 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. The basic idea behind using CSKA structures was based on almost balancing the delays of paths such that the delay of the critical path is minimized. Here we replace some of the middle stages in our proposed structure with a PPA. The proposed hybrid variable latency CSKA structure is shown in Figure 1 where in Mp-bit modified PPA is used for the path stage (nucleus stage). The nucleus stage, which has the largest size among the stages, replacing it by the PPA reduces the delay of the longest off-critical paths.

In addition, note that the projected structure utilizes incrementation blocks that don't exist within the conventional one. These blocks, however, is also implemented with about a similar logic gates (XOR and AND gates) as those used for generating the select signal of the multiplexer in the conventional structure. Therefore, the area usage of the projected CI-CSKA structure is decreased compared with that of the conventional one. The important path of the projected CI-CSKA structure, that contains 3 components, is shown in Fig. 6. These components include the chain of the FAs of the first stage, the path of the skip logics, and therefore the incrementation block in the last stage. The delay of this path (TD) could also be expressed as

$$T_D= [M_1T_{CARRY}] + [(Q-2)\ T_{SKIP}]+ [(M_Q-1)T_{AND} +T_{XOR}]$$

Where the three brackets correspond to the three parts mentioned above, respectively. Here, AND T and XOR T are the delays of the two inputs static AND and XOR gates, respectively.

.

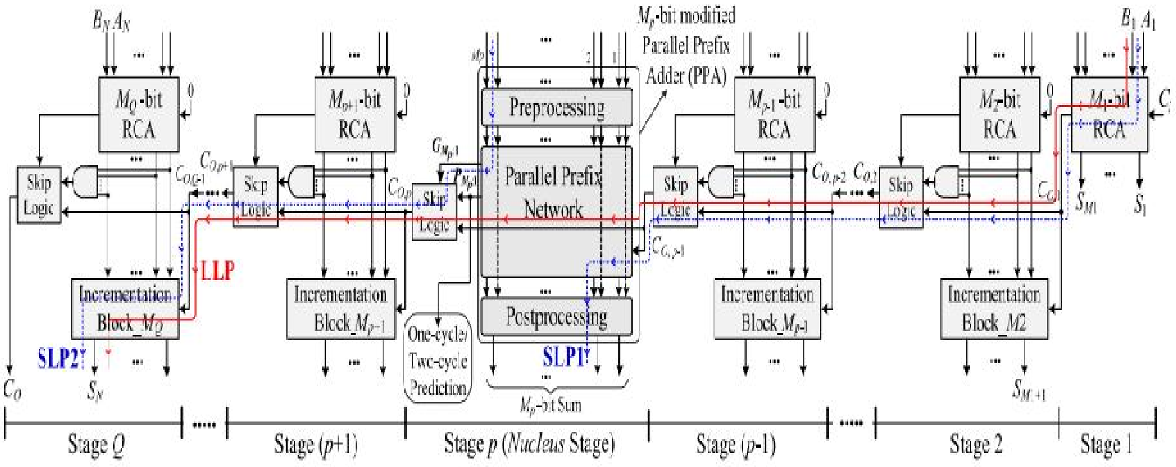


Fig.6. Structure of the proposed modified CSKA

IV. Result

The result analyses of the CSKA and simulation results of the projected system are shown in following figure. The design planned in this paper has been developed victimization MODEL machine. ADDERS are a main building block in ALUs (arithmetic and logic units).Low power arithmetic circuits turn into very necessary in VLSI industry. Adder circuit is that the main building block in DSP processor.

