

Design for Low Power Multiplier Based On Fixed Width Replica Redundancy Block & Compressor Trees

Mariya Stephen¹, Vrinda²

¹M.Tech Student, Department of Electronics and Communication Engineering
SCMS School of Engineering and Technology, Karukuuty, Cochin, Kerala, India

²Assistant Professor, Department of Electronics and Communication Engineering
SCMS School of Engineering and Technology, Karukuuty, Cochin, Kerala, India

Abstract: *This paper establishes designing multipliers that are of high-speed, low power, and regular in layout are of substantial research interest. Multiplier speed can be increased by reducing the generated partial products. Many attempts are done to reduce the number of partial products generated in a multiplication process. One of them is Wallace tree multiplier. Wallace Tree CSA structures are used to sum the partial products in reduced time. Speed can be increased by incorporating compressors with wallace tree technique. Therefore, minimizing the number of half adders used in a multiplier which will reduce the circuit complexity.*

Keywords: Carry save adders (CSA)

1. Introduction

Portable and wireless computing systems are widely growing nowadays. This establishes a need for low power systems. To lower the power dissipation, supply voltage can be thought to scale down, since the power consumption in CMOS circuits is directly connected to the square of supply voltage. However, in deep-submicrometer process technologies, noise interference problems occur. This has increased difficulty to design the reliable and efficient microelectronics systems, hence such design techniques to enhance noise tolerance have been widely developed. A low-power technique, referred to as voltage overscaling (VOS), was proposed to lower supply voltage below critical supply voltage so that throughput is not sacrificed. However, VOS leads to severe degradation in signal-to-noise ratio (SNR). Another technique involves algorithmic noise tolerant (ANT) technique along with VOS main block with reduced-precision replica (RPR), which combats soft errors and helps achieve significant energy saving. However, the RPR designs in the ANT designs are designed in a special manner, which are not easily adopted and repeated. Multiplication is the most critical operation in every computational system

Multiplication is the most important operation in every computational system. Graphics and Process control are two areas where in the multiplier performance plays a crucial role. The bottlenecks posed by multiplication are both temporal and spatial in nature. Hence the only possible alternatives are in the form of Application Specific Integrated Circuits (ASIC) or the DSP processors which are used to address the latency demands of such computationally intensive applications. The multiplier structure will vary depending on the output necessities of the application. The first step of the design process is to select optimum design structure. There are various structures to perform the multiplication operation which starts from serial multipliers

and range up to complex parallel multipliers. Any sought of speed improvement in the multiplier will enhance the frequency of the DSP or can be traded for energy by optimizing circuit sizes and the voltage supply

2. Literature Survey

The fixed-width designs adopted in DSP applications to avoid infinite growth of bit width. Cutting off n-bit LSB output is used to construct a fixed-width DSP with n-bit input and n-bit output. The circuit complexity and consumption of power of a fixed-width DSP is usually about half of the full-length one. However, cutting of LSB part will result in rounding error, which should be dealt separately and compensated precisely. Many literatures have been presented to decrease the truncation error with constant correction value or with variable correction value. The hardware complexity to compensate with constant correction value will be much more simpler than that of variable correction value; but the variable correction method are usually more accurate usually compensation method is to compensate the truncation error between the full-length multiplier and the fixed-width multiplier. In fixed-width RPR of an ANT multiplier, the compensation error we need to correct is the overall truncation error of MDSP block. Unlike, our compensation method is to compensate the truncation error between the full-length Main DSP multiplier and the fixed-width RPR multiplier. However nowadays, there are many fixed-width multiplier structures applied to the full-width multipliers. However, there is still no fixed-width RPR design applied to the ANT multiplier designs.

