International Journal of Scientific Research in Multidisciplinary Studies
Vol.4, Issue.10, pp.18-24, October (2018)

Research Paper
E-ISSN: 2454-9312
P-ISSN: 2454-6143

Low Power Programmable Truncated Multiplier Using Error Tolerant Adder

N. Hemalatha[1*], K. Kavitha[2]

[1]Dept. of ECE, Rathinam Technical Campus, Coimbatore, India
[2]Dept.of ECE, Rathinam Technical Campus, Coimbatore, India

Available online at: www.isroset.org

Received: 10/Oct/2018, Accepted: 22/Oct/2018, Online: 31/Oct/2018

Abstract— In VLSI design and testing using CMOS technology a novel error-tolerant adder (ETA) is proposed. The novel ETA is able to achieve the accuracy, and at the same time achieve more improvements in both performance of power consumption and speed. The proposed truncated multiplication reduces part of the power required by multipliers and by only computing the product of the most-significant bits. A novel approach to truncation using ETA is proposed, where a full precision multiplier is implemented. When compared to its conventional counterparts, the proposed ETA is able to attain more than 65% improvement in the Power-Delay Product (PDP). One important dynamic application of the proposed ETA is in digital signal processing systems that can tolerate particular amount of errors.

Keywords— *Adders, low-power design, Modified Baugh Wooley Algorithm, truncated multiplication.*

I. INTRODUCTION

The data processed by many digital systems may already contain errors. In many applications, such as a digital communication system, the analog signal first be sampled before being converted to digital data. The analog signal are reconstructed from digital data that are be processed and transmitted in a noisy channel. During this conversion process, errors may occur sometimes. Based on the characteristic of digital VLSI design, some novel concepts and design techniques have been proposed. The concept of error tolerance (ET) [3]–[10] and the PCMOS technology [11]–[13] are two of them. According to the definition, a circuit is error tolerant if: 1) it contains defects that may cause both internal and external errors and 2) the system that incorporates this circuit produces acceptable results [3].

The remaining of the paper is organized as follows. Section II proposes the addition arithmetic as well as the structure of the error-tolerant adder (ETA). In Section III, the detailed design of the ETA is explained. In section IV presents a review of the most relevant publications in the field of truncated multiplication. In section V where the programmable truncated multiplier is introduced as a stand-alone unit. The experimental results are shown in Section VI. Lastly, the conclusion of this work is presented in Section VII.

II. ERROR-TOLERANT ADDER

Before detailing the ETA, the definitions of some commonly used terminologies shown in this paper are given as follows.

- ***Overall error (OE):*** where is the result obtained by the adder, and denotes the correct result (all the results are represented as decimal numbers).
- ***Accuracy (ACC):*** In the scenario of the error-tolerant design, the accuracy of an adder is used to indicate how “correct” the output of an adder is for a particular input. *Need for Error-Tolerant Adder*

Increasingly huge data sets and the need for instant response require the adder to be large and fast. The traditional ripple-carry adder (RCA) is therefore no longer suitable for large adders because of its low-speed performance. Many different types of fast adders, such as the carry-skip adder (CSK) [16], carry-select adder (CSL) [17], and carry-look-ahead adder (CLA) [1], have been developed. Also, there are many low-power adder design techniques that have been proposed [19]. However, there are some trade-offs between speed and power. The error-tolerant design can be approximated solution to this problem. By sacrificing some accuracy, the ETA can
attain great improvement in both the power consumption and speed performance

Proposed Addition Arithmetic: In a conventional adder circuit, the delay is mainly attributed to the carry propagation delay that chain along the critical path, from the least significant bit (LSB) to the most significant bit (MSB). Meanwhile, a significant proportion of the power consumption of an adder is due to the glitches that are caused by the carry propagation. Therefore, if the carry propagation can be eliminated or curtailed, a great improvement in speed performance and power consumption can be achieved. In this paper, we propose for the first time, an innovative and novel addition arithmetic that can achieve great saving in speed and power consumption. This new addition arithmetic can be illustrated via an example shown in Fig. 1.It split the first input operands into parts of two operands that is an accurate part that includes several higher order bits and the inaccurate part that is made up of the remaining lower order bits.