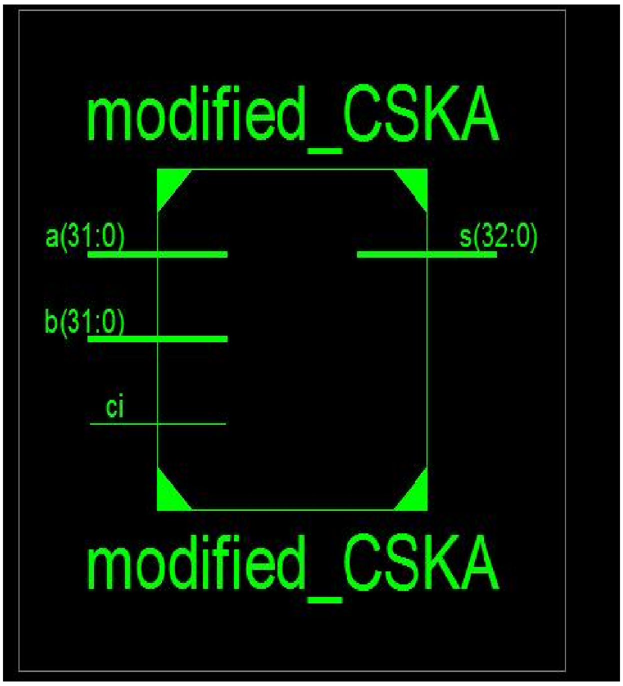


Fig. 7 Block diagram of modified CSKA

Block diagram representation of the Hybrid or proposed model of the CSKA system shown in Figure 7 in which we see three input and two output which is represented in as a sum and carry of the model which is represented as a S and C_o.

The black box or block diagram defines what are the inputs and outputs of the project or design

Inputs = a, b, ci

a,b consists of 32 bits

ci is the single bit

output = s

s consists of 33 bits

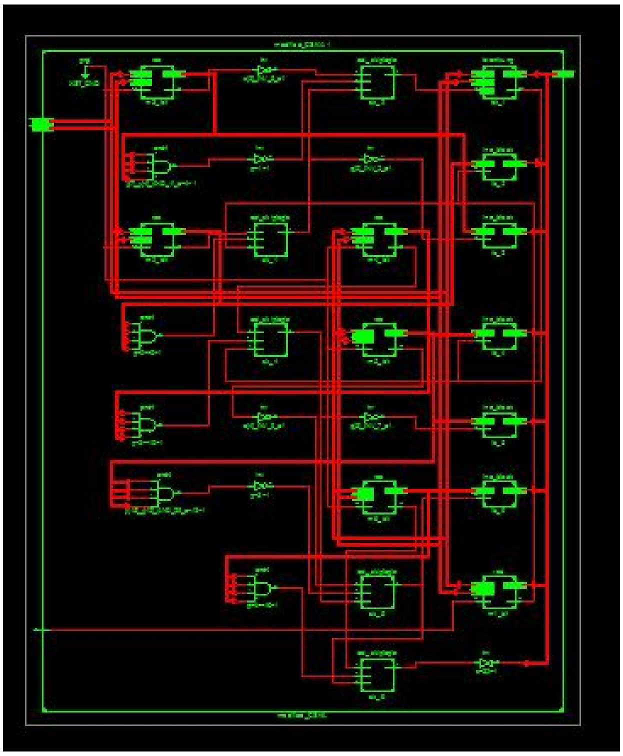


Fig.8 RTL schematic

The Register Transfer level schematic defines the code in (logic diagram) pictorial representation.

It includes rca block, incrementationblock,aoi skip logic and oai skip logic along with brent kung adder block, the code is written in Verilog, synthesized netlist is obtained from Xilinx too.

This schematic is generated after the HDL synthesis phase of the synthesis process. It shows a representation of the pre-optimized design in terms of generic symbols, such as adders, multipliers, counters, AND gates, and OR gates, that are independent of the targeted Xilinx device Which is shown in figure 8.

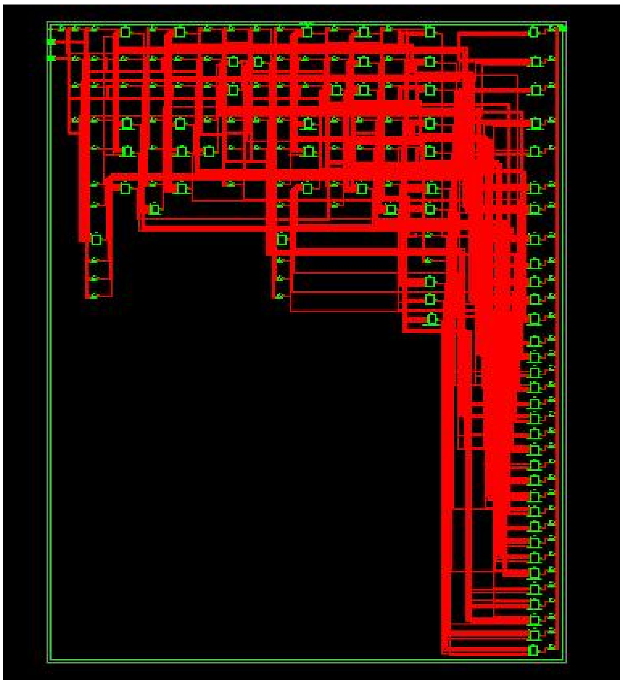


Fig.9 Technology schematic

In the above figure we can see the schematic representation of the applied technique in the CSKA technique. Which represents a applied AND, XOR logic gates.