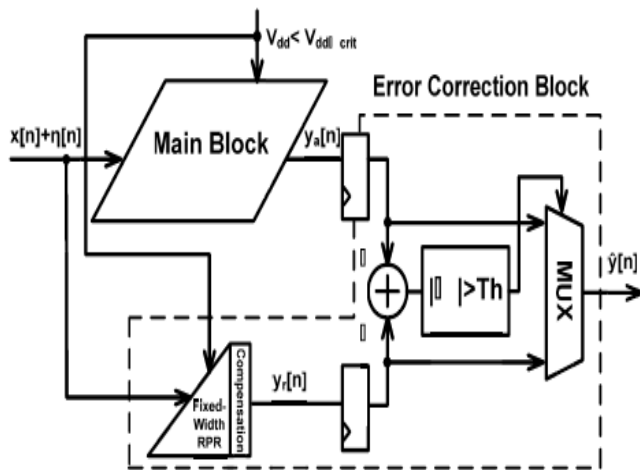


Figure 1: ANT architecture with fixed width RPR

Precise Error Compensation Vector for Fixed-Width RPR Design

In the design, the purpose of RPR is to correct the errors occurring in the output of MDSP and maintain the SNR of whole system and to lower supply voltage. In the case of using fixed-width RPR to realize ANT architecture, we are not only concerned with lowering circuit area and but also deal with power consumption, it also accelerate the computation efficiency as compared with the conventional full-length RPR. However, we need to compensate huge truncation error due to truncating off many hardware elements in the LSB part of MDSP.

The partial product array can be divided into four subsets, which are most significant part (MSP), input correction vector ICV(β), minor ICV [MICV(α)], and LSP. In this design only MSP part is kept and the other parts are removed. Therefore, the other three parts of ICV(β), MICV(α), and LSP are truncated part. The truncated ICV(β) and MICV(α) are the most important parts because they are of highest weight. Thus, they can be applied to construct the truncation error compensation algorithm

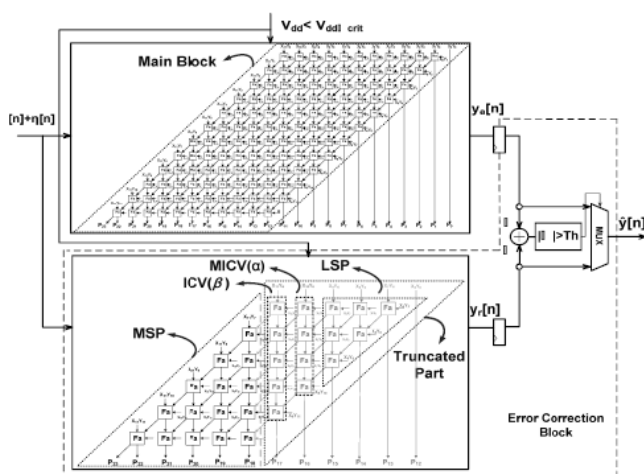


Figure 2: 12 × 12 bit ANT multiplier is implemented with the six-bit fixed width replica redundancy block.

β can be thought as the summation of all partial products of ICV. More precisely, as $\beta = 0$, the average error is close to $\beta + 1$. As $\beta > 0$, the average error is very close to β . If we can select β as the compensation vector, the compensation vector can directly inject into the fixed-width RPR as compensation, which does not need extra compensation logic gates. Now analyze the compensation accuracy by selecting β as the compensation vector. We can find that the absolute average error in $\beta = 0$ is greater than that in other β cases. Therefore, we can apply multiple input error compensation vectors to improve truncation precision. For the $\beta > 0$ case, we can still select β as the compensation vector. For the $\beta = 0$ case, we select $\beta + 1$ with MICV as the compensation vector.