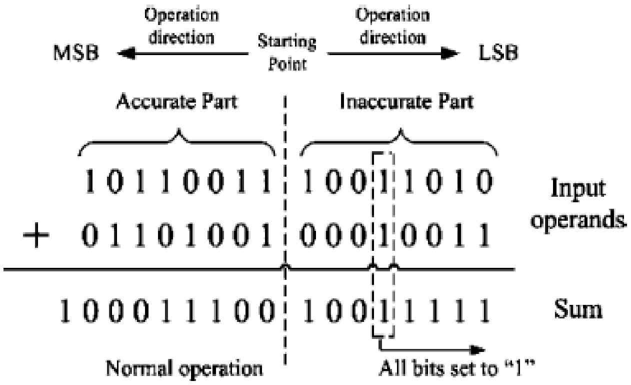


Fig.1. Proposed addition arithmetic

The length of each part need not necessary be equal. The addition process starts from the middle (joining point of the two parts) toward the two opposite directions simultaneously. In the example of Fig.1, the two 16-bit input operands, “1011001110011010” (45978) and “0110100100010011” (26899), are divided equally into 8 bits each for the accurate and inaccurate parts. The addition of the higher order bits (accurate part) of the input operands is performed from right to left (LSB to MSB) and normal addition method is applied. This is to preserve its correctness since the higher order bits play a more important role than the lower order bits. The lower order bits of the input operands (inaccurate part) require a special addition mechanism. No carry signal will be generated or taken in at any bit position to eliminate the carry propagation path.

The overall error is minimized due to the elimination of the carry chain, a special strategy is adapted, and can be described as follow: 1) check every bit position from left to right (MSB to LSB); 2) if both input bits are “0” or different, normal one-bit addition is performed and the operation proceeds to next bit position; 3) if both input bits are “1,” the checking process stopped and from this bit onward, all sum bits to the right are set to “1.” The addition mechanism described can be easily understood from the example given in Fig. 1 with a final result of “10001110010011111” (72863).The example given in Fig. 1 should actually yield “10001110010101101” (72877) if normal arithmetic has been applied. The overall error generated can be computed as OE=72877-72863=14%. The accuracy of the adder with respect to these two input operands is ACC=(1-(14/72877))*100%.By eliminating the carry propagation path in the inaccurate part and performing the addition in two separate parts simultaneously, the overall delay time is greatly reduced, so is the power consumption.

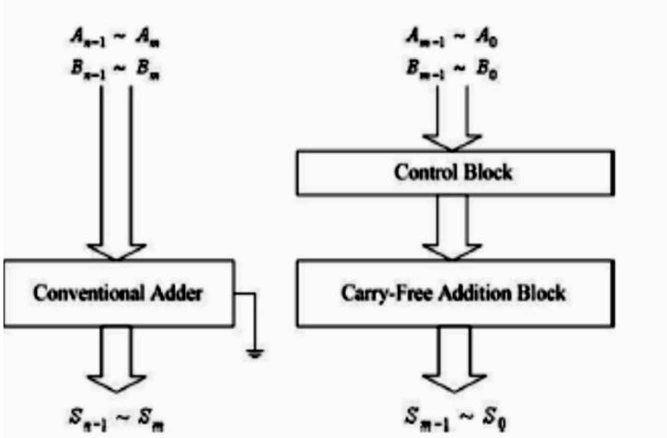


Fig.2. ETA Architecture

Error Tolerant Adder Architecture: The hardware implementation of such an ETA block diagram that adopts our proposed addition arithmetic is provided in Fig. 2.This structure consists of two parts: an accurate part and an inaccurate part. The part of accurate is designed using a conventional adder such as the RCA, CSK, CSL, or CLA. The carry in of this adder is connected to ground. The part of inaccurate constitutes two blocks: a carry-free addition block and a control block. The control block is used to generate the control signals, to determine the working mode of the carry-free addition block. In Section III, a 32-bit adder issued as an example for our illustration the design methodology and circuit implementation of an ETA.

