IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
Vol. 8, Issue 8, August 2020

DOI 10.17148/IJIREEICE.2020.8811

# Low Transition Approximate Multiplier with Multi-mode Error Recovery

**B. Srinivasan[1], Dr.M. Jayasheela[2]**

PG Scholar, Kalaignar Karunanidhi Institute of Technology, Coimbatore, Tamilnadu, India[1]

Professor, Department of ECE, Kalaignar Karunanidhi Institute of Technology, Coimbatore, Tamilnadu, India[2]

**Abstract:** To implement Approximate computing has emerged as a potential solution for the design of energy-efficient digital systems. Applications such as multimedia, recognition and data mining are inherently error-tolerant and do not require a perfect accuracy in computation. Multipliers are key arithmetic circuits in many of these applications including Digital Signal Processing (DSP). Here a novel approximate multiplier design with low power consumption and a short critical path is proposed for high-performance DSP applications. This multiplier leverages a newly designed Spurious-Power Suppression Technique (SPST) approximate adder that limits its carry propagation and dramatically reduces the power dissipation of combinational VLSI designs for multimedia/DSP purposes. The proposed SPST separates the target designs into two parts, i.e., the Most Significant Part and Least Significant Part (MSP and LSP), and turns off the MSP when it does not affect the computation results to save power. Different levels of accuracy can be achieved by using either OR gates or the proposed approximate adder in a configurable error recovery circuit. The approximate multipliers using these two error reduction strategies are referred to as AM1 and AM2, respectively. Both AM1 and AM2 have a low mean error distance, i.e., most of the errors are not significant in magnitude. AM2 has a better accuracy compared with AM1 but with a longer delay and higher power consumption. Image processing applications, including image sharpening and smoothing, are considered to show the quality of the approximate multipliers in error-tolerant applications. By utilizing an appropriate error recovery scheme, the proposed approximate multipliers achieve similar processing accuracy as exact multipliers, but with significant improvements in power.

**Keywords:** Approximate Multiplier, SPST adder and error recovery.

## I. INTRODUCTION

Approximate computing has emerged as a potential solution for the design of energy-efficient digital systems. Applications such as multimedia, recognition and data mining are inherently error-tolerant and do not require a perfect accuracy in computation [1]. For Digital Signal Processing (DSP) applications, the result is often left to interpretation by human perception. Therefore, strict exactness may not be required and an imprecise result may suffice due to the limitation of human perception. For these applications, approximate circuits play an important role as a promising alternative for reducing area, power and delay, thereby achieving better performance in energy efficiency. In this paper [2], a novel approximate multiplier design is proposed using a simple, yet fast approximate adder. This newly designed adder can process data in parallel by cutting the carry propagation chain. It has a critical path delay that is shorter than a conventional one-bit full adder. Albeit with a high error rate, this adder simultaneously computes the sum and generates an error signal; this feature is employed to reduce the error in the final result of the multiplier. In the proposed approximate multiplier, a simple tree of the approximate adders is used for partial product accumulation and the error signals are used to compensate errors for obtaining a better accuracy. The proposed multiplier can be configured into two designs by using OR gates and the proposed approximate adders for error reduction, referred to as Approximate Multiplier 1 (AM1) and Approximate Multiplier 2 (AM2), respectively. Different levels of error recovery can also be achieved by using a different number of MSBs for error recovery in both AM1 and AM2. As per the analysis, the proposed multipliers have significantly shorter critical paths and lower power dissipation.