β can be thought as the summation of all partial products of ICV. More precisely, as $\beta = 0$, To realize the fixed-width RPR, we build one directly injecting ICV(β) to basically meet the statistic distribution and one minor compensation vector MICV(α) to amend the insufficient error compensation cases. The compensation vector ICV(β) is realized by directly injecting the partial terms of $X_{n-1}Y_{n/2}$, $X_{n-2}Y_{(n/2)+1}$, $X_{n-3}Y_{(n/2)+2}$, ..., $X_{(n/2)+2}Y_{n-2}$. These directly injecting compensation terms are labeled as $C_1, C_2, C_3, \dots, C_{(n/2)-1}$ in Fig. The other compensation vector used to mend the insufficient error compensation case is constructed by one conditionally controlled OR gate. One input of OR gate is injected by $X_{(n/2)}Y_{n-1}$, that is designed to realize the function of compensation vector β . The next input is being conditionally controlled by the judgment formula used to judge whether $\beta = 0$ and $\beta_1 = 0$ as well. As shown in Fig the term C_{m1} is used to judge whether $\beta = 0$ or not. The judgment function circuit comprises one NOR gate and its inputs are $X_{n-1}Y_{n/2}$, $X_{n-2}Y_{(n/2)+1}$, $X_{n-3}Y_{(n/2)+2}$, ..., $X_{(n/2)+2}Y_{n-2}$. The term C_{m2} is used to judge whether $\beta_1 = 0$. The judgment function is realized by one OR gate, while its inputs are $X_{n-2}Y_{n/2}$, $X_{n-3}Y_{(n/2)+1}$, $X_{n-4}Y_{(n/2)+2}$, ..., $X_{(n/2)+1}Y_{n-2}$. If both are true, a compensation term C_m is generated via a two-input AND gate. Then, C_m is injected together with $X_{(n/2)}Y_{n-1}$ into a two-input OR gate to correct the insufficient error compensation. Accordingly, in the case of $\beta = 0$ and $\beta_1 = 0$ as well, one additional carry-in signal $C_{(n/2)}$ is injected into the compensation vector to modify the compensation value as $\beta + 1$ instead of β . Moreover, the carry-in signal $C_{(n/2)}$ is injected in to the bottom of the error compensation vector, farthest location away from the critical path. Therefore, not only the error compensation precision in the fixed-width RPR can be enhanced, the computation delay will also not be postponed. Since the critical supply voltage is dominated by the critical delay time of the RPR circuit, preserving the critical path of RPR where delay factor is very important. Finally, the proposed high-precision fixed-width RPR multiplier circuit is shown in fig. In our presented fixed-width RPR design, the adder cells can be saved by half as compared with the conventional full-width RPR. Moreover, the proposed high-precision fixed-width RPR design can even provide higher precision as compared with the full-width RPR design.