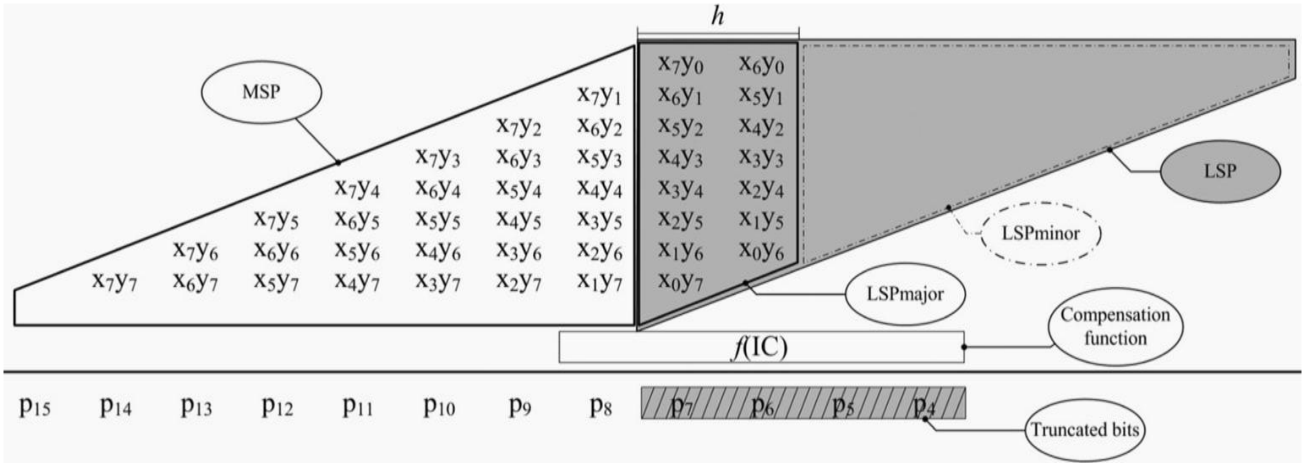

h
MSP
x_7y_0 x_6y_0
x_7y_1 x_6y_1 x_5y_1
x_7y_2 x_6y_2 x_5y_2 x_4y_2
x_7y_3 x_6y_3 x_5y_3 x_4y_3 x_3y_3
x_7y_4 x_6y_4 x_5y_4 x_4y_4 x_3y_4 x_2y_4
x_7y_5 x_6y_5 x_5y_5 x_4y_5 x_3y_5 x_2y_5 x_1y_5
x_7y_6 x_6y_6 x_5y_6 x_4y_6 x_3y_6 x_2y_6 x_1y_6 x_0y_6
x_7y_7 x_6y_7 x_5y_7 x_4y_7 x_3y_7 x_2y_7 x_1y_7 x_0y_7
LSP
LSPminor
LSPmajor
Compensation function
f(IC)
p_{15} p_{14} p_{13} p_{12} p_{11} p_{10} p_9 p_8 p_7 p_6 p_5 p_4
Truncated bits

FIG. 3. partial product matrix for a fixed-width 16-bit truncated multiplier.

III. DESIGN OF A 32-BIT ERROR-TOLERANT ADDER

A.Strategy of Dividing the Adder

The first step of designing a proposed ETA is to divide the adder into two parts in a specific manner. Based on a guess-and-verify stratagem, the dividing strategy is depending on the requirements, such as accuracy, speed, and power. With this partition method is defined and then check whether the accuracy performance of the adder meets the design requirement preset by designer/customer. This can be checked very quickly via some software programs. For a specific application, we require the minimum acceptable accuracy to be 95% and the acceptance probability to be 98%. The proposed partition method must therefore have at least 98% of all possible inputs reaching an accuracy of better than 95%. If this constraint is not met, then one bit should be shifted from the inaccurate part to the accurate part and have the checking process repeated.

B.Design of the Accurate Part

In our proposed 32-bit ETA, the accurate part has 20 bits as opposed to the 12 bits used in the accurate part. The overall delay is measured by the inaccurate part, and so the accurate part need not be a fast adder. The most power-saving conventional adder is RCA, has been considered for the accurate part of the circuit.