Functional and circuit simulations are performed to evaluate the performance of the multipliers. Experimental results indicate that the proposed approximate multipliers perform well in these error-tolerant applications. Section II presents the proposed approximate adder and the design of the multiplier. Section III discusses the error reduction schemes for AM1, AM2 and SPST. Section IV shows the accuracy results and in delay, area and power consumption are and also compares the proposed approximate multipliers with the existing designs in terms of accuracy and hardware costs. Section V discusses the application of the proposed multiplier to image processing applications and concludes the paper.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
Vol. 8, Issue 8, August 2020

DOI 10.17148/IJIREEICE.2020.8811

## II. PROPOSED APPROXIMATE MULTIPLIER

### A. Approximate adder

In this section, the design of a new approximate adder is presented. Approximate half-adder, full-adder is proposed for their accumulation. Carry and Sum are two outputs of these approximate circuits [3],[4]. Since Carry has higher weight of binary bit, error in Carry bit will contribute more by producing error difference of two in the output. Approximation is handled in such a way that the absolute difference between actual output and approximate output is always maintained as one. Hence Carry outputs are approximated only for the cases, where Sum is approximated. In adders and compressors, XOR gates tend to contribute to high area and delay.

$$Sum = x1 + x2$$
$$Carry = x1 \cdot x2 \qquad (1)$$

Table I Truth Table of Approximate Half Adder

| Inputs | | Exact Outputs | | Approximate Outputs | | Absolute Difference |
|---|---|---|---|---|---|---|
| $x1$ | $x2$ | $Carry$ | $Sum$ | $Carry$ | $Sum$ | |
| 0 | 0 | 0 | 0 | 0 ✔ | 0 ✔ | 0 |
| 0 | 1 | 0 | 1 | 0 ✔ | 1 ✔ | 0 |
| 1 | 0 | 0 | 1 | 0 ✔ | 1 ✔ | 0 |
| 1 | 1 | 1 | 0 | 1 ✔ | 1 ✗ | 1 |

For approximating half-adder, XOR gate of Sum is replaced with OR gate as shown in Eq(1). This results in one error in the Sum computation as seen in the truth table of approximate half-adder in Table I. A tick mark denotes that approximate output matches with correct output and cross mark denotes mismatch. In the approximation of full-adder, one of the two XOR gates is replaced with OR gate in Sum calculation as shown in Eq(2). This results in error in last two cases out of eight cases. Carry is modified introducing one error. This provides more simplification, while maintaining the difference between original and approximate value as one. The truth table of approximate full-adder is given in Table II.

$$W = (x1 + x2)$$
$$Sum = W \oplus x3$$
$$Carry = W \cdot x3 \qquad (2)$$

Table II Truth Table of Approximate Full Adder

| Inputs | | | Exact Outputs | | Approximate Outputs | | Absolute Difference |
|---|---|---|---|---|---|---|---|
| $x1$ | $x2$ | $x3$ | $Carry$ | $Sum$ | $Carry$ | $Sum$ | |
| 0 | 0 | 0 | 0 | 0 | 0 ✔ | 0 ✔ | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 ✔ | 1 ✔ | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 ✔ | 1 ✔ | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 ✔ | 0 ✔ | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 ✔ | 1 ✔ | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 ✔ | 0 ✔ | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 ✗ | 1 ✗ | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 ✔ | 0 ✗ | 1 |

The proposed adder can process data in parallel by cutting the carry propagation chain. Let A and B denote the two input binary operands of an adder, S be the sum result, and E represent the error vector.

$$S_i = (A_i \oplus B_i) + A_{i-1}B_{i-1},$$
$$E_i = (A_i \oplus B_i)A_{i-1}B_{i-1}, \qquad (3)$$

where i is the bit index, i.e., i = 0, 1, • • •, n for an n-bit adder. Let A-1 = B-1 = 0 when i is 0, thus, S0 = A0 ⊕B0 and E0 = 0. Also, Ei = 0 when Ai-1 or Bi-1 is 0. Consider an n-bit adder, the inputs are given by A =An-1 • • • A1 A0 and B =

IJIREEICE
ISSN (Online) 2321-2004
ISSN (Print) 2321-5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
Vol. 8, Issue 8, August 2020
DOI 10.17148/IJIREEICE.2020.8811

Bn-1 • • • B1B0, the exact sum isS = Sn-1 • • •S1S0. Then, Si can be computed as Si + Ei and thus, the exact sum of A and B is given by S = S + E. In above equation (3) '+' means the addition of two binary numbers rather than the 'OR' function. The error E is always non-negative and the approximate sum is always equal to or smaller than the accurate sum. This is an important feature of this adder because an additional adder can be used to add the error to the approximate sum as a compensation step.