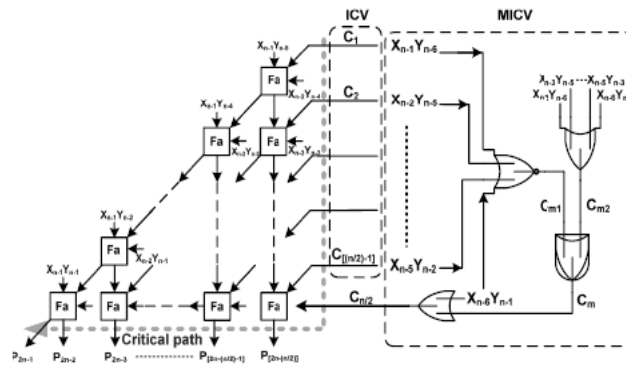


Figure 3: high-accuracy fixed-width RPR multiplier with compensation constructed by the multiple truncation EC vectors combined ICV together with MICV

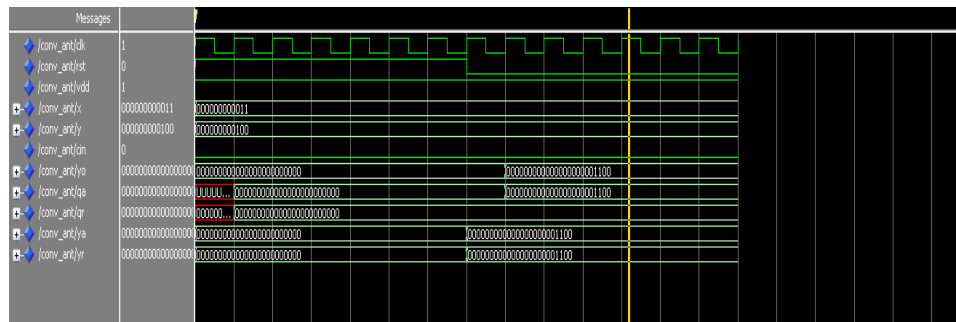


Figure 4: Conventional ANT simulation

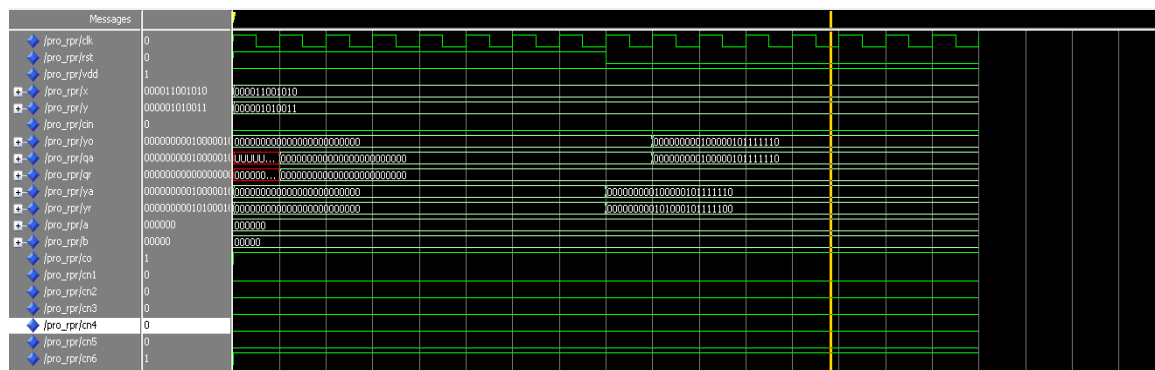


Figure 5: Fixed width RPR simulation

3. Wallace Tree Multiplier with Compressors

Basically multiplier consists of three parts 1. partial product generation 2. partial product addition and 3. final adding part. A multiplier essentially consists of two operands, a multiplicand "Y" and a multiplier "X" and produces a product. Initially x & y are multiplied bit by bit to generate the partial products. Second stage is the most important, because it consists of the most complicated and determines the speed of the overall multiplier to add these partial products to generate the Product "P". Modification will be focused on the optimization of this stage, it contains the addition of all the partial products. In case speed is not of main concern then partial products can be added serially, it reduces the circuit complexity. However, in high-speed design, in the Wallace tree construction method addition of partial products occurs in a tree-like fashion that produces two rows of partial products which can be added in the final stage. Although fast, since its critical path delay is directly in proportion with the logarithm of the number of bits in the multiplier, the Wallace tree introduces other problems such

as layout area wastage and increased complexity. So compressor trees are used to perform high-speed addition with less area complexity. In the last stage, addition can be performed by using a high-speed adder for example carry save adder to generate the output result.

A. Wallace tree multiplier

In this technique, a three-step process is used to multiply 2 numbers; products are formed, the bit product matrix is reduced to a two-row matrix where the sum of the row will be equal to the sum of bit products, & resulting rows are summed with a fast adder to generate a final product. In the Wallace Tree method, three-bit signals are passed to a one-bit full adder ("3W") this is called a 3-input Wallace Tree circuit, and the output signal (sum signal) is supplied to the coming stage full adder of the same bit and carry output signal thereof is passed to the next stage full adder of the same number of bits, & then the carry output signal will be supplied to the next stage of the full adder located at a one-bit higher position.

B. Compressors

Compressors- arithmetic components, which is similar in principle to parallel counters, but with two distinct differences: (1) explicit carry-in and carry-out bits; and (2) there may be some redundancy among the ranks of the sum and c

1) 4:2 Compressor

The 4:2 compressor it has 4 input bits and produces 2 sum output bits (out0 and out1), it also has a carry-in and a carry-out bit (thus, the total number of input/output bits are 5 and 3); All input bits, including cin, will posses rank 0; the two output bits will posses ranks 0 and 1 respectively, while cout

has rank 1 as well. 4:2 compressor output will be a redundant- number; for example, out1 = 0 and cout = 1 is equivalent to out1 = 1 and cout = 0 in all cases

2) 5:2 Compressor

It has 5 input bits and produces 2 sum output bits (sum and cout3), it also has a carry-in (cin1, cin2 and a cout1, cout2,cout3 bit thus, the total number of input/output bits are 7 and 4); All input bits, including cin1 will have rank 0 and cin2 has rank 2 ; the two output bits have ranks 0 and 1 respectively, while cout2 has rank 1 and cout1 has rank 2