C.Design of the Inaccurate Part

The inaccurate part is the most critical (fault) section in the proposed ETA as it determines the circuit accuracy, performance of circuit speed and power consumption of the adder. The inaccurate part Consists of two blocks: the carry free addition block (CFA) and the control block.

IV. BACKGROUND

A.Two's Complement Multiplication

A bit fractional multiplicand and an bit fractionalmultiplier will be considered as the inputs of the multiplier. Let $X = -x_{N-1} + \sum_{i=0}^{N-2} x_i \, 2^{i-N+1}$, $Y = -y_{N-1} + \sum_{i=0}^{N-2} y_i \, 2^{i-N+1}$ where $x_i, y_i \in 0{,}1$. Also, for the sake of generality, their *2N* –bit product $\mathrm{P}_{standard}$ will be maintained in full precision, as:
$\mathrm{P}_{standard} = XY = -p_{2N-1^2} + \sum_{i=0}^{2N-2} p_i \, 2^{i-2N+2}$.

Applying the Baugh-Wooley algorithm [3] to the partial product matrix of all-positive partial product matrix that generates $\mathrm{P}_{standard}$ results in a final matrix of all-positive partial product bits such as

$$\begin{aligned} \mathrm{P}_{standard} = XY = \ & x_{N-1}y_{N-1} + \sum_{i=0}^{N-2} \sum\nolimits_{j=0}^{N-2} x_i \, y_j 2^{i+j-2N+2} \\ & + 2^1 + \ 2^{1-N} + \sum_{i=0}^{N-2} \overline{x_{N-1} y_i} \, 2^{i+1-N} \\ & + \sum_{i=0}^{N-2} \overline{y_{N-1} x_i} \, 2^{i+1-N}. \end{aligned}$$

Once the partial product matrix of the parallel multiplier is generated, partial product terms have to be combined into the final product result. Tree structures are often preferred to array implementations due to their shorter critical path [4]–[6].

B.Truncated Multiplier

Multiplication is a common requirement in DSP systems and plays an important role in terms of power consumption. In those systems where it is not necessary to compute the exact least significant part of the product, truncated multipliers achieve power, area and timing improvements by skipping the implementation of a part of the least significant part of the partial product matrix. Fig. 3 displays a generic partial product matrix, where the partial product matrix is split into two main regions, the least significant part (LSP), which contains the N least significant columns of the partial product matrix, and the most significant part (MSP), that includes the most significant columns of partial product terms. LSP can be also be split in two regions, LSP major being the $h$most significant column, and LSP minor being the least significant columns of the partial product matrix. Truncation allows a way of reducing the complexity of the multiplier unit by discarding the lower parts of the partial product matrix. This results in most of the error being generated in the lower weighted bits of the output that are discarded when converting the output back to the original bit width. By doing so, significant savings in power and complexity can be achieved, at the expense of signal degradation. The simplest scheme to obtain a truncated multiplier consists of removing the lower N columns of the partial product matrix that form the LSP, and is referred to in the literature as direct truncation (D-Truncation). It results in the multiplier requirements being almost halved in both area and power, at the price of large errors with a strong negative bias introduced in the multiplier output.

In order to reduce both the magnitude and bias of the error introduced by the truncation process, most of the references available in the literature study fixed width multiplying structures where the partial-product bits belonging to LSP minor are not implemented. A compensation term is added to compensate for the missing partial products, usually in the form of a function of the input correction partial-products (IC), where the IC terms are the partial product bits in the leftmost column of LSP minor. The constant compensation method introduced in [2], was further improved and explored for fixed width multipliers in [3], [4].Variable correction structures calculate probabilistic estimates of the sum of the elements of the LSP minor by using partial-products in the leftmost column of LSPminor. These schemes differ from each other by different compensation functions, $f(IC)$ applied to generate the compensation value [2], [5], [6], [10]. Review papers [7] and [8] present reviews of existing art, with the latter presenting an analysis on synthesized versions of the referenced technologies, resulting in the first analysis of the electrical benefits derived from truncation. Recent publications in the field of truncated multipliers [10]–[15] include improvements in the truncation error estimation by optimizing the carry estimation mechanism, reducing the average error bias by using symmetric schemes and minimizing the mean square error.