**B. Approximate Multiplier**

A distinguishing feature of the proposed approximate multiplier is the simplicity to use approximate adders in the partial product accumulation. It has been shown that this may lead to low accuracy, because errors may accumulate and it is difficult to correct errors using existing approximate adders. However, the use of the newly proposed approximate adder overcomes this problem by utilizing the error signal. The resulting design has a critical path delay that is shorter than a conventional one-bit full adder, because the new n-bit adder can process data in parallel. The approximate adder has a rather high error rate, but the feature of generating both the sum and error signals at the same time reduces errors in the final product. An adder tree is utilized for partial product accumulation; the error signals in the tree are then used to compensate the error in the output to generate a product with a better accuracy. The architecture of the proposed approximate multiplier is shown in Fig.1.

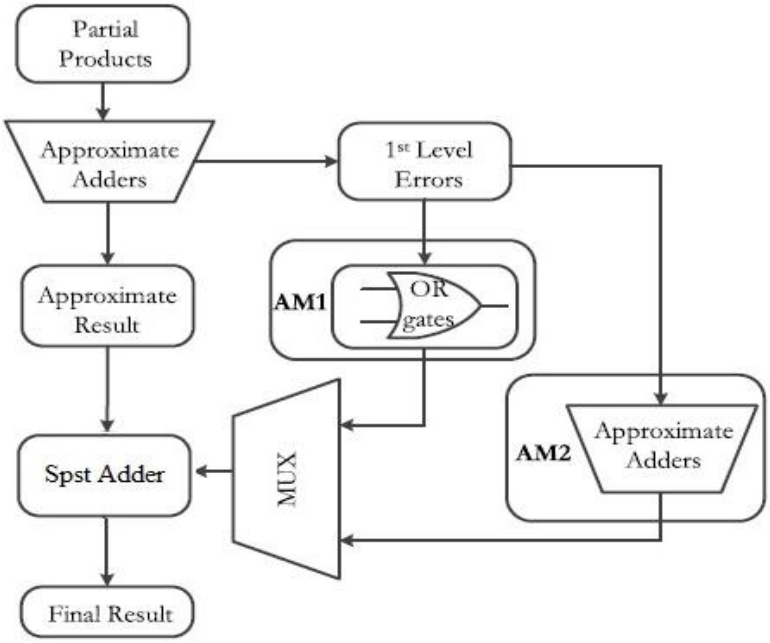


Fig. 1. Block Diagram of Proposed system

In the proposed design, the simplification of the partial product accumulation stage is accomplished by using an adder tree, in which the number of partial products is reduced by a factor of 2 at each stage of the tree as shown in Fig.2.

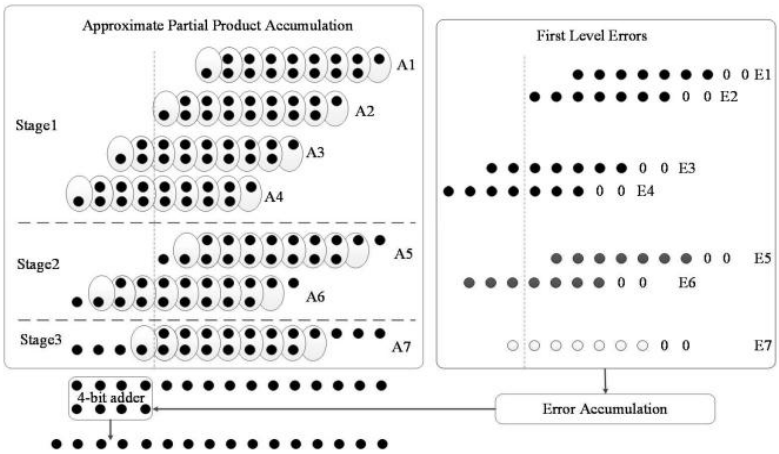


Fig. 2. Approximate Multiplier with Partial Error Recovery

IJIREEICE
ISSN (Online) 2321-2004
ISSN (Print) 2321-5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
Vol. 8, Issue 8, August 2020
DOI 10.17148/IJIREEICE.2020.8811

This adder tree is usually not implemented using accurate multi-bit adders due to the long latency. However, the proposed approximate adder is suitable for implementing an adder tree, because it is less complex than a conventional adder and has a much shorter critical path delay.