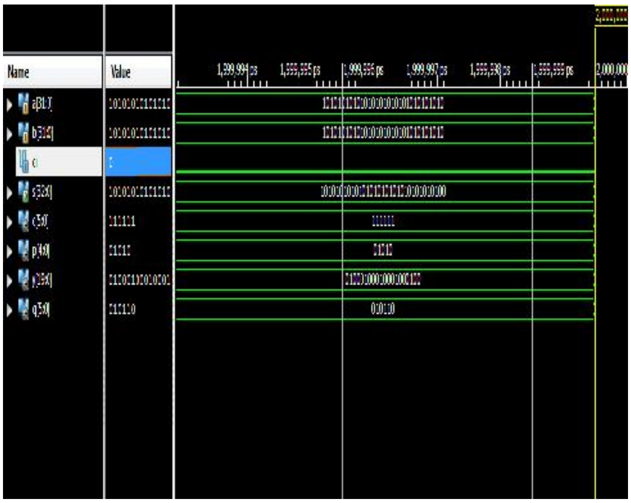


Fig. 10 Simulation results

Figure 10 represents the simulation results come after the applied schematic digram as all Parameters applied on it. Simulation result is comes out by the use of Xilinx software.

Comparison table

The conventional CSKA contains the number of LUTs 72 and delay is 15.386 ns.

The concatenation and incrimination CSKA contains the number of LUTs75 , delay 10.731 ns.

The proposed modified CSKA contains the number of LUTs 68 and delay 10.112 ns.

TABLE II Comparison table

	CONV -CSKA	CI-CSKA	MODIFIE D-CSKA
No. Of LUTs	72	75	68
No. Of bonded IOBs	98	98	98
Delay(ns)	15.386	10.731	10.112

V. Conclusion

In this paper, a static CMOS CSKA structure known as CI-CSKA was proposed, that exhibits a higher speed and lower energy consumption compared with those of the conventional one. The results also suggested the CI-CSKA structure as a really good adder for the applications wherever both the speed and energy consumption are crucial. Additionally, a hybrid variable latency extension of the structure was planned. It exploited a modified parallel adder structure at the middle stage for increasing the slack time, which provided us with the chance for lowering the energy consumption by reducing the supply voltage. The effectiveness 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 creating itself as a better condition for high-speed low-energy applications.

References

[1] Milad Bahadori, Mehdi Kamal, Ali Afzali-Kusha & Massoud Pedram "High-Speed and Energy-Efficient Carry Skip Adder Operating Under a Wide Range of Supply Voltage Levels ", IEEE transactions on very large scale integration (vlsi) systems, 2015.

[2] I. Koren, Computer Arithmetic Algorithms, 2nd ed. Natick, MA, USA:A K Peters, Ltd., 2002.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] M. Vratonjic, B. R. Zeydel, and V. G. Oklobdzija, "Low- and ultra low-power arithmetic units: Design and comparison," in Proc. IEEEInt. Conf. Comput. Design, VLSI Comput. Process. (ICCD), Oct. 2005, pp. 249–252.

[8] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders," IEEE Trans. Circuits Syst. II, Analog Digit. SignalProcess., vol. 43, no. 10, pp. 689–702, Oct. 1996.

[9] 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.

[10] C.-H. Chang, J. Gu, and M. Zhang, "A review of 0.18 μm full adder performances for tree structured arithmetic circuits," IEEE Trans. VeryLarge Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 686–695, Jun. 2005.

[11] D. Markovic, C. C. Wang, L. P. Alarcon, T.-T. Liu, and J. M. Rabaey, "Ultralow-power design in near-threshold region," Proc. IEEE, vol. 98, no. 2, pp. 237–252, Feb. 2010.

[12] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, andT. Mudge, "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," Proc. IEEE, vol. 98, no. 2, pp. 253–266, Feb. 2010.

[13] S. Jain et al., "A 280 mV-to-1.2 V wide-operating-range IA-32 processor in 32 nm CMOS," in IEEE Int. Solid-State Circuits Conf.Dig. Tech. Papers (ISSCC), Feb. 2012, pp. 66–68.