C.Reconfigurable Power Reduction

Fixed-width truncated multipliers minimize power consumption and optimize silicon area and timing, at the price of adding a truncation error, by not implementing the LSPminor region of the partial product matrix. Being purpose-specific and optimized in hardware, they do not provide any flexibility to operate at different configuration modes in systems where the power and performance goals vary with time. Reconfigurable and programmable multipliers are structures that allow hardware multiplier architecture to be configured to operate in different modes. Such structures can be split in two main groups: i) Reconfigurable, (or configurable multipliers) that can dynamically divide a large full precision of fixed-width multiplier into several smaller units working in parallel [2], [3], [8] and ii)

Programmable multipliers, where parts of the partial product matrix can be disabled at run-time by reducing or eliminating switching activity. In [7], dynamic power consumption reduction in a multiplier is achieved by using arbitrary levels of bit precision on a full $N \times N$ bit multiplier. Here reducing the input width results in a reduction on the toggling patterns of different parts of the multiplier, effectively disabling them when left-shifting the input operands.

D. Modified Baugh Wooley two's Complement efficient Signed Multiplier.

One important complication in the development of the efficient multiplier implementation is the multiplier's two's complement signed numbers. Modified Baugh Wooley Two's Complement Signed Multiplier is the best known algorithm for signed multiplication because it maximizes the regularity of the multiplier logic and allows all the partial products to have positive sign bits [8]. To design direct multipliers for two's complement numbers Baugh wooley technique was developed. When multiplying two's complement number directly, each of the partial products to be added is a signed number. Thus each of the partial product has to be sign-extended to the width of the final product in order to form the correct sum by the carry save adder tree. According to the existing approach, an efficient method of adding extra entries to the corresponding bit matrix is suggested to avoid having to deal with the negatively weighted bits in the partial product matrix. Modified form of the Baugh Wooley method, is more preferable since it does not increase the height of the column in the matrix. However, this type of the multiplier is suitable for applications where operands with less than 32 bits are processed, like digital filters where small operands like 6,8,12