## III. ERROR REDUCTION AND SPST ADDITION

The approximate adder generates two signals: the approximate sum S and the error E; the use of the error signal is considered next to reduce the inaccuracy of the multiplier. The sum of every single approximate adder in the tree, an error reduction circuit is applied to the final multiplication result rather than to the output of each adder. Two steps are required to reduce errors: i) error accumulation and ii) error recovery by the addition of the accumulated errors to the adder tree output using a CPA. In the error accumulation step, error signals are accumulated to a single error vector, which is then added to the output vector of the partial product accumulation tree. Two approximate error accumulation methods are proposed, yielding the approximate multiplier 1 (AM1) and approximate multiplier 2 (AM2). Fig. 3. shows the symbols for an a) OR gate, b) full adder and half adder cell and c) an approximate adder cell used in the error accumulation tree.

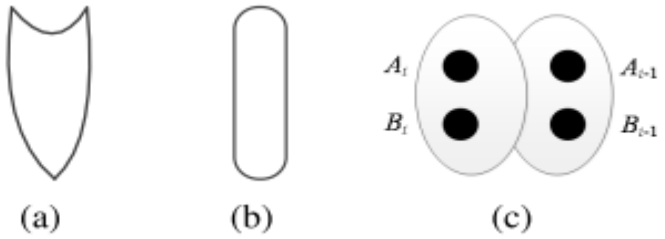


Fig. 3. Symbols for (a) OR gate, (b) A full adder or a half adder and (c) An approximate adder cell

### A. Error Accumulation for Approximate Multiplier 1and 2

Each approximate adder Ai generates a sum vector Si and an error vector Ei, where i = 1, 2, • • • , 7.If the error signals are added using accurate adders, the accumulated error can fully compensate the inaccurate product; however to reduce complexity, an approximate error accumulation is introduced. Consider the observation that the error vector of each approximate adder tends to have more 0's than 1's. Therefore, the probability that the error vectors have an error bit '1'at the same position, is quite small. Hence, an OR gate is used to approximately compute the sum of the errors for a single bit as shown in Fig. 4(a). If m error vectors (denoted by E1, E2, .., Em) have to be accumulated, then the sum of these vectors is obtained as Ei = E1i OR E2i OR ... OR Emi.