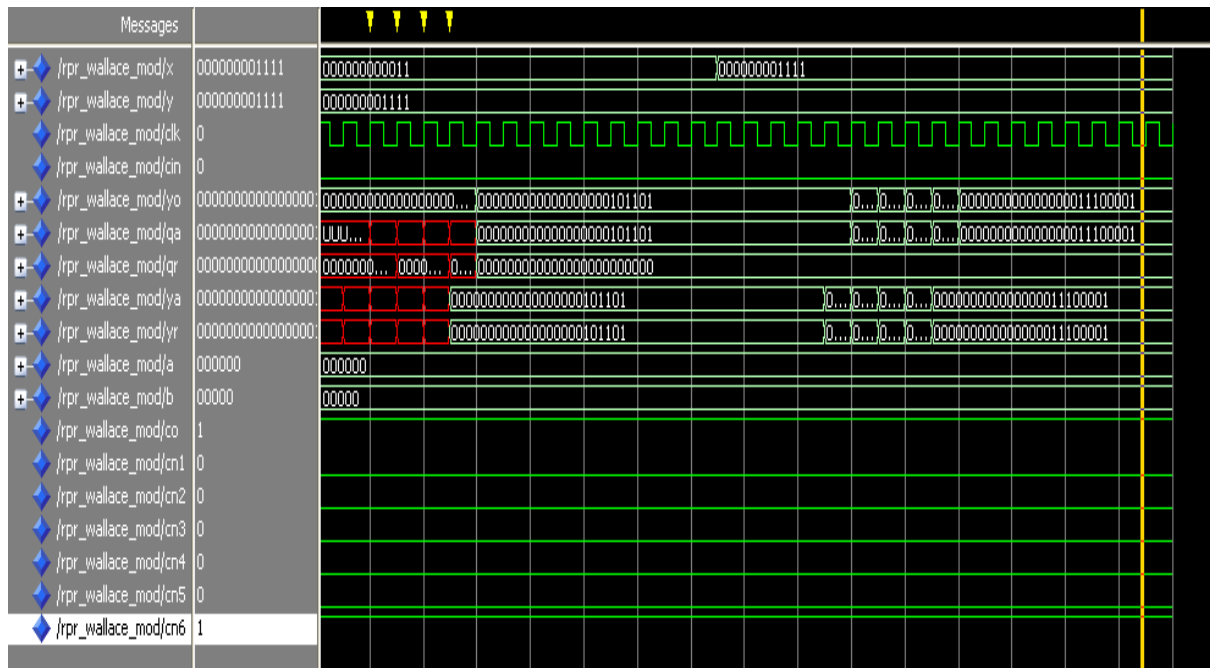


Figure 6: Simulation of compressor output

4. Application

Multiply & accumulate unit

Multiply accumulate unit(MAC) uses low power truncated baugh-wooley & compressor tree multiplier is also used and compare the performance. In computing, especially digital signal processing, the multiply-accumulate operation is a usual step that computes the product of 2 numbers and adds that product to an accumulator. The circuit performing this operation is called multiplier-accumulator; The MAC operation modifies an accumulator a:

$$a \leftarrow a + (b \times c)$$

When done with floating point numbers, it might be performed with 2 roundings (typical in many DSPs), or with a single rounding. When performed with a single rounding, it is termed as a fused multiply-add or fused multiply-accumulate. . Modern computers may contain a dedicated MAC, which consists of a multiplier implemented in combinational logic followed by an adder and an accumulator register that holds the result. Now output of the register is fed back to one input of the adder, so that on each clock cycle, multiplier output is added to the register. Combinational multipliers require a large amount of logic, but product calculated quickly than the method of shifting

and adding typical of earlier computers. The 1st processors having MAC units were DSPs, now it is common in general-purpose processors.

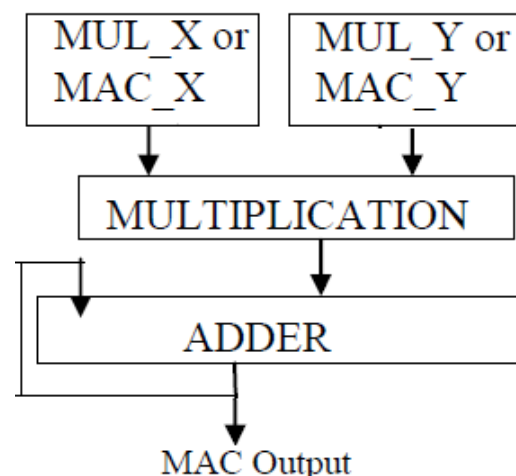


Figure 7: MAC

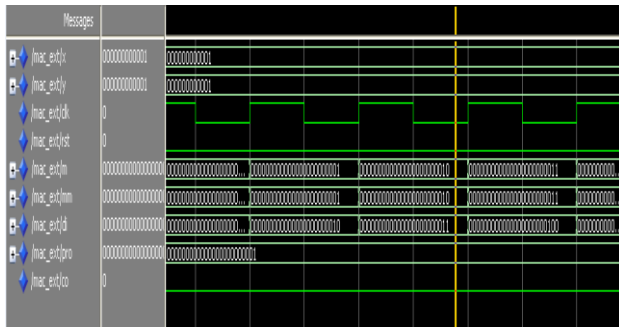


Figure 8: Simulation of MAC unit

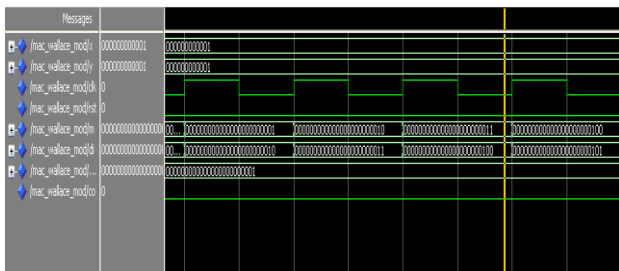


Figure 9: MAC using compressor trees

5. Performance Comparison

DELAY

Table 1: Delay comparison

	Fixed width RPR	RPR with compressor tree
Minimum period	41.488ns	10.032ns
Minimum input arrival time before clock	4.736ns	4.376ns
Maximum output time after clock	6.140ns	7.157ns
Maximum combinational path delay	No path found	No path found