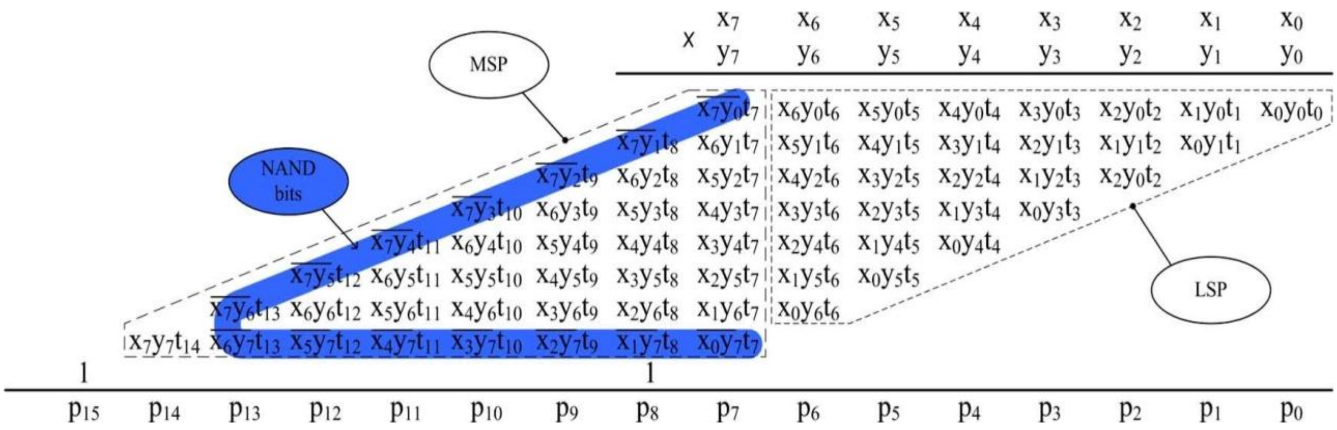

x7 x6 x5 x4 x3 x2 x1 x0
X
y7 y6 y5 y4 y3 y2 y1 y0
MSP
NAND bits
LSP
x7y0t7 x6y0t6 x5y0t5 x4y0t4 x3y0t3 x2y0t2 x1y0t1 x0y0t0
x7y1t8 x6y1t7 x5y1t6 x4y1t5 x3y1t4 x2y1t3 x1y1t2 x0y1t1
x7y2t9 x6y2t8 x5y2t7 x4y2t6 x3y2t5 x2y2t4 x1y2t3 x2y0t2
x7y3t10 x6y3t9 x5y3t8 x4y3t7 x3y3t6 x2y3t5 x1y3t4 x0y3t3
x7y4t11 x6y4t10 x5y4t9 x4y4t8 x3y4t7 x2y4t6 x1y4t5 x0y4t4
x7y5t12 x6y5t11 x5y5t10 x4y5t9 x3y5t8 x2y5t7 x1y5t6 x0y5t5
x7y6t13 x6y6t12 x5y6t11 x4y6t10 x3y6t9 x2y6t8 x1y6t7 x0y6t6
x7y7t14 x6y7t13 x5y7t12 x4y7t11 x3y7t10 x2y7t9 x1y7t8 x0y7t7
1 1
p15 p14 p13 p12 p11 p10 p9 p8 p7 p6 p5 p4 p3 p2 p1 p0

Fig.4. Partial product matrix for a 8 bit programmable truncated multiplier

and 16 bits are used. Baugh Wooley scheme becomes slow and area consuming when operands are greater than or equal to 32 bits.

V.PROPOSED ARCHITECTURE

A.The Programmable Truncated Multiplier

While many specific applications require the inputs and the output of the multiplier to have the same bit width, general purpose digital signal processors need flexibility to support the generation of large output results in accumulators where the magnitude of the output is bigger than the multiplier inputs. They also need to deal correctly with small size operands that do not optimize the input bit width Taking as a starting point a full $2N$-bit multiplier, where the full partial product matrix is computed in order to produce an exact two's complement result, the architecture proposed in this paper provides a method to adjust the active width of the multiplier following a column-based strategy, thus allowing a flexible truncation scheme capable of adapting the power consumption to the requirements of the application. Equation (1) shows the mathematical representation of the partial product matrix of a programmable truncated multiplier. Using external control signals, the active toggling area can be defined in real time.

$$P_{PTM} = X \odot Y = 2^{1} + 2^{-N} + \sum_{i=0}^{N-2}\sum_{j=0}^{N-2} t_{(i+j)} \cdot ppt_{[i,j]} \cdot 2^{i+j-2N+2} \quad (1)$$

Where $\odot$ indicates a truncated multiplication, $t_{(i+j)}$ is an input vector of $2N-1$ elements aligned with the output product bits, and $ppt_{[i,j]}$ are the terms that form the partial product matrix. For a Baugh-wooley implementation $ppt_{[i,j]}$ terms can be obtained as indicated in Equation (2)

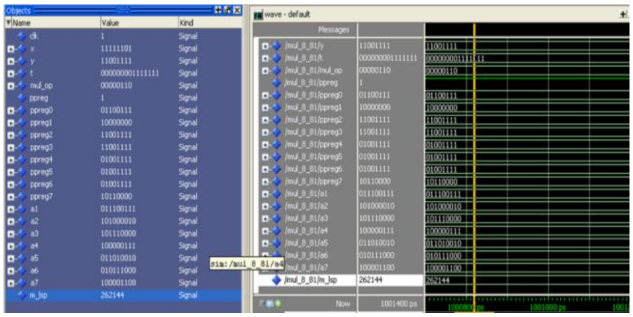
$$ppt[i,j] = \begin{cases} x_i.\, y_j if\ 0 < i < (N-2) and\ 0 < j < (N-2) \\ \overline{x_i.\, y_j} if\ 0 < i < (N-2) and j = (N-1) \\ \overline{x_i.\, y_j} if i = (N-1) and\ 0 < j < (N-2) \\ x_i.\, y_j if\ i = (N-1) and\ j = (N-1) \end{cases}$$