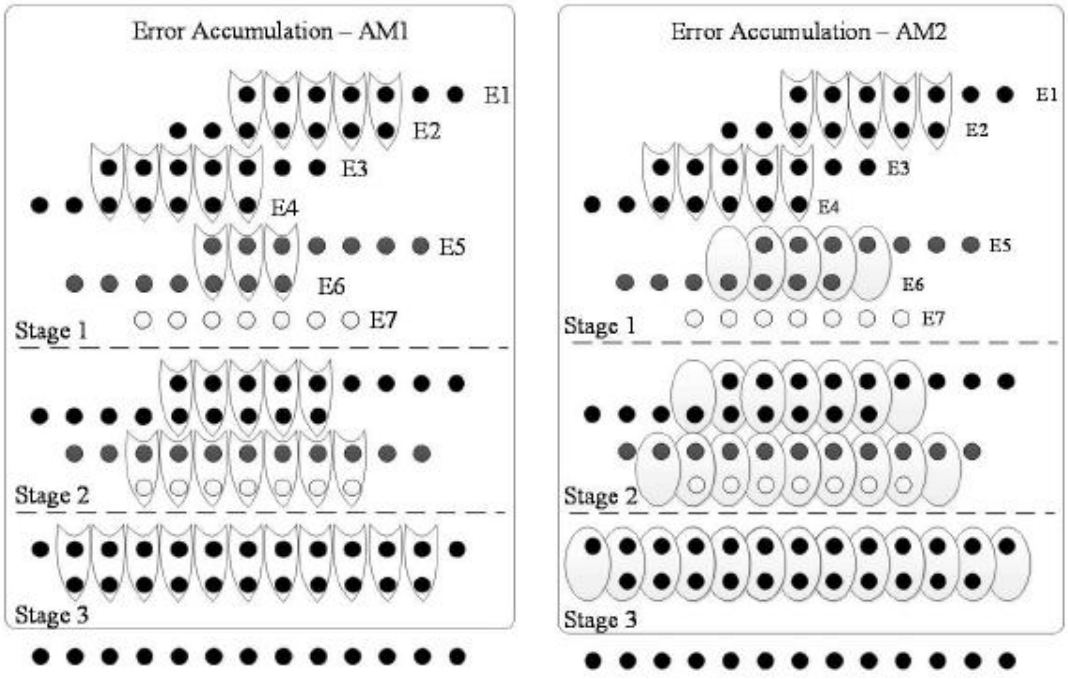


Fig. 4(a). Error accumulation tree for AM1, (b). Error accumulation tree for AM2

IJIREEICE
ISSN (Online) 2321-2004
ISSN (Print) 2321-5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
Vol. 8, Issue 8, August 2020
DOI 10.17148/IJIREEICE.2020.8811

An OR gate can be used to accumulate the two bits in the error vectors E1 and E2 in Fig.4(b). After applying the OR gates to accumulate E1 and E2 as well as E3 and E4, the four error vectors are compressed into two. For E5, E6 and E7, they are generated from the approximate sum of the partial products rather than the partial products. Therefore, they cannot be accurately accumulated by OR gates. E5 and E6 are accurately accumulated by one approximate adder in the first stage of the error accumulation. After the first stage of error accumulation, three vectors are generated, and another two approximate adders are then used to accumulate these three vectors as wellas the error vector remaining from the previous stage (E7).

**B. Spurious-power suppression technique**

The proposed SPST separates the target designs into two parts, i.e., the most significant part and least significant part (MSP and LSP), and turns off the MSP when it does not affect the computation results to save power. Here we explore five cases of 16-bit additions as shown in Fig.5. The cases of exchanging the operands A and B in additions lead to the same spurious transitions. Hence, there is probably no other case beyond these five based on this design. The first case illustrates a transient state in which spurious transitions of carry signals occur in the MSP, although the final result of the MSP is unchanged. Meanwhile, the second and third cases describe situations involving one negative operand adding another positive operand without and with carry-in from the LSP, respectively. Moreover, the fourth and fifth cases demonstrate the addition of two negative operands without and with carry-in from the LSP, respectively. In those cases, the results of MSP are predictable; therefore, the computations in MSP are useless and can be neglected.