AREA

Table 2: Area comparison

	Fixed width RPR	RPR with compressor tree
Number of Slice Flip Flops	132 out of 13,824	294 out of 13,824
IOB flipflops	36	12
Total equivalent gate count for design	4671	4032
Peak Memory Usage	138MB	152MB

POWER

Table 3: Power Consumption

	Fixed width RPR	Fixed width RPR with compressor tree
Power (mW)	58mW	63mW

6. Conclusion

Fixed-width RPR-based ANT multiplier design is presented. The Wallace tree multipliers can be solved & analysed using a new version of Wallace tree construction using compressors. The modified tree has a slightly smaller critical path, a little higher wiring overhead but gives high speed. This modified design of multiplier which consist of

7:2 compressor, 6:2 compressor, 5:2 compressor, 4:2 compressor, 3:2 compressor, full adders and reduced no. of half adder and reduces the complexity and reduce the time delay. Multiplier using Compressor have small increase in area and power but the time delay is lower to conventional Wallace Tree Multiplier. As the Compressor order is increased the time delay reduces respectively. Hence for small delay requirement Wallace Tree Multiplier using compressor is used.

References

- [1] I-Chyn Wey, Chien-Chang Peng, and Feng-Yu Liao "Reliable Low-Power Multiplier Design Using Fixed-Width Replica Redundancy Block" IEEE transactions on very large scale integration (vlsi) systems, vol. 23, NO. 1, JANUARY 2015
- [2] B. Shim, S. Sridhara, and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 497–510, May 2004.
- [3] B. Shim and N. R. Shanbhag, "Energy-efficient soft-error tolerant digital signal processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 4, pp. 336–348, Apr. 2006.
- [4] R. Hedge and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in Proc. IEEE Int. Symp. Low Power Electron. Des., Aug. 1999, pp. 30–35.
- [5] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.
- [6] Y. Liu, T. Zhang, and K. K. Parhi, "Computation error analysis in digital signal processing systems with overscaled supply voltage," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 4, pp. 517–526, Apr. 2010.
- [7] J. N. Chen, J. H. Hu, and S. Y. Li, "Low power digital signal processing scheme via stochastic logic protection," in Proc. IEEE Int. Symp. Circuits Syst., May 2012, pp. 3077–3080.
- [8] J. N. Chen and J. H. Hu, "Energy-efficient digital signal processing via voltage-overscaling-based residue number system," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 7, pp. 1322–1332, Jul. 2013.
- [9] P. N. Whatmough, S. Das, D. M. Bull, and I. Darwazeh, "Circuit-level timing error tolerance for low-power DSP filters and transforms," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 6, pp. 12–18, Feb. 2012.
- [10] G. Karakonstantis, D. Mohapatra, and K. Roy, "Logic and memory design based on unequal error protection for voltage-scalable, robust and adaptive DSP systems," J. Signal Process. Syst., vol. 68, no. 3, pp. 415–431, 2012.
- [11] Y. Pu, J. P. de Gyvez, H. Corporaal, and Y. Ha, "An ultra low energy/frame multi-standard JPEG co-processor in 65-nm CMOS with sub/near threshold

power supply,” IEEE J. Solid State Circuits, vol. 45, no. 3, pp. 668–680, Mar. 2010.

- [12] H. Fuketa, K. Hirairi, T. Yasufuku, M. Takamiya, M. Nomura, H. Shinohara, et al., “12.7-times energy efficiency increase of 16-bit integer unit by power supply voltage (VDD) scaling from 1.2V to 310mV enabled by contention-less flip-flops (CLFF) and separated VDD between flip-flops and combinational logics,” in Proc. ISLPED, Fukuoka, Japan, Aug. 2011, pp. 163–168.

Author Profile



Mariya Stephen received the B.Tech degree in Electronics And Communication Engineering from Mahatma Gandhi University, Kerala at Federal Institute of Science and Technology 2013 and now she is pursuing her M.Tech degree in VLSI and Embedded systems under the same university in SCMS School of Engineering and Technology, Cochin.