Where x_i and $y_j$terms correspond to the bits in the inputs of the multiplier X and Y. No gating is applied to the constant terms added by the Baugh-wooley algorithm as, being constant these bits not contribute to increase the dynamic power consumption of the multiplier [9]. Fig. 4 illustrates the concept of programmable truncated multiplication in an 8×8 bit two's complement multiplier. The column-wise controllability of the active state of the partial product terms is achieved by substituting the 2-input AND gates that form the original partial product terms by 3-inputAND gates, assigning one of the inputs to the corresponding t_{i+j} bit. Since 3-input AND gates require an area overhead of 20% of 2-input ones, adding programmable truncation results in a maximum area penalty of 20% of the partial product matrix. Post synthesis area reports indicate that the area overhead added due to the addition of programmable truncation in the whole partial product matrix represents approximately 4% of the multiplier area in the proposed 16-bit multiplier, increasing the logic gate count from 886 to 905 equivalent logic gates. The multiplier input controls the functionality of the programmable partial product cells that generate the final result. A value $t = 0x7F00$result in half the partial product matrix being disabled, which is comparable to results obtained by direct truncation for fixed-width multipliers, and equivalent to those in [4] plus a negative bias.

VI. SIMULATION RESULTS

Results for different post-layout simulations are presented in this section. Tools used---XILINX.

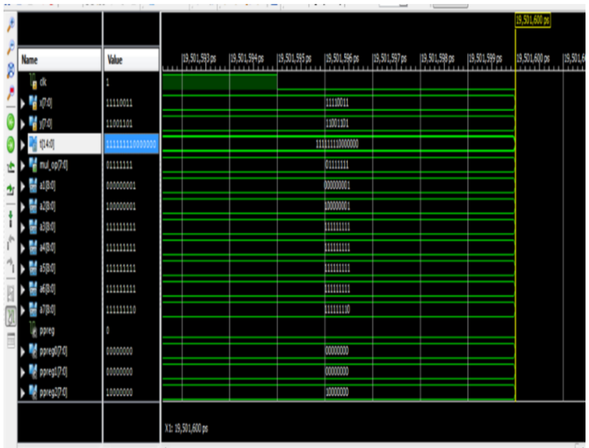
A.Existing PTM using Carry Save Adder Simulation Output



Name	Value	Kind
clk	1	Signal
x	11111101	Signal
y	11001111	Signal
t	000000001111111	Signal
mul_op	00000110	Signal
ppreg	1	Signal
ppreg0	01100111	Signal
ppreg1	10000000	Signal
ppreg2	11001111	Signal
ppreg3	11001111	Signal
ppreg4	01001111	Signal
ppreg5	01001111	Signal
ppreg6	01001111	Signal
ppreg7	10110000	Signal
a1	011100111	Signal
a2	101000010	Signal
a3	101110000	Signal
a4	100000111	Signal
a5	011010010	Signal
a6	010111000	Signal
a7	100001100	Signal
m_lsp	262144	Signal

Fig. 5. Existing PTM using Carry Save Adder Simulation Output

B.Proposed PTM using Error Tolerant Adder Simulation Output



Name	Value
clk	1
x[7:0]	11110011
y[7:0]	11001101
t[14:0]	111111110000000
mul_op[7:0]	01111111
a1[8:0]	000000001
a2[8:0]	100000001
a3[8:0]	111111111
a4[8:0]	111111111
a5[8:0]	111111111
a6[8:0]	111111111
a7[8:0]	111111110
ppreg	0
ppreg0[7:0]	00000000
ppreg1[7:0]	00000000
ppreg2[7:0]	10000000

Fig. 6. Proposed PTM using Error Tolerant Adder Simulation Output

C.Experimental Results

Power Measurement of both PTM using Carry Save Adder and PTM using Error Tolerant Adder measured and their area, delay also obtained. Comparison table is shown in table 1.