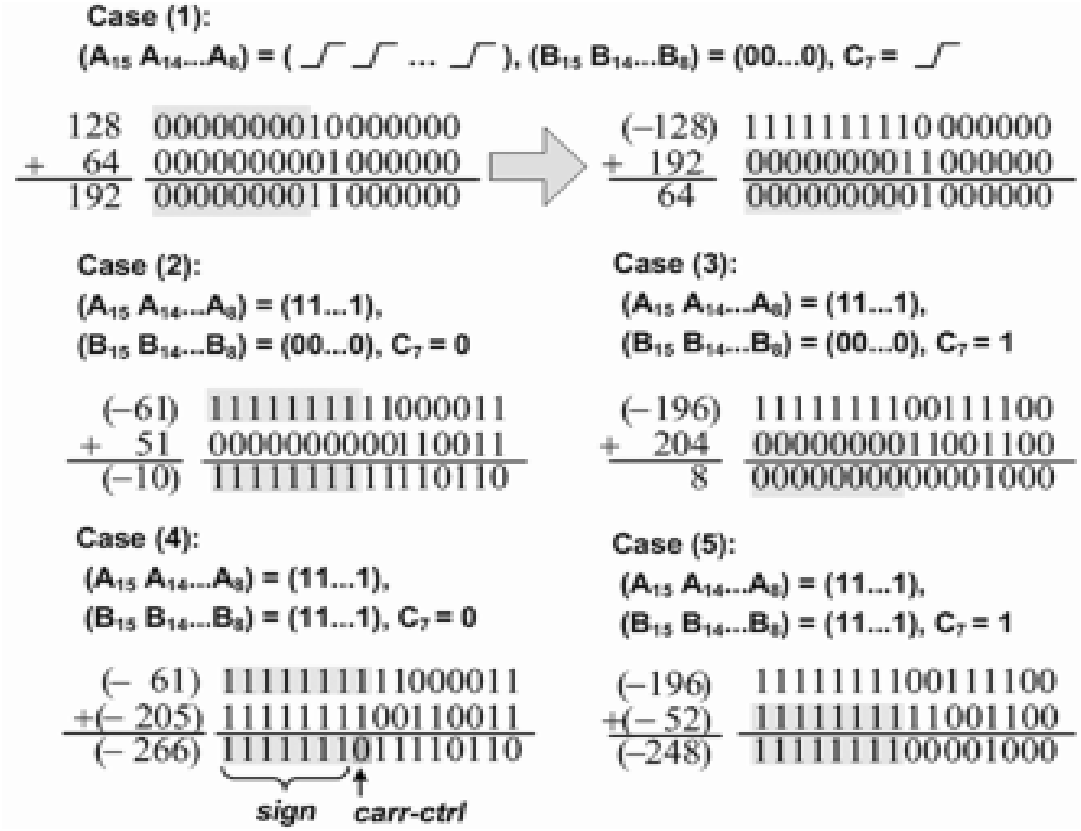


Fig. 5. Different cases of operands A and B

Where operand A and operand B, and AMSP and BMSP, respectively, denote the MSP parts, i.e., the 9th bit to the 16th bit, of the operands A and B. When the bits in AMSP and/or in BMSP are all ones, the value of Aand and/or that of Band, respectively, become one, while when the bits in AMSP and/or in BMSP are all zeros, the value of Anor and/or that of Bnor, respectively, turn into one. Being one of the three outputs of the detection-logic unit, closed notes whether the MSP circuits can be neglected or not. When the two input operands can be classified into one of the five cases, the value of close becomes zero, which indicates that the MSP circuits can be closed to save power dissipation. This design intends to close the MSP circuits by feeding zero inputs into them, which may freeze the switching activities in the MSP circuits to avoid dynamic power consumption.

IJIREEICE
ISSN (Online) 2321-2004
ISSN (Print) 2321-5526
International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
Vol. 8, Issue 8, August 2020
DOI 10.17148/IJIREEICE.2020.8811

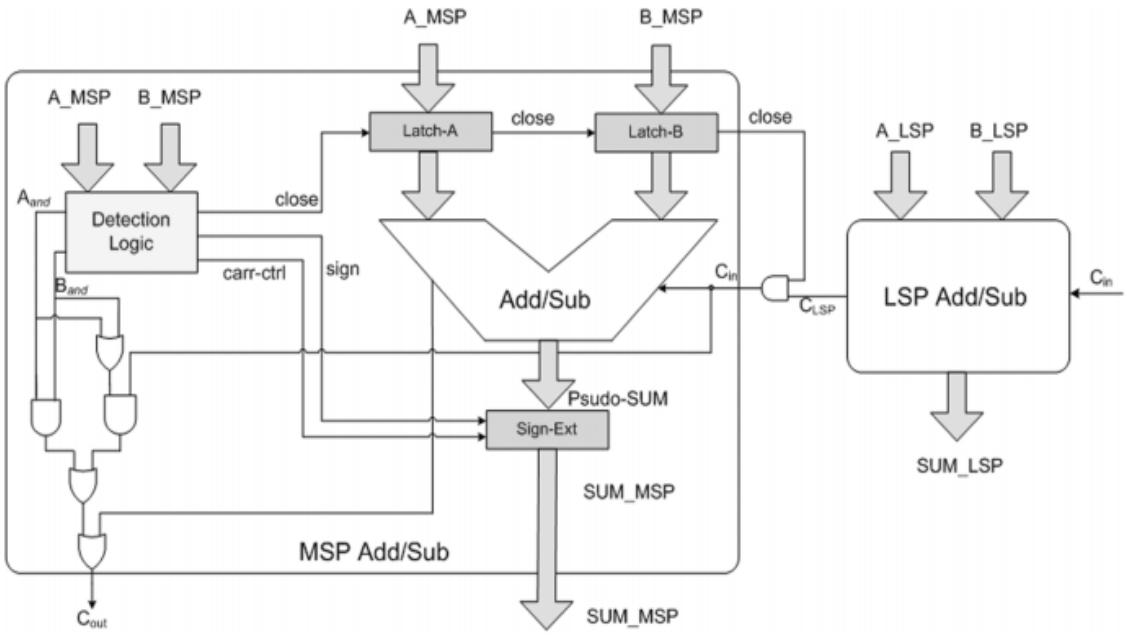


Fig. 6. Proposed SPST Adder

The above Fig.6 shows a 16-bit adder/subtractor design example adopting the proposed SPST. In this example, the 16-bitadder/subtractor is divided into MSP and LSP between the eighth and the ninth bits. Latches implemented by simple AND gates are used to control the input data of the MSP. When the MSP is necessary, the input data of MSP remain unchanged. However, when the MSP is negligible, the input data of the MSP become zeros to avoid glitching power consumption. The two operands of the MSP enter the detection-logic unit, except the adder/subtractor, so that the detection-logic unit as shown in Fig.7 can decide whether to turn off the MSP or not.

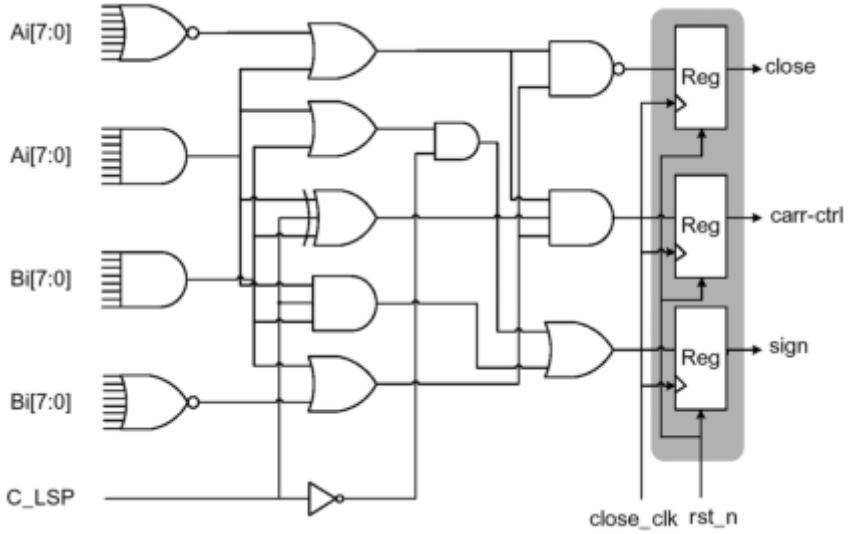


Fig.7. Detection-logic unit

The SE circuits can be intuitively implemented by multiplexers to compensate for the sign signals of the MSP shows the input data of the SE circuits are pseudo summations (PS) from the MSP adder/subtractors.

## IV. RESULTS AND DISCUSSIONS

The simulation result of the low transition approximate multiplier with SPST- Proposed system is shown in Fig.8. 8 × 8 multiplier have been implemented in VHDL and simulated using Modelsim. RTL design is synthesized using Xilinx ISE and its parameters are shown in Table III.

IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering
Vol. 8, Issue 8, August 2020

DOI 10.17148/IJIREEICE.2020.8811

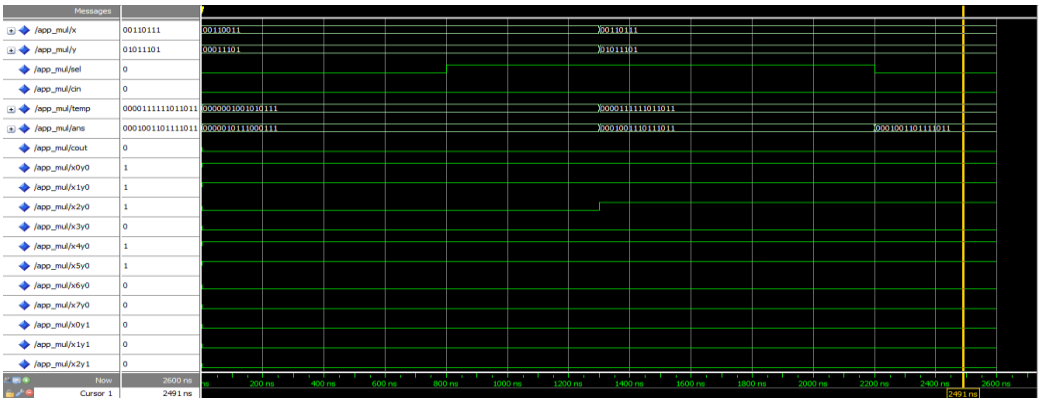


Fig. 8. Simulation result of proposed system

Table III Performance Comparison between existing method and proposed method

| Parameters | Existing Method | Proposed System |
|---|---|---|
| Area (GC) | 1452 | 1524 |
| Delay (ns) | 32.763 | 32.664 |
| Power(mW) | 232 | 211 |

## V. CONCLUSION

From the software simulation and synthesis results we obtained existing method and proposed method power, area, and transition. By using SPST adder we can able to reduce the delay and power when compared to the existing method using Ripple carry adder. Future work includes the further improvement of the output obtained by proposed method is applied to the hardware and applications to be implemented such as Image sharpening and Image smoothening. This project consists of the main technologies approximate computing technology and SPST Technique combined using VHDL algorithms. Comparing with the existing systems this proposed method provides actual results and more advantages like it reduces transition count due to SPST Technique, less computational complexity, less delay and low power dissipation.

## REFERENCES

[1]. Suganthi Venkatachalam and Seok-Bum Ko, " Design of Power and Area Efficient Approximate Multipliers,"2011.
[2]. Honglan Jiang, Cong Liu, Fabrizio Lombardi and Jie Han "Low-Power Approximate Unsigned Multipliers With Configurable Error Recovery".2019
[3]. J. Han and M. Orshansky, "Approximate computing: An emergingparadigm for energy-efficient design," in Proc. 18th IEEE Eur. TestSymp., May 2013, pp. 1–6
[4]. S.-L. Lu, "Speeding up processing with approximation circuits," Computer, vol. 37, no. 3, pp. 67–73, Mar. 2004.
[5]. V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," inProc. IEEE/ACM Int. Symp. Low Power Electron. Design, Aug. 2011, pp. 409–414.
[6]. Kuan-Hung Chen, and Yuan-Sun Chu, "Spurious-Power Suppression Technique forMultimedia/DSP Applications,"2009.
[7]. B. Shao and P. Li, "Array-based approximate arithmetic computing: A general model and applications to multiplier and squarer design," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 62, no. 4, pp. 1081–1090, Apr. 2015.
[8]. O. Chen, S. Wang, and Y. W. Wu, "Minimization of switching activities of partial products for designing low-power multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 3, pp. 418–433, Jun.2003.
[9]. L. Benini, G. D. Micheli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Glitch power minimization by selective gate freezing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 8, no. 3, pp. 287–298, Jun. 2000.
[10]. J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol.63, no.9, pp. 1760–1771, Sep. 2013.