Table I: Comparison Between Existing And Proposed Work

Low Power DSP Truncated Multipliers	Programmable Truncated Multiplier Using CSA	Programmable Truncated Multiplier Using ETA
Power (mw)	40(mw)	34(mw)
Area (Total gate count)	1278	1922
Delay(ns)	0.68(ns)	0.19(ns)

D.Comparison Chart

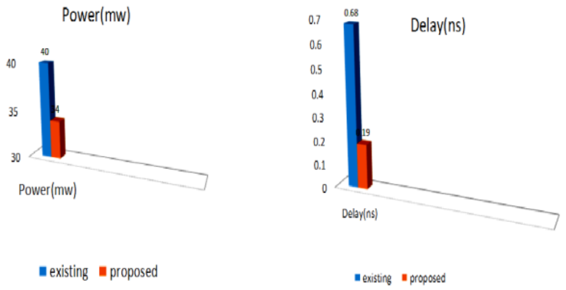


Fig. 7. Comparison Chart between Existing and Proposed Work

VII. CONCLUSION

In this paper, the concept of error tolerance is introduced in VLSI design. A novel type of ETA adder is proposed, which trades certain amount of accuracy for significant power consumption and performance improvement, is proposed. Extensive comparisons with conventional digital adders showed that the proposed ETA is performed better when compared to the conventional adders in both power consumption and speed performance. The dynamic applications of the ETA fall mainly in areas where there is no strict requirement on accuracy or where super low power consumption and high-speed performance are more important than accuracy. A column-wise controllability of the active state of partial product term by adding programmable truncation in the whole partial product matrix is achieved.

REFERENCES

[1] A. B. Melvin, “Let’s think analog,” in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2005, pp. 2–5.

[2] A. B. Melvin and Z. Haiyang, “Error-tolerance and multi-media,” in *Proc. 2006 Int. Conf. Intell. Inf. Hiding and Multimedia Signal Process.* 2006, pp. 521–524.

[3] M. A. Breuer, S. K. Gupta, and T. M. Mak, “Design and error-tolerance in the presence of massive numbers of defects,” *IEEE Des. Test Comput.*, vol. 24, no. 3, pp. 216–227, May-Jun. 2004.

[4] M. A. Breuer, “Intelligible test techniques to support error-tolerance,” in *Proc. Asian Test Symp.*, Nov. 2004, pp. 386–393.

[5] K. J. Lee, T. Y. Hsieh, and M. A. Breuer, “A novel testing methodology based on error-rate to support error-tolerance,” in *Proc. Int. Test Conf.*, 2005, pp. 1136–1144.

[6] I. S. Chong and A. Ortega, “Hardware testing for error tolerant multimedia compression based on linear transforms,” in *Proc. Defect and Fault Tolerance in VLSI Syst. Symp.*, 2005, pp. 523–531.

H.Chung and A. Ortega, “Analysis and testing for error tolerant motion estimation,” in *Proc. Defect and Fault Tolerance in VLSI Syst. Symp.*, 2005, pp. 514–522.

[7] H. H. Kuok, “Audio recording apparatus using an imperfect memory circuit,” U.S. Patent 5414758, May 9, 1995.

[8] T. Y. Hsieh, K. J. Lee, and M. A. Breuer, “Reduction of detected acceptable faults for yield improvement via error-tolerance,” in *Proc. Des., Automation and Test Eur. Conf. Exhib.*, 2007, pp. 1–6.

[9] K. V. Palem, “Energy aware computing through probabilistic switching: A study of limits,” *IEEE Trans. Comput.*, vol. 54, no. 9, pp. 1123–1137, Sep. 2005.

[10] S. Cheemalavagu, P. Korkmaz, and K. V. Palem, “Ultra low energy computing via probabilistic algorithms and devices: CMOS device primitives and the energy-probability relationship,” in *Proc. 2004 Int. Conf. Solid State Devices and Materials*, Tokyo, Japan, Sep. 2004, pp. 402–403.

[11] P.Korkmaz,B.E.S.Akgul,K.V.Palem,andL.N.Chakrapani,“Advocating noise as an agent for ultra-low energy computing: Probabilistic complementary metal-oxide-semiconductor devices and their characteristics,” *Jpn. J. Appl. Phys.*, vol. 45, no. 4B, pp. 3307–3316, 2006.

[12] J. E. Stine, C. R. Babb, and V. B. Dave, “Constant addition utilizing flagged prefix structures,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, 2005.

[13] L.-D. Van and C.-C. Yang, “Generalized low-error area-efficient fixed width multipliers,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 25, no. 8, pp. 1608–1619, Aug. 2005.